Paul Wohlhart
Vincent Lepetit (eds.)

Proceedings of the

20th Computer Vision Winter Workshop

Seggau, Austria
February 9 - 11, 2015

Graz University of Technology

Paul Wohlhart, Vincent Lepetit (eds.)

Proceedings of the

20th Computer Vision Winter Workshop

Seggau, Austria, February 9-11, 2015

Graz University of Technology 2015

Volume Editors

Paul Wohlhart, Vincent Lepetit
Graz University of Technology
Institute for Computer Graphics and Vision
Inffeldgasse 16/II, 8010 Graz, Austria
Email: {wohlhart, lepetit}@icg.tugraz.at

Layout and Cover

Markus Oberweger
Email: oberweger@icg.tugraz.at

Cover Image

© Kristoferitsch

Graz University of Technology 2015

Institute for Computer Graphics and Vision
Inffeldgasse 16/II, 8010 Graz, Austria
www.icg.tugraz.at

ISBN 978-3-85125-388-7

DOI 10.3217/978-3-85125-388-7

Preface

The 20^{th} Computer Vision Winter Workshop (CVWW) was organized by the Institute for Computer Graphics and Vision at Graz University of Technology. It took place from 9^{th} to 11^{th} of February 2015 in Seggau, Austria. The Computer Vision Winter Workshop is the annual meeting of several computer vision research groups located in Graz, Ljubljana, Prague, and Vienna. The basic goal of this workshop is to communicate new ideas within the groups and to provide conference experience to PhD students. In this spirit the topics of the workshop were not explicitly limited to a specific topic but include computer vision, image analysis, pattern recognition, medical imaging, 3D vision, human computer interaction, vision for robotics, as well as applications.

We received 25 paper submissions from six countries. Each paper was reviewed by three members of our international program committee. Among these 25 papers, 23 papers were accepted for presentation at the workshop (19 oral presentations and 4 poster presentations). 9 authors took the opportunity to withdraw their paper from the proceedings so that no restrictions on submitting the work to other conferences and journals is imposed.

Besides papers selected in the review process, one invited talk was included in the program. We would like to express our thanks to Prof. Dr. Davide Scaramuzza (University of Zurich). We extend our thanks to the members of the program committee for their time and the their mostly detailed and very helpful feedback to the authors. We would like to extend our sincere thanks to everyone who helped in making CVWW 2015 possible. We are indebted to Eva-Maria Christina Fuchs and Karin Maier for their help with all organizational matters. We also want to thank the sponsors of the workshop for their support: *Federal Government of Styria* and *Vexcel Imaging - a Microsoft company*.

Paul Wohlhart, Vincent Lepetit
CVWW 2015 Workshop Chairs
Graz, Austria, February 2015

Workshop Chair

Paul Wohlhart (Graz University of Technology)
Vincent Lepetit (Graz University of Technology)

Program Committee

Csaba Beleznai (Austrian Institute of Technology)
Horst Bischof (Graz University of Technology)
Nicole Brosch (Vienna University of Technology)
Jan Cech (Czech Technical University Prague)
Luka Cehovin (University of Ljubljana)
Ondrej Chum (Czech Technical University Prague)
Ondrej Drbohlav (Czech Technical University Prague)
Vojtech Franc (Czech Technical University Prague)
Friedrich Fraundorfer (Graz University of Technology)
Margrit Gelautz (Vienna University of Technology)
Václav Hlaváč (Czech Technical University Prague)
Aleš Jaklič (University of Ljubljana)
Stanislav Kovacic (University of Ljubljana)
Matej Kristan (University of Ljubljana)
Walter G. Kropatsch (Vienna University of Technology)
Zuzana Kukelova (Czech Technical University Prague)
Vincent Lepetit (Graz University of Technology)
Rok Mandeljc (University of Ljubljana)
Jiří Matas (Czech Technical University Prague)
Martin Matousek (Czech Technical University Prague)
Matej Nezveda (Vienna University of Technology)
Tomáš Pajdla (Czech Technical University Prague)
Janez Pers (University of Ljubljana)
Thomas Pock (Graz University of Technology)
Daniel Průša (Czech Technical University Prague)
Rene Ranftl (Graz University of Technology)
Peter M. Roth (Austrian Institute of Technology)
Robert Sablatnig (Vienna University of Technology)
Radim Sara (Czech Technical University Prague)

Viktoriia Sharmanska (Institute of Science and Technology, Austria)
Alexander Shekhovtsov (Graz University of Technology)
Danijel Skočaj (University of Ljubljana)
Tomáš Svoboda (Czech Technical University Prague)
Paul Wohlhart (Graz University of Technology)

Contents

20^{th} *Computer Vision Winter Workshop*
Paul Wohlhart, Vincent Lepetit (eds.)
Seggau, Austria, February 9-11, 2015

Towards Agile Flight of Vision-controlled Micro Flying Robots: from Frame-based to Event-based Vision

Davide Scaramuzza
University of Zürich, Switzerland
sdavide@ifi.uzh.ch

Abstract. *Autonomous quadrotors will soon play a major role in search-and-rescue and remote-inspection missions, where a fast response is crucial. Quadrotors have the potential to navigate quickly through unstructured environments, enter and exit buildings through narrow gaps, and fly through collapsed buildings. However, their speed and maneuverability are still far from those of birds. Indeed, agile navigation through unknown, indoor environments poses a number of challenges for robotics research in terms of perception, state estimation, planning, and control. In this talk, I will give an overview of my research activities on visual navigation of quadrotors, from slow navigation (using standard frame-based cameras) to agile flight (using event-based cameras).*

20^{th} *Computer Vision Winter Workshop*
Paul Wohlhart, Vincent Lepetit (eds.)
Seggau, Austria, February 9-11, 2015

Towards Segmentation of Human Teeth Contours in Dental Radiographs Using Active Shape Models

Michael Sprinzl, Walter G. Kropatsch
Vienna University of Technology
Institute of Computer Graphics and Algorithms
Pattern Recognition and Image Processing Group
Vienna, Austria
msprinzl@prip.tuwien.ac.at, krw@prip.tuwien.ac.at

Robert Sablatnig
Vienna University of Technology
Institute of Computer Aided Automation
Computer Vision Lab
Vienna, Austria
sab@caa.tuwien.ac.at

Georg Langs
Medical University of Vienna
Department of Biomedical Imaging and Image-guided Therapy,
Computational Imaging Research Lab
Vienna, Austria
georg.langs@meduniwien.ac.at

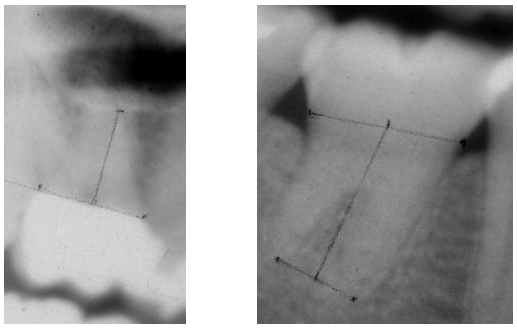
Abstract. *We present a framework for segmentation of human teeth contours in dental radiographs. As all humans share the same tooth structure, but show variation in size and morphology, these variations can be modelled using statistical methods. Therefore we propose "Active Shape Models" (ASM) as segmentation approach. ASM are flexible, statistically based models which iteratively move toward structures in images similar to those on which they were trained in advance and consist of a set of corresponding landmarks. Each landmark represents a part of the tooth's boundary to be located. The training phase of our proposed framework incorporates noise removal, manual segmentation of training images, solving the correspondence problem, aligning the set of training images, and capturing its statistics. For image interpretation, the model of the tooth is placed into the target image. The model parameters are then iteratively adjusted to move the landmarks closer to the contour of the tooth to be segmented. Constraints are applied so that the overall tooth shape to be segmented cannot deform more than the teeth seen in the corresponding training set. Our proposed framework is evaluated using a set of intra-oral dental radiographs containing 60 molars and 70 premolars from 24 patients (22 female, 2 male), taken over a period of ten years.*

1. Introduction

The Department of Oral Surgery of the Bernhard Gottlieb University Clinic for Dentistry (BGUCD) at the Medical University Vienna (MUV) performs more than 2500[1] oral surgery procedures every year. Priorities of the surgical timetable are autotransplantations ("auto" from the Greek meaning for "self"), where the tooth to be transplanted is taken from the same person. In order to determine within the pre-grafting state, which tooth suits best as a donor, and to predict the risk that the grafted tooth will get lost within the post-grafting state, measurements at the dental radiographs of the relevant tooth are performed. As up to now no software exists, which is capable of performing these measurements, they are done manually (see Fig. 1).

Different approaches for segmenting teeth within dental radiograms have been presented in scientific literature so far. In [33], Nomir and Abdel-Mottaleb make use of integral projection. Barboza et al. adopt in [2, 3] a semi-automatic algorithm based on Image Foresting Transform (IFT). The IFT (introduced by Falcão et al. in [17]) defines a robust minimum-cost path in a graph given a set of seed pixels which are the roots of a forest in which the region growth starts.

[1]The number of oral surgery procedures is taken from the homepage of the Department of Oral Surgery.



(a) maxillary molar (b) mandibular molar

Fig. 1: Examples of performing manual measurements on dental radiographs containing molars in the upper jaw (“Maxilla”) and lower jaw (“Mandible”).

The method recommended by Lin et al. in [25] consists of four stages: image enhancement using an adaptive power law transformation, singularity analysis using local Hölder exponent, tooth recognition using Otsu’s threshold, connected component analysis, and tooth delineation using morphological operations. Morphological operations are also used by Said, Nassar, and Fahmy in [38].

The teeth segmentation pipeline proposed by Frejlichowski and Wanat in [18] consists of three stages: it starts with a morphological opening in order to reduce the noise and to create larger areas of similar intensity range. Afterwards, entropy filtering is applied to detect edges of similar areas. Finally, an iterative watershed region growing constrained by ridge information (see [6] for more details) is done.

Chen and Jain contribute two approaches: In [5] they use Gaussian mixture models (GMM), while in [4] generalized fast marching methods (GFMM) are used. GFMM are special cases of level sets and were introduced by Sethian in [40].

By looking at the papers published so far, it can be concluded that the vast majority uses graph-based and/or morphology-based methods. A drawback that all these methods have in common is that due to noise and artefacts within the image, the segmentation results may not show any similarities to shapes of human teeth at all. This motivates our usage of “Active Shape Models” (ASM) as segmentation approach. ASM, introduced by Cootes and Taylor in [9], are flexible, statistically based models, which iteratively move toward structures in images similar to those on which they were trained in advance. Their application to medical images is shown e. g. in [1, 8, 11, 19, 21, 22, 34, 36, 41, 44].

Overview and contribution. Within this paper we present our proposed teeth segmentation framework consisting of noise reduction, building ASM for molars and premolars using corresponding landmarks on training images, and searching for teeth in target images. Our framework can be used either with MATLAB® or GNU Octave.

Within Sec. 2, the medical basics concerning human teeth are presented in a compact manner. Sec. 3 explains our proposed teeth segmentation framework in detail, while the achieved results are presented and discussed in Sec. 4. In Sec. 5, we sum up the conclusions we achieved and address future enhancements.

2. Anatomy of human teeth

According to the definition given by Marcovitch in [27], human teeth are mineralised organs implanted in the jaw, where their visible parts emerge from the bone. The human dentition consists of 20 primary teeth and 32 permanent teeth, which can be classified in incisors, canines, premolars, and molars. Each human tooth has a crown and a root portion. The root portion of the human tooth is implanted into the alveolar jawbone through the periodontal ligament, also called periodontal membrane, and the gum (“Gingiva”), as Nelson explains in [31]. The segmentation is done at this transition between the tooth and its surrounding gingivial tissue, which has a size of 2-4 mm, according to Newman et al. in [32].

3. Teeth Segmentation Framework

ASM consists of a sequence of landmarks, each representing corresponding points between similar shapes. During training, a model for molars and premolars is built using the statistics of landmark points within a set of training images. For image interpretation, the model of the tooth to be segmented, is placed into the target tooth image. The tooth model parameters are then iteratively adjusted to move the landmarks closer to the contour of the tooth to be segmented. Constraints are applied so that the overall tooth shape cannot deform more than the teeth seen in the corresponding training set.

3.1. Training phase

The training phase of our proposed teeth segmentation framework incorporates five steps: removing the impulsive noise, manual segmentation of training images, solving the correspondence problem, aligning the training images, and capturing its statistics.

Impulsive Noise Reduction. The dental radiographs we use for training our proposed teeth segmentation framework are analogue X-ray films that were scanned by means of a charge-coupled device (CCD) based X-ray image scanner. This conversion introduces impulsive noise, which appears as random patterns of light and dark pixels ("Salt-and-pepper noise"). Median filtering is used in digital image processing, because it preserves edges while removing impulsive noise. Lin states in [26] that his proposed Adaptive Centre Weighted Median (ACWM) filter *"outperforms eight well-accepted alternative median-based filters in terms of both noise suppression and detail preservation. It also provides excellent robustness at various percentages of impulsive noise."* Therefore we use his proposed ACWM filter in order to reduce the impulsive noise in our training images.

Segmentation of Training Images. To speed up the manual segmentation of the training images, we utilise an interactive graph-based image segmentation technique called "Intelligent Scissors", proposed by Mortensen and Barrett in [29, 30]. The underlying mechanism for Intelligent Scissors is the "Live-Wire" path selection tool. The Live-Wire tool allows the user to interactively select the optimal boundary from a source pixel to a target pixel. To minimise user interaction, seed points are generated automatically along the current active boundary segment via "boundary cooling". Boundary cooling occurs, when a section of the current portion of the boundary has not changed recently and consequently "freezes", depositing new seed points, while continuing the optimal boundary expansion.

Solving the Correspondence Problem. After the manual segmentation of the teeth contours, a problem arises when a set of sample points has to be chosen that is placed exactly at corresponding locations within the training set. This problem is known as "correspondence problem", and is discussed e. g. by Kotcheff and Taylor in [24] and Davies et al. in [15]. One way of solving the correspondence problem is using anatomical landmarks. Kotcheff and Taylor point out in [24] that this manual process is slow, introduces an operator bias and – especially in medical applications – requires expert knowledge of the anatomical structures being dealt with. These problems motivate our search for a method that is capable of solving the correspondence problem without any user intervention.

We use the approach proposed by Davies et al. in [15, 16], which incorporates the Minimum Description Length (MDL) principle (introduced by Rissanen in [37]), for finding pseudo-landmarks automatically within our n_s manually segmented training images.

Aligning a Set of Training Images. In order to be able to compare training shapes containing an equal number of pseudo-landmarks, it is important that the shapes are represented in the same coordinate frame, as Cootes et al. point out in [12]. Therefore, the shapes have to be aligned with respect to a set of axes, in order to remove any kind of variation, which could be attributable to the allowed global transformation. We solve this problem by minimising a sum of squared differences between corresponding pseudo-landmarks on different shapes, which corresponds to a Generalised Procrustes Analysis (GPA) as proposed by Gower in [20], and define x_i as vector containing n_{lm} pseudo-landmarks of the i-th tooth in the training set X such that

$$\begin{aligned} x_i = (&x_{i1}, x_{i2}, \ldots, x_{ik}, \ldots, x_{in_{lm}}, \\ &y_{i1}, y_{i2}, \ldots, y_{ik}, \ldots, y_{in_{lm}})^T. \end{aligned} \tag{1}$$

When two shapes x_i and x_j have to be aligned ($x_i, x_j \in X$), GPA determines a linear transformation of the landmarks in x_j to best conform to the landmarks in x_i. More formally, GPA aligns two shapes by choosing a rotation θ, a scale s, and a translation $t = (t_x, t_y)^T$, mapping x_j onto x_i, so that the resulting dissimilarity measure

$$\begin{aligned} D = \sum_{k=1}^{n_{lm}} \Bigg(&\Big(\begin{bmatrix} x_{ik} \\ y_{ik} \end{bmatrix} - M(s,\theta)[x_{jk}] - t \Big) \\ &\Big(\begin{bmatrix} x_{ik} \\ y_{ik} \end{bmatrix} - M(s,\theta)[x_{jk}] - t \Big)^T \Bigg) \end{aligned} \tag{2}$$

is minimised, where

$$M(s,\theta) \begin{bmatrix} x_{jk} \\ y_{jk} \end{bmatrix} = \begin{pmatrix} x_{jk}a_x - y_{jk}a_y \\ x_{jk}a_y + y_{jk}a_x \end{pmatrix}, \tag{3}$$

$$\begin{aligned} a_x &= s\, cos\theta, \\ a_y &= s\, sin\theta. \end{aligned} \tag{4}$$

Computing the derivatives of D shown in Eq. 2 wrt. t_x, t_y, a_x, a_y leads us to A, a set of four linear equations, such that

$$A = \begin{pmatrix} B_1 & -B_2 & n_{lm} & 0 \\ B_2 & B_1 & 0 & n_{lm} \\ B_3 & 0 & B_1 & B_2 \\ 0 & B_3 & -B_2 & B_1 \end{pmatrix} \begin{pmatrix} t_x \\ t_y \\ a_x \\ a_y \end{pmatrix} = \begin{pmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \end{pmatrix}, \tag{5}$$

where

$$B_1 = \sum_{k=1}^{n_{lm}} x_{ik},\ B_2 = \sum_{k=1}^{n_{lm}} y_{ik},\ B_3 = \sum_{k=1}^{n_{lm}} (x_{ik}^2 + y_{ik}^2),$$
$$C_1 = \sum_{k=1}^{n_{lm}} x_{jk},\ C_3 = \sum_{k=1}^{n_{lm}} (x_{ik}x_{jk} + y_{ik}y_{jk}),$$
$$C_2 = \sum_{k=1}^{n_{lm}} y_{jk},\ C_4 = \sum_{k=1}^{n_{lm}} (x_{ik}y_{jk} - y_{ik}x_{jk}). \tag{6}$$

As long as the set of four linear equations shown in Eq. 2 has a non-singular matrix ($det(A) \neq 0$), it can be solved using standard matrix methods resulting in a single unique solution for t_x, t_y, a_x, a_y. We use an iterative approach for aligning all training shapes within X. It consists of four steps:

1. $\forall x \in X$: align x_i with current $\bar{x}$.
2. re-calculate $\bar{x}$ using Eq. 7.
3. align current $\bar{x}$ with initial $\bar{x}$, set current $|\bar{x}| = 1$.
4. $d\bar{x}$ = current $\bar{x}$ − previous $\bar{x}$.

Our iterative approach is repeated until $d\bar{x}$ drops under a predefined threshold or the maximum number of iterations is reached.

Capturing the Training Images Statistics. After alignment, all training images are centred and share a common coordinate frame. But one problem remains: each landmark within the training set forms a cloud of corresponding points in a $2n_{lm}$-dimensional space. To simplify this problem, we apply Principal Component Analysis (PCA) on the aligned shapes in order to reduce their dimensionality. Therefore we calculate the mean shape vector $\bar{x}$ such that

$$\bar{x} = \frac{1}{n_s} \sum_{i=1}^{n_s} x_i \tag{7}$$

and determine the covariance matrix S such that

$$S = \frac{1}{n_s} \sum_{i=1}^{n_s} (x_i - \bar{x})(x_i - \bar{x})^T. \tag{8}$$

Now PCA can be applied on S, resulting in p_k ($k = 1, 2, \ldots, 2n_{lm}$) eigenvectors of S such that

$$Sp_k = \lambda_k p_k, \tag{9}$$

where λ_k is the k^{th} corresponding eigenvalue of S (sorted so that $\lambda_k \geq \lambda_{k+1}$).

In order to reduce the dimensionality of the data, the number of eigenvectors (and their corresponding eigenvalues) has to be reduced. Using the fact addressed by Johnson and Wichern in [23] that the variance explained by each eigenvector is equal to the corresponding eigenvalue, the total variance σ^2 is the sum of all eigenvalues, λ_T such that

$$\sigma^2 = \sum_{k=1}^{2n_{lm}} \lambda_k. \tag{10}$$

We choose t, the number of eigenvalues to retain, such that

$$\sum_{i=1}^{t} \lambda_i \geq f_v \sigma^2, \tag{11}$$

where f_v defines the proportion of the total variance of the training shapes that shall be explained (e. g. 95.45%, which is equivalent to $\pm 2\sigma$ standard deviation of σ^2).

When new shapes are created using the statistics captured above, it is worth noticing that precautions have to be taken in order to ensure that they are similar to the shapes already present within the training data. Cootes et al. name this in [12] as *"creating new allowable shapes"* or *"producing plausible shapes"* that lie within the Allowable Shape Domain (ASD) of the training data. Any shape within the ASD can be approximated by taking $\bar{x}$ and adding a linear combination of the first t eigenvectors multiplied by a vector of weights such that

$$x_{new} \approx \bar{x} + P_t b_t, \tag{12}$$

where $P_t = (p_1; p_2; \ldots; p_t)$ is a matrix of the first t eigenvectors, and $b_t = (b_1, b_2, \ldots, b_t)^T$ a t-dimensional vector of weights.

3.2. Image Interpretation

Having generated ASM for molars and premolars, we can use them to segment examples of teeth within dental radiographs. This involves - after removing the impulsive noise from the target image, which is done using our proposed ACWM filter - finding shape, scale, and pose parameters which cause the tooth model to coincide with the structures of interest in the dental radiograph containing the tooth to be segmented. According to the definition given by Cootes et al. in [12], an instance of the tooth model is given by

$$X = M(s, \theta)[x] + X_c, \tag{13}$$

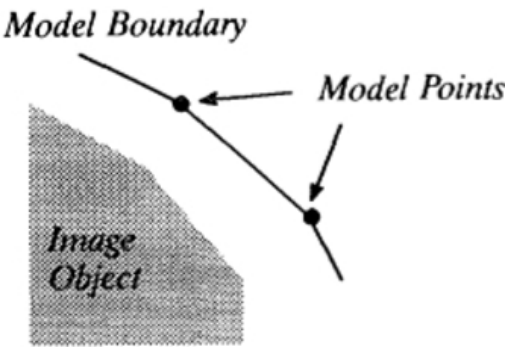


Fig. 2: Part of a model boundary created by connecting the model points (landmarks) approximating to the edge of an image object (Fig. courtesy of [12]).

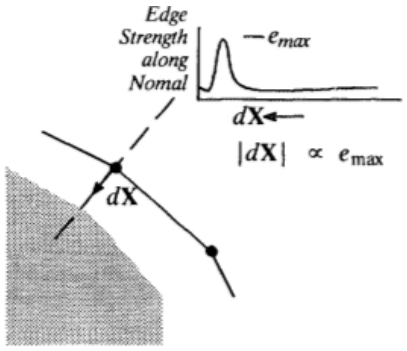


Fig. 3: Suggested movement dX of a model point along a normal to the boundary proportional to the edge strength (Fig. courtesy of [12]).

where $M(s,\theta)[x]$ is a scaling by s and a rotation by θ as defined in Eq. 3, and X_c incorporates the position of the centre of the corresponding tooth model in the image frame such that $X_c = (x_{c1}, x_{c2}, \ldots, x_{cn_s}, y_{c1}, y_{c2}, \ldots, y_{cn_s})^T$. We use an iterative approach for refining the shape, scale, and pose parameters in order to give a better match to the tooth to be segmented. It consists of three steps:

1. Examine a region around each landmark to calculate the displacements in order to move the landmarks closer to the boundary of the tooth.
2. Use these proposed displacements to calculate adjustments to the shape, scale, and pose parameters of the tooth model.
3. Update the tooth model parameters. By enforcing limits on the shape parameters, global shape constraints can be applied ensuring that the current instance of the tooth model cannot deform more than the teeth seen in the corresponding training set.

Our iterative approach is repeated until either the Sum of Squared Errors (SSE) between the current and the previous instance of the model drops under a predefined threshold or the maximum number of iterations is reached.

Move landmarks closer to the boundary. To start the segmentation process, the user has to place an estimation of the mean shape vector $\bar{x}$ within the dental radiograph containing the tooth to be segmented, which leads to an initial situation similar to the one shown in Fig. 2. As the pseudo-landmarks within an ASM represent the boundaries of image objects, they have to be moved towards the contour of the tooth to be segmented in order to give a better match within the next iteration. In the examples Cootes et al. mention in [12], they use an adjustment perpendicular to the model boundary toward the strongest image edge, with a magnitude proportional to the strength of the edge, as illustrated in Fig. 3. This approach results in a vector of adjustments, dX, such that $dX = (dX_1, dX_2, \ldots, dX_{n_{lm}}, dY_1, dY_2, \ldots, dY_{n_{lm}})^T$.

Calculate adjustments of model parameters. Adjusting the scale and pose parameters of the tooth model means moving the landmarks from their current locations X to the suggested better locations $X+dX$. If we assume that X, the current instance of the tooth model, is centred at X_c with orientation θ and scale s, a set of residual adjustments dx in the local tooth model coordinate frame can be achieved by finding a translation dX_c, a rotation $d\theta$, and a scaling factor $1 + ds$, which best map the landmarks from X to $X + dX$ using Eq. 2-6 such that

$$X + dX = M(s(1+ds), (\theta + d\theta))[x + dx] + (X_c + dX_c). \quad (14)$$

Inserting Eq. 13 in Eq. 14, eliminating the term X_c, and moving the term dX_c to the left results in

$$M(s,\theta)[x] + dX - dX_c = M(s(1+ds), (\theta + d\theta))[x + dx], \quad (15)$$

and since $M^{-1}(s,\theta)[\ldots] = M(s^{-1}, -\theta)[\ldots]$ holds, we obtain

$$dx = M((s(1+ds))^{-1}, -(\theta + d\theta))[y] - x, \quad (16)$$

where $y = M(s,\theta)[x] + dX - dX_c$. It can be concluded that these adjustments to pose and scale parameters will never be optimal, leaving residual adjustments which can only be satisfied by deforming the shape parameters.

However, it has to be ensured that the tooth model only deforms into shapes consistent with the training set. In order to apply these shape constraints, we transform dx into the parameter space of the model ("tangent space"). This transformation is needed, because dissimilarities between two shapes are not euclidean within the parameter space and therefore cannot be isometrically embedded in a euclidean space, as Wilson et al. point out in [45]. The mapping to tangent space results in db, the changes in model parameters required to adjust the landmarks as closely to dx as allowed. Using Eq. 12, we wish to find db such that

$$x + dx \approx \bar{x} + P_t(b_t + db). \tag{17}$$

Substracting Eq. 12 from Eq. 17 gives

$$dx \approx P_t db. \tag{18}$$

As the columns of P_t are orthonormal, we are able to calculate $P_t^T = P_t^{-1}$ using the Moore-Penrose pseudo-inverse ([28, 35]), and finally achieve

$$db \approx P_t^T dx. \tag{19}$$

Update the model parameters. Eq. 16 allows us to calculate changes and adjustments dX_c, $d\theta$, and ds, to the scale and pose parameters. Applying Eq. 19, we achieve the updates to the shape parameters db, to adjust the landmarks as closely to dx as allowed. We apply these changes and adjustments in an iterative scheme, such that

$$\begin{aligned} X_c &= X_c + w_t dX_c, \\ \theta &= \theta + w_\theta d\theta, \\ s &= s(1 + w_s ds), \\ b_t &= b_t + W_b db, \end{aligned} \tag{20}$$

where w_t, w_θ, and w_s are scalar weights, while W_b is a diagonal matrix of weights consisting of one weight for each mode, where we choose each weight such that it is proportional to the standard deviation of the variance of its corresponding shape parameter. This allows faster adjustments in modes showing larger shape variations, as Cootes et al. propose in [12]. In order to ensure that the tooth model only deforms into shapes consistent with its training set, we place limits on the values of b_t such that we consider a new shape unacceptable, if the Mahalonobis distance D_m from $\bar{x}$ is greater than D_{max}, such that

$$D_m = \sqrt{\sum_{k=1}^{t} \left(\frac{b_t^2}{\lambda_k} \right)} > D_{max}. \tag{21}$$

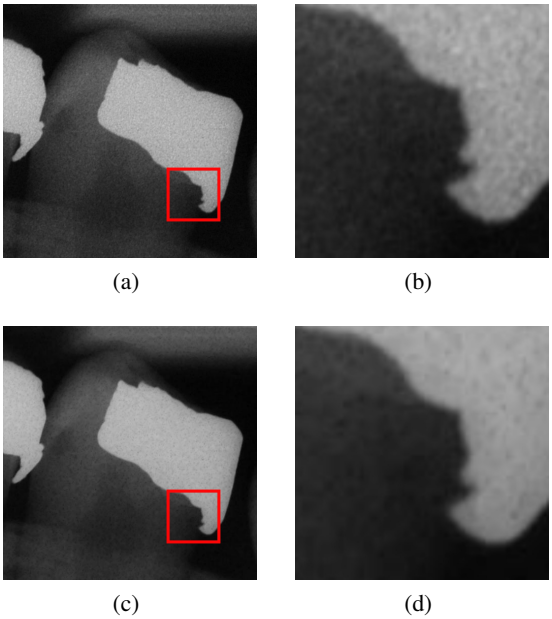
In such a case, b_t has to be rescaled in order to produce a plausible shape using

$$b_t' = b_t \left(\frac{D_{max}}{D_m} \right). \tag{22}$$

Finally, after the scale, pose and shape parameters have been updated, and limits applied where necessary, we move the landmarks from their current locations to the suggested better locations.

4. Results and Discussion

As the development of the segmentation framework that we propose in Sec. 3 is still ongoing due to erroneous results we achieve after calculating the adjustments of the model parameters, we present the results that we obtained so far. The results are evaluated using a set of intra-oral dental radiographs containing 60 molars and 70 premolars from 24 patients (22 female, 2 male), taken over a period of ten years [39], which were scanned using a resolution of 300 dots per inch (dpi) and stored as JPEG-compressed images with a bit depth of 8 bits.



(a) (b) (c) (d)

Fig. 4: Dental radiograph of a premolar. The red highlighted areas are zoomed in order to show the amount of impulsive noise present before (a, b) and after filtering (c, d).

Impulsive Noise Reduction. Fig. 4 shows the results of applying impulsive noise reduction using our proposed ACWM filter with five adaptive centre weights and a median filter incorporating a 5-by-5 neighbourhood on a dental radiograph of a premolar.

To evaluate the performance of our ACWM filter, we calculate the mean structural similarity (MSSIM) between the original and the de-noised dental radiograph. The results we achieve can be found in Tab. 1. The definition and a detailed explanation of MSSIM are given by Wang et al. in [43]. We expect our ACWM filter to perform comparable on molars and premolars (null hypothesis, H_0). Running a two-tailed Welch t-test with $\alpha = 0.05$ on our achieved MSSIM values gives $p = 2.287^{-06}$. Therefore we reason that the performance of our proposed ACWM filter is significantly lower on molars. Whether this is due to the different anatomical structure or if another filter parametrisation would have given better results was not evaluated further.

MSSIM	Min. [1]	Median [1]	Mean [1]	Max. [1]
Molar	0.5619	0.7247	0.7414	0.9059
Premolar	0.5973	0.8332	0.8181	0.9267

Tab. 1: Comparison of the MSSIM values we achieve applying our proposed ACWM filtering procedure.

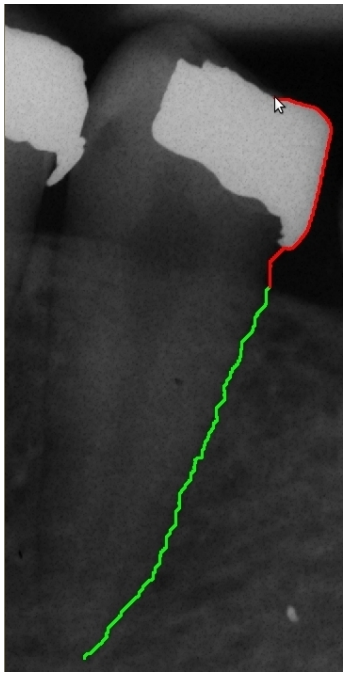


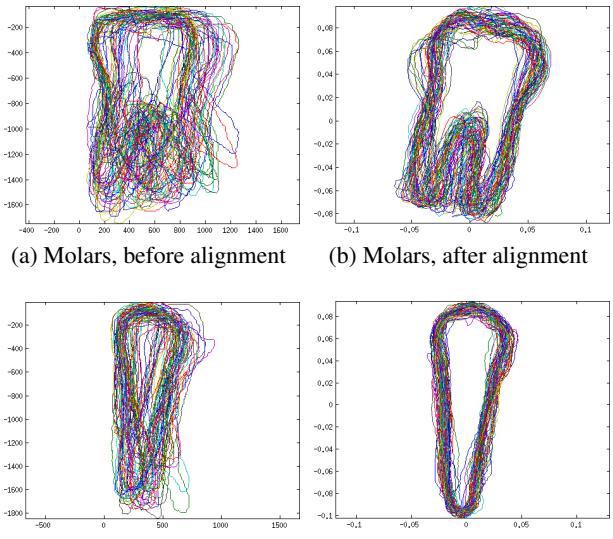
Fig. 5: Screenshot captured during segmentation of a premolar. The segmentation was started at the tip of the premolar and moved upwards in counter-clockwise direction. The green part of the boundary consists of seed points that are already "frozen", while the red part shows the current active boundary segment proposed by the Live-Wire tool.

Segmentation of Training Images. We use the implementation of the Live-Wire tool published by Hamarneh[2] et al. in [7] for segmenting the teeth needed to train our proposed teeth segmentation framework. Fig. 5 shows a screenshot captured during manual segmentation of a premolar.

Solving the Correspondence Problem. We use the MDL implementation published by Thodberg in [42] for solving the correspondence problem. We achieve a sequence of n_{lm} pseudo-landmarks placed at corresponding positions within the n_s training shapes, whose arc lengths along the contour are normalised to run from zero to one and whose centres of origin are moved to their respective centres of gravity.

	n_{lm}, [1]	n_{Iter}, [1]	D, [1]
Molar	64	3	2.265^{-06}
	128	3	2.257^{-06}
	256	3	2.265^{-06}
Premolar	64	2	2.429^{-06}
	128	2	2.545^{-06}
	256	2	2.515^{-06}

Tab. 2: Comparison of the alignment iterations and the dissimilarity measure D we achieve after applying our proposed shape alignment procedure.



(a) Molars, before alignment (b) Molars, after alignment

(c) Premolars, before alignment (d) Premolars, after alignment

Fig. 6: 60 molar and 70 premolar shapes (with 64 landmarks each) before (left) and after (right) applying our proposed shape alignment procedure.

[2]Hamarneh's Live-Wire implementation for MATLAB® is available for download at http://tinyurl.com/osdkr5h/.

Aligning a Set of Training Images. The alignment of the images needed for training our proposed teeth segmentation framework is done using the GPA approach discussed in Sec. 3. It can be concluded by looking at the results we achieve in Tab. 2 and Fig. 6 that our approach is not only fast (it does not need more than three iterations), but also produces accurately aligned shapes ($D \leq 2.75^{-06}$).

λ_k	$n_{lm} = 64$, [%]	$n_{lm} = 128$, [%]	$n_{lm} = 256$, [%]
1	57.338	57.375	57.391
2	15.348	15.305	15.281
3	8.666	8.675	8.674
4	6.651	6.677	6.666
5	3.184	3.172	3.174
6	1.727	1.720	1.721
7	1.470	1.469	1.475
8	0.830	0.826	0.825
9	0.683	0.688	0.691
$\sum \lambda_k$	95.897	95.907	95.898

(a) Molars

λ_k	$n_{lm} = 64$, [%]	$n_{lm} = 128$, [%]	$n_{lm} = 256$, [%]
1	43.111	43.026	43.082
2	27.793	27.804	27.757
3	8.316	8.373	8.366
4	4.374	4.371	4.363
5	2.978	2.975	2.969
6	2.665	2.674	2.675
7	1.824	1.825	1.830
8	1.683	1.669	1.667
9	1.094	1.090	1.089
10	0.858	0.857	0.858
11	0.626	0.624	0.627
12	0.573	0.578	0.577
$\sum \lambda_k$	95.895	95.866	95.861

(b) Premolars

Tab. 3: Percentage of the variance explained by each λ_k in order to reach 95.45% of the total variance of the captured statistics of 60 molar shapes (above) and 70 premolar shapes (below) containing 64, 128, and 256 pseudo-landmarks.

Capturing the Training Images Statistics. In order to reduce the dimensionality of our training shapes, we capture the image statistics using PCA, as discussed in Sec. 3.

It can be concluded by looking at the results in Tab. 3 that we achieve a huge data compression, as we just need nine eigenvectors in order to reach 95.45% of the total variance of the captured statistics for molars. For premolars, we need only twelve eigenvectors (and their corresponding eigenvalues).

5. Conclusion and Future Work

We presented a framework for segmentation of human teeth contours in dental radiographs using ASM as segmentation approach. We showed the necessary steps to build an ASM (removing the impulsive noise, manual segmentation of training images, solving the correspondence problem, aligning the set of training images, and capturing its statistics). Using our set of dental radiographs containing 60 molars and 70 premolars, we achieved a MSSIM of 0.7414 for molars and 0.8181 for premolars using our proposed ACWM filter. We searched for 64, 128, and 256 corresponding pseudo-landmarks within the manually segmented training images. Aligning them using our proposed GPA approach took three iterations at maximum and produced accurately aligned shapes ($D \leq 2.75^{-06}$). Finally, we were able to reduce the dimensionality of our training images by applying PCA, which resulted in nine remaining eigenvectors for molars and twelve for premolars, in order to reach 95.45% of the total variance of the captured statistics.

For image interpretation, we explained in a theoretical manner how to find shape, scale, and pose parameters, which cause an ASM to coincide with the structures of interest in the dental radiograph containing the tooth to be segmented, as this part of our framework is still in development. Finishing this task has top priority on our list of additions that are foreseen in the future. As soon as image interpretation is working as expected, we plan to incorporate the statistics of local grey levels in regions around each pseudo-landmark. More details regarding local grey levels can be found in [10, 14]. We also consider to enhance our ASM implementation with a multi-resolution approach using image pyramids similar to the one described by Cootes et al. in [13].

Acknowledgements

We would like to thank Dr. Georg D. Strbac from the Department of Oral Surgery of the BGUCD at the MUV for providing the set of dental radiographs used within this paper.

References

[1] P. D. Allen, J. Graham, D. J. J. Farnell, E. J. Harrison, R. Jacobs, K. Nicopolou-Karayianni, C. Lindh, P. F. van der Stelt, K. Horner, , and H. Devlin. A Generalized Inverse for Matrices. *IEEE Transactions on Information Technology in Biomedicine*, 11(6):601–610, November 2007. 2

[2] E. B. Barboza and A. N. Marana. A Multibiometric Approach in a Semi Automatic Dental Recognition Using DIFT Technique and Dental Shape Features. In H. Lopes and N. Hirata, editors, *Workshop of Theses and Dissertations (WTD) within the 25th Conference on Graphics, Patterns and Images (SIBGRAPI '12)*, pages 13–18. SBC, August 2012. 1

[3] E. B. Barboza, A. N. Marana, and D. T. Oliveira. Semiautomatic Dental Recognition Using a Graph–Based Segmentation Algorithm and Teeth Shapes Featuress. In A. K. Jain, A. Ross, S. Prabhakar, and J. Kim, editors, *Proceedings of the 5th IAPR International Conference on Biometrics (ICB '12)*, pages 348–353. IEEE Press, March/April 2012. 1

[4] H. Chen. *Automatic Forensic Identification based on Dental Radiographs*. PhD thesis, Michigan State University, East Lansing, MI, USA, 2007. 2

[5] H. Chen and A. K. Jain. Dental Biometrics: Alignment and Matching of Dental Radiographs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1319–1326, August 2005. 2

[6] L. Chen, M. Jiang, and J. Chen. Image Segmentation Using Iterative Watersheding Plus Ridge Detection. In *Proceedings of the 16th IEEE International Conference on Image Processing (ICIP '09)*, pages 4033–4036. IEEE Press, November 2009. 2

[7] A. Chodorowski, U. Mattsson, M. Langille, and G. Hamarneh. Color Lesion Boundary Detection using Live Wire. In *Proceedings of SPIE Medical Imaging 2005: Image Processing*, volume 5747, pages 1589–1596. SPIE, April 2005. 7

[8] T. F. Cootes, A. Hill, C. J. Taylor, and J. Haslam. The Use of Active Shape Models for Locating Structures in Medical Images. *Image and Vision Computing*, 12(6):355–365, July 1994. 2

[9] T. F. Cootes and C. J. Taylor. Active Shape Models – "Smart Snakes". In *Proceedings of the 3rd British Machine Vision Conference (BMVC '92)*, pages 266–275. Springer-Verlag, September 1992. 2

[10] T. F. Cootes and C. J. Taylor. Active Shape Model Search Using Local Grey–Level Models – A Quantitative Evaluation. In *Proceedings of the 4th British Machine Vision Conference (BMVC '93)*, pages 639–648. BMVA Press, September 1993. 8

[11] T. F. Cootes and C. J. Taylor. Statistical Models of Appearance for Medical Image Analysis and Computer Vision. In *Proceedings of SPIE Medical Imaging 2001: Image Processing*, volume 4322, pages 236–248. SPIE, February 2001. 2

[12] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active Shape Models – Their Training and Application. *Computer Vision and Image Understanding*, 61(1):38–59, January 1995. 3, 4, 5, 6

[13] T. F. Cootes, C. J. Taylor, and A. Lanitis. Multi–Resolution Search with Active Shape Models. In *Proceedings of the 12th IAPR International Conference on Pattern Recognition (ICPR '94)*, volume 1, pages 610–612. IEEE Press, October 1994. 8

[14] T. F. Cootes, C. J. Taylor, A. Lanitis, D. H. Cooper, and J. Graham. Building and Using Flexible Models Incorporating Grey–Level Information. In *Proceedings of the 4th International Conference on Computer Vision (ICCV '93)*, pages 242–246. IEEE Press, May 1993. 8

[15] R. H. Davies, T. F. Cootes, and C. J. Taylor. A Minimum Description Length Approach to Statistical Shape Modelling. In *Proceedings of the 17th International Conference on Information Processing in Medical Imaging (IPMI '01)*, pages 50–63. Springer-Verlag, June 2001. 3

[16] R. H. Davies, C. J. Twining, T. F. Cootes, J. C. Waterton, and C. J. Taylor. A minimum description length approach to statistical shape modelling. *IEEE Transactions on Medical Imaging*, 21(5):525–537, May 2002. 3

[17] A. X. Falcão, J. Stolfi, and R. de Alencar Lotufo. The Image Foresting Transform: Theory, Algorithms, and Applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(1):19–29, January 2004. 1

[18] D. Frejlichowski and R. Wanat. Extraction of Teeth Shapes from Orthopantomograms for Forensic Human Identification. In P. Real, D. Diaz-Pernil, H. Molina-Abril, A. Berciano, and W. G. Kropatsch, editors, *Computer Analysis of Images and Patterns*, volume 6855 of *Lecture Notes in Computer Science*, pages 65–72. Springer-Verlag, 2011. 2

[19] B. van. Ginneken, M. B. Stegmann, and M. Loog. Segmentation of anatomical structures in chest radiographs using supervised methods: a comparative study on a public database. *Medical Image Analysis*, 10(1):19–40, February 2006. 2

[20] J. C. Gower. Generalized Procrustes Analysis. *Psychometrika*, 40(1):33–51, March 1975. 3

[21] T. J. Hutton, S. Cunningham, and P. Hammond. An evaluation of Active Shape Models for the automatic identification of cephalometric landmarks. *European Journal of Orthodontics*, 22(5):499–508, October 2000. 2

[22] T. J. Hutton, P. Hammond, and J. C. Davenport. Active Shape Models for Customised Prosthesis Design. In *Proceedings of the 7th Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making (AIMDM '99), LNAI 1620*, pages 448–452. Springer, June 1999. 2

[23] R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice Hall, 6th edition, March 2007. 4

[24] A. C. W. Kotcheff and C. J. Taylor. Automatic Construction of Eigenshape Models by Direct Optimization. *Medical Image Analysis*, 2(4):303–314, December 1998. 3

[25] P. L. Lin, P. Y. Huang, P. W. Huang, H. C. Hsu, and C. C. Chen. Teeth Segmentation of Dental Periapical Radiographs Based on Local Singularity Analysis. *Computer Methods and Programs in Biomedicine*, 113(2):433–445, February 2014. 2

[26] T.-C. Lin. A new Adaptive Centre Weighted Median Filter for Suppressing Impulsive Noise in Images. *Information Sciences*, 177(4):1073–1087, February 2007. 3

[27] H. Marcovitch. *Black's Medical Dictionary*. A&C Black Publishers, 42nd edition, September 2009. 2

[28] E. H. Moore. On the Reciprocal of the General Algebraic Matrix. *Bulletin of the American Mathematical Society*, 9(26):394–395, September 1920. 6

[29] E. N. Mortensen and W. A. Barrett. Intelligent Scissors for Image Composition. In *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*, pages 191–198. ACM, December 1995. 3

[30] E. N. Mortensen and W. A. Barrett. Interactive Segmentation with Intelligent Scissors. *Graphical Models and Image Processing*, 60(5):349–384, September 1998. 3

[31] S. J. Nelson. *Wheeler's Dental Anatomy, Physiology and Occlusion*. Saunders, 9th edition, June 2009. 2

[32] M. G. Newman, H. Takei, F. A. Carranza, and P. R. Klokkevold. *Carranza's Clinical Periodontology*. Saunders, 10th edition, July 2006. 2

[33] O. Nomir and M. Abdel-Mottaleb. A System for Human Identification From X–ray Dental Radiographs. *Pattern Recognition*, 38(8):1295–1305, August 2005. 1

[34] A. D. Parker, A. Hill, C. J. Taylor, T. F. Cootes, X. Y. Jin, and D. G. Gibson. Application of point distribution models to the automated analysis of echocardiograms. In *Proceedings of the 21st International Conference on Computers in Cardiology (CinC '94)*, pages 25–28. IEEE Press, September 1994. 2

[35] R. Penrose. A Generalized Inverse for Matrices. *Proceedings of the Cambridge Philosophical Society*, 51(3):406–413, July 1955. 6

[36] A. L. Redhead, A. C. W. Kotcheff, C. J. Taylor, M. L. Porter, and D. W. L. Hukins. An Automated Method for Assessing Routine Radiographs of Patients with Total Hip Replacements. *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, 211(2):145–154, January 1997. 2

[37] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465–471, September 1978. 3

[38] E. H. Said, D. E. M. Nassar, and G. Fahmy. Teeth Segmentation in Digitized Dental X–Ray Films Using Mathematical Morphology. *IEEE Transactions on Information Forensics and Security*, 1(2):178–189, June 2006. 2

[39] B. Schwinner. Vertical Bone Height and Apex Growth of Autografted Teeth. Master's thesis, Medical University Vienna, Bernhard Gottlieb University Clinic for Dentistry, Department of Oral and Maxillofacial Surgery, May 2007. 6

[40] J. A. Sethian. A Fast Marching Level Set Method for Monotonically Advancing Fronts. *Proceedings of the National Academy of Sciences of the United States of America*, 93(4):1591–1595, February 1996. 2

[41] S. Solloway, C. E. Hutchinson, J. C. Waterton, and C. J. Taylor. The use of Active Shape Models for making thickness measurements of articular cartilage from MR images. *Magnetic Resonance in Medicine*, 37(6):943–952, June 1997. 2

[42] H. H. Thodberg. Minimum Description Length Shape and Appearance Models. In *Proceedings of the 18th International Conference on Information Processing in Medical Imaging (IPMI '03)*, pages 51–62. Springer-Verlag, July 2003. 7

[43] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004. 7

[44] P. A. Widhalm. Automatic Assessment of the Knee Alignment Angles on Full-limb Radiographs. Master's thesis, Vienna University of Technology, Institute of Computer Aided Automation, Pattern Recognition and Image Processing Group, October 2008. 2

[45] R. C. Wilson, E. R. Hancock, E. Pekalska, and R. P. W. Duin. Spherical and Hyperbolic Embeddings of Data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2255–2269, November 2014. 6

20^{th} *Computer Vision Winter Workshop*
Paul Wohlhart, Vincent Lepetit (eds.)
Seggau, Austria, February 9-11, 2015

Hands Deep in Deep Learning for Hand Pose Estimation

Markus Oberweger Paul Wohlhart Vincent Lepetit
Institute for Computer Graphics and Vision
Graz University of Technology, Austria
{oberweger, wohlhart, lepetit}@icg.tugraz.at

Abstract.

We introduce and evaluate several architectures for Convolutional Neural Networks to predict the 3D joint locations of a hand given a depth map. We first show that a prior on the 3D pose can be easily introduced and significantly improves the accuracy and reliability of the predictions. We also show how to use context efficiently to deal with ambiguities between fingers. These two contributions allow us to significantly outperform the state-of-the-art on several challenging benchmarks, both in terms of accuracy and computation times.

1. Introduction

Accurate hand pose estimation is an important requirement for many Human Computer Interaction or Augmented Reality tasks, and has attracted lots of attention in the Computer Vision research community [10, 11, 14, 15, 17, 22, 23, 29]. Even with 3D sensors such as structured-light or time-of-flight sensors, it is still very challenging, as the hand has many degrees of freedom, and exhibits self-similarity and self-occlusions in images.

Given the current trend in Computer Vision, it is natural to apply Deep Learning [18] to solve this task, and a Convolutional Neural Network with a standard architecture performs remarkably well when applied to this problem, as a simple experiment shows. However, the layout of the network has a strong influence on the accuracy of the output [4, 21] and in this paper, we aim at identifying the architecture that performs best for this problem.

More specifically, our contribution is two-fold:

- We show that we can learn a prior model of the hand pose and integrate it seamlessly to the network to improve the accuracy of the predicted pose. This results in a network with an unusual "bottleneck", *i.e*. a layer with fewer neurons than the last layer.
- Like previous work [21, 27], we use a refinement stage to improve the location estimates for each joint independently. Since it is a regression problem, spatial pooling and subsampling should be used carefully for this stage. To solve this problem, we use multiple input regions centered on the initial estimates of the joints, with very small pooling regions for the smaller input regions, and larger pooling regions for the larger input regions. Smaller regions provide accuracy, larger regions provide contextual information.

We show that our original contributions allow us to significantly outperform the state-of-the-art on several challenging benchmarks [22, 26], both in terms of accuracy and computation times. Our method runs at over 5000 fps on a single GPU and over 500 fps on a CPU, which is one order of magnitude faster than the state-of-the-art.

In the remainder of the paper, we first give a short review of related work in Section 2. We introduce our contributions in Section 3 and evaluate them in Section 4.

2. Related Work

Hand pose estimation is an old problem in Computer Vision, with early references from the nineties, but it is currently very active probably because of the appearance of depth sensors. A good overview of earlier work is given in [6]. Here we will discuss only more recent work, which can be divided into two main approaches.

The first approach is based on generative, model-based tracking methods. [15, 17] use a 3D hand

model and Particle Swarm Optimization to handle the large number of parameters to estimate. [14] also considers dynamics simulation of the 3D model. Several works rely on a tracking-by-synthesis approach: [5] considers shading and texture, [1] salient points, and [29] depth images. All these works require careful initialization in order to guarantee convergence and therefore rely on tracking based on the last frames' pose or separate initialization methods—for example, [17] requires the fingertips to be visible. Such tracking-based methods have difficulty handling drastic changes between two frames, which are common as the hand tends to move fast.

The second type of approach is discriminative, and aims at directly predicting the locations of the joints from RGB or RGB-D images. For example, [11] and [13] rely on multi-layered Random Forests for the prediction. The former uses invariant depth features, and the latter uses clustering in hand configuration space and pixel-wise labelling. However, both do not predict the actual 3D pose but only classify given poses based on a dictionary. Motivated by work for human pose estimation [20], [10] uses Random Forests to perform a per-pixel classification of depth images and then a local mode-finding algorithm to estimate the 2D joint locations. However, this approach cannot directly infer the locations of hidden joints, which are much more frequent for hands than for the human body.

[23] proposed a semi-supervised regression forest, which first classifies the hands viewpoint, then the individual joints, to finally predict the 3D joint locations. However, it relies on a costly pixel-wise classification, and requires a huge training database due to viewpoint quantization. The same authors proposed a regression forest in [22] to directly regress the 3D locations of the joints, using a hierarchical model of the hand. However, their hierarchical approach accumulates errors, causing larger errors for the finger tips.

Even more recently, [26] uses a Convolutional Neural Network (CNN) for feature extraction and generates small "heatmaps" for joint locations from which they infer the hand pose using inverse kinematics. However, their approach predicts only the 2D locations of the joints, and uses a depth map for the third coordinate, which is problematic for hidden joints. Furthermore, the accuracy is restricted to the heatmap resolution, and creating heatmaps is computationally costly as the CNN has to be evaluated at each pixel location.

The hand pose estimation problem is of course closely related to the human body pose estimation problem. To tackle this problem, [20] proposed per-pixel semantic segmentation and regression forests to estimate the 3D human body pose from a single depth image. [9] recently showed it was possible to do the same from RGB images only, by combined body part labelling and iterative structured-output regression for 3D joint localization. [27] recently proposed a cascade of CNNs to directly predict and iteratively refine the 2D joint locations in RGB images. Further, [25] used a CNN for part detection and a simple spatial model, which however, is not effective for high variations in pose space.

In our work, we build on the success of CNNs and use them for their demonstrated performance. We observe, that the structure of the network is very important. Thus we propose and investigate different architectures to find the most appropriate one for the hand pose estimation problem. We propose a network structure that works very well, outperforming the baselines on two difficult datasets.

3. Hand Pose Estimation with Deep Learning

In this section we present our original contributions to the hand pose estimation problem. We first briefly introduce the problem and a simple 2D hand detector, which we use to get a coarse bounding box of the hand as input to the CNN-based pose predictors.

Then we describe our general approach which consists of two stages. For the first stage we consider different architectures that predict the locations of all joints simultaneously. Optionally, this stage can predict the pose in a lower-dimensional space, which is described next. Finally, we detail the second stage, which refines the locations of the joints independently from the predictions made at the first stage.

3.1. Problem Formulation

We want to estimate the J 3D hand joint locations $\mathbf{J} = \{\mathbf{j}_i\}_{i=1}^{J}$ with $\mathbf{j}_i = (x_i, y_i, z_i)$ from a single depth image. We assume that a training set of depth images labeled with the 3D joint locations is available. To simplify the regression task, we first estimate a coarse 3D bounding box containing the hand using a simple method similar to [22], by assuming the hand is the closest object to the camera: We extract from

the depth map a fixed-size cube centered on the center of mass of this object, and resize it to a 128×128 patch of depth values normalized to $[-1, 1]$. Points for which the depth is not available—which may happen with structured light sensors for example—or are deeper than the back face of the cube, are assigned a depth of 1. This normalization is important for the CNN in order to be invariant to different distances from the hand to the camera.

3.2. Network Structures for Predicting the Joints' 3D Locations

We first considered two standard CNN architectures. The first one is shown in Fig. 1a, and is a simple shallow network, which consists of a single convolutional layer, a max-pooling layer, and a single fully-connected hidden layer. The second architecture we consider is shown in Fig. 1b and is a deeper but still generic network [12, 27], with three convolutional layers followed by max-pooling layers and two fully-connected hidden layers. All layers use Rectified Linear Unit (ReLU) [12] activation functions.

Additionally, we evaluated a multi-scale approach, as done for example in [7, 19, 25]. The motivation for this approach is that using multiple scales may help capturing contextual information. It uses several downscaled versions of the input image as input to the network, as shown in Fig. 1c.

Our results will show that, unsurprisingly, the multi-scale approach performs better than the deep architecture, which performs better than the shallow one. However, our contributions, described in the next two sections, bring significantly more improvement.

3.3. Enforcing a Prior on the 3D Pose

So far we only considered predicting the 3D positions of the joints directly. However, given the physical constraints over the hand, there are strong correlation between the different 3D joint locations, and previous work [28] has shown that a low dimensional embedding is sufficient to parameterize the hand's 3D pose. Instead of directly predicting the 3D joint locations, we can therefore predict the parameters of the pose in a lower dimensional space. As this enforces constraints of the hand pose, it can be expected that it improves the reliability of the predictions, which will be confirmed by our experiments.

As shown in Fig. 1d, we implement the pose prior into the network structure by introducing a "bottleneck" in the last layer. This bottleneck is a layer with less neurons than necessary for the full pose representation, *i.e.* $\ll 3 \cdot J$. It forces the network to learn a low dimensional representation of the training data, that implements the physical constraints of the hand. Similar to [28], we rely on a linear embedding. The embedding is enforced by the bottleneck layer and the reconstruction from the embedding to pose space is integrated as a separate hidden layer added on top of the bottleneck layer. The weights of the reconstruction layer are set to compute the back-projection into the $3 \cdot J$-dimensional joint space. The resulting network therefore directly computes the full pose. We initialize the reconstruction weights with the major components from a Principal Component Analysis (PCA) of the hand pose data and then train the full network using back-propagation. Using this approach we train the networks described in the previous section.

The embedding can be as small as 8 dimensions for a 42-dimensional pose vector to fully represent the 3D pose as we show in the experiments.

3.4. Refining the Joint Location Estimates

The previous architectures provide estimates for the locations of all the joints simultaneously. As done in [21, 27], these estimates can then be refined independently.

Spatial context is important for this refinement step to avoid confusion between the different fingers. The best performing architecture we experimented with is shown in Fig. 2a. We will refer to this architecture as *ORRef*, for Refinement with Overlapping Regions. It uses as input several patches of different sizes but all centered on the joint location predicted by the first stage. No pooling is applied to the smallest patch, and the size of the pooling regions then increases with the size of the patch. The larger patches provide more spatial context, whereas the absence of pooling on the small patch enables better accuracy.

We also considered a standard CNN architecture as a baseline, represented in Fig. 1b, which relies on a single input patch. We will refer to this baseline as *StdRef*, for Refinement with Standard Architecture.

To further improve the accuracy of the location estimates, we iterate this refinement step several times, by centering the network on the location predicted at the previous iteration.

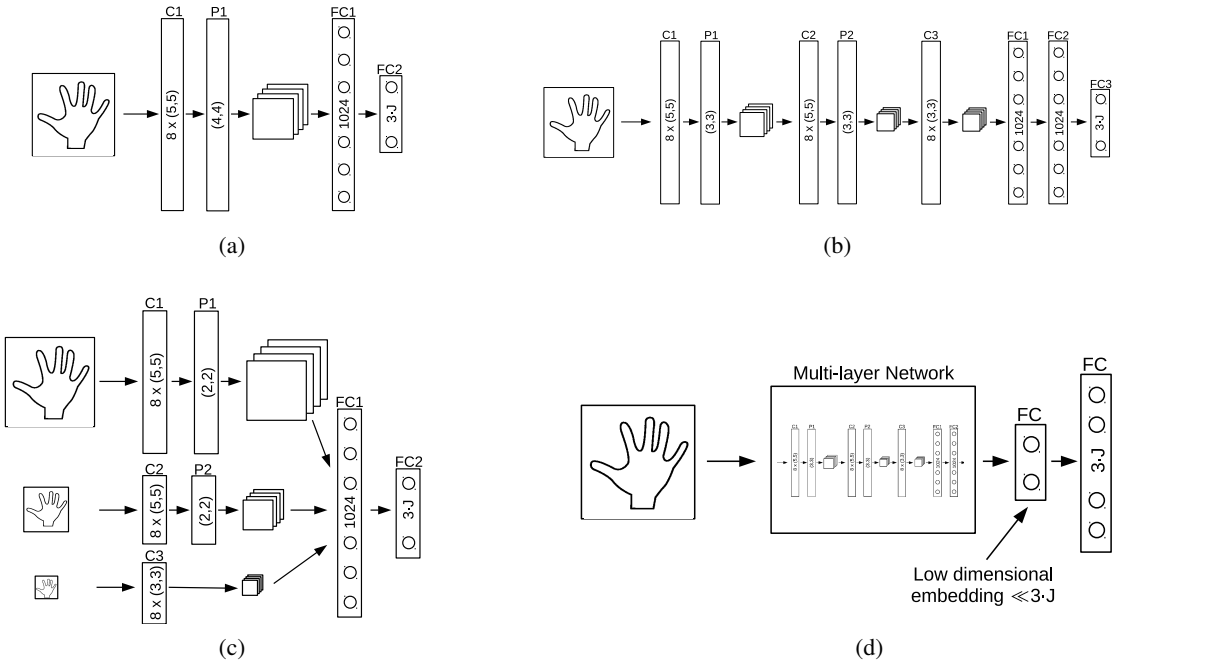


Figure 1: Different network architectures for the first stage. C denotes a convolutional layer with the number of filters and the filter size inscribed, FC a fully connected layer with the number of neurons, and P a max-pooling layer with the pooling size. We evaluated the performance of a shallow network (a) and a deeper network (b), as well as a multi-scale architecture (c), which was used in [7, 19]. This architecture extracts features after downscaling the input depth map by several factors. (d) All these networks can be extended to incorporate the constrained pose prior. This causes an unusual bottleneck with less neurons than the output layer.

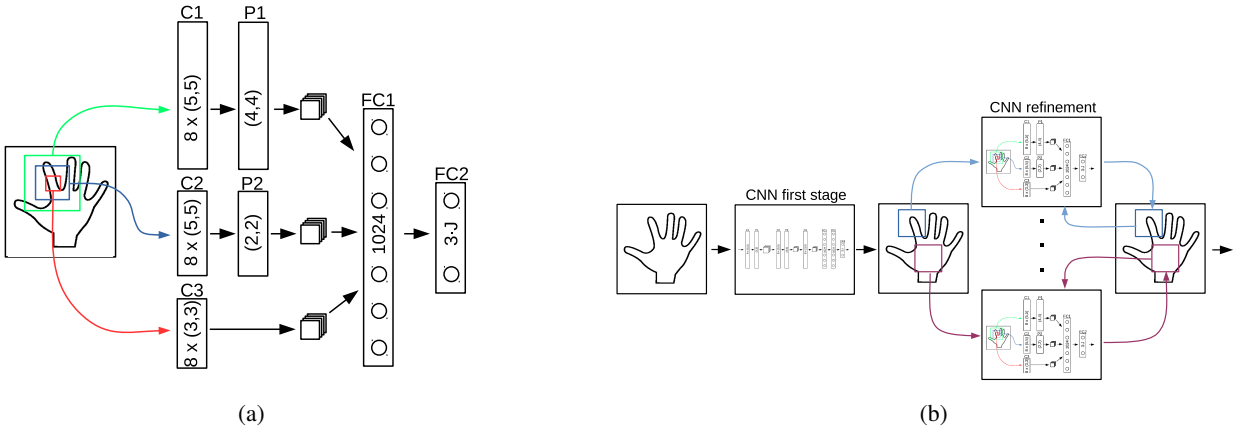


Figure 2: Our architecture for refining the joint locations during the second stage. We use a different network for each joint, to refine its location estimate as provided by the first stage. (a) The architecture we propose uses overlapping inputs centered on the joint to refine. Pooling with small regions is applied to the smaller inputs, while the larger inputs are pooled with larger regions. The smaller inputs allow for higher accuracy, the larger ones provide contextual information. We experimentally show that this architecture is more accurate than a more standard network architecture. (b) shows a generic architecture of an iterative refinement, where the output of the previous iteration is used as input for the next. As for Fig. 1, C denotes a convolutional layer, FC a fully connected layer, and P a max-pooling layer. (Best viewed in color)

4. Evaluation

In this section we evaluate the different architectures introduced in the previous section on several challenging benchmarks. We first introduce these benchmarks and the parameters of our methods. Then we describe the evaluation metric, and finally we present the results, quantitatively as well as qualitatively. Our results show that our different contributions significantly outperform the state-of-the-art.

4.1. Benchmarks

We evaluated our methods on the two following datasets:

NYU Hand Pose Dataset [26]: This dataset contains over 72k training and 8k test frames of RGB-D data captured using the Primesense Carmine 1.09. It is a structured light-based sensor and the depth maps have missing values mostly along the occluding boundaries as well as noisy outlines. For our experiments we use only the depth data. The dataset has accurate annotations and exhibits a high variability of different poses. The training set contains samples from a single user and the test set samples from two different users. The ground truth annotations contain $J = 36$ joints, however [26] uses only $J = 14$ joints, and we did the same for comparison purposes.

ICVL Hand Posture Dataset [22]: This dataset comprises a training set of over 180k depth images showing various hand poses. The test set contains two sequences with each approximately 700 depth maps. The dataset is recorded using a time-of-flight Intel Creative Interactive Gesture Camera and has $J = 16$ annotated joints. Although the authors provide different artificially rotated training samples, we only use the genuine 22k. The depth images have a high quality with hardly any missing depth values, and sharp outlines with little noise. However, the pose variability is limited compared to the NYU dataset. Also, a relatively large number of samples both from the training and test sets are incorrectly annotated: We evaluated the accuracy and about 36% of the poses from the test set have an annotation error of at least 10 mm.

4.2. Meta-Parameters and Optimization

The performance of neural networks depends on several meta-parameters, and we performed a large number of experiments varying the meta-parameters for the different architectures we evaluated. We report here only the results of the best performing sets of meta-parameters for each method. However, in our experiments, the performance depends more on the architecture itself than on the values of the meta-parameters.

We trained the different architectures by minimizing the distance between the prediction and the expected output per joint, and a regularization term for weight decay to prevent over-fitting, where the regularization factor is 0.001. We do not differ between occluded and non-occluded joints. Because the annotations are noisy, we use the robust Huber loss [8] to evaluate the differences. The networks are trained with back-propagation using Stochastic Gradient Descent [3] with a batch size of 128 for 100 epochs. The learning rate is set to 0.01 and we use a momentum of 0.9 [16].

4.3. Evaluation Metrics

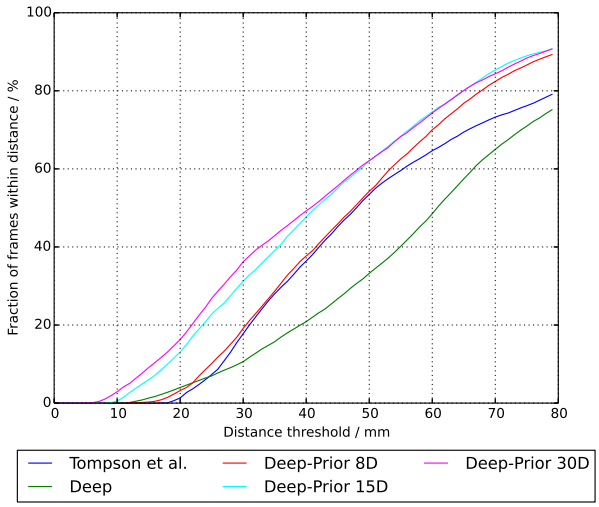
We use two different evaluation metrics:

- the average Euclidean distance between the predicted 3D joint location and the ground truth, and
- the fraction of test samples that have all predicted joints below a given maximum Euclidean distance from the ground truth, as was done in [24]. This metric is generally regarded very challenging, as a single dislocated joint deteriorates the whole hand pose.

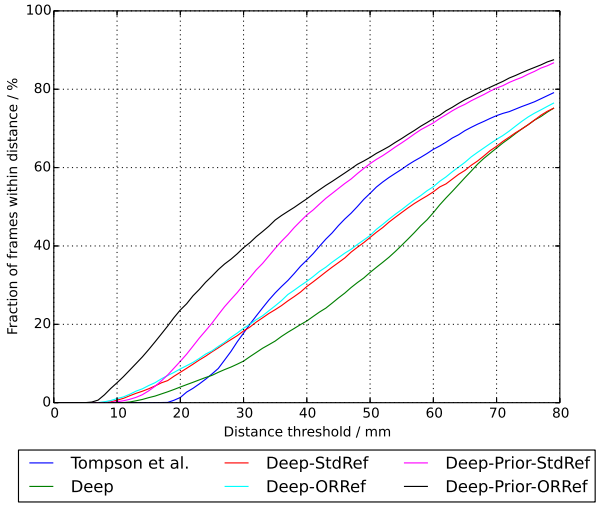
4.4. Importance of the Pose Prior

In Fig. 3a and 3c we compare different embedding dimensions and direct regression in the full $3 \cdot J$-dimensional pose space for the NYU and the ICVL dataset, respectively. The evaluation on both datasets shows that enforcing a pose prior is beneficial compared to direct regression in the full pose space. Only 8 dimensions out of the original 42- or 48-dimensional pose spaces are already enough to capture the pose and outperform the baseline on both datasets. However, the 30-dimensional embedding performs best, and thus we use this for all further evaluations. The results on the ICVL dataset, which has noisy annotations, are not as drastic, but still consistent with the results on the NYU dataset.

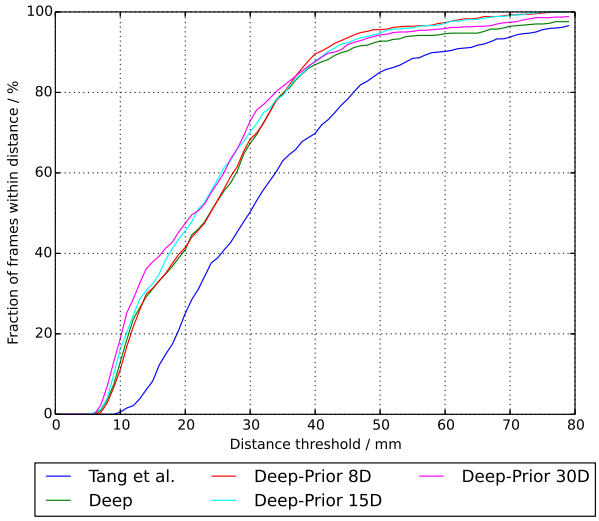
The baseline on the NYU dataset of Tompson *et al*. [26] only provide the 2D locations of the joints. For comparison, we follow their protocol and augment their 2D locations by taking the depth of each joint directly from the depth maps to derive comparable 3D locations. Depth values that do not lie within the hand cube are truncated to the cube's back face to avoid large errors. This protocol, however, has a certain influence on the error metric, as evident in Fig. 4a. The augmentation works well for some joints, as apparent by the average error. However, it is unlikely that the augmented depth is correct for



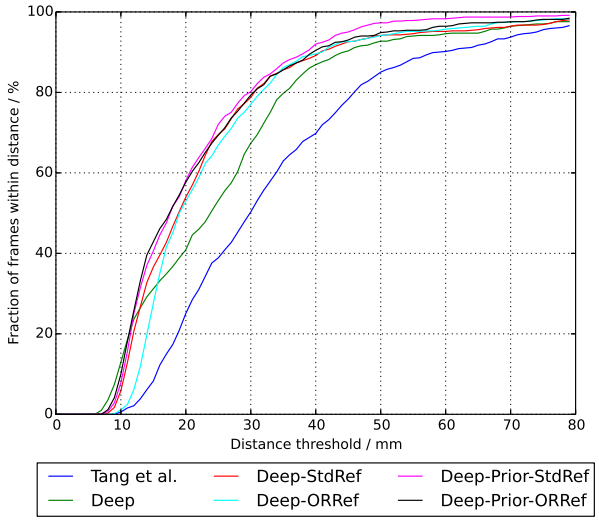
(a) Pose Prior on NYU dataset



(b) Refinement on NYU dataset



(c) Pose Prior on ICVL dataset



(d) Refinement on ICVL dataset

Figure 3: Importance of the pose prior (left) and the refinement stage (right). We evaluate the fraction of frames where all joints are within a maximum distance for different approaches. A higher area under the curve denotes more accurate results. **Left (a), (c):** We show the influence of the dimensionality of the pose embedding. The optimal value is around 30, but using only 8 dimensions performs already very well. The pose prior allows us to significantly outperform the state-of-the-art, even before the refinement step. **Right (b), (d):** We show that our architecture with overlapping input patches, denoted by the *ORRef* suffix, provides higher accuracy for refining the joint positions compared to a standard deep CNN, denoted by the *StdRef* suffix. For the baseline of Tompson *et al*. [26] we augment their 2D joint locations with the depth from the depth maps, as done by [26], and depth values that do not lie within the hand cube are truncated to the cube's back face to avoid large errors. (Best viewed on screen)

all joints of the hand, *e.g*. the 2D joint location lies on the background or is self-occluded, thus causing higher errors for individual joints. When using the evaluation metric of [24], where all joints have to be within a maximum distance, this outlier has a strong influence, in contrast to the evaluation of the average error, where an outlier can be insignificant for the mean. Thus we outperform the baseline more significantly for the distance threshold than for the average error.

4.5. Increasing Accuracy with Pose Refinement

The refinement stage can be used to further increase the location accuracy of the predicted joints. We achieved the highest accuracy by using our CNN with constrained prior hand model as first stage, and

then applying the second iterative refinement stage with our CNN with overlapping input patches, denoted *ORRef*.

The results in Fig. 3b, 3d and 4 show that applying the refinement improves the location accuracy for different base CNNs. From rather inaccurate initial estimates, as provided by the standard deep CNN, our proposed ORRef performs only slightly better than refinement with the baseline deep CNN, denoted by *StdRef*. This is because for large initial errors only the larger input patch provides enough context for reasoning about the offset. The smaller input patch cannot provide any information if the offset is bigger than the patch size. For more accurate initial estimates, as provided by our deep CNN with pose prior, the ORRef takes advantage from the small input patch which does not use pooling for higher accuracy. We iterate our refinement two times, since iterating more often does not provide any further increase in accuracy.

We would like to emphasize that our results on the ICVL dataset, with an average accuracy below 10 mm, already scratch at the uncertainty of the labelled annotations. As already mentioned, the ICVL dataset suffers from inaccurate annotations, as we show in some qualitative samples in Fig. 5 first and fourth column. While this has only a minor effect on training, the evaluation is more affected. We evaluated the accuracy of the test sequence by revising the annotations in image space and calculated an average error of 2.4 mm with a standard deviation of 5.2 mm.

4.6. Running Times

Table 1 provides a comparison of the running times of the different methods, both on CPU and GPU. They were measured on a computer equipped with an Intel Core i7, 16GB of RAM, and an nVidia GeForce GTX 780 Ti GPU. Our methods are implemented in Python using the Theano library [2], which offers an option to select between the CPU and the GPU for evaluating CNNs. Our different models perform very fast, up to over 5000 fps on a single GPU. Training takes about five hours for each CNN. The deep network with pose prior performs very fast and outperforms all other methods in terms of accuracy. However, we can further refine the joint locations at the cost of higher runtime.

4.7. Qualitative Results

We present qualitative results in Fig. 5. The typical problems of structured light-based sensors, such as missing depth, can be problematic for accurate localization. However, only partially missing parts, as shown in the third and fourth columns for example, do not significantly deteriorate the result. The location of the joint is constrained by the learned hand model. If the missing regions get too large, as shown in the fifth column, the accuracy gets worse. However, because of the use of the pose subspace embedding, the predicted joint locations still preserve the learned hand topology. The erroneous annotations of the ICVL dataset deteriorate the results, as our predicted locations during the first stage are sometimes more accurate than the ones obtained during the second stage: see for example the pinky in the first or fourth column.

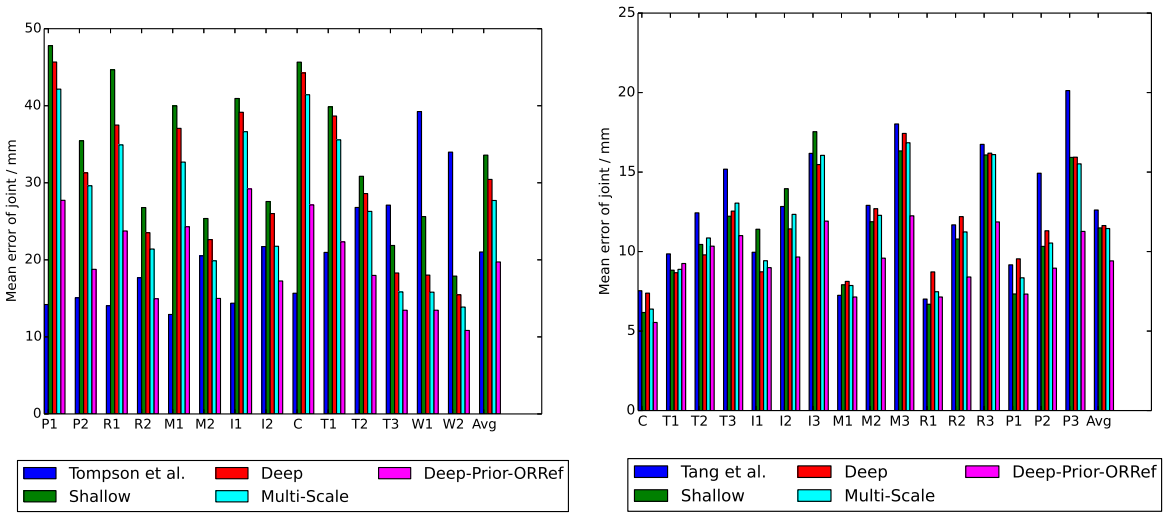
Architecture	GPU	CPU
Shallow	0.07 ms	1.85 ms
Deep [12]	0.1 ms	2.08 ms
Multi-Scale [7]	0.81 ms	5.36 ms
Deep-Prior	0.09 ms	2.29 ms
Refinement	2.38 ms	62.91 ms
Tompson *et al*. [26]	5.6 ms	-
Tang *et al*. [22]	-	16 ms

Table 1: Comparison of different runtimes. Our CNN with pose prior (*Deep-Prior*) is faster by a magnitude compared to the other methods (pose estimation only). We can further increase the accuracy using the refinement stage, still at competitive speed. All of the denoted baselines use state-of-the-art hardware comparable to ours.

5. Conclusion

We evaluated different network architectures for hand pose estimation by directly regressing the 3D joint locations. We introduced a constrained prior hand model that can significantly improve the joint localization accuracy. Further, we applied a joint-specific refinement stage to increase the localization accuracy. We have shown, that for this refinement a CNN with overlapping input patches with different pooling sizes can benefit from both, input resolution and context. We have compared the architectures on two datasets and shown that they outperform previous state-of-the-art both in terms of localization accuracy and speed.

Acknowledgements: This work was funded by the Christian Doppler Laboratory for Handheld Augmented Reality and the TU Graz FutureLabs fund.



(a) NYU dataset (b) ICVL dataset

Figure 4: Average joint errors. For completeness and comparability we show the average joint errors, which are, however, not as decisive as the evaluation in Fig. 3. Though, the results are consistent. The evaluation of the average error is more tolerant to larger errors of a single joint, which deteriorate the pose as for Fig. 3, but are insignificant for the mean if the other joints are accurate. Our proposed architecture *Deep-Prior-ORRef*, the constrained pose CNN with refinement stage, provides the highest accuracy. For the ICVL dataset, the simple baseline architectures already outperform the baseline. However, they cannot capture the higher variations in pose space and noisy images of the NYU dataset, where they perform much worse. The palm and fingers are indexed as C: palm, T: thumb, I: index, M: middle, R: ring, P: pinky, W: wrist. (Best viewed on screen)

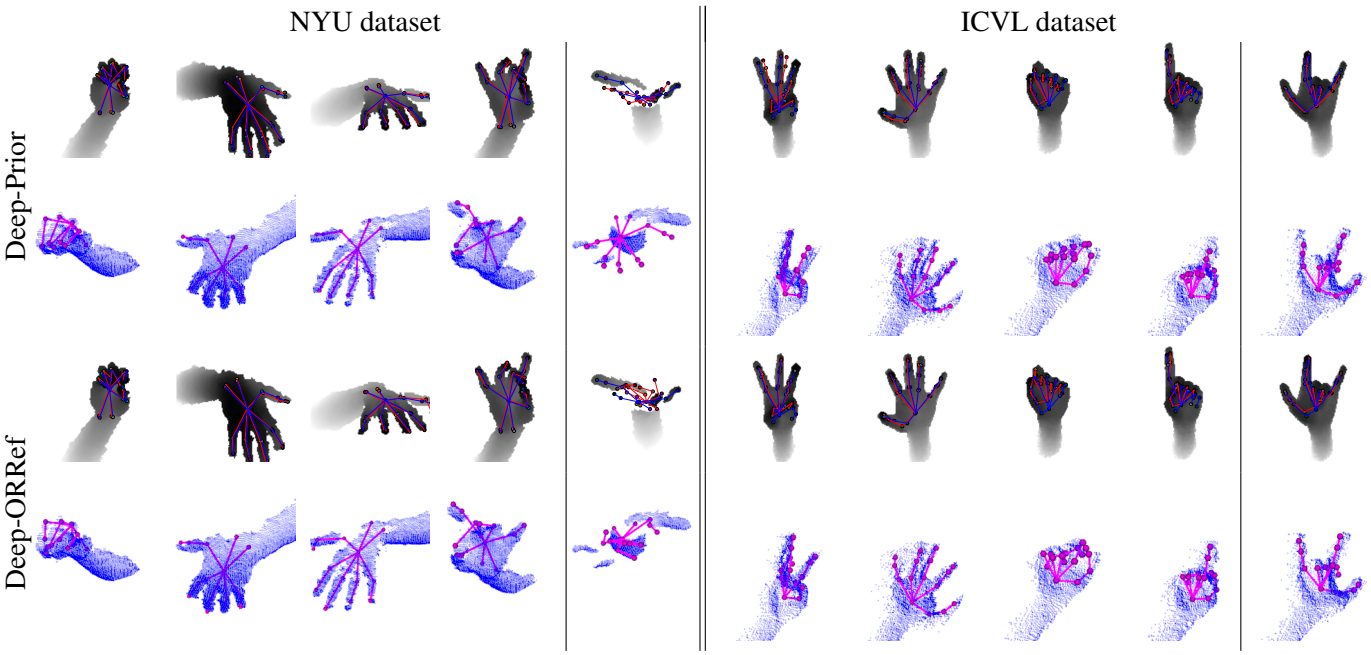


Figure 5: Qualitative results. We show the inferred joint locations on the depth images (in gray-scale), as well as the 3D locations with the point cloud of the hand (blue images) from a different angle. The ground truth is shown in blue, our results in red. The point cloud is only annotated with our results for clarity. The right columns show some erroneous results. One can see the difference between the global constrained pose and the local refinement, especially in the presence of missing depth values as shown in the fifth column. While the global pose constraint still preserves the hand topology, the local refinement cannot reason about the locations without the missing depth data. (Best viewed on screen)

References

[1] L. Ballan, A. Taneja, J. Gall, L. V. Gool, and M. Pollefeys. Motion Capture of Hands in Action Using Discriminative Salient Points. In *European Conference on Computer Vision*, 2012. 2

[2] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: A CPU and GPU Math Expression Compiler. In *Proc. of SciPy*, 2010. 7

[3] L. Bottou. Large-Scale Machine Learning with Stochastic Gradient Descent. In *Proc. of COMPSTAT*, 2010. 5

[4] A. Coates, A. Y. Ng, and H. Lee. An Analysis of Single-Layer Networks in Unsupervised Feature Learning. In *Proc. of AISTATS*, 2011. 1

[5] M. de La Gorce, D. J. Fleet, and N. Paragios. Model-Based 3D Hand Pose Estimation from Monocular Video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9), 2011. 2

[6] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly. Vision-Based Hand Pose Estimation: A Review. *Computer Vision and Image Understanding*, 108(1-2), 2007. 1

[7] C. Farabet, C. Couprie, L. Najman, and Y. LeCun. Learning Hierarchical Features for Scene Labeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013. 3, 4, 7

[8] P. J. Huber. Robust Estimation of a Location Parameter. *Annals of Statistics*, 53, 1964. 5

[9] C. Ionescu, J. Carreira, and C. Sminchisescu. Iterated Second-Order Label Sensitive Pooling for 3D Human Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2014. 2

[10] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun. Real Time Hand Pose Estimation Using Depth Sensors. In *International Conference on Computer Vision*, 2011. 1, 2

[11] C. Keskin, F. Kıraç, Y. E. Kara, and L. Akarun. Hand Pose Estimation and Hand Shape Classification Using Multi-Layered Randomized Decision Forests. In *European Conference on Computer Vision*, 2012. 1, 2

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*, 2012. 3, 7

[13] A. Kuznetsova, L. Leal-taixe, and B. Rosenhahn. Real-Time Sign Language Recognition Using a Consumer Depth Camera. In *International Conference on Computer Vision*, 2013. 2

[14] S. Melax, L. Keselman, and S. Orsten. Dynamics Based 3D Skeletal Hand Tracking. In *Proc. of Graphics Interface Conference*, 2013. 1, 2

[15] I. Oikonomidis, N. Kyriazis, and A. A. Argyros. Full DOF Tracking of a Hand Interacting with an Object by Modeling Occlusions and Physical Constraints. In *International Conference on Computer Vision*, 2011. 1

[16] B. T. Polyak. Some Methods of Speeding Up the Convergence of Iteration Methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5), 1964. 5

[17] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and Robust Hand Tracking from Depth. In *Conference on Computer Vision and Pattern Recognition*, 2014. 1, 2

[18] J. Schmidhuber. Deep Learning in Neural Networks: An Overview. Technical Report 03-14, IDSIA, 2014. 1

[19] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated Recognition, Localization and Detection Using Convolutional Networks. In *Proc. of ICRL*, 2014. 3, 4

[20] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake. Efficient Human Pose Estimation from Single Depth Images. In *Conference on Computer Vision and Pattern Recognition*, 2011. 2

[21] Y. Sun, X. Wang, and X. Tang. Deep Convolutional Network Cascade for Facial Point Detection. In *Conference on Computer Vision and Pattern Recognition*, 2013. 1, 3

[22] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim. Latent Regression Forest: Structured Estimation of 3D Articulated Hand Posture. In *Conference on Computer Vision and Pattern Recognition*, 2014. 1, 2, 5, 7

[23] D. Tang, T. Yu, and T. Kim. Real-Time Articulated Hand Pose Estimation Using Semi-Supervised Transductive Regression Forests. In *International Conference on Computer Vision*, 2013. 1, 2

[24] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The Vitruvian Manifold: Inferring Dense Correspondences for One-Shot Human Pose Estimation. In *Conference on Computer Vision and Pattern Recognition*, 2012. 5, 6

[25] J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation. In *Advances in Neural Information Processing Systems*, 2014. 2, 3

[26] J. Tompson, M. Stein, Y. LeCun, and K. Perlin. Real-Time Continuous Pose Recovery of Human Hands Using Convolutional Networks. *ACM Transactions on Graphics*, 33, 2014. 1, 2, 5, 6, 7

[27] A. Toshev and C. Szegedy. DeepPose: Human Pose Estimation via Deep Neural Networks. In *Confer-*

ence on Computer Vision and Pattern Recognition, 2014. 1, 2, 3

[28] Y. Wu, J. Lin, and T. Huang. Capturing Natural Hand Articulation. In *International Conference on Computer Vision*, 2001. 3

[29] C. Xu and L. Cheng. Efficient Hand Pose Estimation from a Single Depth Image. In *International Conference on Computer Vision*, 2013. 1, 2

20^{th} *Computer Vision Winter Workshop*
Paul Wohlhart, Vincent Lepetit (eds.)
Seggau, Austria, February 9-11, 2015

Biometry from surveillance cameras – forensics in practice

Borut Batagelj
Faculty of Computer and Information Science, University of Ljubljana
Večna pot 113, 1000 Ljubljana, Slovenia
borut.batagelj@fri.uni-lj.si

Franc Solina
Faculty of Computer and Information Science, University of Ljubljana
Večna pot 113, 1000 Ljubljana, Slovenia
franc.solina@fri.uni-lj.si

Abstract. *The article recounts various problems that the authors encountered in biometric face recognition and biometric image interpretation in their experience as court appointed expert witnesses. Before automated face recognition system can be applied on a typical surveillance video, images must be enhanced using various image processing methods or enriched by using computer vision 3D reconstruction methods. Authenticity of video material must also sometimes be verified. If face recognition is not possible or successful then other soft biometric characteristics can be checked. A legal expert witness for image biometry must be able to employ a large array of image processing and computer vision tools and methods. The expert witness must be able to explain how the biometric results were obtained, which were the necessary processing steps and how confident are the final results.*

1. Introduction

The multitude of image and video recording devices ranging from smart phones to the ever more extensive networks of video surveillance cameras produces a massive amount of imagery. Video surveillance is becoming ubiquitous in public and even private spaces. Therefore, the number of cases investigated by law enforcement, which have left some image related traces, is sharply increasing. When such cases subsequently enter some legal process, the need for expert witnesses with a working knowledge of image processing and computer vision is obvious. Interpreation of various security incidents recorded on video clips and photographs is beside the interpretation of material traces (fingerprints, bodily fluids etc.) gaining a steadily more central role in the judicial proceedings.

To correctly and independently interpret that imagery, expert witnesses are needed that can independently evaluate and interpret images. Often the main goal of such video interpretation is to identify or confirm the identity of a person. Researchers from our group have served now for several years as court appointed expert witnesses for interpretation of image and video material. In this article we would like to relate some useful experience from our practice. We discuss in the article only images where persons appear so that the tasks of the expert witness can therefore be consigned to biometry.

The tasks of an image biometry expert witness are much broader than just running a face recognition program [7]. Even if towards the end of the interpretation a face recognition system is used, several other actions on the image data must precede that step.

Since images are often recorded in suboptimal conditions, image enhancement methods must usually be applied (exposure adjustment, contrast improvement, noise filtering, stabilization of video etc.). The faces are often not captured frontally but from the side or from above so that standard frontal face recognition can not be applied directly. Building of a 3D face model is then usually attempted if images of the face from several views are available [3, 11]. Another problem can be a large age difference between the face images that we intend to compare and therefore some compensation for age related

changes must be used. Another practical problem for comparison can be injury related abrasions or tattoos on the face of suspects. If face recognition can not be applied, then some other soft biometric features could sometimes be recovered. A person's height, for example, can be reconstructed even from a single image if enough other geometric information is available in the same image [5].

Sometimes the question of authenticity of the image material can arise. Has somebody tampered with the imagery to change the content of the material? There is a whole expertise area of image forgery detection, ranging from analysis of individual image elements, analysis of the image format, analysis of the input device and finally analysis of the physical and geometrical properties of the captured scene [6].

There are commercial software solutions that cover almost the entire range of forensic tasks [1]. However, the whole area of image biometry is moving so fast that a collection of different software tools, even open source tools, is often more flexible and usable. In any case, an image biometric expert witness must understand when and why certain processing steps or methods should be applied. A court appointed expert witness, in particular, must be able to understand and explain the whole process how he obtained and verified the results.

In this article we discuss only problems related to person identification using different biometric characteristics that we encountered during the past several years as court appointed expert witnesses. By discussing these cases we would like to illustrate the variety of biometric problems encountered in practice and the need to apply methods from a large range of different research results.

2. Face recognition

The most often posed question, that a court appointed expert witness is confronted with, is whether the accused person is really on the examined video clip? This is the problem of person verification. Usually, the expert witness has at his disposal a three-part mug shot from the police records and a video clip from a surveillance camera. Normally, only the face is used for identification. The courts expect that any face comparison should include a careful analysis of individual facial features and distances among them. We will discuss now the most common problems from practice.

2.1. Problems from practice

2.1.1 Poor image quality

Often the video quality of recordings from surveillance cameras is very poor due to low resolution and high compression rates. Such setting are usually chosen to save memory space on recording devices and only rarely due to the initial poor quality of the video signal itself. To save space, some surveillance systems are saving just a limited number of images per second or just images where some movement was detected. Due to all these circumstances, the quality of the video material is on numerous instances so poor that the application of advanced methods for face recognition that are based on facial features or on the integral face appearance is not possible [2].

2.1.2 Small scale face regions

A similar problem in digital face forensics, as poor image quality, is the insufficient size of the face region. The minimal interocular distance for reliable face recognition should be at least 32 pixels. Ideally, the interocular distance should be about 70 pixels. In practice, we often encounter images with a small resolution of 320×240 or 640×480 pixels where the face is furthermore often recorded from a larger distance. On such images the face region might have a size of only 15×15 pixels with interocular distance of mere 8 pixels. Even if the face is well illuminated and in frontal orientation, the success rate of face recognition systems is in such cases very low.

2.1.3 Non-frontal face orientation

Faces on surveillance video are often recorded from above and/or from the side so that the recorded face orientation is not frontal. Persons involved in criminal activity in addition try to evade the surveillance cameras and they tend to never look into the camera. All these circumstances add up to the fact that in the whole video recording of an event there is not even a single frontal image of a face. The faces of persons on such video footage are often partially concealed by sunglasses, hoods or caps which makes face recognition based on facial features even more demanding.

2.2. Possible solutions

Due to all the problems with image quality and face orientation described above, we try to use in

such cases facial features that stand out even in images of poor quality. Such features are the shape of the head, the shape of the chin, the shape of the cheeks, the shape of the hairline and the baldness area, hair color, the shape of ears, nose and the size of the mouth. For recognition it can be beneficial also some irregularities or past injuries of the suspected person. Cases such as a nose deformation, a feature on the front of the adult men's neck (Adam's apple), excessive baldness or a prominent nose, all facilitate recognition. Facial or other visible tattoos can be very usable features for recognition even in images of very poor quality or resolution since they tend to stand out from the background of the skin color quite well.

The familiar feature on the front of the neck that is the forward protrusion of the thyroid cartilage.

2.2.1 Use of a face profile

When we try to analyze facial features on a video recording, it can turn out that the face profile is the most useful face orientation, because in the profile, certain features such as the shape of the nose stand out. As mentioned above, faces on surveillance video are captured from different often atypical viewpoints. This circumstance must be taken into account also in the case of profile views. If a suspect is available, the court can demand photos of the suspected person taken under different viewpoints, similar to those on the surveillance video.

Figure 1. The silhouette of a person in front of an ATM.

The face profile is often usable in surveillance video from ATMs (Automated Teller Machines) where the face is normally backlit, making the face dark on a bright background. Although changing the exposure can help sometimes, often individual face features can not be made visible. Since a person in front of an ATM, who is performing an illegal activity, often looks around, his face profile is usually captured as a silhouette. Such silhouette can serve as a reference image for recognition from profile (Fig. 1).

2.2.2 Use of existing face recognition systems

Despite all the above described problems with different views and poor quality of video recordings, automated face recognition methods for frontal face recognition and from face sketches can be used. Before using such a method or a system, the input face image must be adjusted. Also, the results must be accordingly interpreted. To use a system for frontal face recognition, a 3D model of the corresponding face must be constructed from several viewpoints. The 3D model of the face is then used to generate the frontal view of that face which can subsequently be used as an input image for frontal face recognition [11].

A face recognition system can be used for comparison also on semi rotated faces, however, such a system must also be trained on similarly rotated faces. Another way of using existing systems is by drawing a face sketch or constructing a facial composite, based on the recorded video, and feed the resulting face to a face recognition system which can interpret also sketches [8].

Often offenders who are caught in the act are also suspects for other, similar, but unaccounted offenses. In such cases, the investigation needs to determine if two suspects are similar to each other. Systems for automatic face recognition are for such tasks also very useful.

3. Identification using other biometric features

Since face recognition is often not possible or not reliable enough, other personal features recorded in the surveillance video should be analyzed to help in the identification of a person. We will discuss the physical features of a person and his behavior. The following bodily features can greatly reduce the circle of suspects: body height [5], way of walking [9], way of handling objects and the actual body shape of a person. Most commonly, one tries to establish the body height of persons captured on surveillance video.

3.1. Estimation of body height

To determine the body height of a person on an image the Single View Metrology (SVM) can be used

[5]. Before applying the SVM method the image should be enhanced by increasing the contrast, improving the exposure and enhancing the edges. If several images of the same static scene captured under different illumination conditions are available the image can be improved by averaging the images similar as in high-dynamic-range imaging in order to sharpen and enhance the edges of the static objects on the scene. It is easier to derive the inherent geometric information (i.e. calibrate the space in all three dimensions (x, y, z)) from such enhanced images (Fig. 2).

Figure 2. Calibration of a room should be performed after image distortions are corrected. The figure of the person in the corresponding video is never seen in its entirety. To be able to establish the height of the person, comparison to other calibrated lengths in the image is used.

Before calibration image distortions must be corrected so that objects on the image are correctly displayed. For calibration of the depicted space, portrayal of several rectangular objects aligned with the walls of the space is essential. Very useful for the calibration of the $x - y$ plane are for example quadratic plates in the floor paving. To determine the heights, the vertical axis z must be calibrated also. For this task one can use door and window frames or other vertical objects standing in the room (Fig. 2). Sometimes, if a room was rearranged in between, it is difficult to find a suitable reference object. Usually doors and windows are the most stable features of a room since they are seldom changed.

Sometimes, the video surveillance system was also changed in between. To perform a crime reconstruction or to determine the height of objects, images from the new system must be registered with the images from the old system, using objects that did not change in between. During the actual computation of the calibration one must enter actual measurement of known objects. Therefore a visit to the scene is necessary where as many objects as possible which are seen on the images should be measured to serve for the control of the accuracy of the calibration.

It is also very important that we use the original images when we do the calibration to be able to estimate the actual accuracy of the measurements. Accuracy depends on the resolution of the image and on the height of the person on the image. In normal circumstances, the error in determination of a person's height is about 5 cm. The SVM method therefore enables quite accurate determination of a person's height in an image. In special cases, when a person stands in the door frame or if we would like to estimate the size of an object such as a footprint, calibration of just two dimension of the space suffice, sometimes even just one dimension if the concerned object lies on a calibrated line.

3.1.1 Problems in a person's height determination

Determination of a person's height can be difficult if the person is not visible on the image in its entirety, for example, if the feet or the tip of the person's head are not visible. This can happen quite often if the camera is not mounted high enough or if the person is too close to the camera. In such cases, one can try to reconstruct the hidden body parts with the help of a general body model or the model of the observed person if the missing body part is seen in some other video frames.

Another often problem in determination of a person's height is that the person is on the entire video clip in a hunched posture due to running or brisk walking. If the person does not stop and straighten up, one must take this factor into account and determine at least the smallest possible height. For how much is the person taller, in addition to the determined minimal height, can be estimated using different phases of gait [10].

3.2. Other soft biometric features

If longer video surveillance clips are available the behavior of the observed person should be carefully analyzed. Walking has a certain personal character and can be used for identification [9]. Handling of

Figure 3. Wide hips can be used as a soft biometric feature.

objects can indicate the handedness of a person, for example with which hand one reaches products during shopping, with which hand one pays, reaches for cash on an ATM etc. It is also important how certain objects are carried, in which hand or over which shoulder one carries a bag. All those details can help in a person's identification. In cases of stealing of goods, one needs to check if it can be seen, that the suspect is hiding something under his clothing, or if his way of walking has changed. Any visible signs such as various inborn or injury related handicaps can also be used for identification. Tattoos are are well visible also on images of poor quality.

When we try to identify a person on a video clip, one should not concentrate only on the face features but also on the soft biometric features that can help us to reduce the number of suspects or to exclude a particular person from the list of suspects. Therefore, it is necessary to photograph for police records the entire body of a person where all particularities of that person can be seen. Fig. 3 (left) shows a person with disproportionally wide hips for the person's height. This size ratio can be verified on other images (Fig. 3, center and right).

If a suspect is apprehended immediately after a crime was committed, one can consider also features which can normally change quite rapidly, such as clothing, the shape and color of hair, existence of a mustache or a beard etc.

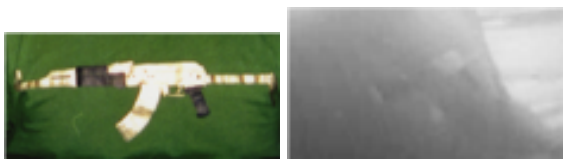


Figure 4. The design of a T-shirt (left) weared by a suspect, apprehended right after the crime took place, was identified on the surveillance video (right).

Clothing features can be used if the suspect could not have changed in between. Fig. 4 shows a case where the design of a T-shirt was used for identification. Special clothing features identified on surveillance video should be described in the report so that later, they can be searched for, for example, during a house search.

4. Other considerations from practice and experience

During a video surveillance system installation, one must mount the cameras so that the camera view angles cover the entire surveilled space and that the image quality is acceptable in all lightning conditions. All circumstances that might influence these two parameters should be considered. Sometimes, the surveillance system needs an extensive long time to adapt to sudden changes in illumination. The view angle of the camera can be obstructed by objects in the surveilled space. When some body parts of the surveilled person are occluded, the estimation of body height, for example, can be much harder. In the video corresponding to Fig. 2, for example, the figure of the person is never seen in its entirety, making the estimation of body height much more complicated.

Fig. 5 illustrates a poor placement of the surveillance camera, since when the door leading into the surveyed space is open, it occludes a large portion of the camera view angle, including the area where the vault was standing.

When the body height of a suspect is measured, it is very important to note if the person was wearing shoes or not. When analyzing events in front of ATMs, it is desirable if the clocks of the video surveillance system and the ATM system are synchronized. If not, then time intervals between ATM transactions should be used instead. Therefore, it is important to recover and save for analysis a much longer segment of the surveillance video where sev-

Figure 5. Bad placement of the surveillance camera–when the door is open, it occludes a large portion of the room, including the vault–the most important object in the room from a security viewpoint.

eral transactions are recorded. Then, several time intervals between transactions can be computed and based on the correspondence with the time intervals between ATM transactions, the timing in the video surveillance system can be aligned with the timing in the ATM system.

In some cases, it turned out, that video recordings from other nearby surveillance cameras would be useful, but it was already too late to obtain them. Namely, the legal obligation for safekeeping surveillance video is time limited, normally, at most up to three months, and then the old video data is usually erased by writing over new video data. Privacy advocates recommend the shortest legally required time for storing surveillance data and most producers of video surveillance equipment enable the storage of data between seven days and three months. Industry standards recommend that the storage capacity in a surveillance recording device should have a capacity to store at least 48 continuous hours of video with the recording parameters that enable a functional reconstruction of the events. For analysis of a crime event, public video surveillance footage can be also helpful, to determine, for example the escape direction or hiding of some material evidence.

Often the poor quality of video footage is a result of inappropriate copying of video data. The original video data can even get lost or stolen. In such cases, sometimes only images printed on paper remain. When original video digital data is not available and only poor quality printed images remain, advanced methods of image enhancement must be used [4].

5. Conclusions

Image material from video surveillance systems, which is used for face recognition, is often not suitable for direct use in automated face recognition systems. Images must usually be enhanced using a variety of image processing and computer vision methods. Sometimes even a manual step is necessary in the chain of recognition if software methods fail at a certain task. A professional sketch artist, for example, can draw a face based on video footage and the resulting sketch can be used as input into a face recognition system that is able to recognize also face sketches. If face recognition fails, then we can attempt to use other soft biometric properties, such as a person's height, for identification. An expert witness for face biometry must therefore have an understanding and a working experience of a very wide range of image processing and computer vision methods and tools.

References

[1] Amped software. http://ampedsoftware.com/five. online; accessed 19-November-2014. 2

[2] B. Batagelj and F. Solina. Face recognition in different subspaces: a comparative study. In *Pattern recognition in information systems : proceedings of the 6th International Workshop on Pattern Recognition in Information Systems, PRIS 2006 in conjunction with ICEIS 2006*, pages 71–80, Paphos, Cyprus, 2006. Insticc Press. 2

[3] V. Blanz and T. Vetter. Face recognition based on fitting a 3D morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, 2003. 1

[4] T. Bourlai, A. Ross, and A. K. Jain. Restoring degraded face images: A case study in matching faxed, printed, and scanned photos. *IEEE Transactions on Information Forensics and Security*, 6(2):371–384, 2011. 6

[5] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *International Journal of Computer Vision*, 40(2):123–148, 2000. 2, 3, 4

[6] H. Farid. A survey of image forgery detection. *IEEE Signal Processing Magazine*, 2(26):16–25, 2009. 2

[7] A. K. Jain, B. Klare, and U. Park. Face matching and retrieval in forensics applications. *IEEE MultiMedia*, 19(1):2–10, 2012. 1

[8] B. F. Klare, Z. Li, and A. K. Jain. Matching forensic sketches to mug shot photos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):639–646, 2011. 3

[9] J. Kovač and P. Peer. Transformation based walking speed normalization for gait recognition. *Transactions on Internet and information systems*, 7(11):2690–2701, 2013. 3, 4

[10] J. Ljungberg and J. Sönnerstam. *Estimation of human height from surveillance camera footage-a*

reliability study. Examensarbete i ortopedteknik, Jönköping University, 2008. 4

[11] U. Park and A. K. Jain. 3D model-based face recognition in video. In *Advances in Biometrics, Proceedings 2nd International Conference on Biometrics, ICB*, pages 1085–1094. Springer, 2007. 1, 3

20^{th} *Computer Vision Winter Workshop*
Paul Wohlhart, Vincent Lepetit (eds.)
Seggau, Austria, February 9-11, 2015

Continuous Hyper-parameter Learning for Support Vector Machines

Teresa Klatzer[1]
[1]Institute for Computer Graphics and Vision
Graz University of Technology
klatzer@icg.tugraz.at

Thomas Pock[1,2]
[2] Safety & Security Department
AIT Austrian Institute of Technology
pock@icg.tugraz.at

Abstract. *In this paper, we address the problem of determining optimal hyper-parameters for support vector machines (SVMs). The standard way for solving the model selection problem is to use grid search. Grid search constitutes an exhaustive search over a pre-defined discretized set of possible parameter values and evaluating the cross-validation error until the best is found. We developed a bi-level optimization approach to solve the model selection problem for linear and kernel SVMs, including the extension to learn several kernel parameters. Using this method, we can overcome the discretization of the parameter space using continuous optimization, and the complexity of the method only increases linearly with the number of parameters (instead of exponentially using grid search). In experiments, we determine optimal hyper-parameters based on different smooth estimates of the cross-validation error and find that only very few iterations of bi-level optimization yield good classification rates.*

1. Introduction

In the field of machine learning much effort is put in developing new algorithms trying to beat the current record on diverse challenges and benchmarks. What all those methods have in common is that they only work as good as they have been fine-tuned by setting sensible parameters affecting the performance of the algorithms. The support vector machine (SVM) [9, 6, 19] as a particular instance of a machine learning algorithm is a very popular method for supervised classification that finds its application in several disciplines including bioinformatics, text and image recognition. Also for the SVM, setting good hyper-parameters strongly influences the classification performance. The aim of model selection is to find the hyper-parameters such that the performance of the learning algorithm is "optimal". Usually this is done manually, or via some combination of grid search and manual search.

Few parameters (1-2) can be set quite successfully based on the evaluation of the cross-validation (CV) error on a grid of possible parameter values. For many parameters, however, the problem is hard to solve because the search space grows exponentially in the number of parameters. Grid search can easily be parallelized, but one would still need access to a massive computational cluster to solve the problem in reasonable time.

In the past, attempts to reduce the complexity of machine learning algorithms in terms of the number of hyper-parameters have been made. E.g., it is common practice to use linear SVMs e.g. for image classification on pre-computed explicit feature maps of the data [22]. Another example is the concept of multiple kernel SVMs where kernels with different fixed bandwidths are combined using weighted sums of them [1, 21, 11]. Here, the weighting factors are directly included in the training objective of the SVM.

More recent literature suggests that especially in the field of computer vision there is increased popularity of large hierarchical models [3] such as Convolutional Neural Networks [14] or Deep Belief Nets [12] which inherently have a large number hyper-parameters to set.

The idea of using bi-level optimization for determining hyper-parameters is not entirely new. Kunapuli et al. [15] have investigated a similar approach to ours, but they use different methods to deal with the optimization problem and only use available standard solvers which limits them to experiments with a linear SVM. Another approach to use gradient methods to solve the parameter selection problem also for kernel SVMs can be found in [7]. They seek to minimize smoothed estimates of the generalization error

of the SVM w.r.t. the hyper-parameters, however, their investigations are restricted to use error measures where the gradient to the hyper-parameters can directly be computed.

Our contribution is an attempt to solve the model selection problem for linear and kernel SVMs, with extension to several kernel parameters using a bi-level optimization approach. We develop a general optimization scheme that allows for continuous hyper-parameter learning based on estimates of the cross-validation error.

Outline. This paper is organized as follows: In Section 2 we discuss typical methods for hyper-parameter optimization such as grid search methods. In Section 3 we develop the bi-level solution for the SVM in general and extend it to optimize several kernel parameters. Furthermore we discuss the choice of a smoothed higher level loss function to estimate the classification performance. In Section 4, we evaluate the proposed method and compare the different performance measures. In Section 5 we conclude the paper.

2. Grid search and random search

Throughout the machine learning literature, grid search is the chosen method to determine hyper-parameters. It is common practice to estimate the performance of a learning algorithm based on a T-fold cross-validation error H (e.g. [10]). Here, the error is determined on the data that has not been used for training in the respective fold. The hope is that the performance of the learning algorithm based on the T validation sets $\text{data}_{t=1,\ldots,T}^{(\text{val})}$ can be successfully transferred to the test set.

Inspired by the discussion about hyper-parameter optimization and grid search/random search in [3], we formalize the problem of hyper-parameter optimization in terms of discrete sets as follows. Let θ be a set of hyper-parameters with cardinality S and θ_k one possible configuration out of K in the discrete search space. Let $w_t(\theta)$ be the separating hyperplane obtained by the SVM training algorithm on training set t using the hyper-parameters θ. The minimization problem addressed by grid search can be written as

$$\underset{\theta \in \{\theta_1, \ldots \theta_k\}}{\arg\min} \sum_{t=1}^{T} H(\text{data}_t^{(\text{val})}, w_t(\theta)). \tag{1}$$

For the SVM a typical set of hyper-parameters is e.g. $\theta = (c, \gamma)$: The regularization parameter c controlling the margin and the bandwidth γ of a Gaussian kernel. From this formulation, we can easily deduce that grid search suffers from the curse of dimensionality: Each hyper-parameter $\theta_1, ..., \theta_S$ from the set θ can take a set of values $V_1, ..., V_S$. Then the number of grid search trials is calculated by counting every possible combination of values:

$$\#\text{trials} = \prod_{s=1}^{S} |V_s|. \tag{2}$$

Often, a grid search procedure is accompanied by some degree of manual search to identify promising value sets V_s for each component of θ. Another practical strategy to alleviate the grid search procedure is to perform first a coarse search to identify interesting parameter ranges, and then consequently re-do the grid search on a finer grid. Given access to a computational cluster, grid search can be easily parallelized and run on the distributed system. It is also common to assign a certain computational budget to perform grid search (e.g. measured in trials).

There have been some attempts to tackle the problem of model selection other than grid search e.g. using Bayesian optimization [20], sequential model based optimization [13], or a random search approach by [3]. Using random search [3] better or equal results in hyper-parameter optimization can be achieved compared to standard grid search, using a reduced computational budget. These approaches are interesting if the cardinality of the set θ exceeds $S = 2$, but they cannot be used for arbitrarily high numbers of parameters (in [3] results are presented for $S \leq 32$; determining hyper-parameters for a Deep Belief Network).

What all those approaches neglect is the fact that e.g. for SVMs the hyper-parameters are continuous. Through the discretization, we always lose accuracy in the possible solution. In our approach to solve the hyper-parameter optimization problem on the example of SVMs we exploit this property.

Moreover, grid search or random search procedures are not adaptive. Only by manual intervention, the course of the experiments can be altered such that irrelevant parameter values are not further explored. For the application to SVMs, we propose a continuous bi-level optimization scheme that is indeed adaptive and performs continuous optimization on the (smoothed) error surface we typically get calculating a full grid search.

We will see another advantage of the bi-level optimization scheme: The complexity only grows linearly in the number of hyper-parameters. For each hyper-parameter we want to determine, we

have one additional gradient to compute (see Eq. 12, 16, 19, 23). This makes the method also applicable to more complex formulations of SVMs such as kernel SVMs with highly parametrized kernels or for learning hyper-parameters for a multiple kernel SVM.

3. Proposed method

3.1. Preliminaries

In this paper we use a soft margin formulation of the support vector machine. Assuming training examples $x_i \in \mathbb{R}^{1\times D}$ with $i = 1, ..., N$ and their labels $y_i \in \{-1, 1\}$ we can write the optimization objective for the linear SVM as follows:

$$\begin{aligned} \min_{w,b,\xi} \frac{c}{2}\|w\|_2^2 + \sum_{i=1}^{N} \xi_i \\ s.t. \quad y_i(\langle w, x_i\rangle + b) \geq 1 - \xi_i \\ \xi_i \geq 0. \end{aligned} \tag{3}$$

Starting from the primal formulation using the slack variables ξ_i we can write the SVM objective function in its unconstrained form in terms of a loss function $\ell(.)$ (like e.g. in [6]):

$$w^*(\theta) \in \arg\min_{w,b} \left\{ \frac{c}{2}\|w\|_2^2 + \sum_{i=1}^{N} \ell(w, b, x_i, y_i) \right\}. \tag{4}$$

Solving the SVM gives us the optimal soft-margin hyperplane defined by w^*. It is influenced by the regularization parameter c which controls the trade-off between maximizing the margin and minimizing the misclassification error. The loss function in Eq. 4 is the exact Hinge loss

$$\ell(w, b, x_i, y_i) = \max(0, 1 - y_i(\langle w, x_i\rangle + b). \tag{5}$$

It will turn out in Eq. 9 that we require the SVM objective to be twice continuously differentiable, thus we introduce a smooth approximation [23] of Eq. 5 parametrized with μ:

$$\ell_\mu(w, b, x_i, y_i) = \frac{1}{\mu} \log(1 + e^{-\mu(y_i(\langle w,x_i\rangle+b)-1)}) \tag{6}$$

In [23] it is shown that $\ell_\mu(.)$ converges to $\ell(.)$ as $\mu \to \infty$. The actual choice of μ will be discussed in Section 4. Solving the SVM we obtain an optimal soft-margin classifier, but the quality of the solution depends on how we choose the hyper-parameter c. We want to set this parameter such that the CV error on the given data is minimal.

3.2. Bi-level formulation

In the following, we want to formulate the model selection problem as a bi-level optimization problem. Bi-level optimization is a mathematical concept involving a higher level optimization problem with another (lower level) optimization problem as its constraint [8]. The aim is to find the hyper-parameters yielding the minimal cross-validation error subject to the SVM solved using those parameters. The challenge is to set the error measure in connection to the hyper-parameters because it is typically not directly dependent on the hyper-parameters, but only via the optimal hyperplane defined by w^* obtained by minimizing the SVM's energy function E. This relation is depicted in Fig. 1.

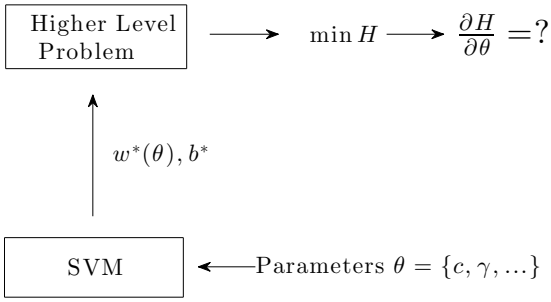


Figure 1. Schema of the bi-level problem.

Formally, we can write:

$$\begin{aligned} \min_{\theta} \sum_{t=1}^{T} H(w_t(\theta), \Xi_t, \eta_t) \\ s.t. \quad w_t(\theta) \in \arg\min_{w_t} E(w_t, \theta, X_t, y_t) \\ t = 1, ..., T. \end{aligned} \tag{7}$$

For simplicity, we use matrix notation in the derivations. Let us define the important symbols. The training set we are given is divided into a training set to calculate the SVM classifier and a validation set to estimate the performance of the trained classifier.

We use an augmented weight vector defined as $w \in \mathbb{R}^{1\times D}$ with D the number of feature dimensions of the input data plus one, including bias b in the end. The training examples $x_i \in \mathbb{R}^{1\times D}$, $i = 1, ..., N$ are condensed in the matrix $X \in \mathbb{R}^{N\times D}$, the validation examples $\zeta_i \in \mathbb{R}^{1\times D}$ are condensed in the matrix $\Xi \in \mathbb{R}^{L\times D}$. Both X and Ξ contain a column of ones in the end for handling the bias implicitly. N and L are the numbers of examples in the training and validation set, respectively. The vectors $y \in \mathbb{R}^{N\times 1}$ and $\eta \in \mathbb{R}^{N\times 1}$ contain the class labels $y_i, \eta_i \in \{-1, 1\}$ for the data. θ is the column vector

of hyper-parameters for the SVM, in the linear case $\theta = c$. There are $t = 1, ..., T$ sets of training and validation data for T-fold cross-validation.

To solve the problem in Eq. 7 we need to reformulate it. We use a Lagrange multipliers λ_t to deal with the lower level constraints:

$$\mathcal{L}(w, \theta, \lambda) = \sum_{t=1}^{T} \left[H(w(\theta)_t, \Xi_t, \eta_t) + \left\langle \lambda_t, \frac{\partial E}{\partial w_t} \right\rangle \right]. \tag{8}$$

We will use the unconstrained form of the bi-level problem from Eq. 8 to calculate the desired gradient $\frac{\partial \mathcal{L}}{\partial \theta}$ via implicit differentiation. This gradient is used to determine the optimal hyper-parameters for the SVM.

For θ^* to be a local minimizer of Eq. 8 the necessary KKT optimality conditions [4] are given by

$$\mathcal{G}(w, \theta, \lambda) = \begin{pmatrix} \frac{\partial H}{\partial w_1} + \frac{\partial^2 E}{\partial w_1^2}\lambda_1 \\ \vdots \\ \frac{\partial H}{\partial w_T} + \frac{\partial^2 E}{\partial w_T^2}\lambda_T \\ \sum_{t=1}^{T} \left(\frac{\partial H(w_t, .)}{\partial \theta} + \frac{\partial^2 E}{\partial w_t \partial \theta}\lambda_t \right) \\ \frac{\partial E}{\partial w_1} \\ \vdots \\ \frac{\partial E}{\partial w_T} \end{pmatrix} = 0. \tag{9}$$

From the structure of Eq. 9 we observe that the SVM's energy function has to be twice continuously differentiable. This fact gives rise to use the smooth approximation of the Hinge loss in Eq. 6. Likewise, we also need a smooth approximate of the CV error as a higher level loss function $H(.)$ (see discussion in Section 3.6).

The system of equations Eq. 9 can be reduced by firstly solving the optimality conditions of the SVM for fixed θ for each fold t up to sufficient accuracy (the last T lines of Eq. 9 are therefore eliminated). Hence we get w_t^* which is then used in the remainder of the equations. From the first T equations we can calculate the Lagrange multipliers λ_t using the inverse Hessian of the SVM's energy function:

$$\lambda_t = -\left(\frac{\partial^2 E}{\partial w_t^{*2}} \right)^{-1} \frac{\partial H}{\partial w_t^*}. \tag{10}$$

Consequently we obtain the main result:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \sum_{t=1}^{T} \left(\frac{\partial H(w_t^*, \Xi_t, \eta_t)}{\partial \theta} - \right. \\ \left. - \frac{\partial^2 E}{\partial w_t^* \partial \theta} \left(\frac{\partial^2 E}{\partial w_t^{*2}} \right)^{-1} \frac{\partial H}{\partial w_t^*} \right). \tag{11}$$

This gradient is used for optimizing the hyper-parameters. Observe that in case of the linear SVM (Eq. 4) the gradient $\frac{\partial \mathcal{L}}{\partial \theta}$ reduces to

$$\frac{\partial \mathcal{L}}{\partial c} = \sum_{t=1}^{T} - \frac{\partial^2 E}{\partial w_t^* \partial \theta} \left(\frac{\partial^2 E}{\partial w_t^{*2}} \right)^{-1} \frac{\partial H}{\partial w_t^*} \tag{12}$$

because the derivative of the higher level loss function $H(.)$ is zero w.r.t. c. So far, we have developed the bi-level solution for the linear SVM. In the following, we show that the concept can easily be extended for kernel SVMs.

3.3. Extension to kernel SVMs

First, we have to formulate the lower level problem - the energy function of the SVM - in terms of a kernel function $k(x, x_i)$. We use again a primal, unconstrained formulation of the SVM's energy (like in [6]). Instead of the weight vector w, we introduce a weight vector $\alpha \in \mathbb{R}^{N \times 1}$ with N the number of training examples.

$$\alpha^*(\theta) = \arg\min_{\alpha} \left\{ \frac{c}{2} \|f\|_2^2 + \sum_{j=1}^{N} \ell_\mu(f(x_j), y_j) \right\} \tag{13}$$

with

$$f(x) = \sum_{i=1}^{N} \alpha_i k(x, x_i) \qquad \text{and} \\ \|f\|_2^2 = \sum_{j=1}^{N} \sum_{i=1}^{N} \alpha_j \alpha_i k(x_j, x_i) = \alpha^T K \alpha. \tag{14}$$

The kernel matrix $K \in \mathbb{R}^{N \times N}$ is composed of matrix elements $k(x_j, x_i)$.

Rewriting Eq. 13 in matrix form using $k_j \in \mathbb{R}^{1 \times N}$ for describing a row of matrix K, we get

$$\alpha^*(\theta) = \arg\min_{\alpha} \left\{ \frac{c}{2} \alpha^T K \alpha + \sum_{j=1}^{N} \ell_\mu(k_j \alpha, y_j) \right\} = \\ \arg\min_{\alpha} E(\alpha, \theta, K, y). \tag{15}$$

For the non-linear case, the SVM's energy function used in the bi-level solution stated in Eq. 9, 10 and 11 is replaced by $E(\alpha, \theta, K, y)$. After the change of the weight vector w to α and of the data matrices X and Ξ to their corresponding kernels $K \in \mathbb{R}^{N \times N}$ and $\mathcal{K} \in \mathbb{R}^{L \times N}$, the former results are directly applicable.

In the case of a simple Gaussian kernel with bandwidth γ we have $\theta = (c, \gamma)^T$ and

$$\frac{\partial \mathcal{L}}{\partial \theta} = \begin{pmatrix} \frac{\partial \mathcal{L}}{\partial c} \\ \frac{\partial \mathcal{L}}{\partial \gamma} \end{pmatrix}. \tag{16}$$

The derivative

$$\frac{\partial H(\alpha(\theta)_t, \mathcal{K}_t, \eta_t)}{\partial \theta} \tag{17}$$

from Eq. 11 is non-vanishing any more due to the dependence of the kernelized data to the kernel parameters.

3.4. Generalization to many kernel parameters

Assuming a Gaussian kernel having $d = 1, ..., D$ parameters, one for each feature dimension of input data, we can write down one element of the kernel matrix:

$$k(x_j, x_i) = \exp(-\sum_{d=1}^{D} \gamma_d (x_{jd} - x_{id})^2). \quad (18)$$

The gradient $\frac{\partial \mathcal{L}}{\partial \gamma}$ is extended to

$$\frac{\partial \mathcal{L}}{\partial \gamma} = \left(\frac{\partial \mathcal{L}}{\partial \gamma_1}, \frac{\partial \mathcal{L}}{\partial \gamma_2}, \ldots, \frac{\partial \mathcal{L}}{\partial \gamma_D} \right)^T \quad (19)$$

and the entries of the gradient vector are computed according to Eq. 11.

3.5. Multiple kernel bi-level SVM

We demonstrate in this section that the bi-level optimization scheme can directly be applied to determine parameters for a multiple kernel model [1, 21, 11]. There are different application scenarios for multiple kernel models: They can be used to combine different subsets of heterogeneous features or to combine different feature representations of the data.

We define the model as follows: Let $p = 1, 2, ..., P$ be the partitions (i.e. equivalent to the number of kernels used) each of which is of dimension D_p. A training example can be written as concatenation of P feature subsets $x_i = \{x_i^1, x_i^2, ..., x_i^P\}$ whereas $x_i^P \in \mathbb{R}^{D_p \times 1}$. A kernel element k_β of the new kernel matrix $K_\beta \in \mathbb{R}^{N \times N}$ is

$$k_\beta(x_j, x_i) = \sum_{p=1}^{P} \beta_p k_p(x_j^p, x_i^p). \quad (20)$$

With $k_{\beta j}$ being a row of the matrix K_β the SVM's energy function becomes

$$\alpha(\theta) = \arg\min_{\alpha} \left\{ \frac{c}{2} \alpha^T K_\beta \alpha + \sum_{j=1}^{N} \ell_\mu(k_{\beta j} \alpha, y_j) \right\}. \quad (21)$$

The vector of hyper-parameters θ now contains the γ_p for each sub-kernel and the weighting factors β_p:

$$\theta = (c, \gamma_1, \gamma_2, ..., \gamma_P, \beta_1, ..., \beta_P)^T. \quad (22)$$

Analogous to the previous derivations, we can write the gradient $\frac{\partial \mathcal{L}}{\partial \theta}$ as follows:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \begin{pmatrix} \frac{\partial \mathcal{L}}{\partial c} \\ \left(\frac{\partial \mathcal{L}}{\partial \gamma_p} \right)_{p=1}^{P} \\ \left(\frac{\partial \mathcal{L}}{\partial \beta_p} \right)_{p=1}^{P} \end{pmatrix}. \quad (23)$$

Our bi-level learning approach makes it possible to treat the kernel combination weights as hyper-parameters and also the parameters for the base kernels can be learnt. Next, we discuss the choice of the higher level loss function $H(.)$.

3.6. Higher level loss function

Due to the nature of our continuous optimization, we need a differentiable estimate of the generalization error. This is ideally a smoothed version of the actual hard classification rate e.g. described by the zero-one loss which assigns constant error to wrongly classified examples and zero error to correct examples.

In this paper we investigate three different higher level loss functions and compare them according to their meaningfulness for estimating the performance of the SVM. We use a smoothed version of the zero-one loss:

$$H(w, \Xi, \eta) = \frac{1}{\exp(\mu[\eta \circ (\Xi w^T)]) + 1} \quad (24)$$

with smoothing parameter $\mu = 12$. However, the zero-one loss is a non-convex function which might be a disadvantage for the optimization process.

The other functions we reviewed were the smoothed Hinge loss function

$$H(w, \Xi, \eta) = \sum_{i=1} \ell_\mu(w, b, \zeta_i, \eta_i) \quad (25)$$

as well as the mean squared error on the classification

$$H(w, \Xi, \eta) = \frac{1}{2L} \|\Xi w^T - \eta\|_2^2. \quad (26)$$

The MSE calculates the mean squared distance of the examples to the class labels (or, otherwise put, to the margins). Intuitively, the smoothed Hinge loss function should yield a better estimate of the hard classification error than the MSE because it assigns no error to correctly classified examples up to the margin and a linear increasing error for examples inside the margin and to wrong examples. Both MSE and Hinge loss are convex functions, and the MSE is particularly easy to differentiate.

On toy experiments, we found that the Hinge loss and the zero-one loss perform better on oddly shaped

datasets (imbalanced, with outliers) than the MSE (see Fig. 2). Using the MSE (green area) the bi-level SVM tends to learn a larger margin than using the Hinge loss (blue area), and the margins are pulled towards the barycenter of the data distribution. There was no difference in the behaviour between Hinge/zero-one loss in this case. However, in our experiments using real world data sets also the MSE performs quite well suggesting a good generalization capability.

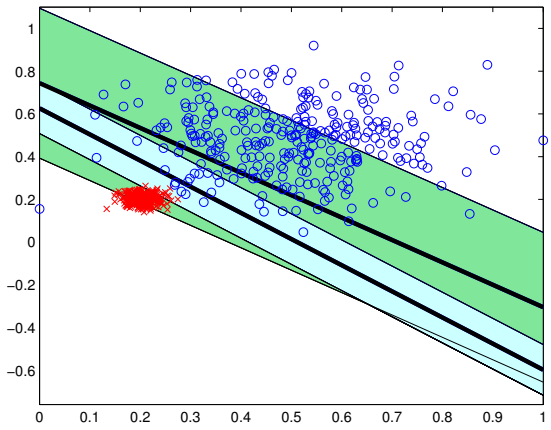


Figure 2. Margins and hyperplanes on an imbalanced toy data set for Hinge loss (blue) and MSE (green) as a higher level loss function.

4. Experimental Results

In our implementation we used the LBFGS-B optimization algorithm to solve the higher level optimization problem, see [5]. For solving the lower level problems (the SVM) we used FISTA [2]. For the experiments, we used several data sets from the UCI machine learning repository [1] (diabetes, ionosphere, heart, seeds, parkinson). The aim of the experiments is to show how the classification results using the hyper-parameters determined via the bi-level optimization scheme compare to the results of the traditional grid search procedure. In particular, we focus on evaluating the effectiveness of the higher level loss function approximations. Furthermore, we show results for two settings using an increased number of hyper-parameters as well as results for an image classification experiment.

The smoothing parameter μ from Eq. 6 and 25 was chosen as big as possible as long as the outer level optimization does not fail (due to the Hessian becoming ill-conditioned when it is very sparse). The initial values θ for the bi-level optimization were set randomly due to the fact that their choice is not critical: Usually the bi-level program converges to the same θ^* for different initial values given sufficient accuracy of the solution of w^*.

[1] http://archive.ics.uci.edu/ml

4.1. Illustrative examples

First, we have a look at how the hard classification rates vary in the hyper-parameters and how the higher level loss functions we mentioned earlier "fit" to the achieved classification performance. For this reason, we show two examples. First, results using a linear bi-level SVM on the diabetes data set are shown in Fig. 3. On the y axis the CV error rate and the test error rate are shown as well as the higher level loss function values. The MSE is plotted in dashed blue, the approximated Hinge loss in solid red and the smooth zero-one loss in solid green. The errors are plotted over the regularization parameter c and have been determined via grid search. We point out that the error values are not directly comparable hence we rescale them for better comparison.

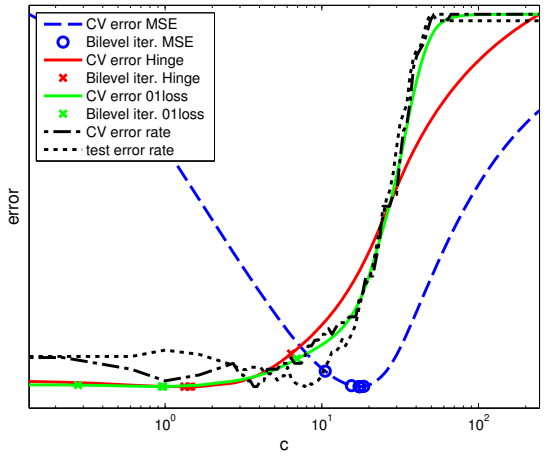


Figure 3. Comparing hard classification rates and the corresponding higher level loss function values over c using a linear SVM on the diabetes data set.

We observe that the minima of the CV and test classification error rates do not coincide exactly but the magnitudes are consistent. The smoothed Hinge loss seems to model the actual CV classification rates quite well, and the zero-one loss approximation fits even better (as expected). Their minima lie in the area of lowest classification error rates. The MSE does not correspond to the error rates, but still has the minimum in a reasonable area.

The second example shown in Fig. 4 illustrates the dependency of the kernel parameter γ of a simple RBF kernel SVM for a fixed c on the same (diabetes) data set. Here, the CV and test error rates have

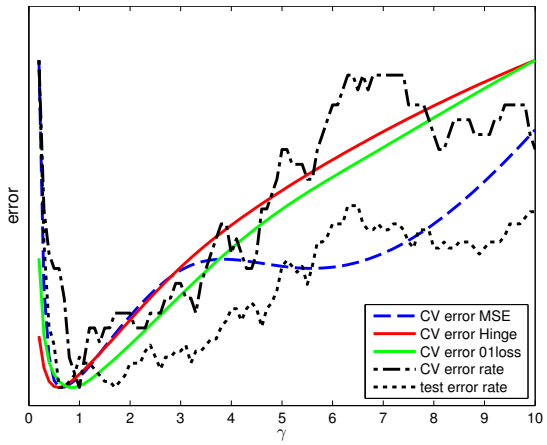


Figure 4. Comparing hard classification rates and the corresponding higher level loss function values with fixed c over γ using a kernel SVM on the diabetes data set.

again similar minima, and in this case, also all three flavours of higher level loss functions share approximately the same minima. For the MSE, we observe and additional local minimum at $\gamma \approx 5.5$ which is an unwanted property for optimization. At this point, no clear answer can be given which of the higher level loss functions is the best.

4.2. Classification rates for different settings

In Tab. 1 we summarize the CV and test error rates obtained by the linear and kernel bi-level SVM as well as the respective rates obtained using grid search on a comparable computational budget measured in trials. Each trial consists of the evaluation of the SVM for T folds for one set of hyper-parameters. The number of folds in our experiments was chosen with $T = 5$. Surprisingly, often the MSE yields good test classification rates, sometimes even best values, even though the CV error does not usually yield lowest rates. Here, often Hinge loss and sometimes zero-one loss lead to better results. In terms of number of trials used for optimization, the zero-one and Hinge loss are the best. Given the low computational budget assigned for grid search, only for one data set better rates were achieved (seeds), even though it is quite possible that grid search can outperform the bi-level approach using more trials in the linear/simple kernel case because the exact classification rates are taken to decide which set of hyper-parameters is best. However, as we will see in the following experiments, the classification rates can be significantly improved by using a more complex kernel for which it will be difficult to achieve a good result using exhaustive grid search.

Data set	Type	CV Err.	Test Err.	Trials
Diabetes	Lin01loss	**24.13**	20.33	10
	LinHinge	**24.13**	20.66	**7**
	LinMSE	25.87	**19.67**	**8**
	LinGrid	*24.34*	*20.66*	*15*
	Ker01loss	22.17	18.03	**8**
	KerHinge	21.30	**17.70**	26
	KerMSE	**18.70**	20.98	11
	KerGrid	*25.00*	*19.02*	*50*
Ionosph.	Lin01loss	12.86	8.57	7
	LinHinge	**11.43**	**7.86**	**5**
	LinMSE	16.19	**7.86**	8
	LinGrid	*14.29*	*8.57*	*15*
	Ker01loss	**0.95**	**2.85**	**17**
	KerHinge	2.86	3.57	32
	KerMSE	3.81	**2.86**	33
	KerGrid	*13.81*	*4.29*	*50*
Heart	Lin01loss	15.79	15	8
	LinHinge	**14.21**	**13.75**	**6**
	LinMSE	17.37	15	8
	LinGrid	*18.95*	***13.75***	*15*
	Ker01loss	15.26	15	**15**
	KerHinge	**14.74**	**13.75**	16
	KerMSE	17.89	**13.75**	34
	KerGrid	*18.95*	*15*	*50*
Seeds	Lin01loss	**6**	10	**5**
	LinHinge	**6**	10	6
	LinMSE	7.33	11.67	11
	LinGrid	*9.33*	***9.33***	*15*
	Ker01loss	2	8.33	**14**
	KerHinge	**1.33**	8.33	15
	KerMSE	7.33	**6.67**	23
	KerGrid	*10.67*	*10*	*50*

Table 1. Summary of classification rates on several datasets comparing the CV and test errors and the number of trials used. Results are reported for the linear ('lin') and kernel ('ker') SVM using the MSE, Hinge or zero-one ('01loss') higher level loss functions.

4.3. Learning multiple parameters

Learning one parameter γ_d per feature dimension. For this experiment, the seeds data set was used ($D = 8$). The results are summarized in Tab. 2.

Learning parameters γ_p and β_p for a multiple kernel SVM. For this experiment the parkinson data set was used [16]. The data contains 21 measurements of different orders of magnitude. Using the multiple kernel SVM we are able to combine the features into P groups of similar magnitude, and set the parameters γ_p and β_p via the bi-level optimization procedure. The results are summarized in Tab. 3. We achieve good results using no pre-processing and no filtering of correlated features compared to the original paper [16] where they report a test classification rate of $8.2\% \pm 2$. We observe an exceptionally low

number of necessary trials using the zero-one loss for both experiments, and very good test classification rates for Hinge and zero-one loss.

Data set	Type	CV Err.	Test Err.	Trials
Seeds	MSE	0.67	**3.33**	62
	Hinge	**0**	**3.33**	119
	01loss	2.67	**3.33**	**59**

Table 2. Results using a bi-level kernel SVM with γ_D parameters.

Data set	Type	CV Err.	Test Err.	Trials
Parkinson	MSE	**0**	9.09	53
	Hinge	8.75	**7.27**	80
	01loss	2.04	**7.27**	**30**

Table 3. Results using a bi-level multiple kernel SVM.

4.4. Image classification

The following image classification experiment was conducted on the Graz02 data set [18]. For feature extraction the VLFeat Library [2] was used. The data was pre-processed according to a bag of visual words model using PHOW features, a variant of SIFT features extracted at several scales [17]. Moreover, for this task we use exponential χ^2 kernels because they show naturally better performance on histogram data compared to RBF kernels [24].

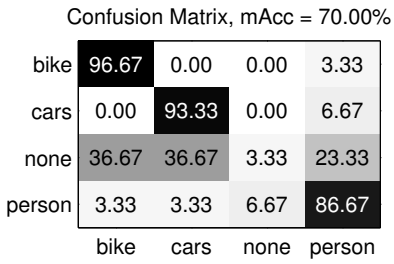


Figure 5. Resulting confusion matrix using a bi-level kernel SVM and the Hinge loss as a higher level loss function.

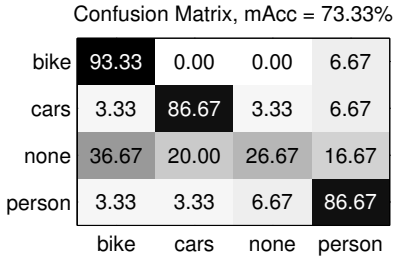


Figure 6. Resulting confusion matrix using a bi-level kernel SVM and the MSE as a higher level loss function.

In Fig. 5 and Fig. 6 we compare the classification results for each of the four classes in the Graz02 data set, namely bike, cars, person and none (the background class). For training we used 60 images per class, and 30 for testing. Overall the accuracy using the MSE is better, but if we do not regard the background class the results using the Hinge loss are superior. By construction, the data for learning 1 vs. rest classifiers is imbalanced due to the low number of positive examples. That might explain why MSE performs worse than Hinge loss in the image classification example. The results via the kernel bi-level SVM were obtained using a mean of 9 trials per each 1 vs. rest classifier that was trained using Hinge loss and 8 trials using the MSE. The results of grid search and evaluating the CV error rate to determine the best hyper-parameters using 50 trials are shown in Fig. 7. We obtain a baseline of classification results on this data set for the relevant classes bike, cars and person.

[2] http://www.vlfeat.org/

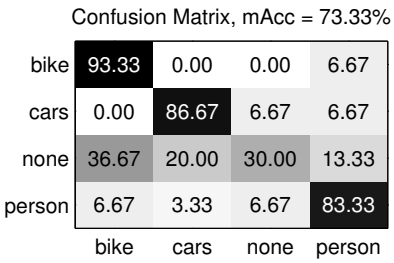


Figure 7. Resulting confusion matrix using a kernel SVM and grid search using 50 trials.

5. Conclusion

In this paper, we presented a novel bi-level optimization scheme that is able to perform continuous hyper-parameter optimization for linear and kernel SVMs based on different smoothed estimates of the CV error rate. Very good test classification rates are obtained using only a tiny fraction of trials that would be necessary to perform exhaustive grid search which makes the method very practical. High potential lies in the optimization of several kernel parameters: The classification rates are better than using only a simple kernel and optimizing the parameters is easy using the bi-level optimization approach. In the case of optimizing one or two parameters only, a very fine grid search might lead to better results than the bi-level approach because the exact classification errors are minimized, but at a much higher computational cost.

Acknowledgements

The authors acknowledge support from the Austrian Science Fund (FWF) under the START project BIVISION, No. Y729.

References

[1] F. R. Bach, G. R. G. Lankriet, and M. I. Jordan. Multiple Kernel Learning, Conic Duality, and the SMO Algorithm. *International Conference on Machine Learning*, 2004. 1, 5

[2] A. Beck and M. Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, Jan. 2009. 6

[3] J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13:281–305, 2012. 1, 2

[4] D. P. Bertsekas. *Nonlinear programming*. Athena Scientific, 1999. 4

[5] R. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995. 6

[6] O. Chapelle. Training a support vector machine in the primal. *Neural Computation*, 19(5):1155–78, May 2007. 1, 3, 4

[7] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, pages 131–159, 2002. 1

[8] B. Colson, P. Marcotte, and G. Savard. An overview of bilevel optimization. *Annals of Operations Research*, 153(1):235–256, Apr. 2007. 3

[9] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sept. 1995. 1

[10] K. Duan, S. Keerthi, and A. N. Poo. Evaluation of simple performance measures for tuning SVM hyperparameters. *Neurocomputing*, 51:41–59, Apr. 2003. 2

[11] M. Gönen and E. Alpaydn. Multiple kernel learning algorithms. *The Journal of Machine Learning Research*, 12:2211–2268, 2011. 1, 5

[12] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006. 1

[13] F. Hutter, H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. *Learning and Intelligent Optimization*, 2011. 2

[14] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 25:1097–1105, 2012. 1

[15] G. Kunapuli and K. Bennett. Classification model selection via bilevel programming. *Optimization Methods & Software*, 23(4):475–489, 2008. 1

[16] M. a. Little, P. E. McSharry, E. J. Hunter, J. Spielman, and L. O. Ramig. Suitability of dysphonia measurements for telemonitoring of Parkinson's disease. *IEEE Transactions on Bio-medical Engineering*, 56(4):1015–1022, Apr. 2009. 7

[17] D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov. 2004. 8

[18] A. Opelt and A. Pinz. Generic object recognition with boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(3):416–431, 2006. 8

[19] B. Schölkopf and A. J. Smola. *Learning with kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2002. 1

[20] J. Snoek, H. Larochelle, and R. Adams. Practical Bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, pages 1–9, 2012. 2

[21] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf. Large scale multiple kernel learning. *Journal of Machine Learning Research*, 7:1531–1565, 2006. 1, 5

[22] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):480–92, Mar. 2012. 1

[23] J. Zhang, R. Jin, Y. Yang, and A. Hauptmann. Modified logistic regression: An approximation to SVM and its applications in large-scale text categorization. *International Conference on Machine Learning*, pages 888–895, 2003. 3

[24] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. *International Journal of Computer Vision*, 73(2):213–238, Sept. 2006. 8

20^{th} *Computer Vision Winter Workshop*
Paul Wohlhart, Vincent Lepetit (eds.)
Seggau, Austria, February 9-11, 2015

Novel Concepts for Recognition and Representation of Structure in Spatio-Temporal Classes of Images

Ines Janusch, Walter G. Kropatsch
Institute of Computer Graphics and Algorithms
Pattern Recognition and Image Processing Group
Vienna University of Technology
Vienna, Austria
{ines, krw}@prip.tuwien.ac.at

Abstract. *This paper discusses open problems and future research regarding the recognition and representation of structures in sequences of either 2D images or 3D data. All presented concepts aim at improving the recognition of structure in data (especially by decreasing the influence of noise) and at extending the representational power of known descriptors (within the scope of this paper graphs and skeletons). For the recognition of structure critical points of a shape may be computed. We present an approach to derive such critical points based on a combination of skeletons and local features along a skeleton. We further consider classes of data (for example a temporal sequence of images of an object), instead of a single data sample only. This so called co-analysis reduces the sensitivity of analysis to noise in the data. Moreover, a representative for a whole class can be provided. Temporal sequences may not only be used as a class of data in a co-analysis process - focusing on the temporal aspect and changes of the data over time an analysis of these changes is needed. For this purpose we explore the possibility to analyse a shape over time and to derive a spatio-temporal representation. To extend the representational power of skeletons we further present an extension to skeletons using model fitting.*

1. Introduction

A single 2D image is defined in the spatial domain. By extending data from a single capturing to a sequence of such data temporal information is added and the data is defined in the spatio-temporal domain [8]. Instead of capturing a single 2D image or 3D data (e.g. a 3D point cloud) the data may be extended to an image sequence or a sequence of 3D data. Temporal information as motion or development over time are thereby added to the representation. Therefore, this paper focuses on novel concepts for the identification of structure from sequences of images or 3D data and on the representation of this structure.

Applications and spatio-temporal datasets for the concepts proposed in this paper can, for example, be found in biology or in medicine. For the latter, spatio-temporal data may describe recurring sequences which may be the motion of an organ or abnormal changes of an organ caused by an illness. In biology, temporal image sequences can, amongst others, be found in plant phenotyping where plants or their roots are imaged on successive days of growth [9]. Furthermore, phenotyping of animals is currently still based on the manual analysis of experts [4]. An analysis of a sequence of 3D scans of an animal may be a future alternative as it provides spatio-temporal data showing the animal as well as its movements.

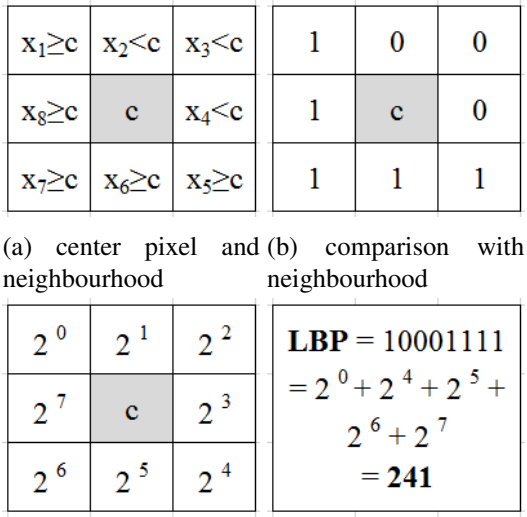
For any analysis of the captured object this object first needs to be detected in the data and processed to compute a suitable representation. Well known representations are Reeb graphs (as described in [1] and skeletons as for example a medial axis or a more sophisticated 3D Curve Skeleton (as described in [2]). For the computation of these representations a binary segmentation of the input data into foreground regions, representing the object to be analysed, and background regions, showing the rest of the data that is not in focus, is needed. However, such a segmentation may introduce artefacts that falsify the representation. We encountered this problem in [10],

where we applied knowledge about the structure to be represented and post-processing methods as for example graph pruning in order to reduce segmentation artefacts kept in the representation. Werghi et al. applied a similar approach in [20]. They handle noise in the input data by knowledge about the structure to be represented and in this way detecting and discarding improper configurations in the representation.
In this paper we discuss general methods to improve representations of data based on a potentially flawed segmentation. In this context we discuss the use of co-analysis for classes of data as well as the application of co-analysis and co-representation for changing, respectively developing shapes. Mitra [15] provides a detailed survey on co-analysis and co-segmentation. Promising methods of co-analysis have for example been presented by Golovinskiy et al. in [6] and van Kaick et al. in [18].
Additionally to co-analysis we propose two novel skeleton based representations: A graph representation using skeletons together with local features and a model based representation that is derived using model fitting to an initial skeleton.

The rest of the paper is structured as follows: Section 2 proposes the use of local features for the computation of critical points while Section 3 bases this computation on a function according to time. The analysis of a whole class of data using so called co-analysis and the representation of such classes using a co-representation is discussed in Section 4. Section 5 introduces extensions to known skeletons that improve their representational power and Section 6 concludes the paper.

2. Critical Points Based on Local Features

Graph based representations or skeletons rely on segmented input data. Thus, for the input data a binary segmentation - a separation between background (not of interest) and foreground (to be represented) - needs to be known. However, such a pre-processing of the data may introduce artefacts. Representations based on flawed segmented data can be improved using post-processing steps that detect and correct spurious parts of the representation. For graph representations a simple graph pruning may for example be applied. However, graph pruning may not remove all spurious branches (false negatives) or discard true branches (false positives).



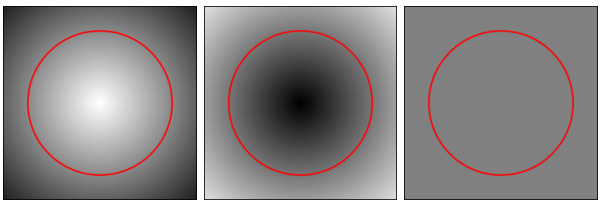
(a) center pixel and neighbourhood (b) comparison with neighbourhood

(c) neighbourhood pattern (d) LBP operator for center pixel c

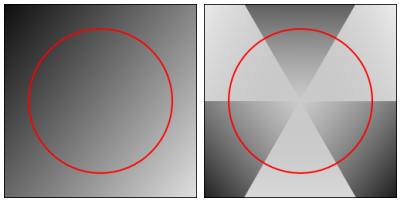
Figure 1: Simple LBP computation.

A graph representation based on segmented data can only provide reliable results for a correct segmentation.
Instead of applying post-processing techniques to reduce artefacts introduced by the segmentation we propose to base the representation on the original unsegmented data. For a Reeb graph representation critical points may be computed on the original data instead of the segmented data. Local Binary Patterns (LBPs) [17] are considered as one method to derive such critical points on an unsegmented image.

LBPs were introduced as as a tool of texture classification and work (in their simplest version) as shown in Figure 1: The center pixel is compared to its neighbourhood. The relations of this comparison are stored as a bit pattern: In case a neighbouring pixel is larger or equal the center pixel its bit is set to 1 otherwise to 0. The neighbourhood pattern is encoded as the position of each neighbourhood pixel in a binary data item [16].
Critical points on a shape according to a Morse function build the nodes in a Reeb graph. Such critical points (in 2D) are minimum, maximum and saddle points. The configuration of the neighbourhood around a pixel encodes the local topology. The region may be a (local) maximum (the bit pattern contains only 0s), a (local) minimum (the bit pattern contains only 1s), a plateau (the bit pattern contains only 1s, but all pixels of the region have the same gray value), a slope (the bit pattern of the region contains one connected component of 1s and



(a) maximum (b) minimum (c) plateau



(d) slope (e) saddle

Figure 2: Neighbourhood configuration detected by LBPS. The red circle indicates the neighbourhood used in the LBP computation for the pixel at the position of the center of the circle.

one connected component of 0s) or a saddle point otherwise [7]. Figure 2 shows examples for all these region configurations that may be encoded by LBPs.

The original LBP operator was defined for the spatial domain only. Similar to the work of Laptev who extended the Harris and Förstner interest point operator to space-time interest points in [12] and [11] the LBP description of local structures was extended in time to describe local features in the spatio-temporal domain [21]. The so called Volume Local Binary Pattern (VLBP) represents dynamic textures as volumes of (X, Y, T), where X and Y are the spatial coordinates, T, as a temporal coordinate, represents points in time. A sequence of dynamic textures over time is therefore represented by a VLBP.

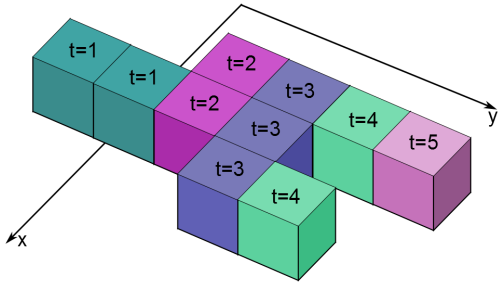
Reeb graphs are derived on binary segmented 2D or 3D data using an analysis based on a Morse function as for example a height function. In order to analyse unsegmented data local descriptors as for example LBPs may be used as Morse function, provided that the descriptors satisfy the necessary conditions, analog to the conditions of Morse functions [3].
Despite the idea to avoid segmentation as a preprocessing step, this approach works on a segmented image as a first input. However, the critical points are computed on the unsegmented data, the segmented image is only needed to guide the computation of the critical points as follows:
On the initial segmentation the medial axis is computed for the foreground region. The medial axis is formed by the centers of maximal circles that cover the shape completely. Therefore, the medial axis implicitly provides a measure of width, as for each point along the medial axis the radius of the inscribed maximal circle (the distance to the boundary) is known [13]. Along this skeleton LBPs are computed for each skeleton pixel. The LBP kernel size is thereby determined by the radius associated with the individual skeleton pixels. Minima, maxima and saddle points that are encountered in this way may then be used as critical points (nodes) in a graph, connections of these nodes can be derived from the skeleton.
In case the position of the skeleton, respectively the critical points, can be estimated (for example in video data based on the position in a previous frame) the segmentation as well as the initial skeleton do not need to be recomputed. Rather this known approximation can be reused to guide the computation of the critical points (in the next frame).

3. Time as Morse Function

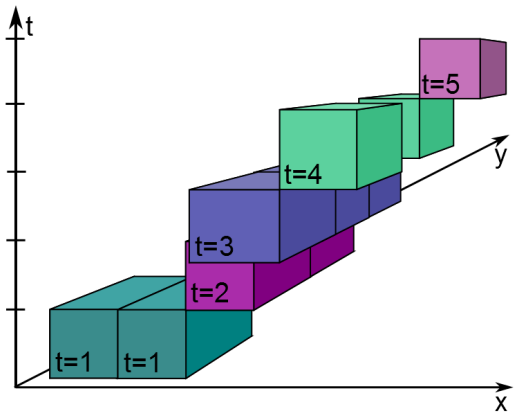
Analysing data over time adds one dimension to the original data domain. Edelsbrunner et al. introduced time varying Reeb graphs in [5]. They present an algorithm to maintain Reeb graphs through time and to store the graph's evolution.
For 2D images of shapes that change over time we can augment the spatial information of the pixels with temporal information by storing as a third coordinate the point in time the according pixel was first

Figure 3: Spatial information of a growing root augmented with temporal information: the image shows a segmented lupine root, the colors indicate measurement time. Image courtesy of Leitner et al. [14]

encountered. Shapes that grow are imaged on several points of time through this development process. After an alignment of the acquired data according to the last acquisition (the most mature one), parts of the growing shape are labeled according to the time they were first encountered. This representation aims specifically at the representation of growth, temporal deformations are not represented. Figure 3 acquired by Leitner et al. [14] shows an example for such a dataset: individual parts of the root are labeled according to the time these parts were first observed. For such a configuration a height function along the temporal axis (time function) may be used to extract level sets. These level sets represent the evolution of the shape over time. Figure 4 illustrates the proposed approach: Figure 4a shows the augmented spatial information, Figure 4b shows level sets of this data according to time.
A Reeb graph can be built, as the time is used equally to a height function as Morse function. In order to build the Reeb graph the individual components are connected by tracing through the spatial information from one time step to the next.



(a) spatial information augmented with temporal information

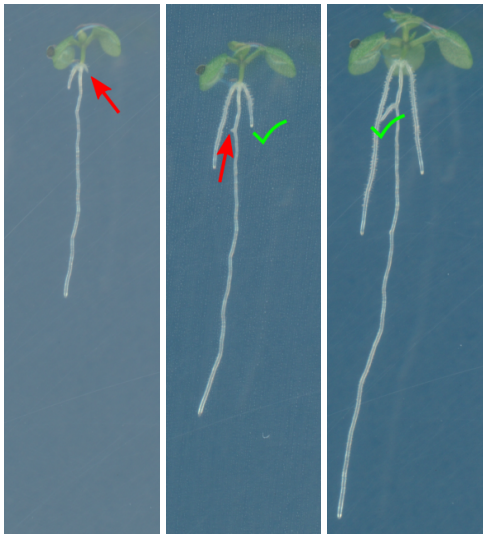


(b) level sets with respect to time

Figure 4: Augmenting the spatial information of a growing structure with temporal information adds one dimension. A height function along this dimension provides a Morse function with respect to time.

4. Co-Analysis and Co-Representation

For the recognition and representation of structure methods based not only on a single object, but on a class of similar objects may be used. This so called co-analysis focuses on a common structure of all objects in the class and on relations between parts of the object and thereby reduces influence of noise in the capturing of a single image. Co-analysis may further reduce the time needed for a training phase, as objects of a class are simultaneously analysed in co-analysis [19].
To perform co-analysis an alignment of the individual data samples is needed first. Van Kaick et al. for example in [18] assume the shapes to be upright-oriented and partitioned into meaningful parts. Golovinskiy et al. in [6] base the shape alignment on the alignment of axis according to a principal component analysis. We propose to align 2D or 3D image data samples using standard representations as for example skeletons or Reeb graphs. Other than the alignment according to an axis, skeletons and graphs allow for an alignment of data samples of articulated or varying objects.
Spatio-temporal data may create classes of images as for example in biological datasets a growing organism may be imaged over time. Therefore, the data consists of sets of related data samples that have certain features in common. An analysis of a collection of data samples is called co-analysis [15] . The aim of this procedure is to label the same entities with the same labels although they may appear in variations. Considering for example drinking glasses: in co-analysis a collection of different glasses is used. All glasses have a body that may hold liquids, some of them may have a stem. Independent on the actual design of the stem (long and thin or short and decorated) this part of the glass should always be detected as stem.
Further knowledge based on co-analysis of a class of data can be used to verify decisions for a single data sample.
Considering the mentioned biological applications, co-analysis of data samples could for example be beneficial in the area of plant phenotyping. Here classes of images are formed as images of several different plants (therefore potentially different



(a) day 12 (b) day 16 (c) day 20

Figure 5: Application for co-analysis: root development. Small branches (indicated by red arrows) may only represent noise. An image of a later day can verify a decision - in this case true branches (indicated by the green tickmark).

phenotypes) exits for the same genotype. Moreover, the plants are imaged on successive days of growth. The temporal stack of images of a single plant can be seen as a class of images. Analysis decisions are not taken for a single image but considering the whole class. Artefacts may therefore be detected and reduced as decisions for a single image are verified considering the remaining images of the class.

After an initial segmentation and alignment of the images, representational decisions, for example whether a branch is a spurious branch or not, can be verified using an image acquired at a later time. Figure 5 shows such a temporal sequence of root images. In Figure 5a and 5b two small branches that may be identified as noise in a single image are indicated by red arrows. The later images in Figure 5b and 5c identify these small branches as true branches (indicated by the green tickmark). In this case the small branches would be kept in the final representations.

Co-analysis may further be used in the development of representations of a whole class of data (co-representation) instead of a single data sample. The reduction of a class of data to its characteristic properties provides a general representation of the whole class of data. Such a co-representation can for example be given by a graph that represents the properties valid for the whole class of data. For shape representation geometric graphs may be used. Especially when analysing and representing the content of an image, a node may be assigned to a pixel, therefore geometry is implicitly represented. In order to use a graph as a co-representation of a class of data, we may represent each data sample using a geometric graph. However, for the general graph derived as a co-representation the graphs representing single data samples may be analysed for common topological structures which are represented in the final co-representation graph. Geometric properties are in this case disregarded in the co-representation.
However, for a developing shape the representation of the latest acquisition may be taken as the co-representation for the whole sequence. In this way geometric properties can be kept in the representation and further data samples can be mapped to the co-representation.

5. Extensions to Skeletons

Skeletons (medial axes) given as a thinned version of a shape with equal distance to the boundary, are widely used shape descriptors. In order to extend the definition and representational power of such a common skeleton, we propose a combination of skeletons and model fitting.

The contemplated approach works as follows:

1. The medial axis is computed for the shape first. On the obtained skeleton a constrained distance transform is performed - the geodesic distance along the skeleton is computed in this way.

2. To allow for the fitting of simple models, the axis needs to be straightened first. Therefore, the medial axis is decomposed into single curve segments at branching points. For each pixel along the skeleton the distance to the starting point of the segment (and further to a starting point of the whole skeleton) as well as the distance to the boundary (the radius of the maximal inscribed circle at this position) are known.

3. Simple models as a parabola, an ellipse, a cylinder or similar (higher order e.g. super-quadrics) models can be fitted to the transformed data.

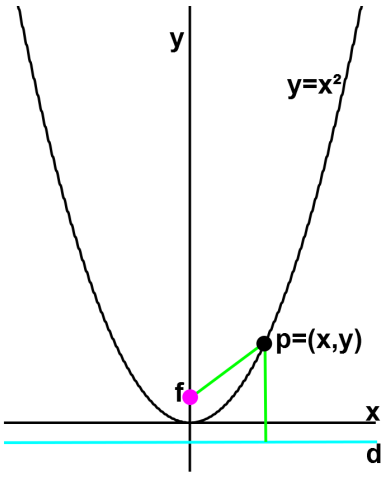


Figure 6: Example parabola, together with focus f and directrix d.

4. Fitted models may further be back-projected to the original domain.

We describe the process of model fitting in more detail using a parabola:

A parabola is defined as the locus of points equidistant from one point (the focus) and one line (the directrix). Figure 6 shows an example.
The straightened skeleton serves as the axis of symmetry of the parabola to be found. In order to fit this parabola to the data points two positions along the axis of symmetry need to be determined:

a. position f along the axis: position of the focus

b. position d along the axis: position of the directrix

For any point $p = (x, y)$ of the parabola the following equation holds:

$$\sqrt{x^2 + (f - y)^2} = |d - y| \qquad (1)$$

Reformulating equation 1 yields the dependency of x from y as follows:

$$x(y)^2 = (d - f)(d + f - 2y) \qquad (2)$$

We determine f and d by minimizing the sum of squared errors between the actual measured values x_i^2 (x_i corresponds to the radius stored with each skeleton pixel) and the value $x(y_i)^2$ given by a model parabola as formulated in equation 2, thus determine the parameters of an optimal fitted parabola.

Skeletons enhanced in this way may amongst others be used to:

- detect artefacts in an image segmentation;
- segment an object into meaningful parts based on fitted models;
- represent a particular shape or its properties using the parameters of the fitted model.

Applications for the above mentioned enhanced skeletons can for example be found in the analysis of biological data: Roots due to their elongated shape and narrow root tip can be approximated by a parabola. The parabola model may be used to improve the segmentation of an image into root region an background. Such a segmentation tends to introduce artefacts due to for example root hairs that falsify the segmented regions. Figure 7 illustrates an example of a parabola fitted to a root branch, according to the described approach. Additionally, to an improvement of the segmentation, the parabola parameters themselves may be used to describe the root and to model its growth.
Another application for enhanced skeletons is presented by the segmentation of 3D objects into rigid

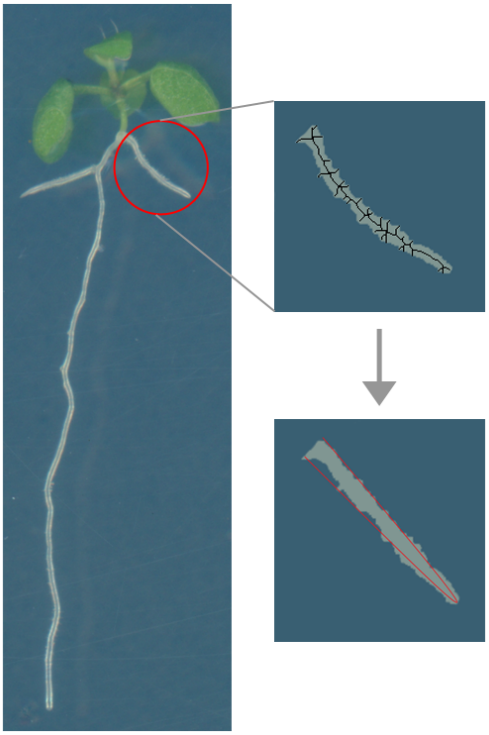


Figure 7: Root segment straightened according to a medial axis and parabola model fitted to the root tip (illustrative model).

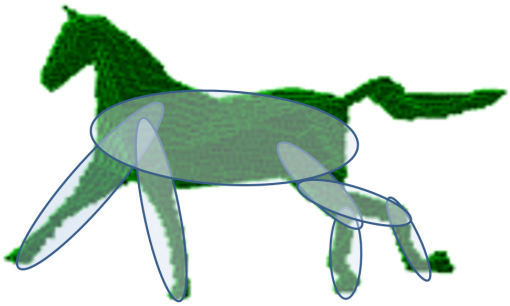


Figure 8: Representation of rigid parts of a horse using fitted ellipses (illustrative model).

parts. Instead of basing this segmentation on a simple medial axis alone, a 3D model of for example a horse can be segmented into rigid parts, by fitting ellipsoids to the individual parts. Elongated shapes as for example the torso of a horse may be better represented by ellipsoids than by spheres which are fitted for the medial axis representation. Moreover, the focal points of the fitted ellipsoids can be overlapped in the individual rigid parts, thereby representing connections between parts (in the horse model case these connections are the joints). Figure 8 illustrates this example.

6. Conclusion

The presented approaches are novel concepts and extensions to the current state-of-the-art. Common approaches for the extraction and representation of structure (for example skeletons and Reeb graphs) are sensitive to noise and depend on the quality of the binary segmented input image. The concepts introduced in this paper aim at decreasing this sensitivity towards noise.

An initial segmentation and an initial skeleton representation may be improved by considering local features along the skeleton or by model fitting using straight segments of the skeleton as axis of symmetry of a model. Both approaches in the end provide compact representations of the input data. A potentially noise flawed segmentation can be used as an initial input as the contemplated approaches can cope with a rough segmentation - while a representation is found, the segmentation may simultaneously be improved.

We further extend the known co-analysis to a more general approach by aligning several data samples according to skeletons or graphs instead of an axis. Therefore, the limitation of the alignment to non-articulated objects only is revoked. Based on graphs or skeletons objects in varying poses may then be used in the co-analysis by aligning their rigid parts.

For well aligned data samples over time we propose to add the temporal information as an additional dimension. A Reeb graph representation may in this case be built by using a height function along the time axis and by tracing back the evolution of the connected components.

These separate ideas may be joined together to improve representations of structures in spatio-temporal data: Robust skeleton representations may be used as an alignment of several data samples respectively their contained structure. Such aligned data samples in turn build the input for a Reeb graph representation over time, representing temporal changes in the structure. As well as for co-analysis which may provide a co-representation - a representation of a whole class of objects. For comparison of data samples, new samples may then be mapped to the co-representation and compared with the class. Furthermore, an initial representation may be improved by fitting a model to it. This model may even be back projected to the input data thereby also improving the segmentation.

Future work includes the implementation of these presented ideas and evaluation on (for example the mentioned biological roots and horses) data sets. An investigation of further approaches to overcome the discussed open problems in the recognition and representation of structures in spatio-temporal data are as well subject to future work.

Acknowledgements

We thank the anonymous reviewers for their constructive comments.

References

[1] S. Biasotti, D. Giorgi, M. Spagnuolo, and B. Falcidieno. Reeb graphs for shape analysis and applications. *Theoretical Computer Science*, 392(1-3):5–22, Feb. 2008. 1

[2] G. S. di Baja, L. Serino, and C. Arcelli. 3d curve skeleton computation and use for discrete shape analysis. In *Innovations for Shape Analysis*, pages 117–136. Springer, 2013. 1

[3] H. Doraiswamy and V. Natarajan. Efficient algorithms for computing Reeb graphs. *Computational Geometry*, 42(67):606–616, Aug. 2009. 3

[4] T. Druml, A. Gabdulkhakova, N. Artner, G. Brem, and W. Kropatsch. The use of image data in the assessment of equine conformation – limitations and

solutions. In R. B. Fisher, J. Hammal, B. Boom, and C. Spampinato, editors, *ICPR 2014 workshop on Visual observation and analysis of Vertebrate And Insect Behavior*, page in press, 2014. 1

[5] H. Edelsbrunner, J. Harer, A. Mascarenhas, V. Pascucci, and J. Snoeyink. Time-varying reeb graphs for continuous space–time data. *Computational Geometry*, 41(3):149–166, 2008. 3

[6] A. Golovinskiy and T. Funkhouser. Consistent segmentation of 3d models. *Computers & Graphics*, 33(3):262–269, 2009. 2, 4

[7] R. Gonzalez-Diaz, W. Kropatsch, M. Cerman, and J. Lamar. Characterizing configurations of critical points through lbp. In *SYNASC 2014 Workshop on Computational Topology in Image Context*, 2014. 3

[8] B. Jähne. *Spatio-temporal image processing: theory and scientific applications*, volume 751. Springer, 1993. 1

[9] I. Janusch, W. G. Kropatsch, and W. Busch. Reeb graph based examination of root development. In *Proceedings of the 19th Computer Vision Winter Workshop*, pages 43–50, Feb 2014. 1

[10] I. Janusch, W. G. Kropatsch, W. Busch, and D. Ristova. Representing roots on the basis of reeb graphs in plant phenotyping. In *ECCV 2014 Workshop on Computer Vision Problems in Plant Phenotyping*, page in press, 2014. 1

[11] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005. 3

[12] I. Laptev and T. Lindeberg. Space-time interest points. In *Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV03)*, pages 432–439, 2003. 3

[13] D. T. Lee. Medial axis transformation of a planar shape. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-4(4):363–369, July 1982. 3

[14] D. Leitner, B. Felderer, P. Vontobel, and A. Schnepf. Recovering root system traits using image analysis exemplified by two-dimensional neutron radiography images of lupine. *Plant Physiology*, 164(1):24–35, 2014. 3, 4

[15] N. J. Mitra, M. Wand, H. Zhang, D. Cohen-Or, V. Kim, and Q.-X. Huang. Structure-aware shape processing. In *ACM SIGGRAPH 2014 Courses, SIGGRAPH '14*, pages 13:1–13:21, 2014. 2, 4

[16] T. Ojala and M. Pietikäinen. Unsupervised texture segmentation using feature distributions. *Pattern Recognition*, 32(3):477–486, 1999. 2

[17] M. Pietikäinen. Image analysis with local binary patterns. In *Image Analysis*, pages 115–118. Springer, 2005. 2

[18] O. van Kaick, K. Xu, H. Zhang, Y. Wang, S. Sun, A. Shamir, and D. Cohen-Or. Co-hierarchical analysis of shape structures. *ACM Transactions on Graphics (TOG)*, 32(4):69, 2013. 2, 4

[19] Y. Wang, S. Asafi, O. van Kaick, H. Zhang, D. Cohen-Or, and B. Chen. Active co-analysis of a set of shapes. *ACM Transactions on Graphics (TOG)*, 31(6):165, 2012. 4

[20] N. Werghi, Y. Xiao, and J. Siebert. A functional-based segmentation of human body scans in arbitrary postures. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 36(1):153–165, 2006. 2

[21] G. Zhao and M. Pietikäinen. Dynamic texture recognition using local binary patterns with an application to facial expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(6):915–928, 2007. 3

20^{th} *Computer Vision Winter Workshop*
Paul Wohlhart, Vincent Lepetit (eds.)
Seggau, Austria, February 9-11, 2015

Signature Matching in Document Image Retrieval

Thomas Schulz and Robert Sablatnig
Computer Vision Lab
Vienna University of Technology
tschulz@caa.tuwien.ac.at

Abstract. *Document image retrieval is a method used for searching through unsorted images of documents to find the ones which are relevant for a given task. This paper presents an approach towards document image retrieval using handwritten signatures as queries. For this purpose a matching algorithm is combined with a pre-filtering method that minimizes the search space. The matching is done using four distance measures which are computed from a Thin-Plate Spline (TPS) transformation and the pre-filtering is based on the shape context distance. The approach is evaluated on a subset of the GPDS960signature database where it is shown that the proposed pre-filtering step results in a significant speed-up factor of 16, as well as slightly better retrieval performance.*

1. Introduction

To analyse libraries of unsorted documents it is helpful to be able to automatically find documents which meet certain criteria (e.g. only documents with handwritten text). In this context there is also interest in finding documents which were authored or authorized by a specific person. An effective means for doing this is the use of signature matching techniques [1, 17].

There is a distinction between offline and online signature matching, where online means that the signature is captured using an electronic device that also captures temporal information about the stroke sequence. In offline signature matching, on the other hand, no electronic device is needed to record the stroke sequence, however, only static information is available for matching [1].

Signature matching is used in areas such as verification [15], identification [13] and retrieval [12]. While signature verification deals with confirming the authenticity of a signature and signature identification tries to find the corresponding author [11], signature retrieval aims to find document images that contain signatures from a specific individual [12]. The differences between the three categories are illustrated in Figure 1. It shows the respective problems that have to be solved for signature verification (left), identification (middle) and retrieval (right).

An early signature retrieval approach by Han and Sethi [8] uses string representations which encode the order of occurrences of events such as branch and crossing points in x and y direction. They compute the *Longest Common Subsequence* (LCS) between the strings which represent the query signature and the strings of the signature images in the data set in order to find the best matches.

Srihari et al. [14] use *Gradient, Structural and Concavity* (GSC) features to capture the image characteristics at local, intermediate and large scales. The resulting binary feature vector is used for signature retrieval by computing distances via a normalized correlation similarity measure.

Zhu et al. [17] propose a signature detection and matching system for document image retrieval that uses analysis of salient structures to locate the signatures in the documents and performs matching using a combination of four distance measures. It is evaluated on the *Tobacco-800* database [9] where it achieves a *Mean Average Precision* (MAP) of 90.5% and a *Mean R-Precision* (MRP) of 86.8%.

Belongie et al. [2] propose the shape context descriptor and the related shape context distance to describe the similarity of shapes and thus help their matching. Lin and Chang [10] extend this method with an indexing approach to minimize the search space which yields a speed-up of factor 5.

The main contribution of this paper is the combination of a *Thin-Plate Spline* (TPS) approach [17]

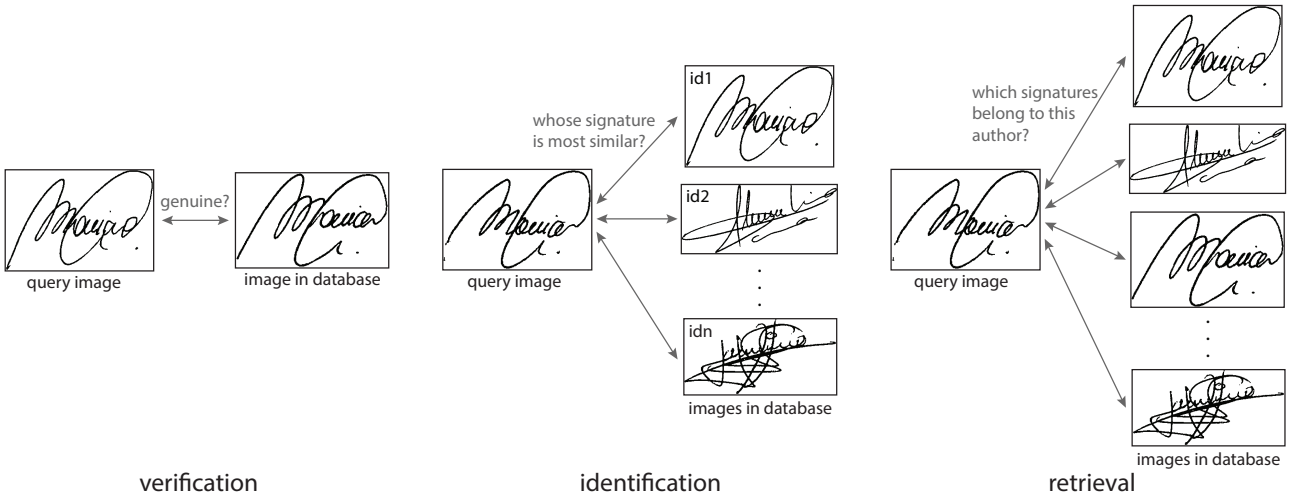


Figure 1. An illustration of the differences between the three areas of application for signature matching. Figure inspired by [11].

with a shape-context-based pre-filtering step to reduce the runtime. Due to the computational load of the approach and the fact that the dissimilarity measures have to be computed for the entire test set for each new query signature, since they depend on the transformations between the query signature and the candidate signatures, the approach becomes infeasible for large datasets (see runtime estimation in Table 1). The runtime reduction achieved by the hybrid approach proposed in this paper therefore extends the retrieval system such that it can be used for large sets of signature images.

A complete document image retrieval system also requires the localization of the signature in the document and its segmentation. This paper, however, only deals with the matching and retrieval part of such a system.

The outline of this paper is as follows. Section 2 describes our document image retrieval algorithm, Section 3 presents the results and their evaluation, and Section 4 concludes the paper.

2. Methodology

The signature retrieval system proposed in this paper is mainly based on the methods presented by Zhu et al. [17] but also introduces modifications that result in reduced computational time and increased matching performance. The main difference is the use of a shape-context-based pre-filtering step that reduces the computational time on a set of 960 signature images by a factor of 16.

First the data are preprocessed similar to the approach of Lin and Chang [10]. In this step the signature images are rotated such that the major axis of the *Best-Fit Ellipse* (BFE) is aligned with the horizontal axis. Then the image is trimmed to fit the size of the signature. Subsequently the image is resized to normalize the length of the diagonal. The point set which represents the signature in the remainder of the algorithm is created by randomly sampling points on the abstract representation of the signature image which is obtained through Canny edge detection [6] or skeletonization (see Section 2.4). An example of the preprocessing steps is given in Figure 2.

Once the data are normalized, the shape context descriptor [2] is computed for each point set which is used to compute the shape context distance to the remaining signature images in the test set. This distance is used in the following pre-filtering step to decide whether the image is processed further or not. In the former case the TPS transformations which best map the point set to the point sets of the other images are computed. Each TPS transformation is then used to compute four distance measures which accumulate to the overall distance of the signature to another in the test set through a weighted sum. The weights that are used to combine the distance measures are obtained using *Linear Discriminant Analysis* (LDA). The retrieval is finally performed by ranking the shape context distances of the filtered images and the combined distance measures of the remaining images. The workflow of the signature retrieval system is illustrated in Figure 3 where the second row depicts the steps that are only performed for the signatures that remain after the pre-filtering step.

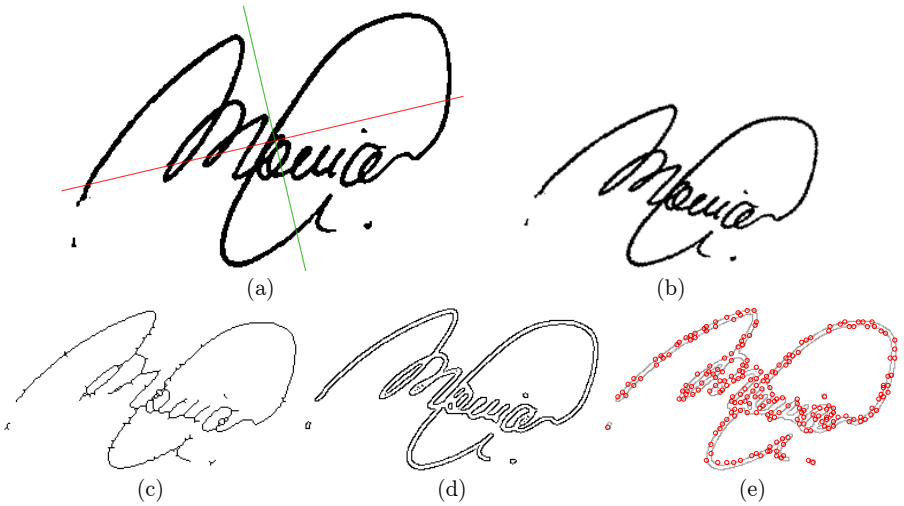


Figure 2. (a) Original signature image with major (red) and minor (green) axis of the BFE. (b) The same image after size and orientation normalization. (c) The skeleton of the normalized image. (d) The edges of the normalized image. (e) Points sampled on the edge image.

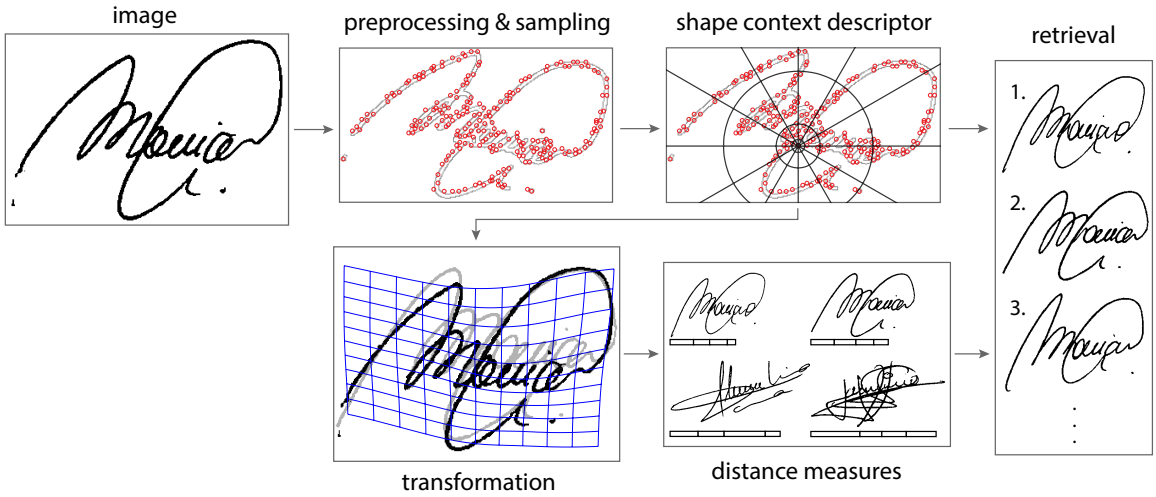


Figure 3. The workflow of the signature retrieval system. The steps in the second row are only performed for the signatures which remain after filtering. The results of the steps in both rows are combined in the retrieval step.

2.1. Thin-Plate Spline – Robust Point Matching Algorithm

The transformation from one signature image to another is computed using the *Thin-Plate Spline – Robust Point Matching* (TPS–RPM) algorithm [7]. A TPS is able to model affine and non-rigid transformations such that they can be separated [4]. It is commonly used for describing flexible transformations [2] which is why it is also applied to handwritten character and signature matching.

The TPS transformation of a point set V in homogeneous coordinates is given as

$$f(V) = Vd + \Phi w, \qquad (1)$$

with the TPS parameters d, w and the TPS kernel matrix Φ. The results of this algorithm are illustrated in Figure 4 using the point sets of two signatures.

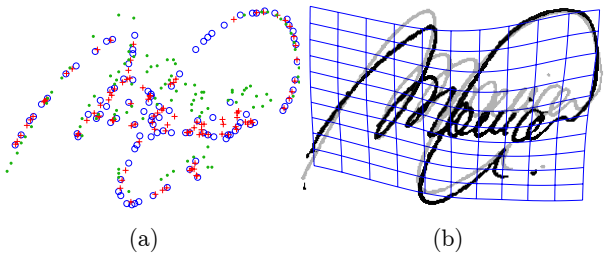


Figure 4. (a) The results of the TPS–RPM algorithm for finding a transformation from point set V (green dots) to X (blue circles). The transformed points $f(V)$ are shown as red crosses. (b) The original signatures from which the point sets are sampled together with the TPS transformation which is represented by a blue grid.

2.2. Dissimilarity Measures

Once the transformation between the query signature and a candidate signature is known, we use it to compute four distance measures as proposed by Zhu et al. [17]; namely the bending energy D_{be}, the shape context distance D_{sc}, the anisotropic scaling D_{as} and the registration residual error distance D_{re}. They are accumulated into the final distance D using the weighted sum

$$D = w_{be}D_{be} + w_{sc}D_{sc} + w_{as}D_{as} + w_{re}D_{re}, \quad (2)$$

where the weights w are estimated via LDA on a random subset of signature images that are not in the test set [17].

2.2.1 Bending Energy

When a TPS is used as a transformation for two-dimensional point matching, the amount of energy that is necessary to deform it such that one point cloud matches the other can be used as an indicator for the quality of the match. This energy – the so-called integral bending norm – is a measure proposed by Bookstein [4] which relates to the amount of non-affine deformation in the transformation. We use the variant of this norm which was proposed by Chui and Rangarajan [7] as

$$D_{be} = \lambda \cdot \mathrm{trace}(w \cdot Y^\top), \quad (3)$$

where λ is the smoothness constraint, w is the TPS parameter describing the non-affine part of the transformation (see Equation 1) and Y is the transformed point set V.

2.2.2 Shape Context

The shape context descriptor [2] is a rich descriptor of the shape of a point set that describes the appearance of the shape. It is computed for each point and represented by a log-polar histogram of lengths and orientations of connecting lines among the points in the set. This representation effectively describes the structural relation of one point to the other points in the set and is therefore used to evaluate the quality of a match.

This descriptor is used to compute the shape context distance D_{sc} between a set P from a query signature with m points and a set Q from a candidate signature with n points as stated in [2]:

$$D_{sc}(P,Q) = \frac{1}{m}\sum_{p\in P} \arg\min_{q\in Q} C(f(p),q) + \frac{1}{n}\sum_{q\in Q} \arg\min_{p\in P} C(f(p),q), \quad (4)$$

where f is the TPS transformation given in Equation 1 and C is the matching cost for two points, defined using the χ^2 test statistic:

$$C(p,q) = \frac{1}{2}\sum_{k=1}^{K} \frac{[h_p(k) - h_q(k)]^2}{h_p(k) + h_q(k)}, \quad (5)$$

where h_p and h_q are the shape context histograms of points p and q, and k specifies the bin with a total number of K bins.

2.2.3 Anisotropic Scaling

The anisotropic scaling is a ratio that measures the isotropy of the scaling in the transformation. It is computed directly from the affine transformation matrix d (see Equation 1) and is defined in [17] as

$$D_{as} = \log \frac{\max(S_x, S_y)}{\min(S_x, S_y)}, \quad (6)$$

where S_x, S_y are obtained by singular value decomposition of d. Here S_x, S_y are the scaling factors of the affine part of the TPS transformation. Thus D_{as} is 0 if there is only isotropic scaling in d (i.e. $S_x = S_y$).

2.2.4 Registration Residual Error

The last distance measure proposed by Zhu et al. [17] is the residual error of the estimated transformation. It describes the quality of the matching by computing the sum of Euclidean distances between corresponding points, normalized by the total number of matches. For a matching assignment M_i it is defined as

$$D_{re}^H = \frac{\sum_{i=1}^{\min(m,n)} ||f(p_i) - q_{M_i}||}{\min(m,n)}, \quad (7)$$

where f is the TPS transformation given in Equation 1 and m, n are the sizes of point sets P and Q respectively.

However, since this formula requires one-to-one correspondences and the TPS–RPM algorithm yields only soft matches (i.e. continuous values in the correspondence matrix instead of binary ones) we use a

different implementation that computes the registration residual error by weighting it with the matching quality from the correspondence matrix of the TPS–RPM algorithm. It is defined as

$$D_{re}^{W} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} M_{ij} \cdot ||f(p_i) - q_j||}{\min(m,n)}. \quad (8)$$

where M is the correspondence matrix of the TPS–RPM algorithm.

2.3. Pre-Filtering

Since the dissimilarity measures are computed from the transformation that best maps a query signature to a candidate signature, the time-consuming TPS–RPM algorithm has to be computed for the entire test set for each new query signature. Therefore it is suggested in this paper to speed up the retrieval process by first reducing the search space. This is done by computing the shape context distance from a query signature to all other signature images in the test set similar to Equation 4 but without prior computation of the transformation (i.e. $f(p) = p$). The results are then sorted and the expensive TPS–RPM algorithm and the dissimilarity measures are computed for only 3% of the highest ranked signatures. The remaining signatures are ranked according to their shape context distance.

2.4. Hybrid Approach

Our experiments show that the shape-context-based pre-filtering step achieves the best results using skeleton images which can be explained by the fact that edge images consist of two edges for each stroke instead of one. Since the shape context descriptor gives more importance to points in close proximity, edge images add potential for noise by having points sampled on both edges of a stroke. The dissimilarity measures on the other hand perform best on Canny edges which matches the observations of Zhu et al. [17]. Regarding the optimal number of sample points the best trade-off between retrieval performance and runtime is achieved when sampling about 200 points for the dissimilarity measures and about 350 points for the pre-filtering step. Sampling more points increases the retrieval performance, however, the runtime increases exponentially. We therefore suggest to use a hybrid approach which performs the pre-filtering step on skeleton images and computes the dissimilarity measures on edge images sampling about 350 and 200 points respectively.

3. Results

The evaluation is done in Matlab on a subset of the *GPDS960signature* database [3]. This database contains binary images of 24 genuine signatures from 960 individuals. Since the computation of the TPS–RPM algorithm and of the dissimilarity measures takes about 2.6 seconds for a single comparison without parallelization (i.e. about 16.6 hours for the evaluation of one query signature on the entire dataset of 960 signers and 24 signatures) an evaluation on the entire set is not feasible (see Table 1). The tests in this section are therefore conducted on a subset of 960 signature images. This set is assembled by simply taking the first 8 signatures of the first 120 individuals in the *GPDS960signature* database. The evaluation is parallelized on six cores to further reduce the runtime.

Method	Test set	Full set
without pre-filtering	17 days	11 years
with pre-filtering	1 day	2 years

Table 1. Comparison of estimated runtimes for a complete evaluation on different sets using parallelization for speed-up.

The performance of the document image retrieval system is evaluated using the same measures as in [17], namely *Average Precision* (AP) and *R-Precision* (RP). The precision of a retrieval system is computed as

$$\text{precision} = \frac{\text{\# of relevant documents retrieved}}{\text{\# of documents retrieved}}. \quad (9)$$

AP is the mean of the precisions at each rank that adds another relevant document, with a precision of zero for relevant documents that are not retrieved [5]. This means that the AP of a retrieval of a total of 3 relevant documents, where only 2 documents are found at positions 1 and 5, is given as $AP = (1/1 + 2/5 + 0)/3 = 46.7\%$. RP is the precision for retrieving R documents where R is the number of relevant documents for the given query. Thus the RP for the example given above is $RP = 1/3 = 33.3\%$. AP rewards higher rankings of relevant documents and penalizes that of irrelevant ones while RP ignores the exact ranking of the results and is more useful when a large amount of relevant documents is present in the dataset [17].

All test runs are conducted using each signature in the test set as query and removing it from the

set for this run. The average of the results for each query signature is then presented as the *Mean Average Precision* (MAP) and the *Mean R-Precision* (MRP). Some of the results are also illustrated by plotting the average recall at each rank. The recall of a retrieval system is defined as

$$\text{recall} = \frac{\text{\# of relevant documents retrieved}}{\text{\# of relevant documents}}. \quad (10)$$

3.1. Comparison with Zhu et al.

Since Zhu et al. [17] evaluate the dissimilarity measures on a different dataset, namely the *Tobacco-800* [9] set which consists of real world documents from US tobacco companies, their results cannot directly be compared to the results in this paper. For this reason both the dissimilarity measures on their own and the hybrid approach using the dissimilarity measures with the pre-filtering step are evaluated on the test set to see how they perform in comparison. Regarding the size of the dataset used by Zhu et al. they state that *Tobacco-800* contains 66 classes with 6-11 signatures each, which results in 396-726 signatures in total. Since 20% are used as training data this leaves 317-581 signatures that are left as test data. The test set used in their evaluation is therefore smaller than our test set. The reason why we evaluate our signature retrieval system on a different set is that we do not have access to the *Tobacco-800* dataset.

The results in terms of MRP and MAP are visualized in Figure 5 (a) and a comparison of the recall of both methods at each rank is given in Figure 5 (b). The exact values including the total runtime of the experiments are shown in Table 2

Method	MRP	MAP	Runtime
DMs	62.4%	66.9%	16.71 days
Hybrid (3%)	64.0%	67.8%	0.99 days
Hybrid (5%)	64.3%	68.2%	1.27 days

Table 2. Retrieval performances and runtime of the Dissimilarity Measures (DMs) and the hybrid approach with a reduced set size of 3% and 5%.

The results show that the hybrid approach with a reduced set of 3% provides a speed-up of factor 16 on the test set and even achieves slightly better retrieval results in terms of MRP and MAP than the dissimilarity measures on their own. It can be seen in Figure 5 (b), however, that the hybrid approach has a lower recall rate when about 20-80 signatures are retrieved which means that the dissimilarity measures are more likely to rank relevant signatures at these positions than the hybrid approach. This effect occurs due to the reduced set which contains only 29 signatures in this case and can be reduced by increasing its size. Using the hybrid approach with a reduced set of 5% still provides a speed-up of factor 13 and achieves a 1.9 percentage points higher MRP and a 1.3 percentage points higher MAP compared to the dissimilarity measures.

3.2. Training Data

As mentioned in Section 2.2 the dissimilarity measures are combined using pre-computed weights, however, they can also be combined without using training data by normalizing each distance measure with its standard deviation:

$$D = \frac{D_{be}}{\sigma_{be}} + \frac{D_{sc}}{\sigma_{sc}} + \frac{D_{as}}{\sigma_{as}} + \frac{D_{re}}{\sigma_{re}}. \quad (11)$$

This section evaluates the impact of using training data on the retrieval performance. For this purpose the hybrid approach with a reduced set of 3% and the dissimilarity measures are both evaluated on the test set with and without weights. The weights are obtained using 25% of training data which are randomly selected from the signatures that are not in the test set. The actual trained weights that are used for the comparison are shown in Table 3. The results of this test in terms of MRP and MAP are shown in Figure 6 and Table 4.

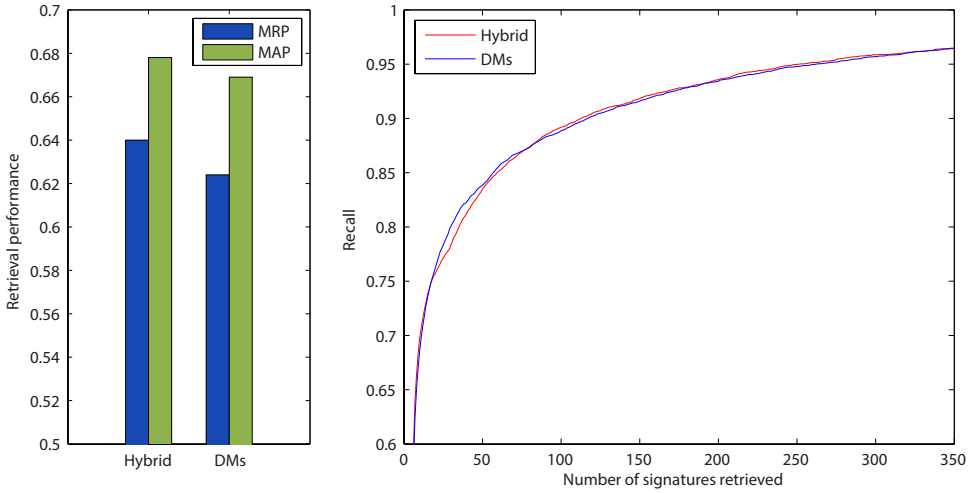
w_{be}	w_{sc}	w_{as}	w_{re}
52.99	0.1104	2.159	1,057

Table 3. Weights that are used to combine the dissimilarity measures

Rates	with weights		without weights	
	Hybrid	DMs	Hybrid	DMs
MRP	64.0%	62.4%	63.7%	62.6%
MAP	67.8%	66.9%	67.7%	66.9%

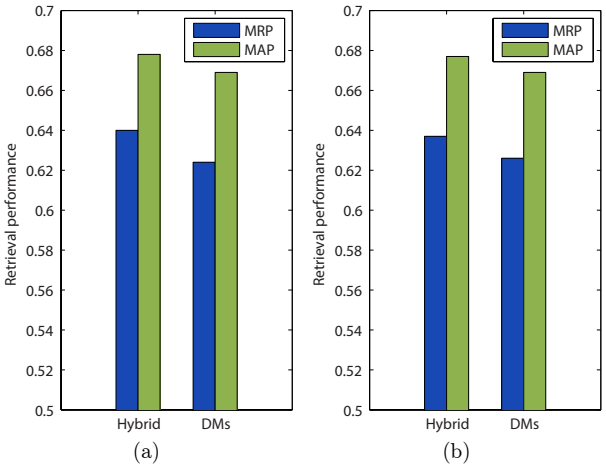
Table 4. Retrieval performances with and without weights for the hybrid approach with a reduced set of 3% and the Dissimilarity Measures (DMs).

It can be seen that the hybrid approach achieves better results with weights than without weights. The results also show that it is possible to obtain only slightly lower performance rates without using any



(a) (b)

Figure 5. The results (a) in terms of MRP and MAP and (b) the average recall of the hybrid approach with a reduced set size of 3% (red) and the dissimilarity measures (blue).



(a) (b)

Figure 6. The retrieval performances in terms of MRP and MAP (a) with weights and (b) without weights for the hybrid approach with a reduced set of 3% and the dissimilarity measures.

training data than with 25% training data. To be precise, the MRP and MAP of the hybrid approach with weights are only 0.3 and 0.1 percentage points higher than without weights.

The results of the dissimilarity measures show even better performance without using weights than the hybrid approach. They achieve a 0.2 percentage points higher MRP and the same MAP without training data as with 25% training data. These results suggest that it is not mandatory for the dissimilarity measures and the hybrid approach to use training data since it reduces the size of the test set. However, the *GPDS960signature* database is several times larger than our test set which means that enough training data is available. The results in this paper are therefore computed using weights.

3.3. Single Distances

In this section we give an overview of the performance of single distances similar to Zhu et al. [17]. The results for the dissimilarity measures and the hybrid approach using single distances on their own are presented in Figure 7 and Table 5.

Distance	DMs		Hybrid (3%)	
	MRP	MAP	MRP	MAP
D_{be}	23.9%	25.9%	56.7%	60.5%
D_{sc}	45.3%	48.8%	55.0%	59.5%
D_{as}	11.0%	13.1%	36.9%	40.5%
D_{re}	33.0%	34.8%	59.9%	63.8%

Table 5. Retrieval performances of single distances for the Dissimilarity Measures (DMs) and the hybrid approach with a reduced set of 3%.

Firstly it can be seen that the order in terms of retrieval performance is different for the two approaches. While for the dissimilarity measures the shape context distance (D_{sc}) performs best, followed by the registration residual error (D_{re}), the bending energy (D_{be}) and the anisotropic scaling (D_{as}), it is D_{re} which performs best for the hybrid approach followed by D_{be}, D_{sc} and D_{as}. The only similarity here is that D_{as} performs worst for both approaches. Comparing the results of the dissimilarity measures to those of Zhu et al., it is also worth noting that D_{re} and D_{as} swapped their position due to the per-

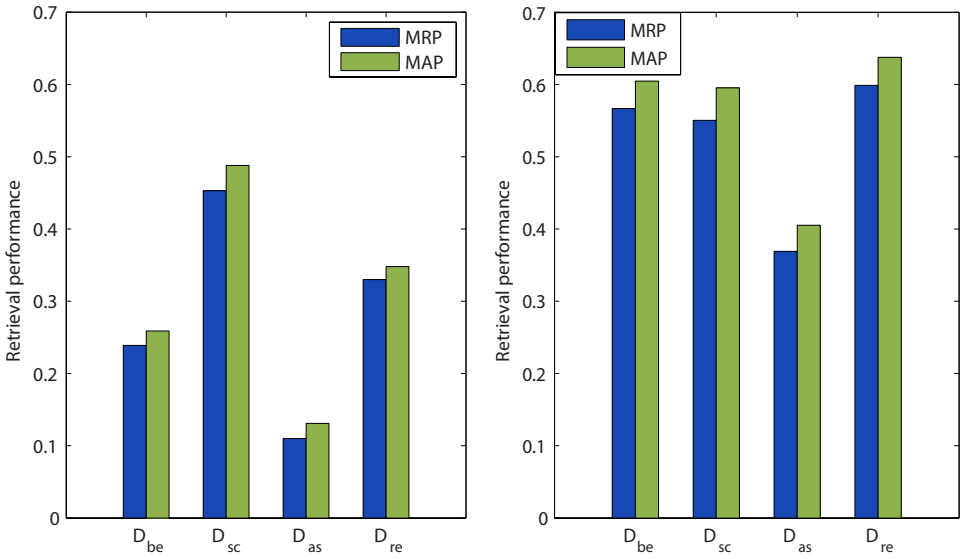


Figure 7. The retrieval performance of single distances in terms of MRP and MAP for (a) the dissimilarity measures and (b) the hybrid approach with a reduced set of 3%.

formance gain from using the weighted registration residual error implementation (D_{re}^{W}).

Secondly the results show that the retrieval performance for single distances is significantly higher (i.e. up to 34.6 percentage points for D_{be}) for the hybrid approach than for the dissimilarity measures. This can be explained by the fact that each distance profits from the pre-filtering step used in the hybrid approach, thus resulting in a better retrieval performance for each distance on its own.

4. Conclusion

In this paper a hybrid approach is proposed that combines a state-of-the-art document image retrieval method with a pre-filtering step. The proposed method first reduces the search space by filtering the test set based on the shape context distance. It then estimates the transformation from a query signature to a candidate using the TPS–RPM algorithm and uses this transformation to compute four dissimilarity measures which are combined to a final distance. The weights for combining the dissimilarity measures are estimated via LDA.

We show that the pre-filtering brings a significant speed-up while providing slightly better retrieval results than the dissimilarity measures on their own. The reason why the shape context distance is used to estimate correspondences is that after the normalization of the images in the preprocessing step similar signatures have a low shape context distance even without knowing the transformation between them.

Additional evaluations demonstrated that the use of training data has only a small effect on the retrieval performance which means that it is not mandatory to train the weights of the signature retrieval system. Finally, the comparison of the performance of single distance measures showed that each distance measure benefits from the pre-filtering step in the hybrid approach, thus achieving significantly better results than without the pre-filtering step.

Future work includes signature detection and preprocessing elements such as printed text removal and filtering of noise. If the system is extended to document image retrieval by adding a signature localization it is also recommendable to improve the TPS–RPM algorithm to support outlier handling in both point sets as proposed by [16] since real world documents contain more noise than the binarized signature images within the *GPDS960signature* database.

Acknowledgements

We would like to thank Miguel A. Ferrer for letting us use the *GPDS960signature* database for our evaluation.

References

[1] G. Agam and S. Suresh. Warping-Based Offline Signature Recognition. *IEEE Transactions on Information Forensics and Security*, 2(3):430–437, Sept. 2007. 1

[2] S. Belongie, J. Malik, and J. Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, Apr. 2002. 1, 2, 3, 4

[3] M. Blumenstein, M. A. Ferrer, and J. F. Vargas. The 4NSigComp2010 Off-line Signature Verification Competition: Scenario 2. In *12th International Conference on Frontiers in Handwriting Recognition*, pages 721–726. IEEE, Nov. 2010. 5

[4] F. Bookstein. Principal Warps: Thin-Plate Splines and the Decomposition of Deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(6):567–585, June 1989. 3, 4

[5] C. Buckley and E. M. Voorhees. Evaluating Evaluation Measure Stability. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 33–40, New York, New York, USA, July 2000. ACM Press. 5

[6] J. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, Nov. 1986. 2

[7] H. Chui and A. Rangarajan. A New Point Matching Algorithm for Non-Rigid Registration. *Computer Vision and Image Understanding*, 89(2-3):114–141, Feb. 2003. 3, 4

[8] K. Han and I. K. Sethi. Handwritten Signature Retrieval and Identification. *Pattern Recognition Letters*, 17(1):83–90, Jan. 1996. 1

[9] D. Lewis, G. Agam, S. Argamon, O. Frieder, D. Grossman, and J. Heard. Building a Test Collection for Complex Document Information Processing. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 665, New York, New York, USA, Aug. 2006. ACM Press. 1, 6

[10] C. Lin and C. Chang. A Fast Shape Context Matching Using Indexing. In *International Conference on Genetic and Evolutionary Computing*, pages 17–20. IEEE, Aug. 2011. 1, 2

[11] I. Pavlidis, N. Papanikolopoulos, and R. Mavuduru. Signature Identification Through the Use of Deformable Structures. *Signal Processing*, 71(2):187–201, Dec. 1998. 1, 2

[12] M. S. Shirdhonkar and M. B. Kokare. Document Image Retrieval Using Signature as Query. In *International Conference on Computer and Communication Technology*, pages 66–70. IEEE, Sept. 2011. 1

[13] M. S. Shirdhonkar and M. B. Kokare. Off-line Handwritten Signature Identification Using Rotated Complex Wavelet Filters. *Journal of Computer Science*, 8(1):478–482, 2011. 1

[14] S. N. Srihari, S. Shetty, S. Chen, H. Srinivasan, C. Huang, G. Agam, and O. Frieder. Document Image Retrieval Using Signatures as Queries. In *International Conference on Document Image Analysis for Libraries*, pages 198–203. IEEE, 2006. 1

[15] J. Vargas, M. Ferrer, C. Travieso, and J. Alonso. Off-line Signature Verification Based on Grey Level Information Using Texture Features. *Pattern Recognition*, 44(2):375–385, Feb. 2011. 1

[16] J. Yang. The Thin Plate Spline Robust Point Matching (TPS-RPM) Algorithm: A Revisit. *Pattern Recognition Letters*, 32(7):910–918, May 2011. 8

[17] G. Zhu, Y. Zheng, D. Doermann, and S. Jaeger. Signature Detection and Matching for Document Image Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):2015–2031, Nov. 2009. 1, 2, 4, 5, 6, 7

20^{th} *Computer Vision Winter Workshop*
Paul Wohlhart, Vincent Lepetit (eds.)
Seggau, Austria, February 9-11, 2015

Using Agglomerative Clustering of Strokes to Perform Symbols Over-segmentation within a Diagram Recognition System

Martin Bresler, Daniel Průša, Václav Hlaváč
Czech Technical University in Prague, Faculty of Electrical Engineering
Department of Cybernetics, Center for Machine Perception
166 27, Praha 6, Technická 2, Czech Republic
{breslmar, prusapa1, hlavac}@cmp.felk.cvut.cz

Abstract. *Symbol segmentation is a critical part of handwriting recognition. Any mistake done in this step is propagating further through the recognition pipeline. It forces researchers to consider methods generating multiple hypotheses for symbol segmentation – over-segmentation. Simple approaches which takes all reasonable combinations of strokes are applied very often, because they allow to achieve high recall rates very easily. However, they generate too much hypotheses. It makes a recognizer considerably slow. This paper presents our experimentation with an alternative method based on a single linkage agglomerative clustering of strokes with trainable distance metric. We embed the method into the state-of-the-art recognizer for on-line sketched diagrams. We show that it results in a decrease in the number of generated hypotheses while still reaching high recall rates. A problem emerges, since the number of bad hypotheses is still significantly higher than the number of symbols and it leads to unbalanced training datasets. To deal with it, we propose to train symbol classifiers with synthesized artificial samples. We show that the combination of these two improvements make the recognizer significantly faster and very precise.*

1. Introduction

Free hand writing and especially drawing are very natural ways how people express their thoughts. Devices allowing users to write and draw with a stylus directly on the surface of a displaying unit became very common. This functionality is in tablets, tablet PCs, or smart white boards. There is a great interest in systems capable to recognize this so called *ink input*. It is also called an on-line input and it is considered to be a sequence of handwritten strokes, where a stroke is a sequence of points captured by a device. Every point is always defined by its coordinates in the plane (drawing canvas). Additional data like a time stamp and a pressure value is usually provided as well. An output of a recognizer is a formal description of the input.

The research in handwritten document analysis and processing has moved from recognition of plain text to recognition of more structured inputs such as mathematical formulas, chemical formulas, music scores, or diagrams. Several recognizers of e.g. mathematical formulas with a good precision were presented in recent years [1, 9, 14]. Moreover, there is a contest in recognition of mathematical expressions [12]. In contrast, availability of diagram recognizers is still limited. The reason might be that there exists a vast of different diagrammatic domains, some of them not being well specified as mathematical formulas. However, there has been an effort to develop recognizers for electric circuits [8], chemical drawings [13], or flowcharts [5].

Although we showed that there exist numerous recognition systems specialized on various domains, they all face a common problem of symbols identification. Symbol segmentation is a crucial part of handwriting recognition where symbols are located in the input so they can be classified later. Ideally, the segmentation output would be disjoint subsets of strokes covering all the strokes. However, the segmentation can be barely done properly without knowledge of the whole structure. In practice, it is not wise to make hard decisions in this early step of the pipeline. A better approach is to perform so called *over-segmentation*. It supplies a larger number of subsets which typically share some strokes.

The final decision, which subsets fits the structure of the input diagram best, is left to the later phases performed by a structural analyzer.

It is important to achieve a high recall rate by the segmentation, which means that there are subsets of strokes representing ideally all of the symbols. Usually, simple over-segmentation methods based on intuitive assumptions that symbols comprise of strokes which are spatially and temporarily close are used. It considers all possible sequences of strokes up to some size. The segments are created iteratively and their number is limited by a maximal number of strokes and also by thresholds saying what is the maximal allowed distance or time difference between strokes in a segment. Variants of described approach are used in all the systems we introduced. Although it can achieve a high recall rate, it usually induces a very poor precision, because it simply generate too many bad hypotheses. Their consideration followed by rejection makes the whole recognition process time consuming.

We designed a diagram recognition system which uses exactly this approach to achieve the over-segmentation [4]. In this paper, we investigate different options which would allow to achieve still high recall rates and generate significantly less segmentation hypotheses and thus to increase the precision. Delaye and Lee [7] showed that objects of interest may be found using Single-Linkage Agglomerative Clustering (SLAC). It is a hierarchical bottom-up clustering where two closest clusters are merged together in each step until there is only one cluster remaining. Singleton clusters consisting of a single stroke are created first and bigger clusters are created iteratively. A link is created at each merging step and it contains information about two clusters it links and a distance between them. The resulting tree structure is called dendrogram and we can get the desired clusters by defining a suitable threshold to cut the tree. For illustration see Figure 1. In case of single-linkage the distance between two clusters is given by the distance between their two closest elements. The tricky part is to find a suitable measure defining a distance between two strokes. The authors use a set of simple features which basically express differences in geometric, spatial, and temporal characteristics of two strokes. The distance between two strokes is given by a weighted sum of these features. Obviously, each feature has different importance and thus it is necessary to find a suitable weights. They proposed an algorithm which is able to train the weights automatically from annotated data. The algorithm finds the best threshold to cut the dendrogram as well. They tested the whole approach on several domains and showed that this approach can find well defined symbols in flowcharts (FC), finite automata (FA), or mathematical expressions as well as loosely defined text blocks and figures in free hand sketches.

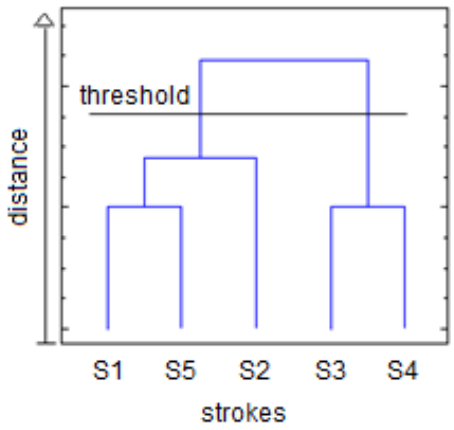
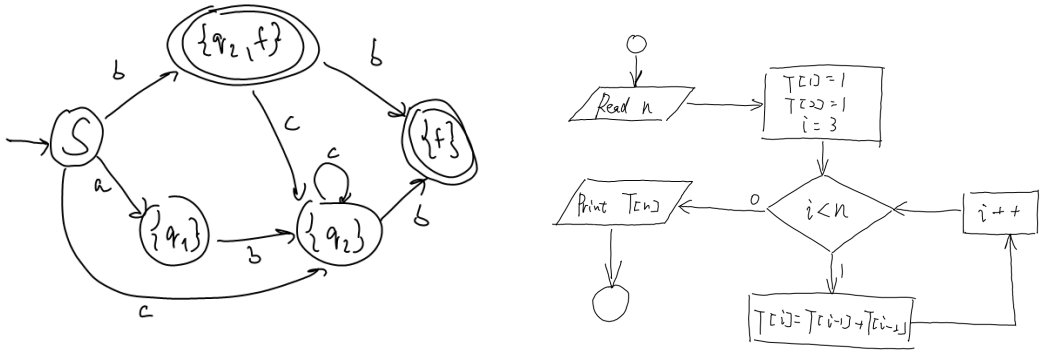


Figure 1: Illustrative example of a dendrogram and its cutting.

Delaye [6] later tried to classify the segmented clusters into corresponding symbol classes. He upgraded the proposed segmentation tool into a diagram recognizer. It is based on Conditional Random Fields (CRF), where the created clusters represent nodes of the graph. The author creates hierarchical model by applying several values of the threshold. The created graphs have a tree structure and thus the problem can be solved efficiently by the Belief Propagation algorithm. It makes the system extremely fast. However, it is a purely statistical approach which gives no further information about the diagram structure and it may produce inconsistent labelings.

There exist benchmark databases for FC [2] and FA [4] domains. We embed the SLAC method proposed by Delaye and Lee into our recognition system and compare the new results with our previous version of the system. We compare it with other two systems – the statistical recognizer by Delaye [6] and the grammar base recognizer by Carton e al. [5]. Examples of diagrams from the two mentioned database are shown in Figure 2.

The rest of the paper is organized as follows. We briefly describe our diagram recognizer in Section 2.



(a) (b)

Figure 2: Examples of diagrams from the two domains – (a) finite automata (FA) and (b) flowcharts (FC).

The proposed improvements are presented in Section 3. Experiments with the improved recognizer follow in Section 4. Finally, we make a conclusion in Section 5.

2. Diagram Recognition System

We have developed a recognition system for online sketched diagrams [4]. It is a general system so far adopted for domains of flowcharts and finite automata. The system consists of several steps of the recognition pipeline, which is depicted in Figure 3. The first one is the input normalization where the points are resampled to remove those points that are too close to each other. Text strokes are removed from the input by the text separator, which is based on the algorithm for mode detection by Phan and Nakagawa [16]. The removed text strokes are going to be put back later when the diagram structure is recognized to form text blocks attached to symbols. Next step is the symbols candidates detection. It is done by over-segmentation and classification of the created groups of strokes. Symbols are divided into two types: *uniform symbols* with relatively stable appearance and *non-uniform arrows* with significantly varying appearance. The recognition is done in two steps. Uniform symbols are found first and arrows are detected afterwards as connectors linking pairs of uniform symbols. Uniform symbols are classified by an SVM classifier based on the trajectory based normalization and direction features proposed by Liu and Zhou [10]. We detect arrows with recently proposed arrow detector based on LSTM RNN classifier [3]. The core of the recognition pipeline is the structural analysis phase. Individual symbol candidates have a score assigned saying how good the hypothesis is without considering any context. Some of the symbol candidates are in relations. Binary predicates are defined to indicate if two symbol candidates can coexist in the solution together or if one symbol candidate can be a part of the solution without the other one, etc. The selection of the best subset of symbol candidates is cast as an optimization problem where the goal is to maximize the sum of scores of selected symbol candidates that fulfil all the constraints given by the predicates. We model this framework as a pairwise max-sum labeling problem. Finally, remaining unused text strokes form text block which can be easily found with the knowledge of the diagram structure.

3. Proposed System Improvements

We propose two improvements of the recognition system. First, we replace the naive strokes grouping by the SLAC. Second, we improve the symbol classifiers by using synthesized samples.

3.1. Over-segmentation Improvement

The old method works with an important assumption that symbols are formed of strokes which are spatially and temporarily close. Strokes grouping is done iteratively. Within the first iteration, every single stroke forms a subset of size 1. Further, subsets of size k are created by adding a single spatially and temporarily close stroke to subsets of size $k-1$. Maximal size of a strokes group k must be derived from knowledge of a domain and it affects a number of generated groups. Threshold used to determine if two strokes are spatially and temporarily close must be derived form data. The advantage of this approach is its simplicity and possibility to achieve 100 % recall using the right parameters. The disadvantage is

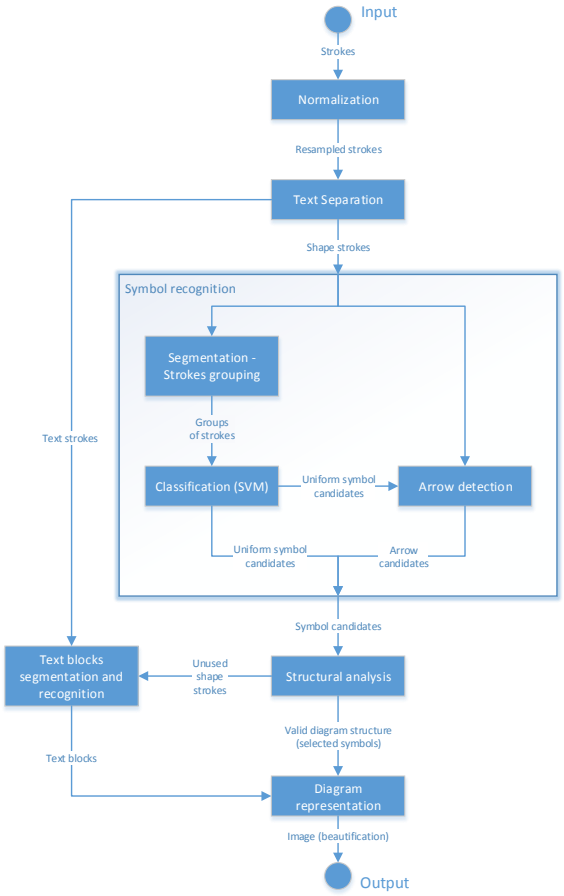


Figure 3: The pipeline of our recognition system.

the fact that the method considers too many combinations of strokes, which are not important, because they can never form a symbol. This inefficiency led us into experimentation with other possibilities.

Single-linkage clustering based on weighted combination of several features with trainable parameters proposed by Delaye and Lee [7] reaches very high precision. Its advantage is that it uses more features combined together and can express more complex relations between strokes than just the Euclidean distance. Another advantage is that single-linkage is a fast clustering algorithm. Time complexity is quadratic in the number of strokes. It is only needed to compute distance between individual strokes once and then the distance between two clusters is given by the distance of their two closest strokes. We reimplemented the method and trained the feature weights and the threshold as is described in the work by Delaye and Lee. We achieved a bit worse precision (cca. 3 % less) on both, FC and FA, databases. We believe that it is caused by slight differences in the input normalization. However, the method is powerful and the result is satisfactory for our purposes.

We perform the clustering with the trained parameters and several values of the threshold to perform an over-segmentation to increase the recall. We obtain various values of the threshold by multiplication of the original threshold h by a changing coefficient c_i: $h_i = h \cdot c_i$. We use various values of c_i from the interval $[c_{min}, c_{max}]$ with step 0.01, where the bounds c_{min} and c_{max} must by found in a validation step. Only uniform symbols are our objects of interest, because our recognition system deals with text and arrows separately. We used the validation dataset of the FA database and train dataset of the FC database to find the bounds of the coefficient. We tried all combinations of c_{min} from the range $[0.1, 1.0]$ and c_{max} from the range $[1.0, 2.0]$. The best combination of the bounds is that one which gives the highest recall. In case that more combinations give the same recall, a combination giving higher precision is taken. We found out that for both domains the best values are $c_{min} = 0.5$ and $c_{max} = 1.2$.

3.2. Improvement of the Symbol Classifier

As we care for the greatest possible universality of our system, we used the most general approach and combined trajectory based normalization and direction features proposed by Liu and Zhou [10] as a descriptor with multiclass classifier implemented as an instance of a structured output SVM learned by BMRM algorithm [15]. We trained the classifier with negative examples to obtain the rejection ability. Dataset of symbols for training has been obtained by applying the over-segmentation implemented as the multi-threshold SLAC. If a group of strokes is annotated as a uniform symbol in the database, it is labeled by that symbol. Otherwise it is labeled as *no_match* which denotes a negative example. Arrows as well as incomplete parts of symbols are labeled as negative examples.

The number of negative examples is much higher than the number of uniform symbols. Moreover, they are greatly inhomogeneous. It is thus necessary to cluster them into several subclasses. We employed k-means base on the descriptor to create m *no_match* subclasses, where m is domain dependent ($m = 30$ for flowcharts, $m = 20$ for finite automata). A greater amount of symbol classes in the flowchart domain results in a greater m. This brings a need

Database – Method	Retrieved	Relevant	Matched	Recall	Precision	F-measure
FC – grouping	19 714	921	878	95.33 %	4.45 %	0.085
FC – clustering	5 245	921	876	95.11 %	16.70 %	0.284
FA – grouping	6 095	488	485	99.39 %	7.96 %	0.147
FA – clustering	1 838	488	487	99.78 %	26.50 %	0.419

Table 1: The results of strokes grouping and clustering on the test datasets of the FC and the FA databases.

for a modified loss function which gives zero penalty when a negative example is classified into a different *no_match* subclass. Additionally, a greater penalty is required for misclassification of a uniform symbol as a negative example than vice versa. The ratio between these two penalties depends also on the ratio between the number of uniform symbols and negative examples. A properly chosen loss function can overcome the problem with unbalanced database, as we showed in [4]. However, our current implementation uses artificially synthesized samples to balance the database. The samples were synthesized using the approach proposed by Martín-Albo et al. [11]. It is based on Kinematic Theory and the distortion of the Sigma-Lognormal parameters in order to generate human-like synthetic samples. We generated up to 20 artificial samples from each uniform symbol taken from the training dataset. From all the synthesized samples of one class we randomly chose a subset to get the desired number of symbols for training. This approach does not only help to balance the dataset, it also supplies additional information on handwriting and makes the classifier more robust. Therefore, we empirically set the smaller penalty to 1 and the bigger penalty to 2 just to increase recall in cost of very small decrease of precision. Unfortunately, the FC database does not contain any information about time – points forming strokes are defined by coordinates only. Since time information is crucial for the synthesization, artificial samples could not be obtained for this database.

4. Experiments

We performed two types of experiments. We report a comparison of the results of the naive strokes grouping and more sophisticated strokes clustering first. We show how the clustering method allows to increase the precision significantly while the recall changes minimally. Later we show how this improvement affects the overall performance of the system. It turns out that time complexity is significantly lowered.

4.1. Strokes Grouping vs. Strokes Clustering

Note that the text separation step precede the over-segmentation step and thus the most of the text strokes are removed. The text separator achieves the precision in *shapes*/*text* class of 99.62 %/94.76 % and 100.00 %/93.31 % for FC and FA, respectively. Since the over-segmentation is used to find uniform symbols only, we do not consider text blocks or arrows as relevant objects. Therefore there are 921 / 488 relevant objects in the test dataset of the FC / FA databases. Results of both over-segmentation methods on both databases are summarized in Table 1. Notice that the clustering method achieved even higher recall than the naive grouping in the case of FA. Obviously, few symbols in the test dataset violated one of the assumptions we use in the process of strokes grouping. Specifically, they comprise of more strokes than is allowed. The advantage of the clustering approach is that we do not need such assumption at all.

4.2. Overall System Performance

We changed the over-segmentation method in the recognition pipeline and made experiments with diagram recognition. We use two standard metrics for system quality assessment – correct strokes labeling and correct symbol segmentation and recognition. We compare the results with the published results of our former system [3], with the grammar based method by Carton et al. [5], and with the purely statistical method by Delaye [6]. Our system achieved the highest precision using both metrics on both domains. For details, see Tables 2, 3. The precision slightly increased in the case of FA and slightly decreased the case of FC. However, the main benefit of the new over-segmentation method is the difference in the performance in the term of the running time. Our system is implemented in C# and we ran the experiments on a standard tablet PC Lenovo X230 (In-

Class	Correct stroke labeling [%]				Correct symbol segmentation and recognition[%]			
	Carton	Delaye	WACV 2015	Proposed	Carton	Delaye	WACV 2015	Proposed
Arrow	83.8	–	88.7	87.5	70.2	–	78.1	76.6
Connection	80.3	–	94.1	94.1	82.4	–	95.1	95.1
Data	84.3	–	96.4	95.3	80.5	–	90.6	90.5
Decision	90.9	–	90.9	88.2	80.6	–	75.3	72.9
Process	90.4	–	95.2	96.3	85.2	–	88.1	88.6
Terminator	69.8	–	90.2	90.7	72.4	–	88.9	89.0
Text	97.2	–	99.3	99.2	74.1	–	89.7	89.5
Total	**92.4**	**93.2**	**96.5**	**96.3**	**75.0**	**75.5**	**84.4**	**84.2**

Table 2: Recognition results for the FC database. We compared the proposed system with the grammar based method by Carton et al. [5], with the statistical method by Delaye [6], and with our previous work presented at WACV 2015 [3].

Class	Correct stroke labeling [%]			Correct symbol segmentation and recognition[%]		
	Delaye	WACV 2015	Proposed	Delaye	WACV 2015	proposed
Arrow	–	94.9	98.0	–	92.8	97.5
Initial arrow	–	85.0	98.6	–	84.0	97.3
Final state	–	99.2	99.2	–	98.4	99.2
State	–	96.9	98.3	–	97.2	98.2
Label	–	99.8	99.7	–	99.1	99.2
Total	**98.4**	**97.4**	**99.0**	**97.1**	**96.4**	**98.5**

Table 3: Recognition results for the FA database. We compared the proposed system with the statistical method by Delaye [6] and with our previous work presented at WACV 2015 [3].

tel Core i5 2.6 GHz, 8GB RAM) with 64-bit Windows 7. Detailed results with performance of all the systems are in Table 4. We reduced the running time significantly and made the system useful for a real-time applications. However, the purely statistical approach by Delaye is much faster. On the other hand, the author probably used more optimized implementation and more powerful machine, because our system spent more time on feature extraction solely than his system did on the whole recognition.

System	FC	FA
Carton [5]	1.94 s	-
Delaye [7]	80.8 ms	52.0 ms
WACV 2015 [3]	1.06 s	2.03 s
proposed	0.78 s	0.69 s

Table 4: Average running time for diagram recognition by different systems.

5. Conclusion

Naive over-segmentation approach considering all combination of spatially and temporarily strokes is simple and achieves a very high recall. It is possible to apply several restrictions like maximal number of strokes in a segment to suppress the number of created segmentation hypotheses. However, the number of generated hypotheses is still too big and the precision is limited. Even though the symbol classifier can reject most of the hypotheses in the early stage, it might be still time consuming.

We experimented with over-segmentation method based on agglomerative clustering of strokes. It creates hypotheses in a smarter way, avoiding the consideration of all strokes combinations. We combined clusters obtained by cutting the dendrogram with various thresholds. It allows to increase the recall at the cost of decreased precision. However, the achieved

precision is still much higher than in the case of naive strokes grouping and the recall is comparable. This approach generally does not lead to 100 % recall even when all possible values of the threshold are tried. The reason is that all threshold values always produce nested clusters. Their characteristics is given by the set of used distance features, sets of their weights, and the principle of the single-linkage clustering itself. Different clustering methods could be probably combined together to further increase the recall. An intuitive idea is to combine together other agglomerative clustering methods like average or complete linkage. Unfortunately, this methods have higher time complexity than single linkage. However we leave this for a future work.

Acknowledgements

The first author was supported by the Grant Agency of the CTU under the project SGS13/205/OHK3/3T/13. The second and the third authors were supported by the Czech Science Foundation under the project P103/10/0783.

References

[1] F. Álvaro, J.-A. Sánchez, and J.-M. Benedí. Recognition of on-line handwritten mathematical expressions using 2d stochastic context-free grammars and hidden markov models. *Pattern Recognition Letters*, 35(0):58 – 67, 2014. Frontiers in Handwriting Processing. 1

[2] A.-M. Awal, G. Feng, H. Mouchere, and C. Viard-Gaudin. First experiments on a new online handwritten flowchart database. In *DRR'11*, pages 1–10, 2011. 2

[3] M. Bresler, D. Průša, and V. Hlaváč. Detection of arrows in on-line sketched diagrams using relative stroke positioning. In *WACV 2015: IEEE Winter Conference on Applications of Computer Vision*, pages 610–617. IEEE Computer Society, January 2015. 3, 5, 6

[4] M. Bresler, T. Van Phan, D. Průša, M. Nakagawa, and V. Hlaváč. Recognition system for on-line sketched diagrams. In J. E. Guerrero, editor, *ICFHR 2014: Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition*, pages 563–568, 10662 Los Vaqueros Circle, Los Alamitos, USA, September 2014. IEEE Computer Society. 2, 3, 5

[5] C. Carton, A. Lemaitre, and B. Couasnon. Fusion of statistical and structural information for flowchart recognition. In *International Conference on Document Analysis and Recognition (ICDAR), 2013 12th*, pages 1210–1214, 2013. 1, 2, 5, 6

[6] A. Delaye. Structured prediction models for online sketch recognition. Unpublished manuscript, `https://sites.google.com/site/adriendelaye/home/news/unpublishedmanuscriptavailable`, 2014. 2, 5, 6

[7] A. Delaye and K. Lee. A flexible framework for online document segmentation by pairwise stroke distance learning. *Pattern Recognition*, 2014. 2, 4, 6

[8] G. Feng, C. Viard-Gaudin, and Z. Sun. On-line hand-drawn electric circuit diagram recognition using 2D dynamic programming. *Pattern Recogn.*, 42(12):3215–3223, Dec. 2009. 1

[9] A. D. Le, T. Van Phan, and M. Nakagawa. A system for recognizing online handwritten mathematical expressions and improvement of structure analysis. In *11th IAPR International Workshop on Document Analysis Systems (DAS), 2014*, pages 51–55, April 2014. 1

[10] C.-L. Liu and X.-D. Zhou. Online Japanese Character Recognition Using Trajectory-Based Normalization and Direction Feature Extraction. In G. Lorette, editor, *Tenth International Workshop on Frontiers in Handwriting Recognition*, La Baule (France), Oct. 2006. Université de Rennes 1, Suvisoft. 3, 4

[11] D. Martín-Albo, R. Plamondon, and E. Vidal. Training of on-line handwriting text recognizers with synthetic text generated using the kinematic theory of rapid human movements. In J. E. Guerrero, editor, *ICFHR 2014: Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition*, pages 543–548, 10662 Los Vaqueros Circle, Los Alamitos, USA, September 2014. IEEE Computer Society. 5

[12] H. Mouchère, C. Viard-Gaudin, R. Zanibbi, and U. Garain. ICFHR 2014 Competition on Recognition of On-line Handwritten Mathematical Expressions (CROHME 2014). In J. E. Guerrero, editor, *ICFHR 2014: Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition*, pages 791–796, 10662 Los Vaqueros Circle, Los Alamitos, USA, September 2014. IEEE Computer Society. 1

[13] T. Y. Ouyang and R. Davis. Chemink: A natural real-time recognition system for chemical drawings. In *Proceedings of the 16th International Conference on Intelligent User Interfaces*, IUI '11, pages 267–276, New York, NY, USA, 2011. ACM. 1

[14] J. Stria, D. Průša, and V. Hlaváč. Combining structural and statistical approach to online recognition of handwritten mathematical formulas. In Z. Kúkelová and J. Heller, editors, *CVWW2014: Proceedings of the 19th Computer Vision Winter Workshop*, pages 103–109, Pod Vodárenskou věží 4, 182 00, Prague, Czech Republic, February 2014. Czech Society for Cybernetics and Informatics, Center of Machine

Percep tion at CTU in Prague, Czech Society for Cybernetics and Informatics. 1

[15] C. H. Teo, A. J. Smola, and Q. V. Le. Bundle Methods for Regularized Risk Minimization. *Journal of Machine Learning Research*, 11:311–365, 2010. 4

[16] T. Van Phan and M. Nakagawa. Text/non-text classification in online handwritten documents with recurrent neural networks. In J. E. Guerrero, editor, *ICFHR 2014: Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition*, pages 23–28, 10662 Los Vaqueros Circle, Los Alamitos, USA, September 2014. IEEE Computer Society. 3

20^{th} *Computer Vision Winter Workshop*
Paul Wohlhart, Vincent Lepetit (eds.)
Seggau, Austria, February 9-11, 2015

Segmentation of Depth Data in Piece-wise Smooth Parametric Surfaces

Aitor Aldoma, Thomas Mörwald, Johann Prankl and Markus Vincze
Vision4Robotics, ACIN, Vienna University of Technology
{aldoma, moerwald, prankl, vincze}@acin.tuwien.ac.at

Abstract. *This paper describes an efficient algorithm to extract piece-wise smooth surfaces from depth images. The algorithm is based on the Mumford-Shah (MS) functional. A solution is obtained by means of a multi-model and multi-scale region merging strategy that does not require to define the number of regions in advance. Our current formulation allows smooth regions to be modeled either as planar or B-splines surfaces and thus provides a parametric representation of the scene upon convergence. Additionally, we propose a final refinement step that corrects initial region boundaries obtained by means of supervoxel segmentation. This final stage results in smooth boundaries (due to the boundary length penalization in the MS) that better separate different regions in the scene. We demonstrate the performance of the proposed algorithm in indoor scenes, acquired with RGB-D sensors, showcasing man-made objects and structures.*

1. Introduction

Segmentation of images into meaningful structures is a major research area in the field of computer vision. Even though segmentation has been predominantly investigated for intensity and color images, the recent appearance of RGB-D sensors has sparked a renewed interest among roboticists. Clearly, the availability of depth data in conjunction with color images provides additional cues to aid in segmentation. Segments cannot only be assessed by their similarity in color space, but also by their continuity and smoothness in Euclidean space. Nevertheless, while computer vision scientists have adopted energy minimization techniques (which in some cases consider the whole extend of the image as well as interaction among segments) to address the challenges present in segmentation, recent approaches making use of depth information still rely strongly on local heuristics (in particular during the initial stages of the segmentation pipeline) to determine the extend of individual regions in an image. While these algorithms perform well in the envisioned situations, their strong dependence on local properties of the data results in an undesired lack of robustness to local perturbations. This results in complex pipelines that are difficult to adapt to novel situations or slightly different sensors.

Because of the aforementioned caveats and inspired by recent trends in the segmentation of intensity images, this paper formalizes the segmentation of depth images into piece-wise smooth surfaces within the Mumford-Shah framework (see Section 3). We propose an algorithm (based on Koepfler *et al.* [6]) to obtain an approximated solution of the functional that upon convergence results in a parametric surface representation of the input data (see Section 4). We demonstrate the performance of the proposed approach in Section 5 on two datasets acquired with RGB-D sensors but with different characteristics vouching for the generalization capabilities of the proposed framework.

2. Related work

Various approaches to segment images into larger patches exist. Most of them are based on simple color and edge features [4, 17, 21, 22, 2, 19], some include depth information [7, 10, 20] and others rely on the estimation of shape primitives [9, 5] or combine 3d-shape with color information [16, 11]. In the following paragraph we review aspects of these approaches starting with

algorithms relying on appearance cues.

Many approaches formulate image segmentation as energy minimization with a MRF [17, 21, 22]. In addition to an appearance model computed from color and texture Werlberger et al. [22] introduce a shape prior which is modeled as a Geodesic Active Contour energy. In [2] and [19] the objective function is formalized with the Mumford-Shah functional [12]. Bernard et al. [2] introduce a continuous parametric function using B-splines to model a contour energy term. Strekalovskiy and Cremers [19] rewrite the proximal operator in a primal-dual algorithm using Moreau's identity to achive real-time performance.

A graph cut is also used in [7, 10]. While Kootstra et al. [7] include the disparity deviation of pixels to the dominant plane and solve an MRF-formulation using α/β swap [3], Mishra et al. [10] use fixation points and a shortest path in a log polar transformed edge image. Ückermann et al. [20] propose a model-free algorithm which subsequently combines smooth surface patches, directly computed in depth images, to form object hypothesis. The approach by Hager et al. [5] is able to segment objects from cluttered scenes in point clouds by using a strong prior 3d model. Hence, it is limited to parametric models such as boxes and cylinders. The problem of fitting higher order surfaces to point clouds was addressed by Leonardis et al. [9]. They segment range images by estimating piecewise linear surfaces, modeled with bivariate polynomials. A Model Selection framework based on the Minimum Description Length (MDL) principle is used to find the best interpretation of the scene. MDL for Model Selection is also used in [11]. Instead of piecewise linear surfaces Mörwald et al. use planes and B-spline surfaces.

Like Mörwald et al. the approach in this paper uses basic surface models, such as planes and B-splines. Instead of using Model Selection and MDL where the complexity for each model needs to be defined with respect to their number of parameters, we integrate these surface models into the Mumford-Shah functional [12] and model complexity is implicitly encoded by the curvature of the regional surfaces.

3. Piece-wise Smooth Segmentation

This section briefly reviews the Mumford-Shah framework for image segmentation. Then, we propose an adaptation of the functional for the segmentation of depth images into piece-wise smooth parametric surfaces.

3.1. Mumford-Shah framework

In a nutshell, the celebrated Mumford-Shah functional [12] is used to establish an optimality criterion to segment an image into a disjoint set of sub-regions. The aim of the functional is to find an approximation I of an input image I_o such that (i) I is similar to I_0, (ii) I is smooth within the different sub-regions and (iii) the boundaries between regions are of minimal length. In the continuous setting, the functional is formulated as

$$E(I, C_i) = \int_\Omega \|I - I_0\|^2 \, \mathrm{d}x + \beta \int_{\Omega \setminus C_i} \|\nabla I\|^2 \, \mathrm{d}x + \alpha \int_{C_i} \mathrm{d}s, \tag{1}$$

where Ω is the image domain and C_i represents the boundaries of the different sub-regions in the image. α and β are parameters (≥ 0) penalizing lack of smoothness within regions and boundary length, respectively. Of special interest is the piecewise constant Mumford-Shah model when $\beta \to \infty$ enforcing the different regions in the image approximation, I, to be constant.

3.2. Multi-model MS for depth images

This section proposes a set of modifications to the MS framework in order to extract piece-wise smooth parametric surfaces from a depth image. *Multi-model* refers to the availability of different parametric surface models (with increasing expressiveness and potentially decreasing smoothness) to approximate piece-wise smooth subregions in the input data. In our current formulation, surfaces can be represented by planar or B-splines (with 3x3 control points) surfaces. Please note that these two parametric models of surfaces are by construction smooth and differentiable. Thus, (1) becomes in our setting:

$$E(D, C_i) = \int_\Omega \|D - D_0\|^2 \, \mathrm{d}x + \beta \int_{\Omega \setminus C_i} \kappa^2 \mathrm{d}x + \alpha \int_{C_i} \mathrm{d}s. \quad (2)$$

where D_0 represents the input depth image and D represents an approximation of the input depth and is composed by different piecewise smooth regions parametrically modeled either as planar or B-spline surfaces. Note that in our specific setting, the second term of (2) penalizes the curvature κ of the approximating surface instead of $\|\nabla D\|$. This formulation allows on one hand to overcome the problem of favoring fronto-parallel planar surfaces (with $\|\nabla D\| = 0$) over equally planar but slanted surfaces (with $\|\nabla D\| > 0$) [15, 8]. On the other hand, it favors regional models with less expressiveness (e.g. planar surfaces) over richer models (e.g. B-splines) presenting higher complexity. Intuitively, we would prefer segmentations that use simpler parametric models unless there is a good reason to increase the model complexity, such as low regional data fidelity and/or reduction of the boundary length.

4. Implementation

This section revolves around the implementation of the piecewise smooth surface segmentation framework proposed in the previous section. In particular, we address the problem of minimizing the functional in (2).

4.1. Overview

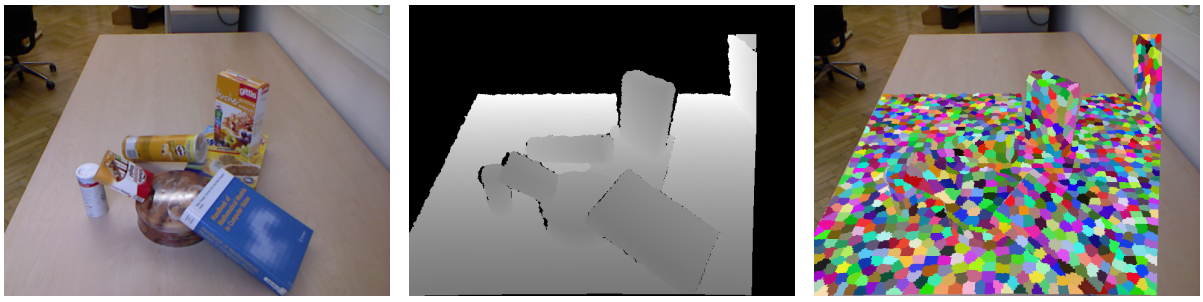
Provided with a depth image of a scene, D_0, the algorithm starts by computing an oversegmentation of the scene in terms of supervoxels. These small regions are the basis to minimize (2) and provide as well the initial boundaries between different sub-regions. In particular, a solution is obtained by incrementally merging pairs of adjacent regions (i.e., sharing a boundary) that improve the functional energy. Our multi-model scheme is introduced here by first trying to reduce the energy by fitting planes to neighboring regions. Once the energy cannot be further reduced, the model expressiveness is increased and the merging process is restarted by fitting B-splines surfaces to connected regions. Upon convergence, a refinement stage is performed that swaps the associated region at pixels located at the boundary between two regions. This final stage aims at improving the initial boundaries provided by the over-segmentation in the scene in situations where they do not adhere properly to the actual boundaries of the smooth surfaces in the scene. The outcome of the proposed algorithm at different stages is depicted in Figure 1.

4.2. Oversegmentation

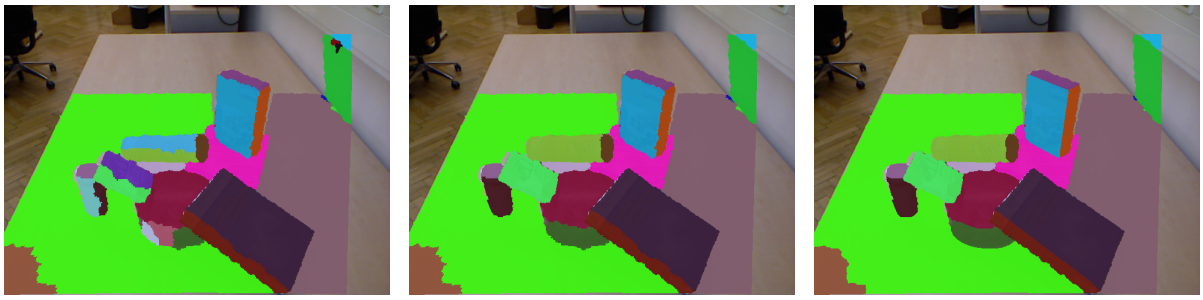
Over-segmentation of an image into regions of similar pixels, known as superpixels, is a widely used preprocessing step in oder to reduce the amount of data for subsequent computationally expensive algorithms. We use the method of Papon et al. [13], which is able to cluster a set of points using color and the 3D information. The main idea is to select spatially uniform distributed (in Euclidean space instead of image space) seed points and to iteratively cluster neighboring points enforcing spatial connectivity and smoothness. In contrast to traditional superpixel algorithms working on image space [1], this results in supervoxels which do not flow across boundaries in 3D space and are smooth by considering surface normals. The implementation used in this paper is the one provided by the original authors within the Point Cloud Library. The supervoxel extraction is governed in our case by two parameters indicating spatial compactness and smoothness. Please note that supervoxels provide on one hand an initial reduction of the number of regions and on the other hand, pixels get grouped together in larger regions that allow the extraction of parametric surface models.

4.3. Multi-scale and multi-model region merging

The previous stage results in an oversegmentation of the image domain into a disjoint set of regions $\Omega = \{R_1 \cup R_2 \cup ... \cup R_i \cup ... \cup R_n\}$. $C_i \in R_i$ is defined as the boundary between R_i and adjacent regions. Provided with this initial set of regions and boundaries, this section describes the algorithm to minimize the functional in (2). To this end, we propose an adaption of the multi-scale algorithm by Koepfler *et al.* [6] that is reviewed in the following for completeness. They minimize the piece-wise constant Mumford-Shah model for an intensity image I_0



(a) Color image (b) Depth map (c) Over-segmentation



(d) Planes (e) Planes + B-splines (f) Refinement

Figure 1: Overview of the different stages of the method

$$E(I, C_i) = \int_{\Omega} (I - I_0)^2 \, \mathrm{d}x + \alpha \int_{C_i} \mathrm{d}s, \quad (3)$$

where the smoothness term has been dropped by letting $\beta \to \infty$ in (1). The algorithm in [6] proceeds by iteratively merging adjacent regions whereby the different regions composing the piece-wise constant approximation $I = \{R_1 \cup ... \cup R_n\}$ are modeled by the average intensity of all pixels within each region. In a nutshell, at each iteration, the algorithm selects the merging move with minimal $\hat{\alpha}_k$. The $\hat{\alpha}_k$ of a certain move k representing the merging of two regions R_i and R_j is defined as:

$$\hat{\alpha}_k = \frac{-\Delta\mathcal{E}_{region}}{\Delta\mathcal{E}_{length}} = \frac{-\left(\mathcal{E}_{R_i} + \mathcal{E}_{R_j} - \mathcal{E}_{R_i \cup R_j}\right)}{|C_i| + |C_j| - |C_{ij}|} \quad (4)$$

where C_{ij} represent the boundary length obtained by merging both regions and $\mathcal{E}_{\{R_i, R_j, R_i \cup R_j\}}$ represent the regional error for a piece-wise constant region. The algorithm terminates when all possible merging moves in the current state have an $\hat{\alpha}_k$ larger than the user parameter α, indicating the lack of energetically favorable moves. The multi-scale attribute arises from the fact that as the algorithm proceeds, the boundary length penalizer $\hat{\alpha}_k$ is incrementally increased. Therefore, it is possible to obtain different segmentations at different scales.

In contrast to [6], we propose a modification that is very similar to the original algorithm but with two main differences:

1) We minimize the piece-wise smooth MS instead of the piece-wise constant model by allowing regions to be modeled as parametric smooth surfaces, and
2) We incrementally increase the model complexity representing piece-wise smooth regions once the energy cannot be further reduced by simpler models.

Therefore, (4) becomes:

$$\hat{\alpha}_k = \frac{-\Delta\mathcal{E}_{region} - \beta\Delta\mathcal{E}_{smooth}}{\Delta\mathcal{E}_{length}}, \quad (5)$$

and due to 2), the proposed algorithm works not only at multiple scales but also with different model complexities. In our current implementation, with two piece-wise smooth models (i.e.

planar and B-splines surfaces), our algorithm can be considered a two-pass version of the algorithm of Koepfler (see Algorithm 1). Using the appropriate data structures as well as exploiting incremental computation properties of surface parametric models (see Section 4.4), merge moves can be efficiently implemented.

Algorithm 1 Multi-scale and multi-model region merging

Input: α, β
  $\mathcal{M}odels$={PLANE, BSPLINE_3x3}
  $m \leftarrow 0$
  $\mathcal{C} = \{R_i, R_j, \hat{\alpha}_k\}$ //sorted merging candidates
  converged $\leftarrow$ false
  while not converged **do**
    $c = \{R_i, R_j, \hat{\alpha}_k\} \leftarrow$ pop($\mathcal{C}$)
    if $\hat{\alpha}_k > \alpha$ **then**
      if $m \geq$ length($\mathcal{M}odels$) **then**
        converged $\leftarrow$ true
      else
        //increase model type
        $m \leftarrow m + 1$
        //merging candidates with current model type
        $\mathcal{C} \leftarrow$ comp_candidates($\mathcal{M}odels[m]$, $\mathcal{C}$)
        continue
      end if
    end if
    //apply merge and update structures
    {new_cands, affected} $\leftarrow$ merge(c)
    $\mathcal{C} \leftarrow$ remove_candidates(affected, $\mathcal{C}$)
    new_cands $\leftarrow$ comp_candidates($\mathcal{M}odels[m]$, new_cands)
    insert_sorted($\mathcal{C}$, new_cands)
  end while

4.4. Model fitting

The algorithm proposed in the previous section relies on the ability to fit planar and B-splines models to regions in the scene that build up the piece-wise smooth approximation (D) of the input data D_0. To this end, this section focuses on how to incrementally (whenever possible) and efficiently extract the parametric representation of regions as the algorithm iterates.

4.4.1 Planar surfaces

Planar surfaces are a good initial choice to parametrically approximate unknown surface data:

1) Locally, planar models can approximate almost any structure.
2) Planar surfaces are a recurrent structure in man-made environments.
3) They can be efficiently estimated by first- and second-order moments of the underlying data followed by Eigenvector analysis of the resulting 3x3 covariance matrix.

In addition, because first- and second-order moments can be incrementally computed, planar models become a very efficient model for region merging strategies. In other words, the planar fit of two regions that are to be merged can be efficiently computed by reusing the previously computed statistics of the individual regions.

Relation to (2): The regional fit of a region, R_i, modeled as a planar surface is computed as the squared depth error of the underlying pixels to the model. Regarding the smoothness term, planar surfaces do not present any curvature and thus, the smoothness term has no effect in the energy for any region modeled as a planar surface.

4.4.2 B-spline surfaces

Modeling curved surface areas is a well studied problem and there are many mathematical solutions such as superquadrics, wavelets and bivariate polynomials to name a few. We choose B-splines due to their beneficial properties:

1) They are very flexible w.r.t. the degrees of freedom we wish to model.
2) Derivatives and curvature may be computed explicitly at any point of the surface.
3) The mathematical formulation of fitting a B-spline to a point-cloud or depth map becomes solving a linear system of equations.

A B-spline surface is defined as the sum of weighted basis functions

$$S(\xi, \eta) = \sum_{j=1}^{m} \varphi_{j,p}(\xi, \eta)\mathbf{b}_j \tag{6}$$

where $(\xi, \eta) \in R$ and $\varphi_j(\xi, \eta)$ is a bivariate basis function which can be efficiently evaluated by the *Cox-de-Boor* algorithm. They define the influence of the weights, also called control points $\mathbf{b}_j$. The polynomial order of the basis functions is denoted by p. A full explanation of B-splines is available in the book of Piegl et al. [14]. Note that we embed the B-spline surface into the domain of the depth map thus becoming a function $S : \mathbb{R}^2 \rightarrow \mathbb{R}$.

Fitting to a depth image $D : \mathbb{R}^2 \rightarrow \mathbb{R}$ is the problem of finding control points such that the distance between D and S is minimized. Since we aim for piecewise smooth regions, a least squares optimization w.r.t. the control points is

a sufficiently accurate approximation.

$$\min_{\mathbf{b}} \int_R \|D(\xi,\eta) - S(\xi,\eta,\mathbf{b})\|^2 \qquad (7)$$

where $\mathbf{b}$ denotes a vector collecting the control points. This is equivalent to the first term of Eq. (2). We define the B-spline domain to match the index space of the depth image. This allows to conveniently query surface points and its derivatives (up to order $p-1$) at any image location (ξ,η).

Relation to (2): The attentive reader might already have noticed that the functional we minimize in Eq. (7) is equivalent to the first term of Eq. (2). By using the Greville abscissae and re-projecting into $\mathbb{R}^3$ we obtain the B-spline control points and therefore the surface in Euclidean space ($\mathbf{S} \in \mathbb{R}^3$). We then explicitly evaluate the mean curvature κ for computing the second term as

$$\kappa = \left\langle \frac{\partial^2 \mathbf{S}}{\partial^2 \xi}, \mathbf{n} \right\rangle + \left\langle \frac{\partial^2 \mathbf{S}}{\partial^2 \eta}, \mathbf{n} \right\rangle \qquad (8)$$

with $\mathbf{n}$ being the normal surface of the B-spline

$$\mathbf{n} = \frac{\partial \mathbf{S}}{\partial \xi} \times \frac{\partial \mathbf{S}}{\partial \eta}. \qquad (9)$$

4.5. Refinement stage

So far, we have focused on the minimization of the functional in (2) by merging neighboring regions as described in sections 4.3 and 4.4. While being a successful strategy, it suffers from the inability to change the location of initial region boundaries resulting from the over-segmentation stage. Therefore, if supervoxels flow across object boundaries, the merging moves will not be able to correct these artifacts. Aiming at further minimizing the energy cost, we propose a refinement stage that includes another variety of moves. In particular, the refinement stage aims at swapping the region association of pixels at the boundary between regions provided that this swap minimizes the functional. This simple strategy results in the removal of wiggly boundaries due to a reduction of the overall boundary length as well as a better pixel-wise association due to a reduction of the data error term. By applying this refinement stage after the functional cannot be further minimized by means of merging moves, the overall cost of this stage is computationally acceptable (since the number of boundary pixels has in general been greatly reduced prior to this stage by merging operations).

5. Experimental results

This section provides an initial qualitative evaluation of the proposed method. Figure 2 and 3 show the resulting segmentation for three scenes from the OSD0.2 [16] and three from the NYU-depth (v2) [18] dataset respectively. Both datasets have been acquired indoor using RGB-D cameras but as it can be seen from the images they showcase different scenarios. In particular, OSD focuses on the segmentation of household objects in table-top scenarios. On the other hand, the NYU dataset includes thousands of scenes from domestic environments and its focus lies on larger objects (e.g., furniture, room structure, etc). A major distinctive trait among both datasets is the depth range covered by the datasets. While most of the objects in OSD are to be found not farther away than $1.5m$ from the sensor, the NYU dataset depth range is much larger. It is a well known fact that the quality of RGB-D data degrades rapidly after $2m$ and therefore, segmentation of meaningful structures on the NYU dataset is much more challenging, specially for algorithms like the one proposed in this paper relying solely on depth information.

These differential traits have required a different parameter setting for both datasets (see captions of Figure 2 and 3 for specific values). Overall, we can see that the scenes get segmented into meaningful structures vouching for the efficiency of the proposed method. However, in Figure 3 one can observe how segmentation quality degrades due to the noise in the data as the distance to the camera increases. Figure 4 shows the reconstructed point cloud from the depth data obtained after minimizing the proposed functional. As expected, noise in the data gets smoothed by using smooth parametric surface models. Finally, Figure 5 shows the effect of the boundary regularizer on a scene from the OSD dataset. As boundaries become more costly, larger structures arise.

6. Conclusions and future work

This paper has presented a formulation based on the Mumford-Shah functional to segment depth data into smooth surfaces. Our prelimi-

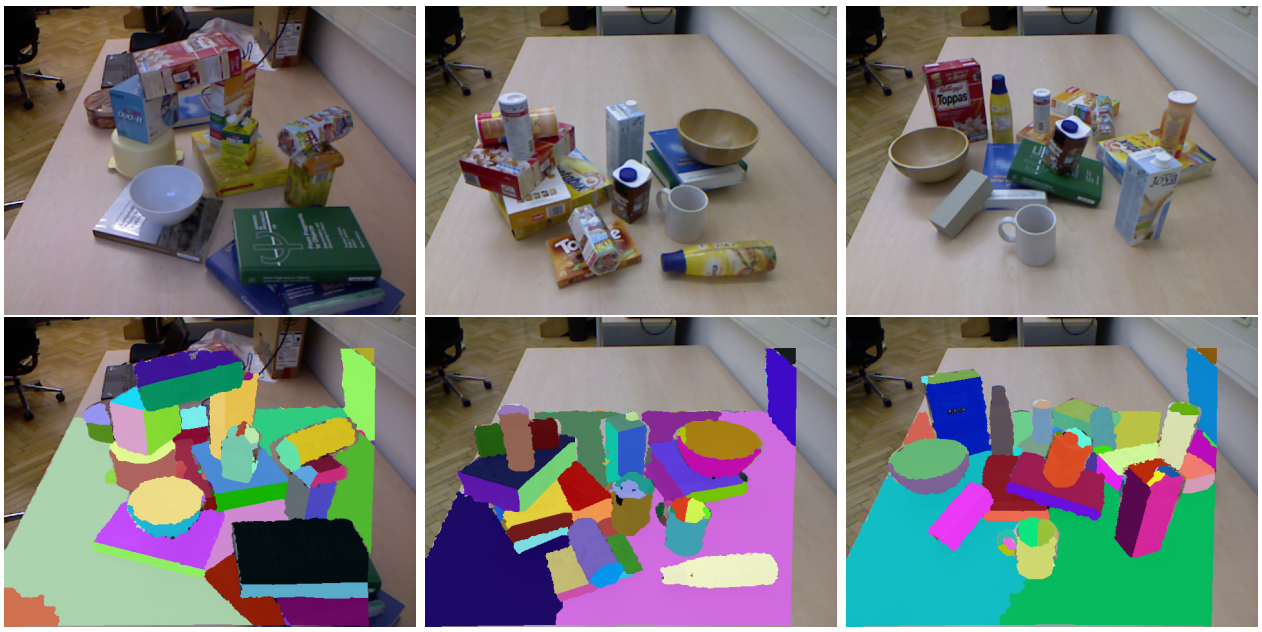


Figure 2: Qualitative results for three scenes in the OSD dataset. $\alpha = 1.5^{-4}, \beta = 1$.

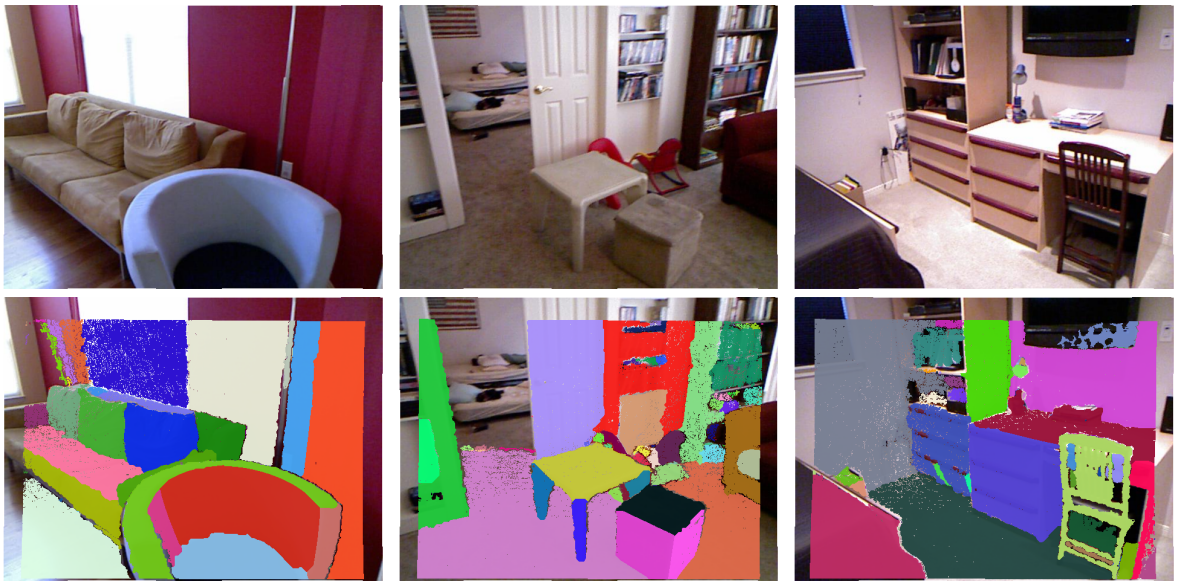


Figure 3: Qualitative results for three scenes in the NYU dataset. $\alpha = 1^{-2}, \beta = 2$. Depth information beyond 3.5m is ignored.

nary results show that the method is an effective and elegant alternative for the task at hand. In the future, we plan to extend our current model to more complex models being able to represent simple objects (i.e. by means of superquadrics) as well as the addition of appearance information to improve segmentation in situations where depth data becomes unreliable. The addition of shape priors based on the knowledge of recurrent objects is another interesting research direction.

Acknowledgements

The research leading to these results has received funding from the European Communitys Seventh Framework Programme FP7/2007-2013 under grant agreement No. 600623, STRANDS and from the Austrian Science Foun-

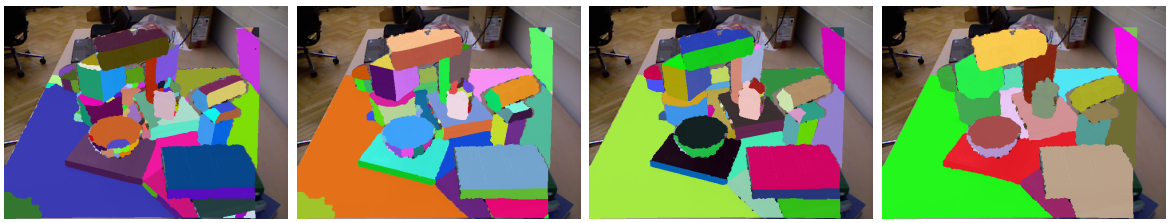


Figure 5: Effects of boundary regularizer ($\alpha = 5^{-5}, 1^{-4}, 2^{-4}, 3^{-4}$)

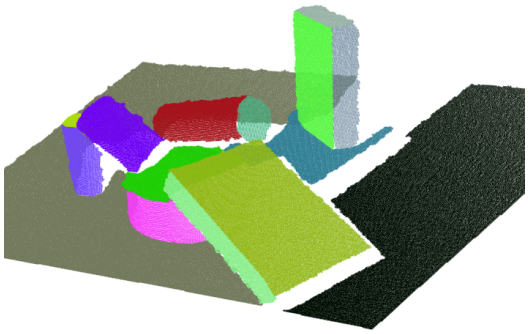


Figure 4: Resulting point cloud after segmentation using the proposed method. The depth of the points has been corrected to lie on the underlying parametric model surface.

dation (FWF) under grant agreement No. I513-N23, vision@home.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Suesstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 2012. 3

[2] O. Bernard, D. Friboulet, P. Thevenaz, and M. Unser. Variational b-spline level-set method for fast image segmentation. In *Biomedical Imaging: From Nano to Macro 5th IEEE International Symposium on*, 2008. 1, 2

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *TPAMI*, 2001. 2

[4] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. *IJCV*, 2004. 1

[5] G. D. Hager and B. Wegbreit. Scene parsing using a prior world model. *The International Journal of Robotics Research*, 2011. 1, 2

[6] G. Koepfler, C. Lopez, and J. Morel. A multiscale algorithm for image segmentation by variational method., 1994. 1, 3, 4

[7] G. Kootstra, N. Bergström, and D. Kragic. Fast and Automatic Detection and Segmentation of Unknown Objects. In *Humanoids*, Bled, 2011. 1, 2

[8] G. Kuschk and D. Cremers. Fast and accurate large-scale stereo reconstruction using variational methods. In *ICCVW*, 2013. 3

[9] A. Leonardis, A. Gupta, and R. Bajcsy. Segmentation of range images as the search for geometric parametric models. *IJCV*, 1995. 1, 2

[10] A. K. Mishra, Y. Aloimonos, L. F. Cheong, and A. Kassim. Active Segmentation. *TPAMI*, 2011. 1, 2

[11] T. Moerwald, A. Richtsfeld, J. Prankl, M. Zillich, and M. Vincze. Geometric data abstraction using b-splines for range image segmentation. In *ICRA*, 2013. 1, 2

[12] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on Pure and Applied Mathematics*, 42(5):577–685, 1989. 2

[13] J. Papon, A. Abramov, M. Schoeler, and F. Wörgötter. Voxel cloud connectivity segmentation - supervoxels for point clouds. In *CVPR*, 2013. 3

[14] L. Piegl and W. Tiller. *The NURBS book*. Monographs in visual communication. Springer, 1996. 5

[15] R. Ranftl, S. Gehrig, T. Pock, and H. Bischof. Pushing the limits of stereo using variational stereo estimation. In *IV*, 2012. 3

[16] A. Richtsfeld, T. Mörwald, J. Prankl, M. Zillich, and M. Vincze. Segmentation of unknown objects in indoor environments. In *IROS*, 2012. 1, 6

[17] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, Aug. 2004. 1, 2

[18] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 6

[19] E. Strekalovskiy and D. Cremers. Real-time minimization of the piecewise smooth mumford-shah functional. In *ECCV*. 2014. 1, 2

[20] A. Ückermann, R. Haschke, and H. Ritter. Real-time 3d segmentation of cluttered scenes for robot grasping. 2012. 1, 2

[21] S. Vicente, V. Kolmogorov, and C. Rother. Joint optimization of segmentation and appearance models. *ICCV*, 2009. 1, 2

[22] M. Werlberger, T. Pock, M. Unger, and H. Bischof. A Variational Model for Interactive Shape Prior Segmentation and Real-Time Tracking. In *SSVM*, 2009. 1, 2

20^{th} *Computer Vision Winter Workshop*
Paul Wohlhart, Vincent Lepetit (eds.)
Seggau, Austria, February 9-11, 2015

Safe Exploration for Reinforcement Learning in Real Unstructured Environments

Martin Pecka
Center for Machine Perception
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
martin.pecka@fel.cvut.cz

Karel Zimmermann
Center for Machine Perception
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University in Prague
http://cmp.felk.cvut.cz/~zimmerk/

Tomas Svoboda, Center for Machine Perception
Department of Cybernetics, Faculty of Electrical Engineering
and
Czech Institute of Informatics, Robotics, and Cybernetics
Czech Technical University in Prague

Abstract. *In USAR (Urban Search and Rescue) missions, robots are often required to operate in an unknown environment and with imprecise data coming from their sensors. However, it is highly desired that the robots only act in a safe manner and do not perform actions that could probably make damage to them. To train some tasks with the robot, we utilize reinforcement learning (RL). This machine learning method however requires the robot to perform actions leading to unknown states, which may be dangerous. We develop a framework for training a safety function which constrains possible actions to a subset of really safe actions. Our approach utilizes two basic concepts. First, a "core" of the safety function is given by a* cautious simulator *and possibly also by manually given examples. Second, a classifier training phase is performed (using Neyman-Pearson SVMs), which extends the safety function to the states where the simulator fails to recognize safe states.*

1. Introduction

Many robotic tasks are tackled by RL with iterative state-action space exploration (RC helicopter acrobacy [1], adaptive traversability [17], etc.). RL essentially needs to exhaustively sample the state-action space (which is called "exploration"), and the exploration strategy is represented by a stochastic policy.

While manually-driven exploration is often prohibitively time consuming, autonomous exploration is usually only applied to inherently safe systems (pendulum) or to simulators [16]. We propose a framework for making autonomous exploration safe even for general systems, and we test it on the task of autonomous control of articulated subtracks (flippers) of the USAR mobile robotic platform depicted in Figure 1.

1.1. Task description

The objective of our algorithm is to train a safety function that will allow to select only those exploration policies, that do not lead to unsafe states. Finding an efficient way to optimize the RL objective while using only safe policies is left for an upcoming research (a relevant approach for policy iteration is shown in [14]).

1.2. Contributions

The contributions of our paper can be summed up as follows. We introduce a novel term "cautious simulator" and show it is both simple to construct and useful in machine learning tasks. Next, we present a safe exploration algorithm based on NP-SVM that gradually discovers the safe region without visiting unsafe states or needing too much prior knowledge.

2. Limits of other safe exploration methods

2.1. Visiting of unsafe states

Many proposed safe exploration techniques require that the robot can visit unsafe states in order to get data with "negative" labels. This may be justifiable only if there is a precise model of the world where the dangerous steps can be simulated (e.g. [11, 10]), or if there is an unlimited number of robots to try with. In our work, the robot is never required to visit an unsafe state.

2.2. Coupling safety with rewards

Many safe exploration approaches try to utilize the existing RL methods to achieve safety. This is usually connected with some consequences unacceptable in USAR cases.

Either they claim a state is safe based on the minimum achievable reward – if it is high enough, the state should be safe [12]. This was proved to be highly non-optimal [8]. Or they just set negative rewards to unsafe states and run standard RL (e.g. [5]). However, no guarantees can be given this way, since RL only optimizes the expected outcome.

Tying safety with rewards seems to be unnecessarily constraining. Especially in the field of safe exploration, it does not hold that what is safe is also good from the task point of view (and vice versa, what is good for the task, is not necessarily safe).

We propose to decouple the terms safety and reward completely, as it is done in [9] (where, however, the safety and reward functions get combined together during learning using a weighted sum). In our work, the safety and reward functions are trained as independently as possible.

2.3. Too optimistic expectations

Our last remark is on what can be achieved at best by any safe exploration algorithm. Geibel mentioned that we can never achieve absolute safety [9]. Not only that the safety guarantees can be often only provided as an estimate (which can be erroneous), but we can also "protect" the robot only against some specified classes of risk.

This issue is covered in Ertle's system paper [7], along with the description of methodology and a general view on how the learning algorithms should look like. One implication of his work is that the safety implemented in robots should be *behavior-based* – e.g. each class of risk should include its own safety function and its own policy to avoid the danger. In our experiment, we only concentrate to the "behavior" of climbing down a step.

3. Precise formulation

We optimize a RL task given in the gradient policy search paradigm presented e.g. in [3]. The robot "lives" in a state space $\mathbf{X}$ and performs actions from $\mathbf{A}$ according to a stochastic policy $\pi : \mathbf{X} \to \mathcal{P}(\mathbf{A})$, and is rewarded by a real-number reward function $\mathbf{R} : \mathbf{X} \to \mathbb{R}$. Every policy can be evaluated by the expected performance given by

$$\mathbf{J}(\pi) = \mathbb{E}_{\xi \sim P_\pi} [\mathbf{R}(\xi)]$$

where ξ is a trajectory (sequence of states) created by following the policy from a common start state, and $\mathbf{R}(\xi)$ is a (possibly discounted) reward for the whole trajectory.

To simplify the learning, the policy is often assumed to be from a parametrized class of functions, and the learning is only performed on the parameter values. Thus we can write the policy as $\pi = \pi(\theta)$, and substitute just θ for the policy, yielding

$$\mathbf{J}(\theta) = \mathbb{E}_{\xi \sim P_\theta} [\mathbf{R}(\xi)]$$

Gradient policy search then searches for the policy parameters $\theta^\star$ which maximize the expected performance [2]:

$$\theta^\star = \arg\max_\theta \mathbf{J}(\theta)$$

And this is where safe exploration comes into play: during the gradient search, the examined values of θ are not restricted in any way, so that it can happen that the robot visits an unsafe state. With just a small alteration to the previous equation, we can "plug in" the safety:

$$\theta^\star = \operatorname*{argmax}_{\theta : \mathbf{S}(\theta) \geq s_{\min}} \mathbf{J}(\theta)$$

where $\mathbf{S}$ is the safety function and $s_{\min} \in \langle 0, 1 \rangle$ is a user-defined safety threshold. Finally, we define the state safety function $\mathbf{s} : \mathbf{X} \to \langle 0, 1 \rangle$, from which $\mathbf{S}$ is "composed" as

$$\mathbf{S}(\theta) = \min_{x \sim \pi(\theta)} \mathbf{s}(x)$$

The task is to construct a safety function closest to the real safety margins, and not to visit any unsafe states during the training.

4. Safe exploration system components

The basic background and motivation to our work has been presented, so now we can describe the main components of the algorithm.

A *cautious simulator* is the main component that differentiates our work from other safe exploration concepts. We use the simulator to predict safe states among the set of unvisited states (it may be e.g. a simple physical simulator). *Cautious* means that if the simulator labels a state as safe, it is also safe in the real world. On the other hand, it is allowed to wrongly label safe states (in the real world) as unsafe. Having a cautious simulator is a key to success of our algorithm, and creating such simulator is (much) easier than constructing a plausible physical simulator. Throughout all this work we suppose that running the simulator is (computationally) expensive, so we try to minimize the number of its uses, and we prohibit sampling the whole state space using the simulator.

Next, we need to have an experienced human operator that is capable of executing safe trajectories on the robot in the real world. We suppose that this operator has much more (prior or sensory) information than the robot has, and thus he or she can assess the safety of intended actions before executing them. These safe trajectories will be used to initialize the *safety function*. If we discover an area in the state space that is misclassified by the safety function as *unsafe*, the operator can reach these areas manually, which forces the algorithm to correct the safety estimates for that region.

Combining the simulator and operator results, we can construct the *safety function*. Such function takes the state of the robot (the extracted features), and labels it either safe or unsafe (by returning a number in the $\langle 0, 1\rangle$ interval, where values greater than s_{min} are considered safe). This component is implemented using Neyman-Pearson SVM classifier.

Finally, we need a safe policy extractor, that takes the current estimate of safety function and chooses a policy going only through safe states. Safe policies are then used to automatically gather new data.

The algorithm that combines all these components into a safe exploration scheme is shown in Alg. 1 and described in detail in the next section.

Algorithm 1 The safety function training algorithm

1. $\mathbf{X}^{real}$ = operator-generated initial trajectories
2. Update $\mathbf{T}$, $\mathbf{S_0} := \texttt{updateSVM}(\mathbf{T})$
3. $i := 0$
4. **while** learning should continue **do**
5. Generate an optimal policy π_i safe on $\mathbf{S_i}$, or use the operator "as a policy"
6. Drive using π_i, record visited states x_{new}
7. $\mathbf{X}^{real} = \mathbf{X}^{real} \cup x_{new}$
8. Update $\mathbf{T}$, $\mathbf{S'_i} := \texttt{updateSVM}(\mathbf{T})$
9. Perturb x_{new} several times, add the perturbed states to $\mathbf{X}^{sim}_{safe}$ or $\mathbf{X}^{sim}_{unsafe}$ depending on the result of simulation
10. Update $\mathbf{T}$, $\mathbf{S_{i+1}} := \texttt{updateSVM}(\mathbf{T})$
11. $i{+}{+}$
12. **end while**

Variable	Definition
n	The dimensionality of feature space
$\mathbf{X}$	$\mathbb{R}^n$, the feature (state) space
$\mathbf{A}$	$\mathbf{X} \times \mathbf{A} \to \mathcal{P}(\mathbf{X})$, the set of actions
$\mathbf{X}^{real}$	$\subset \mathbf{X}$, already visited states
$\mathbf{X}^{sim}_{safe}$	$\subset \mathbf{X}^{sim}$, states labeled *safe* by Sim
$\mathbf{X}^{sim}_{unsafe}$	$\subset \mathbf{X}^{sim}$, states labeled *unsafe* by Sim
$\mathbf{T}$	$\{\mathbf{X}^{real} \times \{\texttt{safe}\}\} \cup \{\mathbf{X}^{sim}_{safe} \times \{\texttt{safe}\}\} \cup \{\mathbf{X}^{sim}_{unsafe} \times \{\texttt{unsafe}\}\}$, the training set for SVM
Sim	$\mathbf{X} \times \mathbf{A} \to \{safe, unsafe\}$, the simulator
π_i	$\mathbf{X} \to \mathcal{P}(\mathbf{A})$, a stochastic safe policy
$\mathbf{S_i}$	$\mathbf{X} \to \{safe, unsafe\}$, a safety function (SVM)

Table 1. Notation used in the algorithm.

5. USAR safe exploration in detail

In this section we're going to go through the algorithm step-by-step and show what exactly is done in each step. In Table 1 we present the basic definitions used in the algorithm.

5.1. Initialization

On line 1 we first require the operator to generate some real-world trajectories. It is generally not necessary for them to be generated by the operator; they can also be substituted by a first run of the simulator or by prior knowledge (e.g. if a small part of safe states can be analytically expressed). It is important for this initial set to be sufficiently large – if it were not, the initial estimate of the safety function would be very poor. All the generated points are inserted into $\mathbf{X}^{real}$ which is represented either as a set of points, or as a spatial search tree (depend-

ing on the expected number of elements). Then we update the training set $\mathbf{T}$ (according to its definition given in Table 1), and update the SVM model of the safety function ($\mathbf{S}_0$). Description of the SVM update is postponed for Section 5.6.

5.2. The stopping criterion

Line 4) represents the stopping criterion. It can be either a subjective measure (trading off safety function accuracy for time available for experimenting), or a qualitative measure (if the algorithm is no longer able to simulate more unvisited states, or if the safety function hasn't changed for some time).

5.3. Generating an optimal safe policy

On line 5 a policy is generated based on $\mathbf{S_i}$. There are several options on how to do that.

If the task is not only to train safety, but also to optimize a given criterion, it is needed to run a modified Reinforcement learning algorithm that optimizes the expected return subject to all the states selected by the policy are safe. Since computing such optimization problem efficiently is a large problem itself, we only give here a simple (and probably inefficient) way to solve the constrained RL problem. The easy solution is to set rewards for all unsafe states to negative infinity. This will surely find a policy that is safe, however we do not say whether it is optimal or not.

The other option is to randomly generate policies and verify their intersection with the safety function (e.g. by sampling). This is good if we are not interested in learning any specific task, and we just want to explore the state space ("optimal" here means any safe policy).

It can happen that there is no safe state for a particular feature value. Then we need to incorporate this into the policy and allow it to answer that a state is unreachable.

5.4. Policy execution

The step to be done next is to execute the safe policy (line 7 and further). This may need some additional work to be done, such as setting the robot to an initial position, changing the environment and so on. After the policy is executed, the newly visited states are added to $\mathbf{X}^{real}$ and an update of $\mathbf{T}$ and the SVM is run.

5.5. Simulation

The loop starting on line 9 specifies that we sample some perturbed states and simulate them in the simulator. Here is one important point – we assume that the further a perturbed state is from the current (real) state of the robot, the less precise the simulation is. Therefore we always try to perturb only in some small local neighborhood of the current state. How to perturb depends on the type of the features – it can be e.g. by Euclidean vector shifting. The magnitude of the shifts is one of the free parameters of this algorithm.

Once we have the simulations done, we record the simulated states to S^{sim}_{safe} or S^{sim}_{unsafe} depending on the results of the simulations (which are either binary classes or numbers from $\langle 0, 1 \rangle$). Then the training set and SVM are updated again (which is described in the next section).

This simulation and perturbation can also be run just after initialization, before the algorithm enters the learning loop. This way the initial estimate of $\mathbf{S}_0$ will be better.

5.6. Updating the safety function (`updateSVM`)

Representation and modification of the safety function are the key points of our algorithm. We need the safety function to generalize the set $\mathbf{X}^{real} \cup \mathbf{X}^{sim}_{safe}$ in continuous space, not containing any point from $\mathbf{X}^{sim}_{unsafe}$.

From our assumptions it follows that it is not necessary that a generalization over this set also denotes only safe regions (because we defined that safe are only visited states, and states tagged *safe* by Sim). However, if we assume continuousness of the safety function, it can be approximated very well.

To describe the representation of the safety function, we first define an auxiliary set $\mathbf{T}^{pruned} \subset \mathbf{T}$, which is basically the set of all visited or simulated states. To avoid serious problems in computation of the safety function, we need to prune $\mathbf{T}^{pruned}$ in such way, that there are no points from $\mathbf{X}^{sim}$ near to any point from $\mathbf{X}^{real}$. This in fact ensures that visited states have "priority" over states just simulated, which allows us to remedy states misclassified by Sim as *unsafe*, although they are *safe* in reality. Again, the distance function is a free parameter of this algorithm.

Now, $\mathbf{T}^{pruned}$ contains states of which no two cover each other, and are tagged either *safe* or *unsafe*. Finding a representation of $\mathbf{S_i}$ is now a binary classification task. To ensure safety of the estimated safety function, the classification has to be done in such a way that it never classifies an *unsafe* state

as *safe*. This can be easily achieved by using the Neyman-Pearson classification [15] with false negative rate limit set to zero (assuming *negative=safe*).

One of the possible implementations of this classification scheme is using $2\nu-$SVM presented in [6] utilizing LIBSVM [4]. There is a set of kernel functions that can be used with SVMs, and which one to choose again depends on the expected structure of the safety function. Preferring SVMs has one good reason against other binary classification tools – SVMs minimize structural risk (error on test data) rather than minimizing the error on training data. This should provide us with a robustly estimated safety function.

5.7. Remarks

The goal of this algorithm is to find a safety function closest to the real safety margins of the robot. The approximation of the real safety with the safety function should get better as the number of visited states increases, which can be confirmed taking into account how the training set for SVMs is built and how SVMs operate (assuming the kernel function is rich enough to represent the safety function).

Also we can conclude that the number of simulator runs is less than if we sampled the state space regularly, which could be another method of estimating the safety function. Furthermore, our approach has the advantage that it is always sufficient to simulate in local neighborhood of the state the robot is in, allowing for better simulations than if we ran the simulator in distant states.

6. Experiments

6.1. Platform description

To prove this concept of safe exploration we have set up an experiment on a real robot. In the experiment we train a safety function for the task of climbing down steps with various heights. The robot is in front of a terrain step and it receives the "go forward" command. The task is to find the safe flipper angles using which the robot climbs down safely (if it is possible at all).

The robot we used is the Absolem platform from EU FP7 projects NIFTi and TRADR (see Fig. 1). This is an actively articulated tracked platform with size about $60\,\text{cm}\times30\,\text{cm}\times30\,\text{cm}$ and weight $25\,\text{kg}$. The four articulated subtracks (two on each main track) are called *flippers*. The robot can actively control the rotation of each of the flippers (independently).

From the point of view of this experiment, the robot has two important sensors - an IMU (measuring rotation and acceleration), and a laser range finder with broad field of view (270° both horizontally and vertically). There is also a 3D map incrementally built from the laser data, so the robot knows the terrain under itself (which is occluded for the laser).

6.2. Experiment setup

For the experiment we have chosen the task of controlling front flippers when driving down a step (both flippers the same angle). This action is interesting because for different step heights there are different safe flipper configurations, and from a particular height up, there is no safe flipper configuration. The potentially unsafe states cover robot body breakage due to flipping over, gaining too high speed, or touching the terrain with one of its fragile parts (e.g. the laser scanner or camera).

So the state space consists of all possible step heights (also drop heights; measured at the point where the flipper is attached to the main track). The robot generates multiple data when driving down a step – first for height 0, then for the maximum height, and then for all the heights until it finishes climbing down the step (however, we assume only limited sampling capabilities, and this is why the data in Fig. 3 are that sparse). The action space then covers all possible flipper angles the robot can set when climbing down the step. For simplicity, we assume the robot can switch quickly between two different flipper configurations.

The policies are from the deterministic linear policy class of the form $\pi(x) = \theta_0 + \theta_1 x$. The reinforcement learning objective we minimize is $\mathbf{J}(\theta) = \theta_1^2$ (to prefer policies with less flipper motion, e.g. to save power). We seek for a safety function that would discriminate which flipper configurations are safe for which step heights. The safety function is represented by a 2C-SVM (equivalent to 2ν-SVM) with Radial Basis Function kernel.

For executing the simulations, we created a simple model of the robot for use with the Gazebo simulator. Gazebo is a physical simulation library, however our model contains only the basic physical properties. Namely, we have created plausible collision links for the real robot links (simple enough to allow for fast collision checking, though they are still

triangle meshes and not primitive objects). For each of the links we have estimated the weight, center of mass position and inertial properties. Specifically, we have not estimated or set any properties regarding the motors, friction, slippage or other dynamic properties.

Similarly, we put in the simulator a rough terrain representation that is created directly by triangulating the point cloud (either from the laser scanner or from the point map). Such map is in no means smooth, rigid or regular. It contains triangles with wrongly estimated normals or even corner positions and it is non-continuous. Creating a more sophisticated map is an option for improving the estimated safety function, but it is difficult and we want to show that this algorithm works well even with the cluttered map and simplistic robot model. Thus, the task environment can be considered unstructured.

The simulation is then done in the following manner: first we get the triangulated map and place the robot to the position corresponding to the real world. Then we shift it forward 30 cm, set the desired flipper angle and let the robot "fall" on the ground, adjusting the flipper angle according to the given policy. If the flipper policy is safe, then the robot only falls a few millimeters and remains in a stable state, and we can mark all passed state-action pairs as *safe*. The policy is considered unsafe if the robot touches the terrain with its fragile parts, if it turns over or if it ends up too far from the desired $[x, y]$ coordinate – then the simulator tags all the state-action pairs as *unsafe*. For a reference on how the robot looks visualized by Gazebo, refer to Figure 1.

Fortunately, physical simulations in this setting proved to satisfy the requirement on cautiousness of the simulator. To even more ensure cautiousness, we perturb each simulated state several times and return the ratio of safe simulations to all simulations as the final result (thus our simulator returns values from $\langle 0, 1 \rangle$). Here the great advantage of our algorithm showed up – the simple physical model, as well as the triangulated map, are matters of hours to create. If we should create a precise physical model (of both the robot and the terrain), it would still have cases where it fails, and it would have needed much more effort to be done. Moreover, there are properties of the terrain that cannot be modeled in advance, and our perturbation approach could overcome some of them.

It is important to notice that the simulations are performed in a space much larger than the feature space (which is 1-dimensional). The simulations are performed with full 3D models (triangle meshes) incorporating physical influences of forces. So what we do is simulate the problem in its full description, and then map the result of the simulation to the problem projected to a 2D subspace consisting of features and actions. If the projection is chosen wise, there should be no problem with this dimension shrinking.

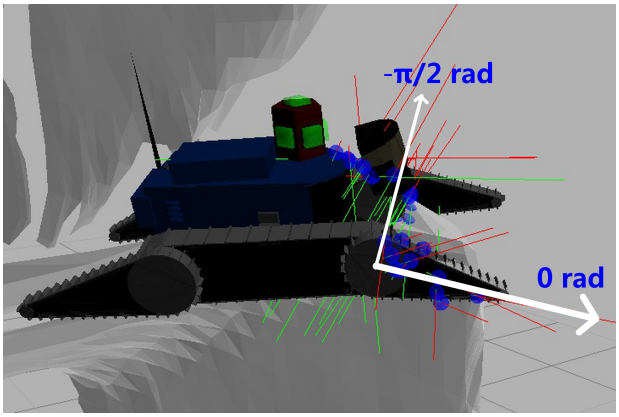


Figure 1. Robot simulation in the Gazebo simulator. Four articulated subtracks (flippers) can be seen in the image – the front ones on the right, and the rear ones on the left. All flippers are in a configuration corresponding to flipper angle 0 rad, and the white arrows symbolize some basic flipper configurations. So lifting up the flippers decreases the flipper angle. In the image there is also shown the triangulated terrain. The red and green segments denote detected robot-terrain collisions.

6.3. Realization of the experiment

To verify the safe exploration algorithm in practice, we drove the robot on several steps of different heights, running the algorithm after each trial. After each teleoperated trial there was an autonomous test of the generated policy. We always chose the policy that intersects the largest area of safe states.

6.4. Evaluation of the experiment

During the realization phase, the robot never tried to enter an unsafe state (both from the estimated unsafe set, and from the real unsafe states). It always managed to add new points to the safety function representation and enlarge the area of state space covered with the safe region. The safe and optimal policy did not change during the experiment, it was always a constant policy $\pi = 1.1 + 0x$.

The progress of the safety function, as well as its support vectors is shown in Fig. 3, note how the safety function's safe area growed gradually with

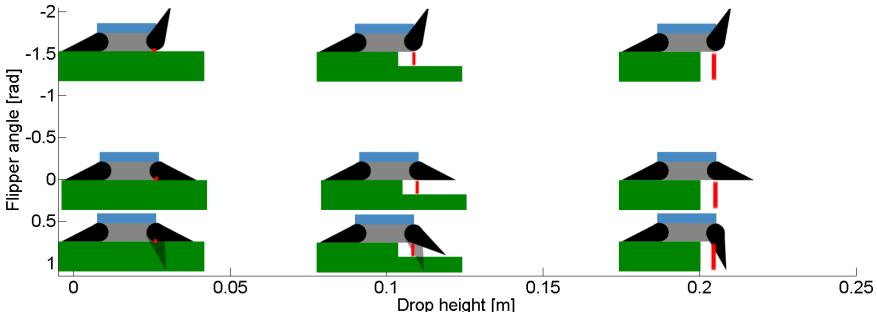


Figure 2. Poses of the robot to better illustrate the meaning of data points in Figure 3. The robot icons are placed approximately with their center on the data point (the left column represents drop height 0). The "ghost" flippers for angle 1 rad denote that the robot pushes to get to that angle, but the applied power is not sufficient (the flippers are compliant). The red bars illustrate the place where the drop height was measured.

each iteration.

After the final iteration, we compared the learned safety function to the limits that an experienced operator would allow for the robot. Actually, in more complex instances of safe exploration, getting such limits is impractical. The comparison is shown in Fig. 3, and Fig. 2 provides a graphical understanding for the data points. It is evident from the figure that we have succeeded keeping the false negative (FN) rate at 0 (here FN denotes *unsafe* states classified as *safe*).

Using the classifier terminology, we can specify true negatives (TN) as the number of *safe* states classified *safe*, false positives (FP) the number of *safe* states classified *unsafe*, and true positives (TP) the number of *unsafe* states classified *unsafe*. Then we may define *accuracy* as $(TP+TN)/(TP+TN+FP+FN)$ and *precision* as $TP/(TP+FP)$. With this terms defined, we may say that the objective of the safe exploration algorithm is to achieve precision as close to 1 as possible, which means to minimize the difference between the estimated and real safety functions, while preserving $FN = 0$.

During the three model updates, the values of accuracy in the individual steps were $[0.70, 0.82, 0.81]$, and precision was $[0.42, 0.66, 0.69]$- Another interesting metric can be seen when we superimpose the last (best) SVM model S_2 over the set of visited points in previous model updates. This shows how the model gets gradually better – accuracy $[0.77, 0.82, 0.81]$, precision $[0.48, 0.66, 0.69]$. We note that compared to the first model S_0, the last model S_2 classifies several previously unsafe points as safe, increasing both accuracy and precision. On the second model S_1 there is no change if superimposed with S_2.

7. Conclusion and further work

In our work, we have presented a novel framework for achieving safe exploration in unstructured environments. Compared to other approaches, our method does not need to visit unsafe states, as well as it can guarantee that the robot doesn't visit unsafe states by accident (this holds only for the unsafe states we provide simulators for). It also allows to train the safety function(s) independently from the robot's other tasks, and such safety functions can be easily composed. The trained safety functions are then used to restrict reinforcement learning and other algorithms to only choose safe actions during exploration.

There are two main prerequisites for our safe exploration approach: having a cautious simulator and knowing how to represent the safety function. For the former, we have shown that creating such simple simulator can be easy at least for some problems. The latter can be circumvented by either analysis and modeling of prior knowledge, or by trial-and-error.

This algorithm can be advanced in several ways. Adjusting parameters of the simulator seems to be an interesting way of increasing performance. However, it is not clear how to do some kind of gradient descent with the whole simulator.

If we could safely visit critical states (those near the decision border), that could also help. This can be for example achieved by implementing a cautious exploration strategy (human operators also slow down in dangerous or unknown situations).

Further improvements can be done in the area of selecting which policy to execute. For example, if we could select a policy that would maximize the increase of the safe area, the exploration could be done faster.

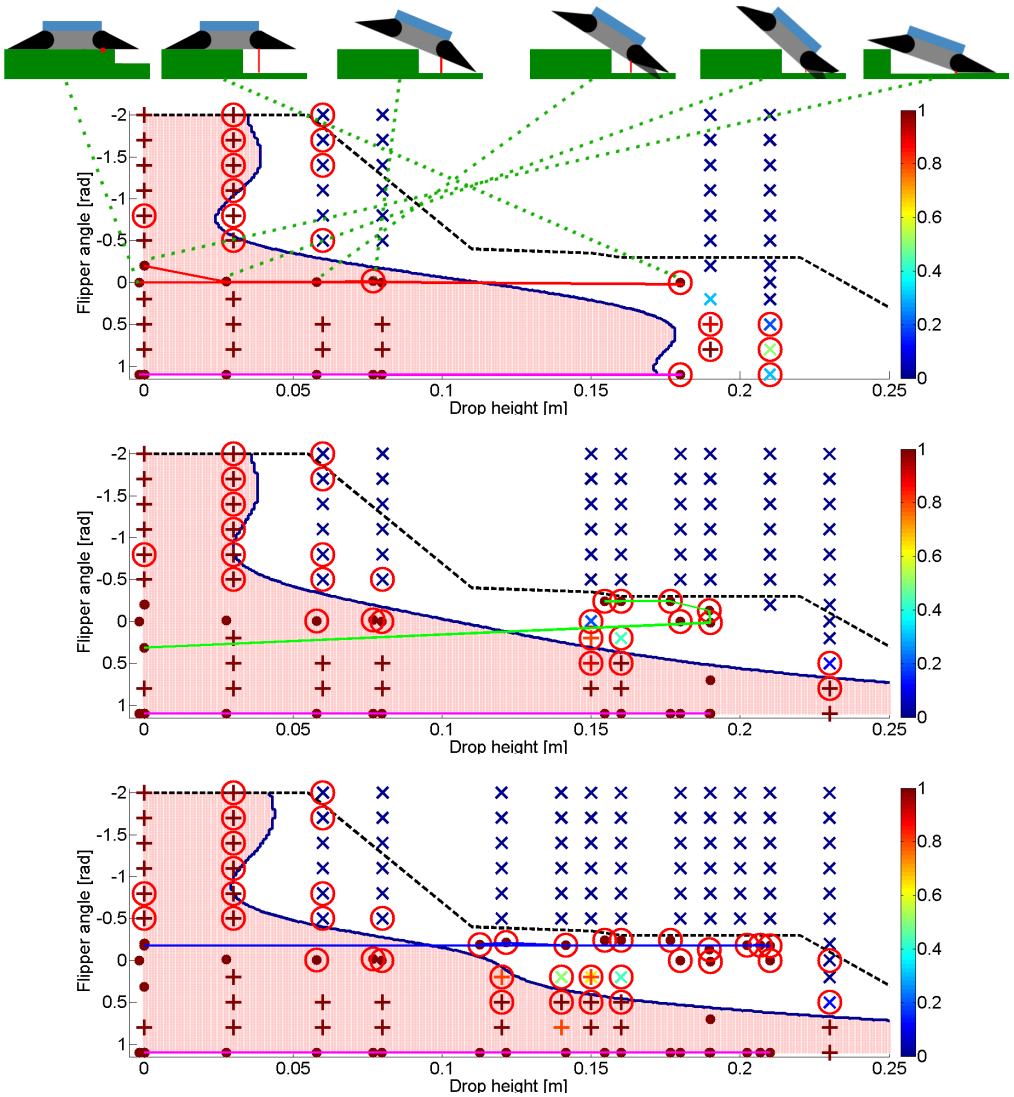


Figure 3. **The progress of learning the SVMs for safety model** (iterations 1, 2 and 3 from the top). The pink area is considered safe by the SVM (the blue solid line is its boundary). The dashed black line denotes the safety boundary estimated by an experienced operator (just for evaluation purposes). Data points from $\mathbf{X}_{real}$ are represented as brown dots, $\mathbf{X}^{sim}_{safe}$ as plus signs and $\mathbf{X}^{sim}_{unsafe}$ as crosses. Safety of $\mathbf{X}_{sim}$ data points is coded by color using the shown color scale (we used safety threshold $s_{min} = 0.7$). Encircled points are the Support Vectors. The thin red, green and blue lines represent the manually driven trajectories, and the magenta line at the bottom is the trajectory executed using π_i. To better understand the visualization of the trajectories, please refer to the robot poses depicted above the first iteration connected by green dotted lines to the corresponding data points (first, the drop height is 0, then it "jumps" to the maximum drop height, and as the robot climbs down, the drop height gets lower and lower). Note that manually visiting the green and azure points in the last step would greatly improve the safety function estimate.

A similar idea is to have an algorithm that would tell the operator which states classified as unsafe by simulator would be worth visiting in the real, if the operator considers them safe. Such approach could both minimize the number of needed human interventions and speed up the exploration process.

Acknowledgments

The 1st author was supported by CTU in Prague under Project SGS13/142/OHK3/2T/13, the 2nd author was supported by The Czech Grant Agency under Project GA14-13876S, the 3rd author was supported by EC project FP7-ICT-609763 TRADR. Any opinions expressed in this paper do not necessarily reflect the views of the European Community. The Community is not liable for any use that may be made of the information contained herein.

References

[1] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng. An application of reinforcement learning to aerobatic helicopter flight. In *Proceedings of the 2006 Conference on Advances in Neural Information Processing Systems 19*, volume 19, page 1, 2007. 1

[2] J. Bagnell. *Learning decisions: Robustness, uncertainty, and approximation*. PhD thesis, Carnegie Mellon University, Pittsburgh, USA, 2004. 2

[3] A. Barto. *Reinforcement learning: An introduction*. MIT Press, 1998. 2

[4] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1—-27:27, 2011. 5

[5] S. P. Coraluppi and S. I. Marcus. Risk-sensitive and minimax control of discrete-time, finite-state Markov decision processes. *Automatica*, 1999. 2

[6] M. Davenport. Controlling false alarms with support vector machines. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages V–V, 2006. 5

[7] P. Ertle, M. Tokic, R. Cubek, H. Voos, and D. Soffker. Towards learning of safety knowledge from human demonstrations. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5394–5399. IEEE, Oct. 2012. 2

[8] J. Garcia and F. Fernández. Safe exploration of state and action spaces in reinforcement learning. *Journal of Artificial Intelligence Research*, 45:515–564, 2012. 2

[9] P. Geibel. Reinforcement learning with bounded risk. In *IEEE International Conference on Machine Learning*, pages 162–169, 2001. 2

[10] J. H. Gillula and C. J. Tomlin. Guaranteed safe online learning of a bounded system. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2979–2984. IEEE, Sept. 2011. 2

[11] A. Hans, D. Schneegaß, A. Schäfer, and S. Udluft. Safe exploration for reinforcement learning. In *Proceedings of European Symposium on Artificial Neural Networks*, number April, pages 23–25, 2008. 2

[12] M. Heger. Consideration of risk in reinforcement learning. In *11th International Machine Learning Conference*, 1994. 2

[13] O. Mihatsch and R. Neuneier. Risk-sensitive reinforcement learning. *Machine learning*, 49(2-3):267–290, 2002.

[14] T. M. Moldovan and P. Abbeel. Safe Exploration in Markov Decision Processes. In *Proceedings of the 29th International Conference on Machine Learning*, May 2012. 1

[15] J. Neyman and E. Pearson. *Joint Statistical Papers*. Cambridge University Press, 1967. 5

[16] S. Ross and J. Bagnell. Agnostic system identification for model-based reinforcement learning. In *Proceedings of the 29th International Conference on Machine Learning*, Edinburgh, Scotland, UK, 2012. 1

[17] K. Zimmermann, P. Zuzanek, M. Reinstein, and V. Hlavac. Adaptive Traversability of Unknown Complex Terrain with Obstacles for Mobile Robots. In *IEEE International Conference on Robotics and Automation*, pages 5177—-5182, 2014. 1

20^{th} *Computer Vision Winter Workshop*
Paul Wohlhart, Vincent Lepetit (eds.)
Seggau, Austria, February 9-11, 2015

Domain-specific adaptations for region proposals

Domen Tabernik, Rok Mandeljc, Danijel Skočaj and Matej Kristan
Faculty of Computer and Information Science, University of Ljubljana, Slovenia
domen.tabernik@fri.uni-lj.si

Abstract. *In this work we propose a novel approach towards the detection of all traffic sign boards. We propose to employ state-of-the-art region proposals as the first step to reduce the initial search space and provide a way to use a strong classifier for a fine-grade classification. We evaluate multiple region proposals on the domain of traffic sign detection and further propose various domain-specific adaptations to improve their performance. We show that edge-boxes with domain-specific learning and re-scoring based on trained shape information are able to significantly outperform remaining methods on German Traffic Sign Database. Furthermore, we show they achieve higher rate of recall with high-quality regions at the lower number of regions than the remaining methods.*

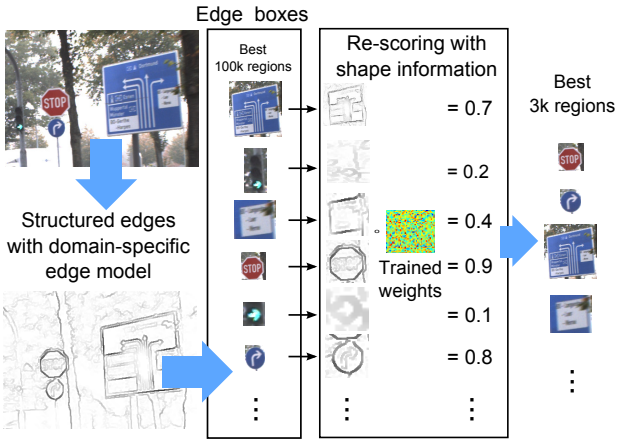


Figure 1: Overview of proposed domain-specific adaptations of edge-boxes using trained structured edges and re-scoring with shape information.

1. Introduction

The problem of detection and recognition of traffic signs has been extensively researched within the field of computer vision [18, 24, 10, 9, 17], with many proposed solutions already being deployed in real-world applications. Such applications are designed mostly for automotive safety and autonomous vehicles, and the main requirements is an excellent detection of only approximately 30 to 50 important traffic sign categories. Out of more than 400 categories, current approaches focus mostly on speed limit signs, stop and yield signs, pedestrian crossing signs and various prohibitory and mandatory signs, while information signs and direction signs are ignored.

Detection of all signs may not be crucial for automotive applications, however, they are important in road maintenance services [21], where an important task is verification of presence or absence of all road-based traffic signs, including verification of various information signs, special road marking signs and various direction signs (see, Figure 2). Extending the detection to the remaining signs is desired, as it would eliminate the tedious work of manual verification. Additionally, remaining signs may also be used in current applications of autonomous vehicles to augment the navigation in case of poor GPS signal. Our work is focused on providing a way towards the detection of all traffic signs by utilizing a fast and general regions proposal algorithms. However, due to the lack of a comprehensive database with such traffic signs, we currently focus only on 40 basic categories contained in the existing datasets.

Specific combination of colors and mathematically well-defined shapes makes traffic signs stand out from the background. Several approaches utilize this information by manually hand-crafting detectors to special colors and shapes [10], and fine tuned the algorithms to them. Hand-crafted features rely on simple techniques, such as color thresholding [19, 15], Hough transform [18] and template matching [15, 14], making them fairly efficient. A downside of hand-crafted approaches are difficulty to scale to potentially very large number of cate-

Figure 2: Examples of traffic signs required for the process of road maintenance.

gories and lack of robustness to real-world changes, where traffic signs are frequently occluded by shadows, trees, vehicles, people or other road signs.

Recent methods improved robustness by relying on machine learning. Liang et al. [14] use SVM to focus on important colors for three main classes from GTSDB [9] dataset and then apply template matching to find the specific shapes. They further use HOG features and SVM with RBF kernel to classify objects. However, they still rely on hand-crafted templates to find interesting regions. This makes extension to the remaining traffic signs difficult as they are of various shapes, sizes and colors. Other state-of-the-art methods avoid hand-crafted features and utilize HOG features to achieve best results [9]. Mathias et al. [17] use integral channel features by Dollar et al. [5] for quick detection and further analyses different discriminative learning approaches of HOG features to refine the object classification. Similarly, Wang et al. [23] find coarse locations in the first stage with LDA classifier and improves accuracy in the second stage using SVM. HOG feature are used in both stages, however, low-resolution features are applied in the first stage and high-resolution in the second. While all approaches with HOG features produce state-of-the-art results [9] they cannot be easily extended to large number of traffic sign categories without creating separate models for each category.

One approach to detect the remaining traffic signs would be to focus on the distinctive color distribution separating all road traffic signs of various shapes and colors from the remaining background object. Following the inspiration of bottom-up visual attention inspired by biological systems various methods used salient region detection to reduce the initial search space to interesting regions [24, 12, 15]. Different approaches were employed to focus on the specific color distribution of traffic signs, ranging from simple thresholding of color values in color-opponent channels [15], to computing saliency map by clustering the color space with Gaussian Mixture Model and calculating per-pixel value distances [12], or to utilizing Phase Spectrum of Quaternion Fourier Transform (PQFT) with additional motion features [13].

Recently, in the field of object detection an increasing interest has been shown in development of new methods that find regions with fully enclosed visual objects [8, 2, 22, 25, 4]. Powerful, but slow, object classification algorithms, such as convolutional neural networks [11], cannot be used in exhaustive search using sliding windows. Instead, they employ pre-processing step to find region proposals, i.e., a small set of regions that may contain objects, and perform classification only on them. Novel approaches where developed with some still relying on sliding windows but using quick computation of objectness measure using single [25, 4] or multiple cues [1], while others utilized hierarchical clustering of segmented regions [22] to generate windows. Their design makes them interesting for limiting the search space in traffic sign detection. As they are class-agnostic they should be able to detect road traffic signs of various sizes, shapes and colors included in various traffic signs. Moreover, they are designed for efficiency and can be employed only once for all categories, therefore, amortizing the computational cost over all categories.

1.1. Our approach and contributions

In this paper we propose to use the region proposal methods to move towards the detection of all road-based traffic signs, including information and various direction signs. We propose a multi-stage approach with region proposals in the first level of cascade to significantly reduce the search space and allow a more powerful but slower classifier to be later used for the fine classification. This paper represents a preliminary work towards that goal and focuses on region proposal part of the cascade. We analyze various region proposal and evaluate how successfully they can be applied to the specific domain of traffic sign detection. Multiple state-of-the-art region proposals are evaluated: Objecness measure [1], a selective search [22], BING [4] and edge-boxes [25].

Furthermore, since none of the evaluated region proposals is able to produce results good enough to enable the whole pipeline to compete with the state-of-the-art traffic sign detectors, we present domain-specific adaptations as our main contribution of this work. Out of multiple domain-specific adaptations evaluated, we propose to use a cascade with domain-specific learning of edge-boxes and additional re-

scoring based on learning of shape information with linear SVM (see, Figure 1). We show that domain-specific adaptation improves both the accuracy and the quality of region proposals for traffic signs. Although this method is applied to traffic sign detection, it does not use hand-crafted features that limit the method to this specific domain and may be easily applied to various other domains.

The paper is structured as follows: in Section 2 state-of-the-art region proposal algorithms are described, followed by proposed domain-specific adaptations in Section 3 . In Section 4 both state-of-the-art region proposals and domain-specific adaptations are evaluated. Conclusions are drawn in Section 5.

2. Region proposal algorithms

This section provides an overview of various state-of-the-art region proposals. For a comprehensive overview of region proposals see Houben et al. [9].

2.1. Window objectness

This early region proposal algorithm was proposed by Alexe et al. [2]. The algorithm is based on a fast evaluation of sliding windows to quickly reduce the search space of potential objects. Windows are evaluated using an objectness measure that integrates multiple weak cues. It utilizes saliency cue computed from the residual spectra of the FFT, additionally modified to bias larger windows and applied to multiple scales. The second cue, color contrast, measures the dissimilarity of the window compared to its immediate surrounding. The measure utilizes color histogram of LAB channels and computes Chi-squared distance between the window and its surrounding. The third cue captures edge density and computes the share of the edges found at the borders compared to ones at the window's center. Canny detector is used to detect the edges. The last cue measures closed boundary characteristics of the object by using superpixel straddling. Since superpixels will over-segment the object there will be a small probability that window containing the object will break the superpixel. All four measures are complementary to each other and are best integrated using Naive Bayes model.

2.2. Selective search

The approach proposed by Uijlings et al. [22] clusters individual pixels into object hypotheses using hierarchical grouping. Bottom-up approach enables objects to group from smaller regions up to bigger regions as they are being grouped together higher in the layers. This captures objects at different scales without sliding windows. Due to hierarchical segmentation, the approach favors objects with homogeneous regions. This may be well suited for traffic signs where homogeneous regions with one or two main colors are normally present in the center.

Hierarchical clustering uses segmentation by Felzenszwalb and Huttenlocher [7] to obtain initial regions and merging of two regions is performed when they are the most similar. Similarity between them is computed from four complementary measures. First measure is defined as a sum of differences between their normalized color histograms, where color histogram is created from three quantized channels. The second measure utilizes texture information by histogramming quantized edge orientations for each channel individually. The third measure computes similarity based on region sizes to encourage the merging of small regions as early in the hierarchy as possible. The last measure checks how well the two regions fit each other in order to avoid regions with holes and irregular shapes. All four measures can be efficiently propagated through the hierarchy to enable fast computation.

In [22] different strategies of combining all four measures are considered. Out of eight different color channels considered (HSV, LAB, RGB, normalized RGB, intensity and individual color channels) HSV channels performed the best. Out of different possible ways to combine similarity measure, combining all four also performed the best. We consider only HSV and all four similarity measures combined in our evaluation. Authors also evaluated combining multiple strategies together, using different color channels, combination of similarities and parameters for segmentation. However, combining multiple strategies can take more than 10-times longer as each strategy has to be run individually, thus taking significantly longer to compute. In our evaluation we consider only one strategy as our database already contains high-resolution images that take more time to process.

2.3. BING

Authors of BING [4] propose to capture objectness using the 64D norm (i.e. magnitudes) of the gradients feature. The method is based on the finding that stand-alone objects with well-defined bor-

ders and centers have a clear correlation in normed gradient space, particularly when objects are resized to small fixed sizes. The method proposes to normalize the size of object to multiple quantized sizes and collect a feature vector containing 8x8 norm of the gradients. Linear SVM is further used to find the set of weights that capture windows with fully enclosed object. In the second stage of learning, a linear SVM is used to calibrate the scores from different window sizes and to suppress the sizes that have low probability of containing an object.

The method is applied to densely sampled windows using sliding window approach and handle hundreds of thousands of windows with an efficient computation of feature vector using binarized normed gradients.

2.4. Edge boxes

Region proposals by Zitnick et al. [25] are based on a realization that a window with one fully enclosed object does not have many strong edges at the borders. In particular, strong edges rarely intersect with the borders as this would be an indication that window breaks the main outline of the object. The method computes a score based on this premise by first grouping edges into small segments, ensuring that group's sum of orientation differences does not exceed $\pi/2$. The score of the window is then computed as a weighted sum of magnitudes of segments that intersect with the borders of the window. The magnitudes are weighted based on how much of a segment lies outside of the window. Additionally, magnitudes within the window center are ignored as only border edges have been shown to be important. The authors propose an efficient way to compute this score using integral images.

The edges used by this method are extracted using structured edges by Dollar et al. [6]. This can reduce the presence of noisy edges as structured edges can be learned from the object borders.

3. Domain-specific adaptations

The domain to which we are applying the region proposals is very specific. The colors of traffic signs are designed to be very distinctive and the signs contain many homogeneous regions that are designed to pop out from the background. While the shape of traffic signs can vary, they are still designed around a small set of shapes, such as triangles, squares or circles, that fairly well separate traffic signs from the background objects. In this section we detail several proposed adaptations of region proposals that can exploit the specific characteristics of our target domain.

3.1. Saliency-based region proposals

We evaluate two region proposals generated from salient regions. Salient regions can be often present in traffic signs, particularly in the center of the sign, where homogeneous regions with single color are prominent. We consider region proposals generated by two salient region detectors: MSER [16] and WaDe key-point detector [20].

3.2. Selective search with SEED superpixels

In the selective search [22], the size of the smallest region detected is determined by the size of the initial segmentation segments. Since many traffic signs are only 20-30 pixels wide, the size of initial segments is even more important for this domain than for generic objects. We consider replacing the segmentation with the SEED superpixels [3] to obtain finer initial regions.

3.3. Domain-specific learning of BING

Many region proposals rely on a learning stage that is normally performed on a generic set of classes. We propose to utilize region proposal learning procedure to capture the visual characteristic of our target domain.

In BING [4] learning is performed on gradient features that are resized to specific sizes and aspect ratios. Particularly, window normalization is important in our domain as it will normalize traffic signs of different sizes, such as different information and directions board, to a specific size. Moreover, learning will capture rectangular, circular or triangular shape structures predominantly present in all traffic signs.

3.4. Domain-specific learning of edge-boxes

We propose two improvements to the traffic sign proposals for edge-boxes [25]. First, we can implement an adaptation of edge-boxes in a similar fashion as with BING [4] and use its own learning procedure to capture domain specific characteristics. Secondly, we propose to run region proposal algorithm in a cascade, with edge-boxes providing a bigger set of regions in the first stage, and using re-scoring with the shape information to further reduce the set of interesting regions. In the first stage of the cascade edge-boxes is set to return 10 - 20% of best region proposal. Results show (see, Figure 3) that at this

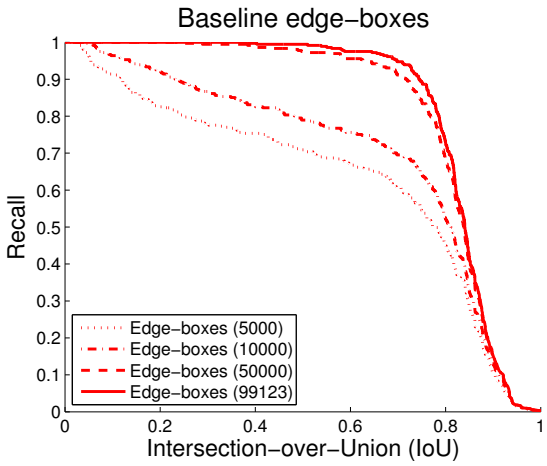


Figure 3: Performance improvements on edge-boxes when increasing the number of regions to 100.000.

range the positive samples are covered with fairly good windows. Note that due to high resolution images in our dataset and a small size of target objects, sliding window generates 250k to 500k regions, thus first stage with edge-boxes reduces a set of interesting regions to 50k - 100k regions.

3.4.1 Learning

Although edge-boxes do not perform any explicit learning, they rely on structured edges by Dollar et al. [6] that are normally trained on generic object boundaries. We train structured edges on traffic signs boundaries and allow the method to focus on borders around specific color distributions. A similar information is usually captured in various hand-crafted traffic sign detectors, however, those methods have edge detectors tuned to specific color-opponent channels [15, 13]. Instead of tuning to specific color-opponents we allow structured edges to learn which color channels are the most appropriate to find the borders of traffic signs. We trained structured edges on first 100 images from GTSDB and have manually segmented their boundaries to provide groundtruth for structured edges.

3.4.2 Re-scoring with shape information

We propose to use shape information for the re-scoring. By default, trained structured edges capture shape information fairly well. However, this information is not fully utilized in edge-boxes as the method focuses only on edges around the borders that lead out of the window, while ignoring the central edges that carry shape information.

We also perform normalization of window to specific size as windows with uniform size are invariant to changes in sizes, aspect ratios and also small degrees of rotation. Invariance to the aspect ratio is important in our domain with various rectangular boards, such as directions signs, city limits signs or information signs, which appear in multiple sizes and aspect ratios. We use simplified norming of windows by simply resizing them to specific size.

We propose the following procedure to capture shape information. Region proposals are resized to a smaller size, specifically we use 40x40 pixels that can capture enough shape information. Next, we obtain edges for each region and create feature vector from them. We can reuse domain-specific structured edges from edge-boxes and avoid additional computational cost. Feature vector is created directly from structured edges using both edge magnitudes and orientations, thus producing 3200 dimensional vector. Finally, linear SVM is trained to separate between traffic sign and non-traffic sign regions. Due to linear implementation of SVM, classification can be implemented as a dot-product between feature vector and a vector of weights.

4. Evaluation

We evaluate region proposal methods on German Traffic Sign Database [9], which contains 600 testing and 300 training images taken from vehicle mounted camera in city and countryside settings. All images have 1360x800 pixels and depict 43 different annotated traffic signs. All algorithms were tested on the testing set, while the training set was used only by the methods that require domain-specific adaptations. Baseline methods that require learning are trained on a generic dataset.

The standard evaluation of region proposals focuses on several metrics: recall, which represent the ratio of positive samples detected, the number of all regions returned by the method and the intersection-over-union (IoU) of the detected regions. The last measure is important since it captures the quality of the region proposals. Regions that cover object with only low IoU will introduce an error that propagates onwards. To capture both performance and quality of the region proposals we measure (a) recall versus IoU and (b) recall versus the number of regions proposed.

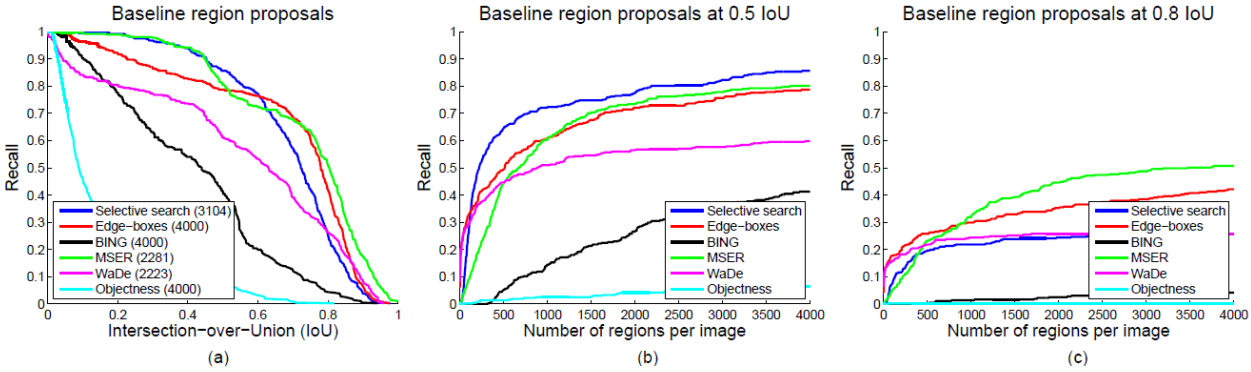


Figure 4: Results of evaluating baseline region proposals on GTSDB [9] dataset with recall over various Intersection-over-Union overlaps in (a) and recall over various number of regions in (b) for 0.5 IoU and in (c) for 0.8 IoU overlap. Note, values next to legend names in (a) are the number of regions used.

We compute the recall versus IoU measure by sorting the detection based on best IoU and thresholding IoU at various points. To ensure valid comparison between different methods the number of regions have to be fixed. For some algorithms this may not be easily achievable, however, where this is possible we set the parameters accordingly. In practice, the selective search variants produced at most approximately 3000 to 4000 regions. We adjusted the parameters of remaining methods to closely match this number. Note that standard region proposal evaluations consider only 1000 regions, however, they typically evaluate 4-times smaller images. Image samples in other evaluations also contain larger objects that are mostly present in the foreground, while GTSDB contains many small objects that are often barely visible. We account for this discrepancy by taking more than 3000 regions.

We compute recall versus the number of proposed regions by sorting the region proposals based on their score and limiting the number of regions. Note that this measure is not fully appropriate for MSER and WaDe detectors as they do not return any score. We measure recall versus the number of regions at two IoU values. One at 0.5 based on PASCAL overlap criteria and another at 0.8 to capture high-quality regions.

4.1. Baseline region proposals

Results of four baseline state-of-the-art region proposals and two salient region detectors are shown in Figure 4. Three methods, namely selective search [22], edge-boxes [25] and MSER [16], performed similarly. MSER covers most positive samples at low and high quality regions, while selective search is competitive at mid-quality regions and edge-boxes are competitive at high-quality regions. The selective search appears to perform best only at IoU of approximately 0.5 where it outperforms both MSER and edge-boxes. On the other hand, MSER performs the best at higher-quality regions, which are more important for successful classification.

More than half of the traffic signs are still not covered by any of the high-quality region proposals. This makes region proposals difficult to compete with the state-of-the-art traffic sign detectors that achieve 98 to 100% detection rate on this database [17, 23]. Both MSER and selective search have a difficulty at competing as they achieve 99% coverage at only 0.2 IoU. However, edge-boxes can achieve better coverage when enough regions are generated as 98 - 100% of samples can be covered with high-quality regions when 100k regions are generated (see, Figure 3). With MSER and selective search this is not possible as they both generate a fixed number of windows depending either on salient regions found or depending on the number and quality of initial segmentation segments.

Note that in our evaluation the objectness measure performed the worst, mainly due to a poor set of initial windows. The method constructs a dense set of initial windows, however, the implementation [1] has difficulties generating smaller windows that cover smaller traffic signs, therefore, we excluded this method from further evaluation.

4.1.1 Finer resolution

We additionally evaluate proposals at finer resolution by doubling the size of each input image. Finer reso-

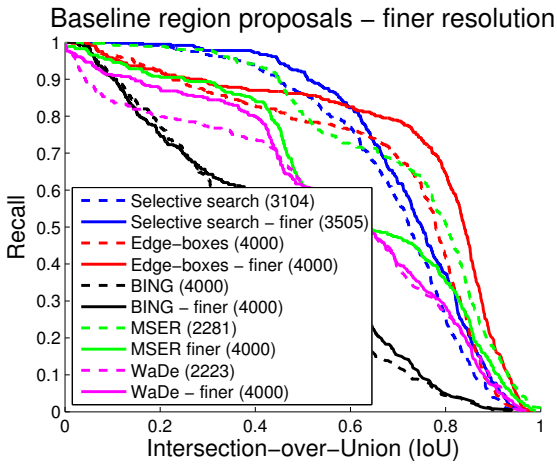


Figure 5: Comparing baseline region proposals at double the resolution. Note, values next to legend names are the number of regions used.

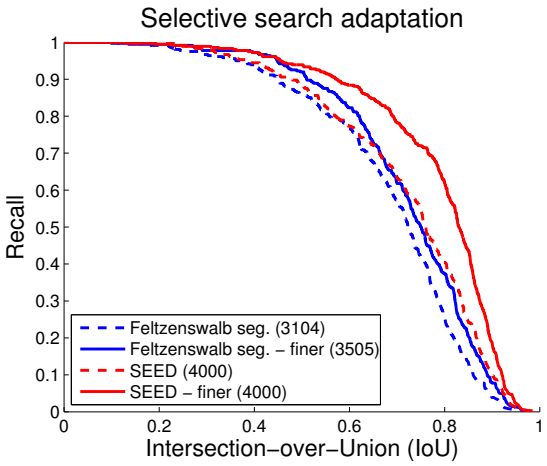


Figure 6: Results of evaluating selective search [22] using SEED [3] superpixels instead of Felzenszwalb and Huttenlocher [7] segmentation. Note, values next to legend names in are the number of regions used.

lution is better suited for our domain due to relatively small size of traffic signs. The results are shown in Figure 5, where improvements in almost all proposals can be observed. High resolution images improve edge-boxes the most, while the performance of MSER actually worsens. This happens due to a higher number of salient regions being returned, but as only the first 4000 of regions are selected to fairly evaluate all algorithms, some correct regions are discarded.

Result at the finer resolution also need to be normalized with the additional computational cost. Selective search is already slow at normal resolution, therefore, using finer resolution makes it even less useful. Higher resolution has little computational cost for BING, however, this method has the worst recall. The highest benefit is observed in edge-boxes, where multi-scale edge detection can be replaced with a single finer scale at little computational cost and a significant improvement in the performance.

4.2. SEED superpixels in selective search

We further evaluate replacing segmentation in selective search with SEED [3] superpixels to generate a higher number of regions. The results can be seen in Figure 6. A finer control over the size of initial segmentation when using SEED superpixels generates windows that capture smaller regions and improves overall performance. The performance is improved even further with finer resolution, matching the performance of edge-boxes. However, this improvement comes at a higher computational

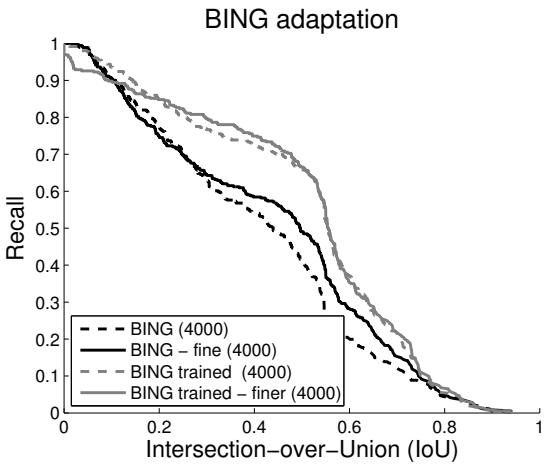


Figure 7: Results of evaluating domain-specific adaptation of BING [4] with gradient features trained on traffic signs. Note, values next to legend names are the number of regions used.

cost compared to edge-boxes, thus making selective search less attractive.

4.3. BING adaptation

Next, we evaluate the effect of domain-specific adaptation of BING with the results reported in Figure 7. The graph shows improved performance when training BING features on traffic signs over all IoU, with the highest improvement observed at the low quality regions. Despite improving the overall performance, the results are still significantly worse than in selective search or edge-boxes. The reason for

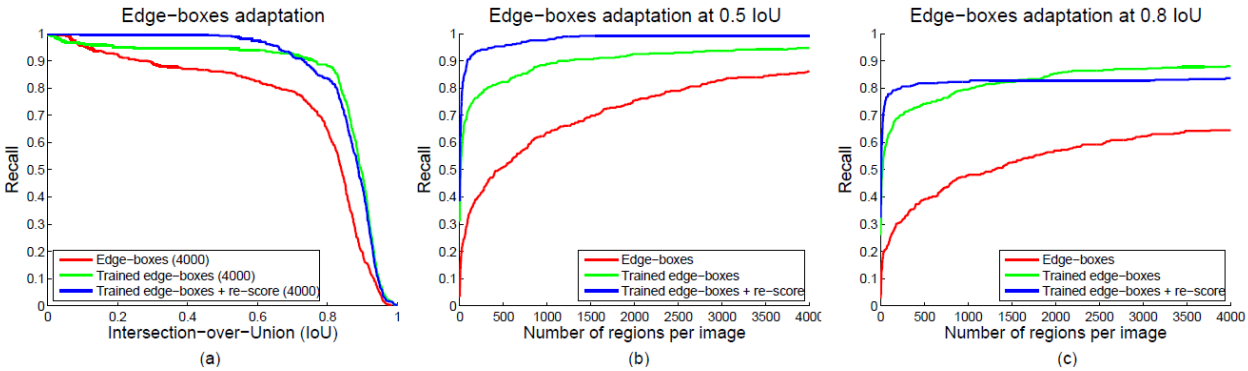


Figure 8: Results of evaluating domain-specific adaptation of edge-boxes [25] using domain-specific learning and re-scoring with shape information. Recall versus Intersection-over-Union overlaps is in (a) and recall versus number of regions in (b) for 0.5 IoU and in (c) for 0.8 IoU overlap. Note, values next to legend names in (a) are the number of regions used.

poor performance is a low resolution gradient feature that cannot sufficiently capture enough details in our domain.

4.4. Edge-boxes adaptation

With the final experiment we evaluate the effects of domain specific-adaptation applied to edge-boxes [25] as proposed in Section 3.4. The results can be observed in Figure 8. Both our proposed adaptations have proven to significantly boost the performance of region proposals, achieving recall of 0.99 at 0.5 IoU overlap and 0.90 at 0.8 IoU overlap. Learning structured edges alone is already able to capture 30% more traffic signs compared to generic structured edges. Moreover, all 90% of traffic signs are covered with high-quality regions with IoU over 0.8. Adding re-scoring with shape information further improves the region proposal, as almost 100% of traffic signs can be covered with 0.5 IoU overlap.

Additionally, an excellent performance can be achieved at a small number of windows, as can be observed in Figure 8. At both 0.5 and 0.8 IoU overlap the recall quickly converges to 0.9, requiring only between 1000 and 2000 region proposals to achieve this score.

5. Conclusion

In this paper we explored multiple region proposals in the context of traffic sign detection. We proposed to use region proposals as a first step in detection of all traffic signs to reduce the initial search space to a promising set of regions. Multiple state-of-the-art region proposals were evaluated: Objectness measure [1], selective search [22], BING [4] and edge-boxes [25]. To further increase the performance we proposed additional improvements in a form of domain-specific adaptation. Multiple adaptations where evaluated: two salient region detectors, MSER [16] and WaDe [20], replacing segmentation in selective search [22] with SEED superpixels [3], learning BING [4] feature on traffic signs and proposing domain-specific learning of edge-boxes [25] with re-scoring. The latter proved to be the most effective. We performed learning of edge-boxes by training structured edges on traffic signs, while for re-scoring we captured shape information with the magnitudes and orientations of structured edges and used linear SVM to learn the specific shape information. We showed that proposed method captures 99% of traffic signs on GTSDB [9], with 90% of objects covered with a high-quality regions. Furthermore, our proposed approach does not use hand-crafted features and is general enough to be applied to other domains as well.

In future, we will further extend the cascade using re-scoring based on trained color information. We will also evaluate the whole pipeline and explore the effects of region quality on various classifiers. We are also planing on assembling a new dataset containing traffic signs with additional categories, including direction signs, information signs and various road marking signs.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, pages 73–80, 2010.

This work was supported by ARRS research program P2-0214 and ARRS research project L2-6765.

[2] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2189–202, Nov. 2012.

[3] M. V. D. Bergh, X. Boix, and G. Roig. SEEDS: Superpixels extracted via energy-driven sampling. *European Conference on Computer Vision*, pages 1–19, 2012.

[4] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr. BING: Binarized Normed Gradients for Objectness Estimation at 300fps. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3286–3293, June 2014.

[5] P. Dollár, R. Appel, and W. Kienzle. Crosstalk cascades for frame-rate pedestrian detection. *European Conference on Computer Vision*, pages 1–14, 2012.

[6] P. Dollár and C. Zitnick. Structured forests for fast edge detection. *International Conference on Computer Vision*, 2013.

[7] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181, Sept. 2004.

[8] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? *British Machine Vision Conference*, pages 1–25, June 2014.

[9] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel. Detection of traffic signs in real-world images: The German traffic sign detection benchmark. *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, Aug. 2013.

[10] J. F. Khan, S. M. a. Bhuiyan, and R. R. Adhami. Image Segmentation and Shape Analysis for Road-Sign Detection. *IEEE Transactions on Intelligent Transportation Systems*, 12(1):83–96, Mar. 2011.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, pages 1–9, 2012.

[12] A. C. Le Ngo, L.-M. Ang, K. P. Seng, and G. Qiu. Colour-based bottom-up saliency for traffic sign detection. *2010 International Conference on Computer Applications and Industrial Electronics*, (Iccaie):453–457, Dec. 2010.

[13] C. Li, W. Song, L. Xiao, Y. Hu, and X. Pan. Salient traffic sign video detection based on hypercomplex frequency domain. *Proceedings of the 33rd Chinese Control Conference*, pages 7379–7382, July 2014.

[14] M. Liang, M. Yuan, X. Hu, J. Li, and H. Liu. Traffic sign detection by ROI extraction and histogram features-based recognition. *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, Aug. 2013.

[15] K. Lim, K. Seng, and L. Ang. Intra color-shape classification for traffic sign recognition. *Computer Symposium (ICS), 2010*, 2010.

[16] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, Sept. 2004.

[17] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool. Traffic sign recognition How far are we from the solution? *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, Aug. 2013.

[18] F. Moutarde, A. Bargeton, A. Herbin, and L. Chanussot. Robust on-vehicle real-time visual detection of American and European speed limit signs, with a modular Traffic Signs Recognition system. *IEEE Intelligent Vehicles Symposium*, 51(33):1122–1126, 2007.

[19] C. F. Paulo and P. L. Correia. Automatic Detection and Classification of Traffic Signs. *Eighth International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS '07)*, pages 11–11, June 2007.

[20] S. Salti, A. Lanza, and L. Di Stefano. Keypoints from symmetries by wave propagation. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2898–2905, June 2013.

[21] S. Segvic and K. Brkic. A computer vision assisted geoinformation inventory for traffic infrastructure. *13th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 66–73, 2010.

[22] J. Uijlings and K. van de Sande. Selective search for object recognition. *International Journal of Computer Vision*, 2013.

[23] G. Wang, G. Ren, Z. Wu, Y. Zhao, and L. Jiang. A robust, coarse-to-fine traffic sign detection method. *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–5, Aug. 2013.

[24] W.-j. Won, M. Lee, and J.-w. Son. Implementation of road traffic signs detection based on saliency map model. *2008 IEEE Intelligent Vehicles Symposium*, pages 542–547, June 2008.

[25] C. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. *European Conference on Computer Vision*, 2014.

20^{th} *Computer Vision Winter Workshop*
Paul Wohlhart, Vincent Lepetit (eds.)
Seggau, Austria, February 9-11, 2015

Cuneiform Character Similarity Using Graph Representations

Bartosz Bogacz[1], Michael Gertz[2] and Hubert Mara[1]

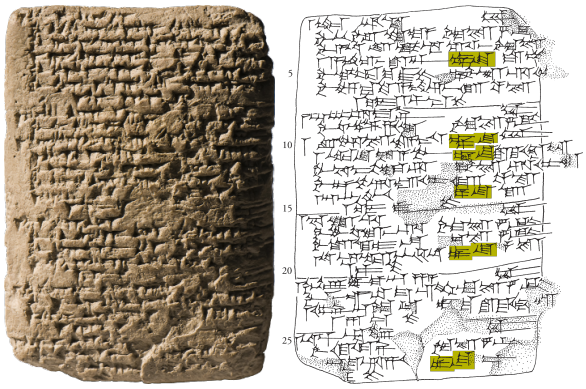
[1]Interdisciplinary Center for Scientific Computing (IWR),
Forensic Computational Geometry Laboratory (FCGL)
[2]Institute of Computer Science
Heidelberg University, Germany
{bartosz.bogacz|hubert.mara}@iwr.uni-heidelberg.de
gertz@informatik.uni-heidelberg.de

Abstract.

Motivated by the increased demand for computerized analysis of documents within the Digital Humanities we are developing algorithms for cuneiform tablets, which contain the oldest handwritten script used for more than three millennia. These tablets are typically found in the Middle East and contain a total amount of written words comparable to all documents in Latin or ancient Greek. In previous work we have shown how to extract vector drawings from 3D-models similar to those manually drawn over digital photographs. Both types of drawings share the Scalable Vector Graphic (SVG) format representing the cuneiform characters as splines. These splines are transformed into a graph representation and extend these by triangulation. Based on graph kernel methods we show a similarity metric for cuneiform characters, which have higher degrees of freedom than handwriting with ink on paper. An evaluation of the precision and recall of our proposed approach is shown and compared to well-known methods for processing handwriting. Finally a summary and an outlook are given.

1. Introduction

Cuneiform tablets are one of oldest textual artifacts comparable in extent to texts written in Latin or ancient Greek. Since those tablets were used in all of the ancient Near East for over three thousand years [22], many interesting research questions can be answered regarding the development of religion, politics, science, trade, and climate change [9]. These tablets were formed from clay and written on by impressing a rectangular stylus [2]. The result is a wedge shaped impression in the clay tablet. The word cuneiform derives from the Latin word "cuneus" *wedge* and "forma" *shaped*.



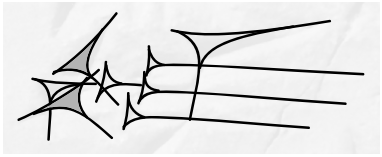
(a) (b)

Figure 1. Cuneiform tablet No. TCH92, G127 [8]: (a) Photograph and (b) its drawing. Six instances of the same two character tuple have been highlighted in yellow. A method for cuneiform character recognition would ideally classify those wedge configurations as highly similar.

There is an increasing demand in the Digital Humanities domain for handwriting recognition focusing on historic documents [20]. Even the recognition of ancient characters sharing shapes with their modern counterparts e.g. ancient Chinese Sutra [14] is a challenging task. For digitally processing cuneiform script there exist only a few recent related approaches like proposed in [6] using geometric features of cuneiform tablets acquired with a 3D-scanner [16].

However, with the aim of building a search tool for cuneiform tablets we have to consider the complexity of cuneiform characters in their de facto standardized

(a)



(b)

Figure 2. The cuneiform character for the syllable "zum" and its drawing.

2D-representation. Figure 1 shows a photograph of a cuneiform tablet and its drawing. Properties like the lack of fixed word length together with a wide variety of infixes, suffixes and prefixes prevents the application of existing machine learning methods on pictographs.

The extraction of the wedge shaped impressions of a cuneiform character is currently being approached by manually tracing a photograph of a cuneiform tablet using tools like Inkscape or by automatically extracting the boundary of a wedge configuration on basis of a 3D-model of the cuneiform tablet [15]. In either case, the result of an extraction is a document in the Scalable Vector Graphic (SVG) format. Figure 2 shows the cuneiform character for syllable "zum" and its drawing. The extraction of the wedges, looking like Ys, is challenging because these wedges are described with splines to retain all the damage and complexity of being written by hand. A clean extraction of the wedges is not sufficient to easily compare cuneiform characters. The configuration (position, orientation, grouping and overlap) and the shape of the wedges varies among different instances of the same character to a degree that requires a sophisticated character model to properly classify such characters.

2. Character recognition in raster images

Virtually all related research uses raster data as input. Word spotting is performed either on segmented lines [4, 24, 10] or on whole documents [18, 17]. The usage of Hidden Markov Models (HMMs) in these approaches circumvents the problem of learning fixed-length features for words or characters by decomposing the document or its lines into smaller features. The observations of the HMM are thin slices of a word, less than a character in width but with the same height as a line. A word is the represented as a succession of hidden states, each emitting a set of word slices. The advantage of this representation is that each slice is a fixed-length feature.

Wshah and colleagues [24] use direction gradients and a set of four intensities as features for a sliding window approach over already segmented lines. The query word is modeled to match a complete line by beginning and ending with filler characters modeling non-keywords to reduce the false positive rate.

To reduce the amount of required training on words lexicon-free handwritten word spotting approach using character HMMs [4] is applied. Then, the training of the HMM classifier only requires a small number of character classes. Just like in the work of Wshah and colleagues [24] filler models are employed, now consisting of a space character and all other character classes, to improve the retrieval precision.

Instead of directly using features extracted from the bitmap data, Rothaker and colleagues [17] use a Bag-of-Features representation with densely sampled Scale Invariant Feature Transform (SIFT) descriptors. These descriptors are then clustered into a dictionary with a limited set of words and quantized onto a regular grid overlapping the top of the document. No preprocessing of the document is necessary because the SIFT descriptors work directly on gray-scale data. A HMM classifier determines the most probable positions for the query word for all possible positions on the aforementioned grid. A segmentation of the document is therefore not necessary.

The work presented by Fischer and colleagues in [5] uses graphs as features to describe characters and measure similarity. Their approach requires an document already segmented into words. Images are first transformed into a color-binary representation and then thinned to one pixel medial axis curves. Graph vertices are created at endpoints, intersections and corner points of the medial axis curves. A HMM classifier is trained on thin slices of these word graphs.

The nature of writing cuneiform script poses a problem for HMM based classifiers. Cuneiform character traces have significantly more foreground-background transitions in the vertical axis than a word written in Latin. Classification with a HMM based approach would necessitate a larger feature

space of thin slices and therefore more training data for robust classification. Training data in the form of traced clay tablets is not readily available.

Furthermore, these approaches assume that word slices are always rigidly in the same order. Wedge shaped impressions, on the other hand, can locally interchange position, both in the vertical direction as well as in the horizontal direction, and yet still describe the same word. Graph based methods are more robust against such changes in topology.

A method for segmentation free word spotting is presented by Almazan and colleagues in [1] that uses exemplary SVMs to train one positive sample versus many negative samples. The document and the query are represented by grid of Histogram of oriented Gradient (HoG) descriptor cells. Training the SVM is done by using slightly translated windows of the query as positive examples. Negative examples are randomly selected windows of the document.

Although this approach does neither require any labeled samples nor a segmented document, the resulting SVMs only work very well on typeset or script written without much variation. Cuneiform text is highly variable in the expression of the wedges due to various factors such as the age of the clay tablet or the nature of the tool being used to impress wedges. An approach is necessary that offers more flexibility with respect to the deformation of the query word.

Howe presents a one-shot word spotting approach in [7] that does not require any training data. Words are binarized and represented as a tree of points connected by spring-like potentials. The document itself is then transformed using the structure of the tree of the query word. Locations where the transformation leads to a local energy maximum are those where the query word can be found.

Leydier and colleagues [12] use basic visual features found in written text to spot words in a document. The first order oriented image gradient is compared in specific image patches of the query word and document, so called zones of interest, to asses the similarity of words. The zones of interest themselves allow for an initial rough matching. The query word zones of interest are aligned to those locations of the document that share the same shape.

Both approaches do not assume a specific writing direction nor do they require segmented documents, but they do not allow for enough variation in character shape. The approach presented by Leydier and colleagues in [12] allows changes in positioning of the structuring elements using elastic cohesive matching, but does not allow for sufficient variability in the structuring elements themselves. Cuneiform wedges can be slightly rotated or elongated for aesthetic purposes and deform the zones of interest enough to preclude a successful match. Conversely, Howe [7] allows local variability but does not account for swapped characters.

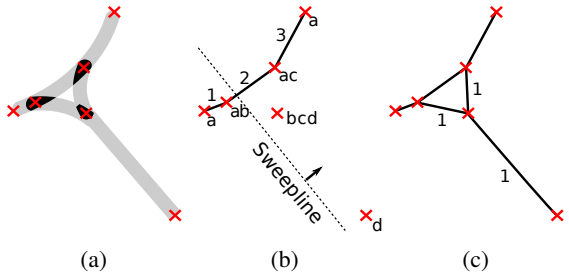


Figure 3. A cuneiform character in spline representation. (a) Closed spline paths forming strokes (gray) are pairwise intersected (black). (b) The vertices (marked X) of a stroke are first ordered from bottom right to top left. (c) Edges of the final graph are created by connecting the vertices. The sequence is indicated by the numbers.

3. From splines to graphs

Before any graph matching methods can be applied, the cuneiform characters first need to be transformed into well-formed graphs. Further, we segment cuneiform characters manually since cuneiform script has no visual word boundaries that would allow for automatic segmentation. The recognition of a word in the Assyrian language requires the knowledge of its grammatical case and context in which it is used. The clusters of wedges that have been manually segmented do not necessarily represent distinct characters in the original text. Nevertheless, they will be referred to as character in the following. A character consists of a set of strokes. Each stroke is a geometric shape bounded by a closed path of splines. These strokes are drawn in an vector graphic editor by assyriologitsts tracing a cuneiform tablet.

All strokes are pairwise intersected to create a set of key points. Most strokes are drawn in a way that there is only one closed unambiguous area of intersection. A vertex is placed at the center of such an area. Since more than two strokes can intersect in the same vicinity, the set of key points is pruned so that no two points are closer than some threshold ϵ. Figure 3 and Figure 4 illustrate this process. We set

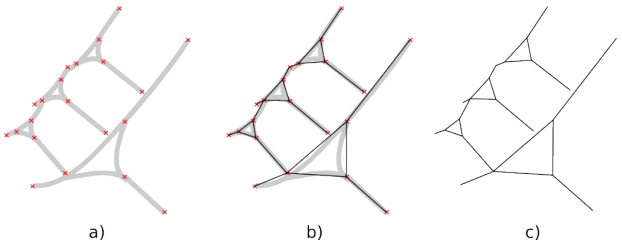


Figure 4. This figure shows the steps from Figure 3 applied to a complex set of strokes representing a character. In the majority of cases a small intersection of the arms is not semantically relevant.

$\epsilon = 1$. The threshold ϵ is expressed in tenths of a millimeter. The typical line height of cuneiform script is 5 millimeters. The choice of ϵ did not matter in our experiments as long as it was two orders of magnitude smaller than a character. We add the two endpoints of each stroke to the set of key points. Endpoints are calculated by finding the two most distant points on the spline enclosing a stroke.

The extracted points are not yet connected by edges. There is no inherent order of key points on a stroke between its endpoints. All points belonging to a stroke, that is its endpoints and points created with intersections with other strokes, are order geometrically and connected in sequence. In rare cases this may create incorrectly connected points if the stroke is slightly curved and the geometric ordering does not correctly represent the curvature of the stroke. Only few instances have been observed where this is the case.

4. Graph Similarity

After transforming the cuneiform characters from a collection of strokes into a graph representation, the characters can now be compared in similarity using common graph matching methods. Since small differences in position and orientation in wedge impressions do not change the meaning of a character, we assume that the graph topology is sufficiently descriptive to measure the similarity of cuneiform characters. We present three graph matching methods and extend each to work on the Delaunay triangulation of the cuneiform character graph. The Delaunay triangulation is used to catch big structural differences, significant translation or rotation of wedges, or wedge impressions in distinct graph components that do not modify the topology of the cuneiform character graph.

4.1. Weisfeiler-Lehman graph kernel

The graph kernel presented by the authors in [19] is an extension of the graph isomorphism test introduced by Weisfeiler and Lehman [23]. The kernel works by by counting how many subtrees the two graphs being compared share.

Each vertex of a cuneiform character graph is assigned a unique label. (Using the same label for each vertex results in significantly worse results.) On every iteration of the algorithm each vertex label is expanded with the labels of adjacent vertices. To increase computational performance all vertex labels can be converted into a shorter representation using hashing. Adjacent vertex labels have been, in turn, extended in an earlier iteration by their adjacent vertex labels. The label of each vertex is therefore an enumeration of a subtree rooted at this specific vertex.

The label, and therefore the subtree, contains multiple repetitions of itself since the root vertex is adjacent to each of its adjacent vertices. This behavior is called *tottering* [19] and degrades the quality of the labels and the quality of the similarity metric.

The similarity of two graphs G_A and G_B and their label sets N_A^k and N_B^k is the count of matching labels at iteration k. We denote the labels of N_A^k and N_B^k with e and f.The graphs are considered to be highly similar if most of the subtrees extracted from either graph are present in both graphs. $\delta(e, f)$ is the Kronecker delta, that is, $\delta(e, f) = 1$ if $e = f$, and 0 otherwise. We perform four relabeling iterations $n = 4$. More relabeling iterations ($n = 10$) did not result in better classification performance.

$$K = \sum_{k}^{n} \sum_{e \in N_A^k} \sum_{f \in N_B^k} \delta(e, f) \qquad (1)$$

4.2. Spectral decomposition

The spectral decomposition [3] of a graph has a variety of applications in the field of graph matching [13]. A graph is decomposed by computing the eigenvectors and eigenvalues of its adjacency matrix that has been at first converted into a normalized Laplacian matrix.

The resulting multi-set of eigenvalues and eigenvectors have many interesting properties [3] and are often used for clustering where the multi-set of eigenvectors can be used to find a nearly minimal cut. Additionally, the spectral decomposition of a graph

is used as an approximation for the random walk kernel on a graph [21]. We make use of the property of the spectral decomposition that two identical graphs have the same multi-set of eigenvalues. Such graphs are called to be *isospectral*. Small changes to those graphs result in only small changes to the multi-set of eigenvalues.

We compute the normalized Laplacian matrix L with components l_{uv} from the adjacency matrix A of a graph G with vertices u, v and vertex degrees d_u, d_v. L has the eigenvectors ϕ_i and the multi-set of eigenvalues λ_i.

$$\begin{aligned} L =& [l_{uv}] \\ l_{uv} =& \begin{cases} 1, & \text{if } u = v \text{ and } d_u \neq 0 \\ -\frac{1}{\sqrt{d_u d_v}}, & \text{if } u \neq v \text{ and } a_{uv} = 1 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{2}$$

$$\begin{aligned} \phi_i^T L \phi_i =& \lambda_i \\ & \lambda_1 \leq \cdots \leq \lambda_i \leq \cdots \leq \lambda_n \end{aligned} \tag{3}$$

The multi-set of eigenvalues is usually used as an embedding into a feature space where graphs are compared using Euclidean distance. We have found that using the cosine similarity to measure the angle between the eigenvalues of both decomposed graphs yields better classification performance.

The similarity of two graphs G_A and G_B can therefore be computed by measuring the angle between the features vectors λ^A and λ^B (the multi-sets of eigenvalues of the respective graphs A and B).

$$K = \frac{\langle \lambda^A, \lambda^B \rangle}{|\lambda^A||\lambda^B|} \tag{4}$$

4.3. Random walk graph kernel

The random walk kernel is based on the idea that two similar graphs share many identical walks [21] and their similarity can be measured by counting the number of identical walks.

A naive and computationally very expensive approach would be to generate walks for two graphs randomly, and to compare all pairs of walks. A faster approach makes use of the properties of the product graph of the two graphs being compared. The product graph is constructed by computing the Kronecker product of the two graph adjacency matrices. The product graph has an edge only if the corresponding nodes in both of the original graphs are adjacent.

Exponentiating an adjacency matrix of graph is used to count the number of walks in a matrix. Exponentiating the adjacency matrix of a product graph therefore leads to the number of shared walks in both original graphs. The computation of such a random walk kernel is a deterministic process that converges towards the stochastic solution with each iteration of the exponentiation. The count of iterations is also the maximal walk length to be found and is denoted by n. We set $n = 10$. We found that higher values ($n = 20$) did not improve classification performance. A_A and A_B are the adjacency matrix of the graphs G_A and G_B being compared. The operator $\otimes$ is the Kronecker product of two matrices. K is the resulting kernel computing the similarity of two graphs.

$$\begin{aligned} R &= A_A \otimes A_B \\ K &= \sum_k^n R^k \end{aligned} \tag{5}$$

4.4. Delaunay triangulation

Transforming cuneiform characters in spline representation to a representation as graphs can result in a graph with multiple disconnected components. Topology based graph kernels as described in the previous sections do not pick up differences in graphs if one of these components is geometrically translated or rotated.

We extend all three presented methods by triangulating the extracted points using the Delaunay triangulation and additionally measuring the similarity between the Delaunay triangulated characters graphs.

The Delaunay triangulation should also consider geometrical translations and rotations where the spatial relationship (a wedge is below/above/right of/left of another wedge) significantly changes between the wedges. Small changes in position or shape do not matter for the classification of a character.

The graph kernel methods are extended by computing a new adjacency matrix (therefore new edges) from the key point set without considering the strokes the key points originate from. Edges are created instead by the Delaunay algorithm. Let D_A be a adjacency matrix of a Delaunay triangulated character graph G_A and A_A be the original adjacency of character graph G_A and K a graph kernel from one of the presented methods. K' is then a graph kernel that computes the similarity between two character graphs G_A and G_B.

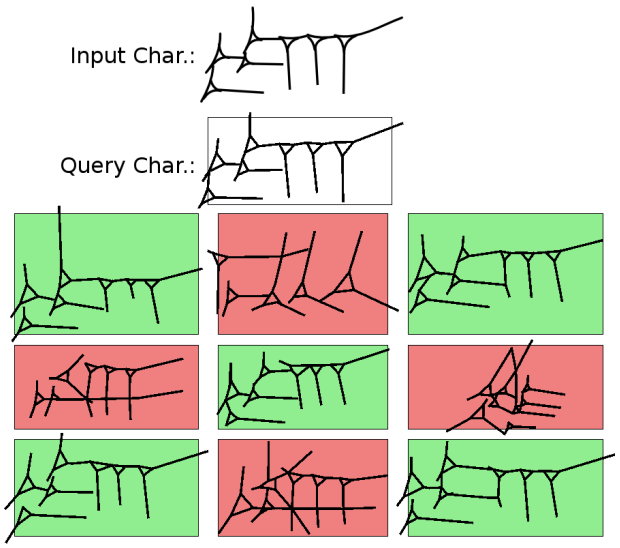


Figure 5. The top 9 results for the task to retrieve the Query Character (the prototype) are displayed. The Input Character is represented as a set of strokes. This set is then transformed into the graph representation of the Query Character. The kernel used for similarity is the spectrum kernel extended with Delaunay triangulation. The results are ordered from best (top left) to worst (bottom right). Characters with a green background have been correctly classified, characters with a red background have been incorrectly classified.

$$K' = \max\{K(A_A, B_A), K(D_A, D_B)\} \quad (6)$$

We also tried the $\min$ operator but the classification performance was worse for all kernels except for the random walk kernel where the improvement in performance was negligible.

5. Experimental Evaluation

The data set used are a subset of several hundred 3D-scanned cuneiform tablets and tablets provided and manually transcribed into a vectorized file format by assyriologists. Only a subset of the words has been segmented manually since the tablets were partly damaged. There are 23 distinct word classes and 73 word instances used in the data set.

The task to test the classification performance of the presented methods was performed by hiding a prototype instance of the segmented words and comparing the remaining word instances against the prototype instance. The retrieved candidates were ranked by similarity from most similar to least similar.

The classification performance of the presented methods was then compared using a precision recall graph.

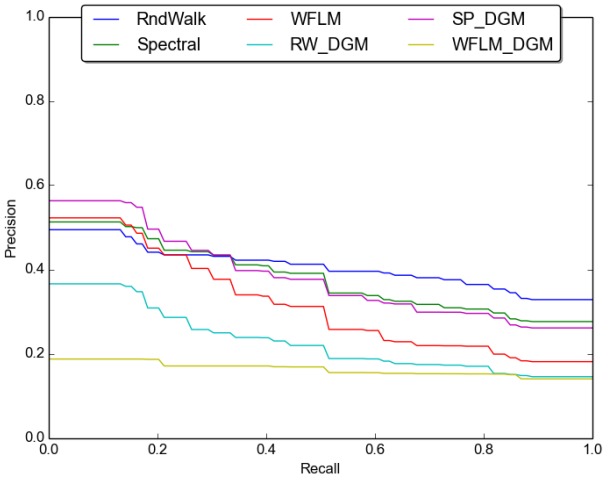


Figure 6. The precision recall graph for all the presented methods. A high precision implies that most of the retrieved character have the correct label, the false positive rate is low. If we increase the recall (the count of true positives and false positives, therefore ask for more results) the false positive rate climbs and the precision falls.

5.1. Precision and Recall

Figure 6 shows the classification performance of the various methods. The three basic methods are: the Weisfeiler-Lehman Graph Kernel (*WFLM*), the spectral decomposition (*Spectral*) and the random walk graph kernel (*RndWalk*). Then, the methods extended with the delaunay triangulation are *WFLM_DGM*, *SP_DGM* (for the spectral decomposition) and *RW_DGM* (for the random walk kernel), respectively.

The Delaunay transformation reduces precision greatly for the Weisfeiler-Lehman graph kernel. This kernel counts the number identical subtrees in both graphs. Since many vertices in a Delaunay triangulated graph have the same degree, two geometrically dissimilar triangulated graphs will share a high number of subtrees rendering them indistinguishable for the Weisfeiler-Lehman graph kernel.

The decrease in performance for the random walk kernel can be attributed to the same problem. Dissimilar triangulated graphs share a lot of random walks since most vertices are reachable by a high number of different walks.

The spectral decomposition, on the other hand, has better precision when extended with delaunay transformed graphs. The spectral decomposition can be seen as a series of minimal cuts [3] of a graph where the edge density is lowest. Translation and ro-

tation of wedges are therefore detectable by changes in connectivity of the graph partition leading to a better classification performance than just using the graph topology.

The random walk method and the Weisfeiler-Lehman graph kernel achieve better classification performance when the untransformed cuneiform graphs are used. Much more variating node degree and unique walks in the untransformed graphs enable those methods to differentiate cuneiform characters graphs better.

6. Conclusions and Outlook

Common handwriting recognition methods are not applicable to cuneiform characters. The Assyrian language has no means of separating words, thus making word segmentation very difficult. Cuneiform characters are very variable with respect to the positioning and rotation of their wedges and also exhibit a lot of complexity in the vertical direction without having a fixed shape that can be used by fixed-length feature vector classification methods.

We transform cuneiform characters into graphs and find that such a representation does not lose any structural elements of cuneiform and is very suitable for further analysis of the characters.

We applied graph kernels to classify cuneiform characters with the result that the random walk kernel performs best. The spectral decomposition, on the other hand, performs best when extended with the Delaunay triangulation and achieves the highest classification precision of all the presented methods.

We are currently working on using the wedge shaped impressions as a basic structural feature of cuneiform characters. A template shaped like an ideal wedge is used to match and extract wedges in a cuneiform character. The characters, decomposed into wedge shaped templates, are compared based on the similarity of their wedge shapes and the quality of the matching of the wedge configuration (position, orientation, overlap). To support our claim that conventional OCR methods are not suitable for cuneiform script we are currently investigating the classification performance of standard HMM and DTW methods on rasterized cuneiform script.

Additionally, we are investigating a method that represents the cuneiform characters as point clouds. Query word and candidate alignment and subsequent matching is performed with Iterated Closest Points [11].

Acknowledgements

We thank Prof. Stefan M. Maul and the members of the *Assur-Forschungsstelle* in Heidelberg, Germany for their support and fruitful discussions. This work is partially funded by the *Massnahme 5.4 – Zukunftskonzept* (institutional strategy) of the 2nd *German University Excellency Initiative*. Within this initiative the work is part of the *Field of Focus 3: Cultural Dynamics in Globalised Worlds*. Additional support is provided by the *Heidelberg Graduate School of Mathematical and Computational Methods for the Sciences* (HGS MathComp).

References

[1] J. Almazan, A. Gordo, A. Fornes, and E. Valveny. Segmentation-free word spotting with exemplar svms. *Pattern Recognition*, 47(12):3967–3978, 2014. 3

[2] R. Borger. *Mesopotamisches Zeichenlexikon*. Alter Orient und Altes Testament / Alter Orient und Altes Testament. Ugarit Verlag Münster, 2004. 1

[3] F. R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997. 4, 6

[4] A. Fischer, A. Keller, V. Frinken, and H. Bunke. Lexicon-free handwritten word spotting using character hmms. *Pattern Recogn. Lett.*, 33(7):934–942, May 2012. 2

[5] A. Fischer, K. Riesen, and H. Bunke. Graph similarity features for hmm-based handwriting recognition in historical documents. In *Proceedings of the 2010 12th International Conference on Frontiers in Handwriting Recognition*, ICFHR '10, pages 253–258. IEEE Computer Society, 2010. 2

[6] D. Fisseler, F. Weichert, G. G. Mller, and M. Cammarosano. Extending Philological Research with Methods of 3D Computer Graphics Applied to Analysis of Cultural Heritage. In *Proc. of 12th Eurographics Workshop on Graphics and Cultural Heritage*, pages 165–172, Darmstadt, Germany, 2014. 1

[7] N. R. Howe. Part-structured inkball models for one-shot handwritten word spotting. In *Proceedings of the 2013 12th International Conference on Document Analysis and Recognition*, ICDAR '13, pages 582–586. IEEE Computer Society, 2013. 3

[8] S. Jakob. *Die mittelassyrischen Texte aus Tell Chuera in Nordost-Syrien*. Harrassowitz, O., July 2009. Tafel 26. 1

[9] D. Kaniewski, E. V. Campo, and J. G. et al. Environmental roots of the late bronze age crisis. *PLoS ONE*, 8(8), 2013. 1

[10] Y. Kessentini, C. Chatelain, and T. Paquet. Word spotting and regular expression detection in handwritten documents. In *ICDAR*, pages 516–520. IEEE, 2013. 2

[11] W.-S. Kim and R.-H. Park. Fast icp algorithm using a one-dimensional search. In *MVA*, pages 435–438, 2000. 7

[12] Y. Leydier, F. Lebourgeois, and H. Emptoz. Text search for medieval manuscript images. *Pattern Recognition*, 40(12):3552 – 3567, 2007. 3

[13] B. Luo, R. C. Wilson, and E. R. Hancock. Spectral embedding of graphs. *Pattern Recognition*, 36(10):2213 – 2230, 2003. 4

[14] H. Mara, J. Hering, and S. Krömker. GPU based Optical Character Transcription for Ancient Inscription Recognition. In *Proc. of 15th International Conference on Virtual Systems and Multimedia (VSMM) – "Vision or Reality? Computer Technology and Science in Art, Cultural Heritage, Entertainment and Education"*, pages 154–159, Vienna, Austria, September 2009. 1

[15] H. Mara and S. Krömker. Vectorization of 3D-Characters by Integral Invariant Filtering of High-Resolution Triangular Meshes. In *Proc. of the 12th Int. Conf. on Document Analysis and Recognition (ICDAR)*, pages 62–66, Washington D.C., USA, 2013. IEEE. 2

[16] H. Mara, S. Krömker, S. Jakob, and B. Breuckmann. GigaMesh and Gilgamesh - 3D Multiscale Integral Invariant Cuneiform Character Extraction. In A. Artusi, M. Joly, G. Lucet, D. Pitzalis, and A. Ribes, editors, *Proc. VAST Int. Symposium on Virtual Reality, Archaeology and Cultural Heritage*, pages 131–138, Palais du Louvre, Paris, France, 2010. Eurographics Association. 1

[17] L. Rothacker, M. Rusiol, and G. A. Fink. Bag-of-features hmms for segmentation-free word spotting in handwritten documents. In *Proceedings of the 2013 12th International Conference on Document Analysis and Recognition*, ICDAR '13, pages 1305–1309. IEEE Computer Society, 2013. 2

[18] M. Rusiol, D. Aldavert, R. Toledo, and J. Llads. Efficient segmentation-free keyword spotting in historical document collections. *Pattern Recognition*, 48(2):545 – 555, 2015. 2

[19] N. Shervashidze, P. Schweitzer, E. J. van Leeuwen, K. Mehlhorn, and K. M. Borgwardt. Weisfeiler-lehman graph kernels. *J. Mach. Learn. Res.*, 12:2539–2561, Nov. 2011. 4

[20] M. Thaller, editor. *Optical Character Recognition in the Historical Discipline. Proc. of an Int. Workgroup by the Netherlands Historical Data Archive, Nijmegen Institute for Cognition and Information*, volume A18 of *Halbgraue Reihe zur Historischen Fachinformatik*, Göttingen, Germany, 1993. Max-Planck-Institut fr Geschichte. 1

[21] S. V. N. Vishwanathan, N. N. Schraudolph, R. Kondor, and K. M. Borgwardt. Graph kernels. *J. Mach. Learn. Res.*, 11:1201–1242, Aug. 2010. 5

[22] W. von Soden. *The Ancient Orient: An Introduction to the Study of the Ancient Near East*. Lightning Source Incorporated, 1994. 1

[23] B. Weisfeiler and A. A. Lehman. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsia*, 2(9), 1968. 4

[24] S. Wshah, G. Kumar, and V. Govindaraju. Statistical script independent word spotting in offline handwritten documents. *Pattern Recognition*, 47(3):1039–1050, 2014. 2

20^{th} *Computer Vision Winter Workshop*
Paul Wohlhart, Vincent Lepetit (eds.)
Seggau, Austria, February 9-11, 2015

Classification of cellular populations using Image Scatter-Plots

Florian Kromp
Children's Cancer Research Institute
Zimmermannplatz 10, 1090 Wien
florian.kromp@ccri.at

Michael Reiter
Vienna University of Technology
Favoritenstrasse 9-11/183-2, 1040 Wien

Sabine Taschner-Mandl
Children's Cancer Research Institute
Zimmermannplatz 10, 1090 Wien

Peter F. Ambros
Children's Cancer Research Institute
Zimmermannplatz 10, 1090 Wien

Allan Hanbury
Vienna University of Technology
Favoritenstrasse 9-11/188, 1040 Wien

Abstract. *We present a novel semi-automatic approach to classification of biologically meaningful cell populations in imaging cytometry. Of each cell in a set of multispectral microscopic fluorescence image (where each image contains several hundred cells) we obtain morhological, gray level histogram and textural feature measurements. The user provides initial class labels by roughly marking populations with polygons in scatter-plots of two-dimensional feature space projections, a process called "gating". The scatter-plots are enhanced by the multispectral images of each cell, providing the user with visual information about cell morphology to support gating. Typically, gating produces many false assigments which are automatically corrected by a strategy akin to bagging and consensus vote. To prove for validity, we compared our results to annotated samples and to state-of-the-art flowcytometric analysis results.*

1. Introduction

Cytomics deals with the analysis of biological samples consisting of a large number of cells with the goal to classify, and thus, to quantify populations in the sample, i.e. groups of cells of the same cell (sub-)type [10]. The quantitative analysis of samples is important for the understanding of cellular mechanisms including the quantification of protein subcellular appearance or DNA elements [14], [7].
In cytomics, each cell of a sample is described by several descriptive feature measurements. To enable the feature extraction, several parts of the cells have to be visualized, including the cell membrane, the nucleus and the antibody expression. To do so, fluorescent labeled antibodies are used in combination with e.g. DNA binding dyes [19]. Basically, antibodies are proteins attaching to specific cellular targets called antigens representing nuclear or cellular structures or proteins and genes. Antibodies are labeled with a fluorescent protein emitting light of a specific wavelength when illuminated. Using different fluorescence labeled antibodies allows for the visualization of different targets.
There are two basic methods for acquiring measurements on a single cell level: Flow Cytometry Measurement (FCM) and Fluorescence Microscopy (FM). FCM devices measure light intensity of single cells floating in a fluidic stream, whereas FM methods capture multispectral images of cells attached to glass slides. In contrast to FCM, FM methods require image analysis techniques to extract signal intensities of single cells from the resulting multispectral images. Both techniques result in feature vectors extracted on a single cell level. The features have to be processed subsequently to obtain a meaningful interpretation of the underlying data, which is subject of the current article.

When comparing FM and FCM features, the discriminative power of FCM features is higher in general. This is due to the detection method, since the dynamic range of detectors used in FCM devices is

a multiple times higher when compared to sensors used in CCD grayscale cameras or even compared to detectors used in laser scanning microscopes. Nevertheless, image analysis enables the researcher to perform analysis strategies including morphological features [12] as the basis for a subsequent classification of cellular populations. Although a new generation of Imaging Flow Cytometers arose enabling a simultaneous measurement of signal intensities and acquiring images, FM image analysis bears the advantage to relocate cells on the slide, which is not possible when using FCM technologies. Relocation on the slide enables the operator to use high magnifying lenses to investigate a cell in detail, which can be an advantage if antibody expression pattern of the cell is difficult to analyze. Furthermore, grown cells as well as tissue sections can only be analyzed using FM technologies. Ecker et al. proposed the term tissue cytometry for tissue analysis on a single cell level [5]. A comparison outlining the strength of the different methods is presented by Barteneva et. al [1].

In this work, we introduce a novel method for the classification of cellular populations in FM images: we adopt the gating strategies derived from FCM domain to label observations in scatter plots [13] and combine it with the cropped images stored during the process of image analysis. This leads to a powerful visualization we call Image Scatter-Plots (ISP), which is embedded in an analysis workflow used to classify cellular populations sequentially. Since the human brain is able to analyze and classify patterns fast, we use this skill to guide the quantification process in the analysis. In contrast to FCM gating, single observations or also groups of observations gated in the ISP can be relocated in the multispectral images or even directly on the slide. Thus, results of the gating strategies can be verified easily and intuitively increase the researchers confidence in the analyzed data.
Typically, the relevant populations can not be selected by simple polygonal regions. They do not form well-separated clusters in the ISP, and even if multiple gates drawn in different ISP are combined, there are a substantial number of false assignments. This is also due to the use of "weak" image features. To correct for the false assignments, we use random forests or principal component analysis (PCA) for feature selection and compare automated clustering with majority vote of an ensemble of classifiers.
The proposed methodology for classification of cellular populations is embedded in a workflow as presented in Figure 1. While the preprocessing consisting of image segmentation and feature extraction is a widely explored field, we introduce the gating strategies applied on ISPs as a novel method for cellular population classification and thus, quantification. Combining manual input with machine learning strategies to correct for overlapping cellular populations enables a reliable quantification for the samples used in this work.

2. Cellular population classification using ISPs and manual gating

The idea of ISPs was inspired by the work of Hamilton et al. [9], who created representative plots by sorting cellular images according to the distance between their vectors of threshold adjacency statistics [8]. In contrast to the work of Hamilton, we alter the method by enabling the operator to choose which features to use for the two dimensional projection. Furthermore, it is possible to load multispectral images, so the user, further called the operator, can display the opacity for every channel, emphasizing the channel of interest. The underlying aim is to set gates as explained in section 2.5 to outline the population of interest. In contrast to an FCM analysis, the operator is supported by the visualization of the antibody expression pattern and nucleus morphology. Thus, different populations can be classified intuitively and efficiently even if the different populations strongly overlap in the 2-dimensional projection.

2.1. Image Segmentation

A prerequisite for antibody quantification and for a subsequent cellular classification is a robust image segmentation algorithm. Segmentation is the process of outlining distinct objects, resulting in a new image of the same size as the analyzed image, called segmentation mask. In this work, two types of images are segmented, depending on the location of the antibodies of interest: the nuclear image and the cytoplasm image, see Figure 1 B for an example.
We use the Gradient Energy Tensor for nuclear image segmentation as described in [11]. To obtain a cytoplasmic segmentation, we use the image channels visualizing cellular surface markers. These markers are membrane bound, but represent the cytoplasm due to out-of-focus fluorescence signals. The images are simply added and the resulting image is transformed into a binary image using the Otsu threshold

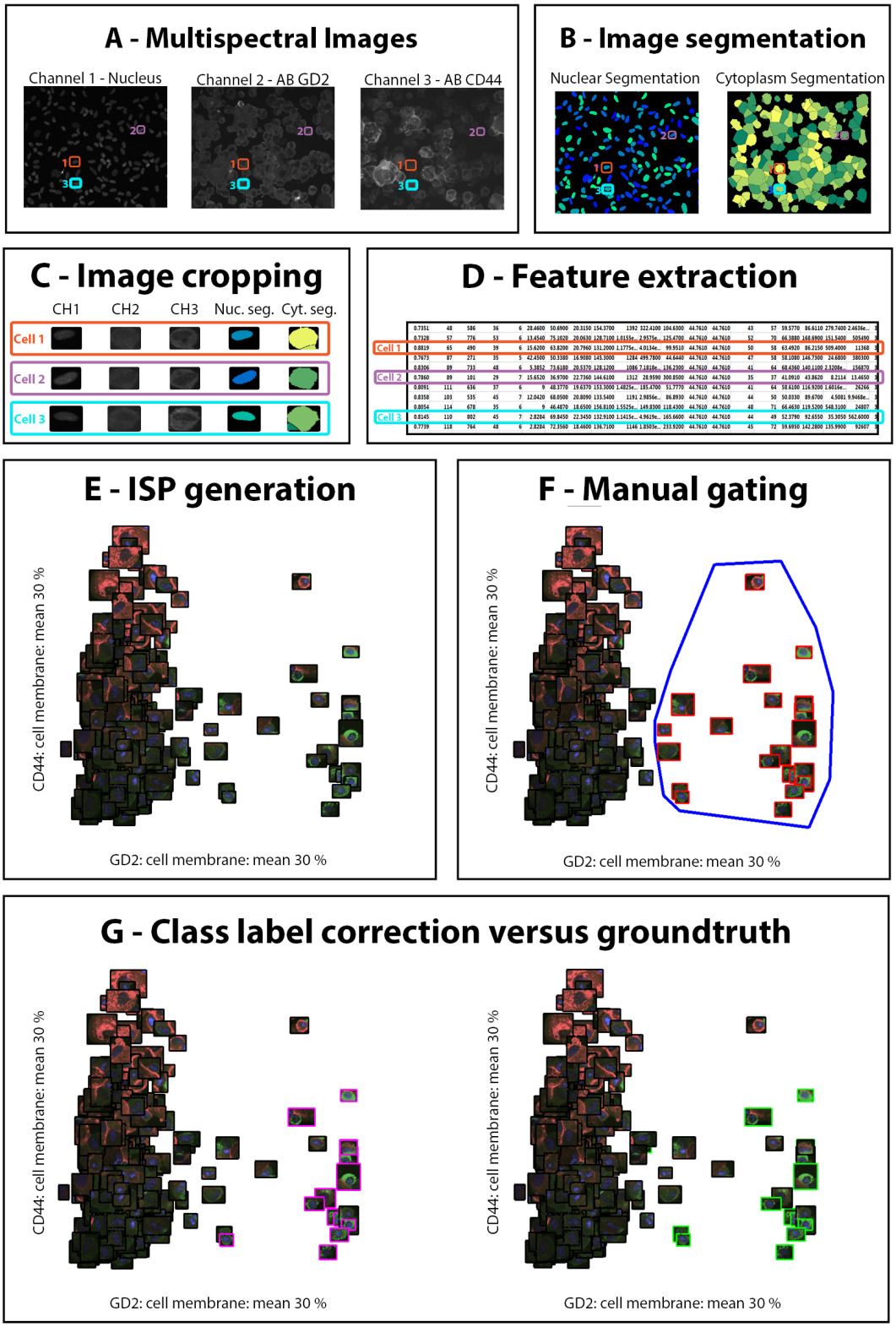


Figure 1. Proposed workflow for cellular population quantification illustrated on a neuroblastoma dataset. Preprocessing (A-D) and contribution of this work (E-G); (A) Channels of the multispectral image each outlining a different cellular target, (B) Segmented nuclear image (left) based on Channel 1 and cytoplasm image (right) based on Channels 2 and 3, (C) Images of observations 1 to 3 cropped in all channels of the multispectral image and in the segmented images, as marked in images in A and B, (D) Extract of the resulting feature matrix, rows corresponding to observations and columns corresponding to feature vectors; features vectors of observations 1 to 3 highlighted, (E) Image Scatter-Plot created using two features selected by the operator; pseudocolors are used to overlay the cropped images of A, (F) Observations were labeled manually (red) by setting a gate (blue) to outline a cellular population, (G) Results of automated label correction to compensate for overlapping populations (violet, left image) and groundtruth (green, right image).

[16]. We use the nuclear segmentation mask as the reference points for a distance transformation subsequently. Finally, the watershed algorithm is used to split touching cells based on the distance transformed image.

2.2. Image cropping

To enable a generation of an ISP after feature measurement, the input images being subject to the analysis have to be cropped. Thus, a minimal bounding box containing the segmented object is cropped for each channel of the multispectral image, for each object separately. Usually, antibodies are located either in the nucleus or in the cytoplasm, depending on the targeted antibody. Thus, we crop a region containing either the nucleus or the cytoplasm region. In Figure 1 C, images of the nuclear as well as of the antibody channels are cropped.

2.3. Feature measurement

We use three types of features, morphological features, features of the gray level histogram and textural features. Morphological features include roundness, perimeter, object size (nuclear and cytoplasm) and solidity. Features of the gray level histogram include mean intensity, median intensity and mean of the 30 percent of brightest pixels. In our experiments, this turned out to be a robust measure of antibody expression which is insensitive to cell size. Each of these features is measured using the nuclear and the cytoplasm mask, respectively, applied to the raw images by simple multiplication. Texture features are features extracted from the gray level histogram of filtered images. We apply local binary patterns (LBP) [15] using a radius of 1, 2 and 3 and either 8, 16 or 32 neighbours to the cropped images. Moreover, we apply Gabor wavelets [4] using three scales and 8 orientations to the cropped images as well.
To compensate for instable texture representations close to the border of the nucleus or the cytoplasm, we only use pixels having a distance of more than 2 pixels to the border of the nucleus or the cytoplasm, indicated by the respective mask, to calculate the gray level histogram. Each of the texture features is measured in all channels of the multispectral image, where the number of channels depends on the number of targeted antibodies. For the samples analyzed in this work we used 3 channels. Currently, a maximum of 6 channels is possible due to the characteristic of overlapping fluorescence spectra [19]. However, there is an ongoing development in this field to increase the number of well discriminating fluorescence dyes.
Thus, we extract 18 LBP features and 48 Gabor features for each channel available. The number of morphological features we extract is between 4 and 5, depending on the location of the antibody, since size of the cytoplasm can only be calculated if a channel representing cytoplasm is available. The number of features we extract from the gray level histogram of the cropped raw images is between 4 and 8, depending on the antibody expression pattern as well as on the availability of a cytoplasm channel. For texture feature extraction, we use the mean and variance of the gray level histogram resulting from image filtering. All of the features are normalized according to mean and standard deviation.

2.4. ISP generation

To generate an ISP, the feature space is projected on two dimensions selected by the operator. Then, the observations are plotted in a scatter plot. In contrast to commonly used scatter plots, we plot the cropped images, see Figure 1 E. To enable the operator to gate cellular populations, the features being most discriminative for the characterisation of the different populations are chosen and assigned to the axes of the ISP. We only provide morphological features and features of the gray level histogram to be selected in the ISPs, since they are meaningful for the expert operator. To obtain the most discriminative features, the operator selects different features for both of the ISP axes repeatedly until obtaining a projection where populations are well separated. In addition to the aforementioned features the first two principal components of the feature vector can be chosen as axis for the ISP. The operator can choose the opacity as well as the pseudocolor for each channel of the multispectral image in order to highlight the antibody channel of interest.

2.5. Setting gates to label cellular populations initially

Once the two suitable axes are chosen, the gate can be drawn in the ISP. Setting a gate is equal to declaring a region containing a population of interest. While setting the gate, the observations being inside the gate are assigned to one class, while all of the other observations are assigned to another class, see Figure 1 F for an example.

3. Automated class label correction

In FCM technology, generally fewer features (up to 20) are used which have high dynamic range and discriminiative power. In contrast, in FM imaging technology a large number of relatively weak FM features is used (populations tend to overlap in the ISP projections). Because polygonal gates are a coarse selection method they will produce a large number of false assignments. The grade of overlap depends on the antibody used, the cells and the quality of the staining. The misclassified observations can be regarded as outliers and have to be removed from the particular populations. Thus, we have to use strategies to detect outliers and reassign them to the correct classes to improve accuracy. We do not assume simple parametric form of the class-conditional distributions, and thus, we choose a non-parametric approach including a consensus vote of an ensemble of classifiers.
The number of features used ranges from 50 to 220 features, depending on the the number of channels used and the features selected by the operator. Sample size used for this type of analysis is between 400 and 5000 observations. Nevertheless, due to the curse of dimensionality the feature space has to be reduced to allow for a reliable classifier training.

3.1. Feature reduction

To reduce the number of features, we compared two methods: PCA and random forest (RDF).
When applying PCA to the sample, we used the first 15 eigenvectors for projection, covering more than 95 percent of the variance, for both samples analyzed. A better way would be to perform a linear discriminant analysis as proposed by Fisher, incorporating the class labels set by manual gating. LDA is searching for a one-dimensional projection maximizing intra-class variance while minimizing inter-class variance. Due to numerical instabilities when applying LDA, we investigated a feature selection method to remove features irrelevant for classification.
We decided to use (RDF) for the task of feature selection, and thus, for feature reduction, as is recommended for use in biological applications with weak discriminative features by Saeys et al. [18]. RDF trees are grown on bootstrap samples randomly created on a training dataset, the method is called Bagging (bootstrap aggregation) [2]. We used the class labels obtained from manual labeling by gating to create the training data. To obtain a measure indicating the importance of the distinct features, observations not being part of the bootstrap sample (out-of-bag observations) used to create a specific tree are predicted. The increase in prediction error when permuting the out-of-bag observations for prediction is the measure of importance, averaged over the ensemble of trees and normalized according to the standard deviation over the ensemble.

We set the parameters of the RDF according to a minimal leaf size of 10 and an ensemble of 40 trees. Since we are only interested in calculating the feature importance, we do not take into account generalization performance of the ensemble of trees. Due to instabilities in the feature importance measure for repeated RDF constructions, we performed the RDF growing 10 times and only kept features for which the overall mean of the importance was above zero. We assume that those features have a positive importance and thus are useful for discriminating the cellular populations. The instabilities most likely occur due to the low number of observations related to the number of features used.

3.2. Automated clustering versus classifier ensemble

The samples we use have groundtruth generated by biologists. They contain two types of cells, the aim is to separate the two cellular populations. Assuming no other cellular populations are present in the sample, we performed a k-means clustering on the RDF or PCA reduced feature space using k=2. We tested the accuracy of k-means within 15 runs using random initialization on one dataset, resulting in an accuracy below 80 percent in 14 cases. In only one attempt we achieved an accuracy of about 98 percent indicating the convergence in a global minima. When choosing the class means of the manual labeled classes for centroid initialization, the k-means was guided to converge into this global minima.
In order to incorporate more information retrieved from the classes labeled by manual gating, we investigated a method proposed by Brodley [3] and compared it to k-means clustering. Class labels of single observations are rated by using n-fold cross-validation on an ensemble of classifiers and majority vote for decision finding. We used an ensemble of eight classifiers including a support vector machine (SVM) with a linear kernel and seven k-nearest neighbor classificators with k=3, 5, 7, 9, 11, 13 and 15. We decided to keep the class label of a single

Feature reduction	accuracy (in %) manual	k-means	majority vote
-		98.82	98.43
PCA	97.45	98.82	98.62
RDF		98.23	97.45

Table 1. Comparison of accuracies for the different strategies on dataset 1.

Feature reduction	accuracy (in %) manual	k-means	majority vote
-		89.96	93.80
PCA	94.02	89.96	94.66
RDF		80.98	96.15

Table 2. Comparison of accuracies for the different strategies on dataset 2.

observation based on the classification results of the ensemble of classifiers trained on all other observations. If more than 50 percent of the classifiers vote for the observation to be part of the assigned class, the class label is kept (majority vote). Otherwise, the observation will be assigned to to the other class. In comparison to the RANSAC algorithm [6], we don't have to know the exact model to use, this is implicitly given by the data analyzed.

4. Results

To validate the proposed method, the biologists in the Tumor Biology Lab at the Children's Cancer Research Institute created the groundtruth for 2 samples consisting of 17 images, labeling a total of 977 observations. Furthermore, we could compare the results of cellular population quantification for one more sample to the results of a state-of-the art flow-cytometric analysis. The manual gates in the ISPs were set by a biologist expert and the author.

4.1. Comparison to annotated groundtruth

The first sample (dataset 1, number of observations n=509) contains 10 images of a co-culture of N cells (neuroblastoma cells) and F cells (flat cells). The cells were marked using GD2, a tumor marker expressed in nearly all neuroblastoma cells, and CD44, a cellular surface molecule associated with certain activities of cancer cells. After applying the RDF for feature reduction, the feature space was reduced from 220 to 17 features. The results are presented in Table 1.

The second sample (dataset 2, n=468) contains 7 images of Schwann cells (cells of the peripheral nervous system) and F cells. The cells were marked using Vimentin, outlining the cytoskeleton and present in all of the cells, and S100, a Schwanncell marker preferentially expressed in Schwann cells. After applying the RDF for feature reduction, the feature space was reduced from 205 to 34 features. The results are presented in Table 2.

As is obvious, the manual labeling already achieves high accuracy of 97.45% and 94.02%, respectively. Since the gating procedure is supported by the cropped images in the ISP, the biologist is enabled to include morphology information and antibody appearance in the process of gating. The two methods, k-means and majority vote, represent unsupervised clustering and outlier removal. The k-means clustering is not unsupervised in general, since it is initialized using the means of the manually labeled classes, and thus, operator interaction is incorporated. Based on the used two datasets, majority vote is more robust than clustering. When using appropriate feature selection, the results are excellent for dataset 2.

4.2. Comparison to a state-of-the-art quantification method (FCM)

To compare the proposed method to a state-of-the-art cytometry analysis method, we performed an analysis on a neuroblastoma sample (dataset 3, n= 1142) and compared it to FCM results. We used CD44 and GD2 for antibody staining, as in dataset 1. In contrast to the analysis of dataset 1 and 2, the aim was to determine the number of cells positive for a certain antibody. A cell is positive for an antibody if the antibody expression pattern is significantly expressed, in the subjective impression of the operator. Thus, four populations have to be classified (GD2 positive, GD2 negative, CD44 positive and CD44 negative cells) preventing the use of k-means with k=2. Due to the results for dataset 1 and 2, we decided to use the majority vote for automated class label correction.

To label CD44 positive cells, the first and the second principal component were set on the ISP axes and the gating was performed subsequently. A cell was declared to be positive for CD44 (and thus, included in the gate) if the mean intensity of the antibody passed a certain threshold, declared by the operator. In contrast, GD2 was more difficult to analyze. GD2 positive cells appear with a granular surface staining,

Antibody stained	positivity: ratio of population (in %) inital gate	majority vote	FCM ref.
GD2	79.25	82.31	85.8
CD44	18.04	16.81	15.3

Table 3. Results of antibody quantification for GD2 and CD44 on dataset 3.

especially in the region of the cell membrane since GD2 is a cellular surface marker. Hence, cells can be negative for GD2 even if the mean intensity is rather high. To label GD2 positive cells (and thus, GD2 negative cells were outside the gate), we chose the mean of the 30 percent of brightest pixels in the nucleus versus the mean of the 30 percent of brightest pixels on the cell membrane for the ISP axes.
Finally, we used majority vote on the PCA reduced feature space to correct for overlapping populations for both antibodies analyzed. The analysis procedure is presented in Figure 2, the results are displayed in Table 3.

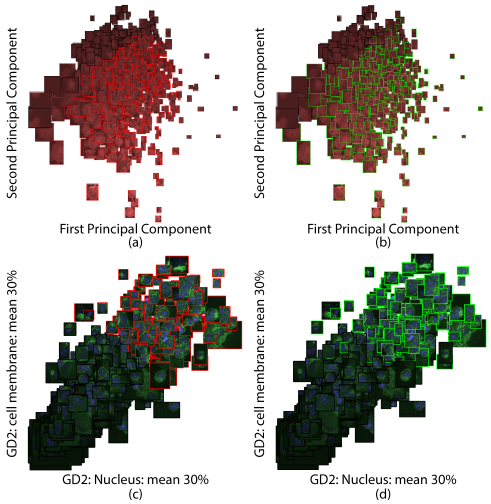


Figure 2. Classification of neuroblastoma cells. (a) Manually labeled CD44 positive cells (red rectangles); only the channel including CD44 is visualized (b) Result of the majority vote to correct the class labels (green rectangles indicate CD44 positive cells) (c) Manually labeled GD2 positive cells (red rectangles); GD2 and nucleus channel are overlaid (d) Result of the majority vote to correct the class labels (green rectangles indicate GD2 positive cells).

As underlined by the results, the analysis procedure is comparable to state-of-the-art FCM analysis with the advantage of directly visualizing the cellular features. Furthermore, the results underline the benefit of incorporating manual input and using majority vote to correct for overlapping cellular populations.

5. Discussion

The ISP method provides the operator with a valuable tool for cellular population quantification by incorporating morphology and antibody expression patterns in the process of analysis. The use of ISPs compensate for the lower discriminative power of FM features when compared to FCM features. Furthermore, using the ability of calculating morphological, gray level histogram and textural features supports in splitting overlapping populations. The comparison of the results of majority vote to those of automated clustering indicates the effectiveness of incorporating operator input for cellular population quantification.

5.1. Number of samples

Comparing k-means and majority vote for class automated class label correction, the use of majority vote seems to be favorable based on the current samples. Nevertheless, we have to incorporate more samples to validate the results on a sound basis, which indeed is a challenge as the generation of groundtruth is a time-consuming task and has to be performed by the biologist experts. Especially, samples including highly overlapping populations, due to a low discriminative power of features, will provide valuable feedback for refining the current approach.

5.2. Texture features used

Aside from the features used in this work, there are other features proposed in the literature that could support the current analysis method but remain unconsidered. Mainly, features based on frequency information, such as fast Fourier transform features, could be used to further improve the accuracy of the automated class label correction. However, due to the curse of dimensionality we are limited in the number of features used. Moreover, the current results based on the investigated features are promising for an application to a similar type of samples.

5.3. Methodological improvements

To perform the automated class label correction, we used a majority vote to decide the final class membership for a single observation. When interpreting the ratio between the number of the votes for a certain class to the number of the members of the ensemble as probabilities for each single observation, one could distinguish three cases: high probability indicating the observation will be part of the partic-

ular class with high certainty, middle probability indicating the observation will be located somewhere close to the decision region in the feature space and low probability indicating the observation will unlikely be part of the particular class. A strategy to further increase the accuracy of automated class label correction would be to determine between the three cases based on training sets and machine learning. For all observations analyzed not passing a lower threshold the class label would be kept. If passing an upper threshold the observations would be assigned to the other class. For all observations having a probability between the lower and the upper threshold, those observations could be subject to an automatic clustering and active learning [17]. By presenting members of each resulting cluster to the operator it could be decided if the observation and thus, the cluster, should be assigned to one or to the other class.

Acknowledgements

The samples and groundtruth used in this work were kindly provided by Sabine Taschner-Mandl, Tamara Weiss, Nelli Frank and Teresa Gerber (Childrens Cancer Research Institute, St. Anna Kinderkrebsforschung, Vienna). This study was supported by an EraSME grant (project TisQuant) of the Austrian Research Promotion Agency (FFG) under the grant no. 844198, the European Union, Marie Curie Industry Academia Partnerships & Pathways (FP7-Marie CuriePEOPLE-2013-IAPP) under the grant no. 610872 and the St. Anna Kinderkrebsforschung.

References

[1] N. S. Barteneva, E. Fasler-Kan, and I. A. Vorobjev. Imaging flow cytometry: Coping with heterogeneity in biological systems. *Journal of Histochemistry & Cytochemistry*, 60(10):723–733, Oct. 2012. 2

[2] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 5

[3] C. E. Brodley and M. A. Friedl. Identifying mislabeled training data. *arXiv preprint arXiv:1106.0219*, 2011. 5

[4] I. Buciu and A. Gacsadi. Gabor wavelet based features for medical image analysis and classification. In *2nd Int. Symp. on Applied Sciences in Biom. and Comm. Technologies*, pages 1–4. IEEE, 2009. 4

[5] R. C. Ecker, R. Rogojanu, M. Streit, K. Oesterreicher, and G. E. Steiner. An improved method for discrimination of cell populations in tissue sections using microscopy-based multicolor tissue cytometry. *Cytometry Part A*, 69(3):119–123, 2006. 2

[6] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6):381–395, 1981. 6

[7] N. Hamilton. Quantification and its applications in fluorescent microscopy imaging. *Traffic*, 10(8):951–961, Aug. 2009. 1

[8] N. A. Hamilton, R. S. Pantelic, K. Hanson, and R. D. Teasdale. Fast automated cell phenotype image classification. *BMC Bioinformatics*, 8(1):110, 2007. 2

[9] N. A. Hamilton, J. T. Wang, M. C. Kerr, and R. D. Teasdale. Statistical and visual differentiation of subcellular imaging. *BMC Bioinformatics*, 10(1):94, 2009. 2

[10] G. Herrera, L. Diaz, A. Martinez-Romero, A. Gomes, E. Villamn, R. C. Callaghan, and J.-E. OConnor. Cytomics: A multiparametric, dynamic approach to cell research. *Toxicology in Vitro*, 21(2):176–182, Mar. 2007. 1

[11] F. Kromp, S. Taschner-Mandl, M. Schwarz, J. Blaha, T. Weiss, P. F. Ambros, and M. Reiter. Semi-automated segmentation of neuroblastoma nuclei using the gradient energy tensor: A user driven approach. In *SPIE Proceedings of ICMV 2014, in Press*, Milano, 2014. 2

[12] S. Liu, P. A. Mundra, and J. C. Rajapakse. Features for cells and nuclei classification. In *Engineering in Medicine and Biology Society, Annual Int. Conf. of the IEEE*, pages 6601–6604. IEEE, 2011. 2

[13] K. Lo, R. R. Brinkman, and R. Gottardo. Automated gating of flow cytometry data via robust model-based clustering. *Cytometry Part A*, 73A(4):321–332, Apr. 2008. 2

[14] R. Narath, T. Lrch, M. Rudas, and P. F. Ambros. Automatic quantification of gene amplification in clinical samples by IQ-FISH. *Cytometry*, 57B(1):15–22, Jan. 2004. 1

[15] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002. 4

[16] N. Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975. 4

[17] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, and H.-J. Zhang. Two-dimensional active learning for image classification. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 8

[18] Y. Saeys, I. Inza, and P. Larranaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, Oct. 2007. 5

[19] C. Vonesch, F. Aguet, J.-L. Vonesch, and M. Unser. The colored revolution of bioimaging. *Signal processing magazine, IEEE*, 23(3):20–31, 2006. 1, 4

20^{th} *Computer Vision Winter Workshop*
Paul Wohlhart, Vincent Lepetit (eds.)
Seggau, Austria, February 9-11, 2015

Sharing local information in scanning-window detection

Jan Pokorný, Jiří Trefný, and Jiří Matas
CTU CMP in Prague, Czech Republic

Abstract. *Object detection is a classic task in computer vision. WaldBoost algorithm is a state-of-the-art method for object detection due its high detection accuracy and real-time speed. However, since the traditional scanning window procedure does not make use of information shared among overlapping windows, there is still a possibility of a significant speed-up by exploiting this property. Zemčík et al. recently proposed to use a second classifier to suppress the neighboring positions with a negligible computational overhead. In this paper we improve upon the work of Zemčík et al. and show that with an improved scanning strategy and predictor selection we outperform it in both geometric accuracy as well as detection rate on the FDDB dataset for face detection, while achieving the same or a higher speed-up.*

1. Introduction

Object detection is a computer vision problem with many applications. Commonly, the applications require not only high accuracy in terms of low false negative and false positive rates but also high processing speed.

The scanning window technique combined with a rejection cascade of classifiers introduced by Viola and Jones [8] represents the state of the art and has been the dominant approach for object detection in recent years. Since its introduction, a large number of follow-up work has appeared in the literature.

In this paper, we focus on the problem of increasing the speed of Viola-Jones type of methods. The WaldBoost [6] algorithm offers a competitive speed-precision trade-off using Wald's quasi-optimal sequential probability test and it achieves high detection rates for various object classes while keeping the ability to process tens of images per second. Recent advances in deep neural networks [3] have influenced state-of-the-art in object recognition significantly, however, fast object detection is still beyond its capabilities.

Recently, Zemčík et al. [9] proposed a method that exploits the fact that information is shared between overlapping scanning windows. The method introduces an auxiliary classifier for suppressing the evaluation at neighboring positions. While a window is being classified with the standard WaldBoost classifier, the response of the suppressing classifier is being computed virtually for free on the same features using only a different look-up table. If the confidence of the suppressing classifier reaches a threshold level, the neighboring position is discarded. However, if the confidence is low, the response of the suppressing classifier is ignored, even though it might contain a valuable information about the neighbor.

Similarly, Dollár et al. [1] use the correlation of pedestrian detector responses in nearby positions to build a sophisticated "crosstalk" cascade which enables neighboring detectors to communicate and achieve major computational gains. The problem we focus on, face detection, differs from pedestrian detection in the average number of evaluated weak classifiers per window – about 3 for face detection, approximately 30 for pedestrian detection – which makes the scheme impractical.

Another feature-centric approach was proposed by Schneidermann [5]. He proposed to pre-compute a set of feature values on a regular grid. The features are available for all the corresponding windows. This resulted in a significant speed-up of the algorithm. However, the reported speed for face detection was about 2 frames per second on 1.8GHz processor, which is not competitive even when the hardware speed-up since the publication of the paper is considered.

In this work we evaluate and improve upon the work of Zemčík et al. [9]. In particular, we:

- propose and test a number of different scanning patterns and predicted neighborhood sets,

as shown in Fig. 3.

- explore the possibility of having a single predictor for multiple positions and thus achieving a significant speed-up.
- propose and evaluate the use of "suppression classifier" as a predictor, i.e. as a weak classifier biasing the original detector.

Evaluating the method on the state-of-the-art FDDB dataset [2], we show experimentally that some of the proposed scanning and neighborhood patterns outperform the original method of Hradiš. The results demonstrate that the method is capable of a 40-50% speed-up without any loss in geometric accuracy and a minimal loss of detection (below 0.5%).

The rest of this paper is structured as follows. The method proposed by Zemčík et al. [9] is reviewed in Section 2. The proposed modifications are described in Section 3. Performance of the method is evaluated in Section 4.

2. Exploiting neighbors for faster scanning window detection in images [9]

Zemčík et al. proposed to learn a classifier for suppression of the evaluation of the detection classifier in the neighborhood of the currently examined window. The detection classifier is a sequential decision strategy based on a majority vote of weak classifier functions $h_t : \chi \to \mathbb{R}$:

$$H_T(\mathbf{x}) = \sum_{t=1}^{T} h_t(\mathbf{x}). \tag{1}$$

The weak classifier usually decides on the basis of a single image feature. Let us denote the *features* as $f : \chi \to \mathbb{N}$. The weak hypotheses are a combination of such features and a *look-up table operation* $l : \mathbb{N} \to \mathbb{R}$

$$h_t(\mathbf{x}) = l_t(f_t(\mathbf{x})). \tag{2}$$

The decision strategy S of a soft cascade is a sequence of decision functions $S = S_1, S_2, \ldots, S_T$, where $S_t : \mathbb{R} \to \sharp, -1$. The $\sharp$ symbol denotes "undecided". The decision functions S_t are evaluated sequentially and the strategy is terminated with negative result when the decision functions outputs -1. The positive result +1 is output if the end of the cascade is reached. Each of functions S_t bases its decision on the comparison of the running sum of the weak hypotheses with a threshold θ_t :

$$S_t(\mathbf{x}) = \begin{cases} \sharp & \text{if } H_t(\mathbf{x}) > \theta_t \\ -1 & \text{if } H_t(\mathbf{x}) \leq \theta_t \end{cases} \tag{3}$$

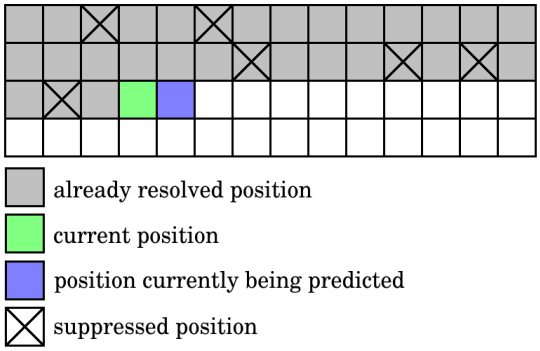


Figure 1: Scanning an image in ordinary line-by-line fashion while using neighborhood suppression [9]

The task of learning the suppressing classifier can be formalized as learning a new soft cascade with a decision strategy S' and hypotheses h'_t, where the weak hypotheses reuse the features f_t from the original classifier, only new lookup-table functions l'_t are learned. The suppression process is visualized in Fig. 1.

2.1. Learning Suppression with WaldBoost

The WaldBoost [6] algorithm was chosen to train the soft cascades. It is relatively simple to implement, it guarantees the classifier in each stage to be quasi-optimal on the training data and the produced classifier is very fast.

Given a weak learner algorithm, training data $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}, \mathbf{x} \in \chi, y \in -1, +1$ and a target miss rate α, the WaldBoost finds such decision strategy that its miss rate α_S is lower than α and the average evaluation time $\bar{T}_S = E(\arg\min_i(S_i \neq \sharp))$ is minimal: $S^* = \arg\min_S \bar{T}_S$, s.t.$\alpha_S < \alpha$.

To create such strategy, WaldBoost combines AdaBoost [4] and Wald's *sequential probability ratio test*. First, AdaBoost selects the most discriminative weak hypothesis h_t. The threshold θ_t is then chosen such that as many negative training samples are rejected as possible while asserting that the likelihood ration estimated on training data

$$\hat{R}_t = \frac{p(H_t(\mathbf{x})|y = -1)}{p(H_t(\mathbf{x})|y = +1)} \tag{4}$$

satisfies $\hat{R}_t \geq \frac{1}{\alpha}$. In the formulation the early termination is not considered for the positive class. Only a

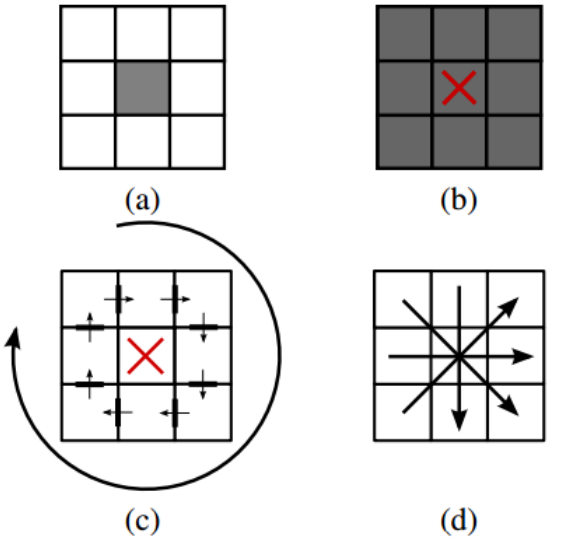


Figure 2: Extended set of LBPs [7]: (a) conventional LBP thresholded by center pixel value; (b) 8-bit coded modified LBP (mLBP) thresholded by pixels mean value; (c) transition coded LBP (tLBP); (d) direction coded LBP

tiny fraction of the tested windows belong to the positive class and the early termination does not have a significant impact on the running time.

3. The proposed method

The proposed method generalizes the method of Hradiš [9]in two ways. First, it breaks away from the top-to-bottom, left-to-right scanning pattern and uses a more efficient strategy instead. Second, it does not use the information for suppression only, but contributes as a weak classifier. The prediction for neighboring positions is assessed like zero-length boosted detector and stops evaluation if the confidence is high enough, otherwise it is reused as the bias for the detection classifier.

Similarly to [9], the predictor reuses the features computed with the original classifier. We use WaldBoost [6] algorithm with the extended set of Local Binary Patterns features [7] (see Fig. 2) for the classification and AdaBoost for prediction. For the LBP features used, the second look-up for the prediction at neighboring positions is about 10 times faster than the feature calculation.

The steps of our method are the following:

3.1. Step 1: 2d partitioning of image

Divide set D of all windows positions in image into 2 disjoint sets C and N such that the Minkowski sum $C \oplus N$ covers the original domain, i.e. $D = C \oplus N$. C is set of all center positions, that will be further used for predicting the responses of their corresponding neighbors. N is a set of all neighboring positions, that will get the prediction from their corresponding center positions. See examples of the neighborhood types in Fig. 3. Each element $\mathbf{x} \in C$ has its corresponding set of neighbors $\mathbf{x}' \in \mathcal{N}(\mathbf{x})$.

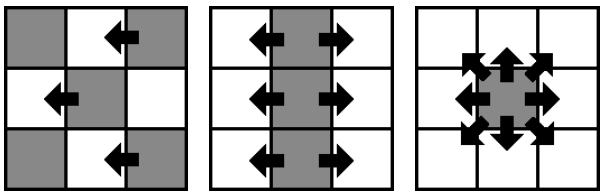


Figure 3: Types of predictors: PL (prediction left), PLR (prediction left & right), P8 (prediction for all 8 surrounding positions). Gray color corresponds to center windows C, white to neighboring windows N.

3.2. Step 2: Windows classification

1. For each $\mathbf{x} \in C$ evaluate $H^f(\mathbf{x})$ and $H^p(\mathbf{x}')$.
2. For each $\mathbf{x}' \in N$ evaluate

$$H_t'^f(\mathbf{x}') = H_t^f(\mathbf{x}') + \min(H^p(\mathbf{x}'), 0), \quad (5)$$

where H^f is the original classifier, H^p is the predictor, $t = 0, \ldots T$ and $H_0^f(\mathbf{x}') = 0$. The algorithm for learning the predictor is described in Algorithm 1.

4. Experiments

We evaluated the performance of our method on FDDB dataset [2]. We trained following predictors: PL-S2:1, PLR-S2:1, PLR-S3:1, P8-S2:2, P8-S3:3, where Sx:y is a step size corresponding to the scanning pattern (see Fig. 4) and Pn is type of neighborhood (PL: single predictor for window on the left, PLR: single predictor for windows on the left and right, P8: single predictor for all 8 surrounding windows). Only the results of best performing predictors are shown in the figures. For neighbors that have a prediction from multiple center windows the predictions value is computed as a mean value of these responses.

We also included the method [Zemcik] in the evaluation, which is a slightly modified version of [9]: the scanning goes from right to left, predictor PL is used to predict the response on a single window and

Algorithm 1 Training predictor H^p

Input:

- original soft cascade $H_T(\mathbf{x}) = \sum_{t=1}^{T} h_t(\mathbf{x})$, its termination thresholds $\theta^{(t)}$ and its features f_t
- training set $\{(\mathbf{x}_1, y_1) \ldots, (\mathbf{x}_m, y_m)\}, \mathbf{x} \in \chi, y \in \{-1, +1\}$ (+1 stands for *at least one positive sample in the neighborhood of* $\mathbf{x}$, -1 for *no positive sample in the neighborhood of* $\mathbf{x}$)

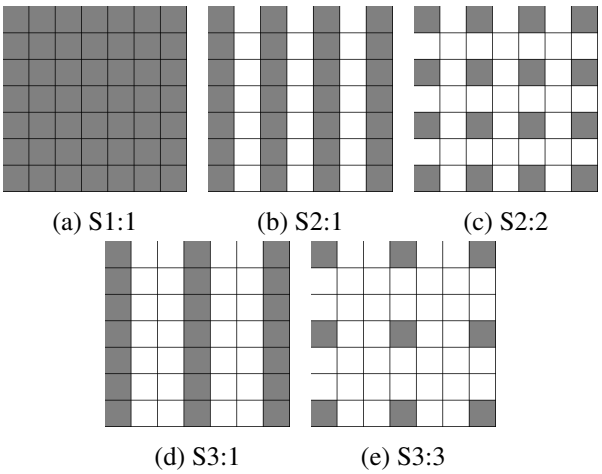
Output:

- look-up table functions l_t^p of the new predictor H^p

Initialize: sample weight distribution $D_1(i) = \frac{1}{m}$
for t = 1, ..., T **do**

1. estimate new l_t^p such that its
$$c_t^{(j)} = -\frac{1}{2} \ln \left(\frac{P_{i \sim D}(f_t(\mathbf{x}_i)=j | y_j=+1)}{P_{i \sim D}(f_t(\mathbf{x}_i)=j | y_j=-1)} \right)$$
2. add l_t^p to predictor
$$H_t^p(\mathbf{x}) = \sum_{r=1}^{t} l_r^p(f_r(\mathbf{x}))$$
3. remove from the training set samples for which $H_t(\mathbf{x}) \leq \theta^{(t)}$
4. update the sample weight distribution
$$D_{t+1}(i) \propto \exp(-y_i H_t^p(\mathbf{x}_i))$$

end for



(a) S1:1 (b) S2:1 (c) S2:2

(d) S3:1 (e) S3:3

Figure 4: Example of scanning patterns. Gray color corresponds to the center positions C, white to the neighboring windows N.

only a single threshold θ_0 for the final predictor response is used. We argue this does not differ significantly from the original version.

We evaluated following metrics: recognition/speed, accuracy/speed. The "detection only" curve corresponds to the reference detector, where the speed-up is achieved by increasing the step size. S1:1 corresponds to an original step size of 2 pixels, S1.5:1.5 corresponds to step size of 3 pixels in both directions.

As one can see in Fig. 5, the best recognition/speed ratio was achieved by P8-S2:2. With this predictor the relative speed 0.73 and 0.5 was achieved with losing 0.2% and 0.5% of recognition respectively (points P1 and P2 in Fig. 5).

Fig. 6 shows that using the predictor response as a starting point of classification for the positions that are not suppressed does not have a dramatic influence on the result. The best accuracy/speed ratio was achieved by PLR-S2:1. Values on Y axis are computed as an average of recognition on ROC curve between 10 and 1000 false positives with logarithmic scale used for false positive axis.

The best result for accuracy/speed ratio was achieved by PLR-S2:1 (see Fig. 7). The results of PLR-S2:1 and [Zemcik] are quite surprising, since on would expect the accuracy to decrease with a decreasing number of evaluated weak classifiers. This could be caused by the fact, that the object "lost" with the decrease in recognition were also the main source of the geometric inaccuracy.

The experiments with a real algorithm speed showed that having one more look-up table increases the processing time to 1.1 of the original value, including 5 more look-up tables increases the processing time to 1.2.

5. Conclusion

The scanning strategy and the selection of predictors is a significant factor in quality of the prediction algorithm.

The final detector using the best performing of the predictors was twice as fast as the detector without prediction while losing only 0.5% of the detection rate. The result outperforms the reference method [9] in both geometric accuracy as well as detection performance on the FDDB dataset, while achieving the same or a higher speed-up.

The topics of future work are: evaluation of the predictor on different object classes (pedestrians, cars) and on multi-view (frontal, profile, half-profile)

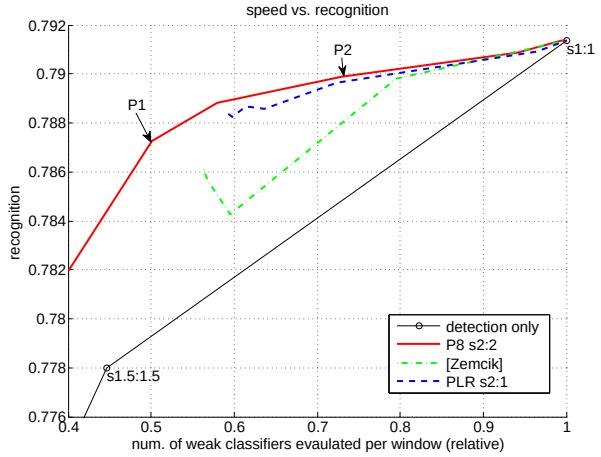


Figure 5: Speed vs. recognition - the best two predictors compared to [Zemcik]. Points were obtained by increasing the θ_0 value from -Inf to 0.

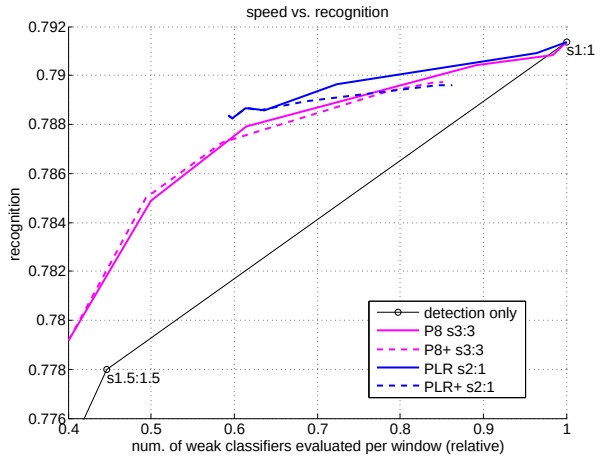


Figure 6: Speed vs. recognition - comparison of not using (P8, PLR) and using (P8$_{pred}$, PLR$_{pred}$) the prediction value as a starting point of the original classifier. Points were obtained by increasing the θ_0 value from -Inf to 0.

face detector.

References

[1] P. Dollár, R. Appel, and W. Kienzle. Crosstalk cascades for frame-rate pedestrian detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 1

[2] V. Jain and E. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010. 2, 3

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012. 1

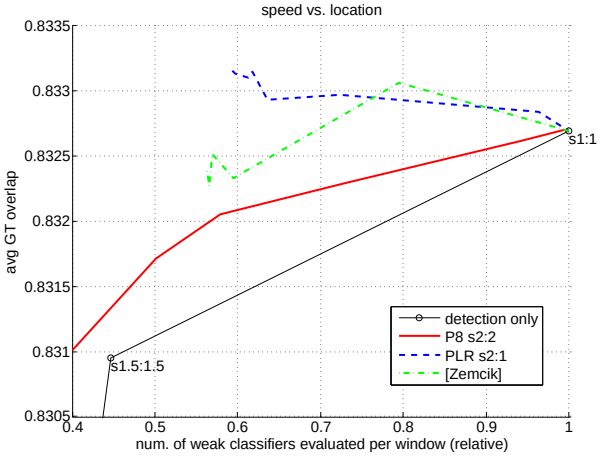


Figure 7: Speed vs. location - detected boxes accuracy. The best two predictors compared to [Zemcik]. Points were obtained by increasing the θ_0 value from -Inf to 0.

[4] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Mach. Learn.*, 37(3):297–336, 1999. 2

[5] H. Schneiderman. Feature-centric evaluation for efficient cascaded object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2004. 1

[6] J. Sochman and J. Matas. Waldboost " learning for time constrained sequential detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '05, pages 150–156, Washington, DC, USA, 2005. IEEE Computer Society. 1, 2, 3

[7] J. Trefny and J. Matas. Extended set of local binary patterns for rapid object detection. In *Computer Vision Winter Workshop, Czech Republic*, 2010. 3

[8] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001. 1

[9] P. Zemčík, M. Hradiš, and A. Herout. Exploiting neighbors for faster scanning window detection in images. In *Advanced Concepts for Intelligent Vision Systems*, LNCS 6475, page 12. Springer Verlag, 2010. 1, 2, 3, 4

20^{th} *Computer Vision Winter Workshop*
Paul Wohlhart, Vincent Lepetit (eds.)
Seggau, Austria, February 9-11, 2015

Multi-view Facial Expressions Recognition using Local Linear Regression of Sparse Codes

Mahdi Jampour Thomas Mauthner Horst Bischof
Institute for Computer Graphics and Vision
Graz University of Technology
{jampour, mauthner, bischof}@icg.tugraz.at

Abstract. *We introduce a linear regression-based projection for multi-view facial expressions recognition (MFER) based on sparse features. While facial expression recognition (FER) approaches have become popular in frontal or near to frontal views, few papers demonstrate their results on arbitrary views of facial expressions. Our model relies on a new method for multi-view facial expression recognition, where we encode appearance-based facial features using sparse codes and learn projections from non-frontal to frontal views using linear regression projection. We then reconstruct facial features from the projected sparse codes using a common global dictionary. Finally, the reconstructed features are used for facial expression recognition. Our regression of sparse codes approach outperforms the state-of-the-art results on both protocols of BU3DFE dataset.*

1. Introduction

Facial expression recognition (FER) has attracted significant interest in the computer vision community because of its applications in human computer interaction, education, robotics, games, medicine and psychology [30]. There are six basic classes in facial expression recognition : anger (AN), disgust (DI), fear (FE), Happiness (HA), sadness (SA) and surprise (SU). Most of the existing approaches work on frontal or near to frontal views [18, 13, 28], whereas in real-world applications, a frontal view is an unrealistic assumption and limits the applicability. For this reason, non-frontal analysis is now one of the active challenges related to facial expression recognition, which needs not only an effective recognition approach, but also a method for compensating missing information (i.e. non-frontal counterpart) [24]. This is a challenging problem because some of the facial features which are necessary for recognition are not available or not completely available due to the face orientation. For example, eyebrows, which are very important for recognizing facial expression, may not be visible in a non-frontal face. On the other hand, we have pairwise sets of non-frontal and frontal views that provide the ability of learning transformations between them to benefit from similar features. For these reasons, we would like to estimate invisible facial features of a face from visible frontal views. To this end, we employ linear regression which can provide an elegant approximate transformation on learning collections (e.g. non-frontal to frontal facial expressions). The idea behind using linear regression is inspired from non-frontal face recognition [16, 3]. In contrast, we apply sparse features instead raw image data, which shows to be more efficient and less sensible to viewpoint changes than raw data. In this work, we aim to address the gap of missed facial features of non-frontal views using linear regression of non-frontal sparse coded features to frontal codes. The motivation behind employing sparse coded features is the robustness on viewpoint variations. Moreover, it has been shown that sparse representation is one of the successful feature-based representation models for facial expression recognition [21]. Therefore, we first create a dictionary using raw training data and then transform raw features to sparse codes using the dictionary. Second afterward, we estimate linear regressions to project non-frontal sparse features to frontal ones and finally we use reconstructed features for expression recognition. In other words, our linear regression-based transformation can approximate a frontal view given a non-frontal view using pairwise collections of frontal and non-frontal training data. Moreover, to show the efficiency of our approach, an

extensive investigation is provided on the BU3DFE dataset. BU3DFE dataset is a popular facial expression dataset which is introduced in section 4.1. We show that our approach outperforms the state-of-the-art results on BU3DFE. Contribution: In this paper we introduce a novel approach for multi-view facial expression recognition. We propose to use a sparse coding representation for facial expression recognition which is efficient and stable with viewpoint variation. We introduce also linear regression of such sparse features to approximate projections in local feature space.

2. Related works

There are significant works on facial expression recognition with many interesting applications in human computer interaction, psychology, games, children education, etc. which could be broadly categorized into the three general categories: 1) Geometric-based models [17, 19, 7, 2], 2) Appearance-based models [32, 6, 14, 9, 21], and 3) hybrid methods which use both texture and shape information [10, 12]. Typically, geometric-based approaches are methods that employ shape information (e.g. facial action units) whereas appearance-based approaches use only texture information. Multi-view facial expression recognition has been attracting increasing attention among the face researcher as well as facial expression recognition. For instance, [17] proposed geometric-based methods that uses 2D facial points to map from non-frontal to frontal view. [7] proposed a computation of 2D facial feature displacement. They normalized extracted distances to zero mean and unit variance to make much discriminative classification. Other approaches rely on the appearance-based model. For instance, a discriminant analysis theory (BDA/GMM) proposed by [32] which optimizes upper bound of the Bayes error derived by Gaussian mixture model. Hesse et al. [6] evaluated different descriptors such as SIFT, LBP and DCT extracted around of facial landmarks and classify then using ensemble SVM. The latter approach proposed by [15] which is a two-step multi-view facial expression recognition model that estimate the pose orientation directly from the image in the first step and then, a pose-dependent expression classifier recognizes facial expressions. Beside of appearance-based multi-view facial expression recognition models, transmutable approaches have been also proposed by [9] and [21], where [9] proposed a multi-view discriminative framework using multi-set canonical correlation analysis (MCCA) and the multi-view model theorem for facial expression recognition with arbitrary views. Their method respects the intrinsic and discriminant structure of samples. They obtained discriminative information from facial expression images based on the discriminative neighbor preserving embedding (DNPE). [21] improved an existing facial expression recognition model using generic sparse coding feature. They applied sparse coding features of dense SIFT on the facial images in a three level spatial pyramid and then encode the local features into sparse codes to make the possibility of multi-view processing.

Sparse coding has been previously used for face recognition [31], facial expression recognition [21] and other applications where it has been shown that sparse coding is a successful encoding technique in [25]. Zhang [31] explained that why sparsity could improve discrimination and how regression could be used to solve a classification problem. [25] proposed an efficient sparse-based model and showed that regression transformation can improve the time complexity in both global and anchored neighborhood regression which are much faster than other related works. An important yet relatively unexplored approach is to employ pose specific linear regression which is challenging due to the partial linear regression. A regression-based approach proposed by ([17]) which employed global transformation however it is not as well as local linear regression of sparse features (LLRSF in section 3.4) on accuracy. Similarly, the approach that used sparse coding feature [21] did not profits the regression transformation. Therefore, to address the above problems, this paper proposes to integrate them in a sequence.

3. Multi-view Facial Expression Recognition

In this section, we describe the proposed approach to multi-view facial expression recognition. Our approach consists of three modules:

a) Feature extraction: We apply a concatenation of HOG and LBP features which are popular in face analysis. HOG [5] is a successful gradient-based descriptor used in different purpose of object detection and recognition that is stable on illumination variation. Moreover, it is a fast descriptor in comparison to the SIFT and LDP (Local Directional Pattern) [11] due to the simple computation. On the other hand, LBP is a common texture-based descriptor which de-

scribes image pixels based on the neighborhood intensities. It has been shown that a concatenation of HOG and LBP can improve human detection performance by [27]. In our experiments, the extracted features are considered as feature vectors for every facial image in every viewpoint without any concern about head pose or expressions. The basic idea to concatenate these two feature descriptors is synthesis of vectors where $I_i = [H_i; L_i]$ is a feature vector with size $(q \times 1)$ concatenated by HOG (H_i) and LBP (L_i) feature vectors, related to the i^{th} face image. The cell size considered for both HOG and LBP is 25 pixels, therefore, the overall dimensionality is 5480 in total where first 2232 dimensions are computed by HOG and the rest 3248 dimensions via LBP.

b) Projections: Linear regression projection is performed to estimate projections based on the Eq. 2, with global projection for global model (GLR) or several projections for local model (LLR). Details are explained within Section 3.1 and 3.3 respectively. In addition, all extracted features are transformed into the sparse representation as described in Section 3.2 and 3.4. The main motivation to use sparse representation is its robustness on the viewpoint variation. In the following, projections of non-frontal to frontal viewpoints are generated using linear regression on the sparse codes. We show that sparse coded facial expression features are much more stable for estimating projections by linear regression than our raw features.

c) Classification: Both testing and training parts of all non-frontal viewpoints are projected to the frontal feature space and a global classifier is used for expression recognition. Linear SVM [4] is applied as our basic classifier to find best facial expression estimation. A strategy of using nearest neighbors is also provided to improve our overall result which is described in Section 3.5.

3.1. Global Linear Regression

Let X be a set of aligned vectorized facial features which have size $(q \times 1)$. X_{θ_i} is a subset of facial features in X from viewing angle θ_i, where $X_{\theta_i} = [I_1^{\theta_i}, I_2^{\theta_i}, \ldots, I_N^{\theta_i}]$ is a matrix of size $(q \times N)$, and refers to the N vectorized facial features denoted by $I_k^{\theta_i} \in \mathbf{R}^{(q \times 1)}$. Note that I_k^0 and $I_k^{\theta_i}$ are vectorized features of the k^{th} facial expression image of the training data from the same person in different poses. Based on this, we define pairwise sets of training data, X_0 and X_{θ_i}, where the former is a set of frontal views and latter is correspondence non-frontal view with angle θ_i. The number of samples in both sets of $X_0, X_{\theta_i}, i = 1, 2, \ldots, M$ is equal. Moreover, $X_\theta = [X_0, X_{\theta_1}, \ldots, X_{\theta_M}]$ contains M sets of non-frontal views and one set of frontal view. Therefore, we need the same number of samples for both sets of frontal and non-frontal views to train a projection between them using linear regression. To this end, we define X_0^M which is the frontal set repeated M+1 times. So, X_0^M and X_θ have same number of samples and we can estimate the projection between them. Mathematically, this can be formulated as:

$$\underset{P}{\operatorname{argmin}} \left\| X_0^M - P X_\theta \right\| \tag{1}$$

Where the global linear projection P can be estimated by Eq. 2, which is the closed form solution for Eq. 1.

$$P = X_0^M (X_\theta^T X_\theta)^{-1} X_\theta^T \tag{2}$$

$$\hat{X}_\theta = P X_\theta \tag{3}$$

Therefore, Eq. 3 is the global linear regression which approximates frontal features $\hat{X}_\theta$ from non-frontal. The overall structure of our globally linear regression of sparse features (GLRSF) is introduced in the following section.

3.2. Global Linear Regression of Sparse Features (GLRSF)

Embedding the feature information within a global code book aims for regularizing the data and therefore being more robust concerning outliers. We are interested in finding a reconstructive dictionary given the training features X by minimizing:

$$\|X - DS\|_2^2 \quad s.t. \, \|s_i\|_0 \leq \Gamma \tag{4}$$

where $D \in \mathbf{R}^{(q \times s)}$ is the dictionary, each column representing a code book vector, and $S \in \mathbf{R}^{(s \times N)}$ the matrix of encoding coefficients. Γ is the sparsity constraint factor, defining the maximum number of non-zero coefficients per sample. We apply K-SVD [1] as dictionary learning algorithm and orthogonal matching pursuit (OMP) [26] as an efficient way for solving the coding of new test samples, given a fixed dictionary. Similarly, S_0^M is frontal sparse coded set repeated M+1 times and $S_\theta = [S_0, S_{\theta_1}, \ldots, S_{\theta_M}]$ is a global collection of one frontal and M sets of sparse features of non-frontal facial expressions where all sets have the same number of samples. Eq. 2 and 3 could be rewritten for sparse representation as:

$$P = S_0^M (S_\theta^T S_\theta)^{-1} S_\theta^T \tag{5}$$

$$\hat{S}_\theta = P S_\theta \tag{6}$$

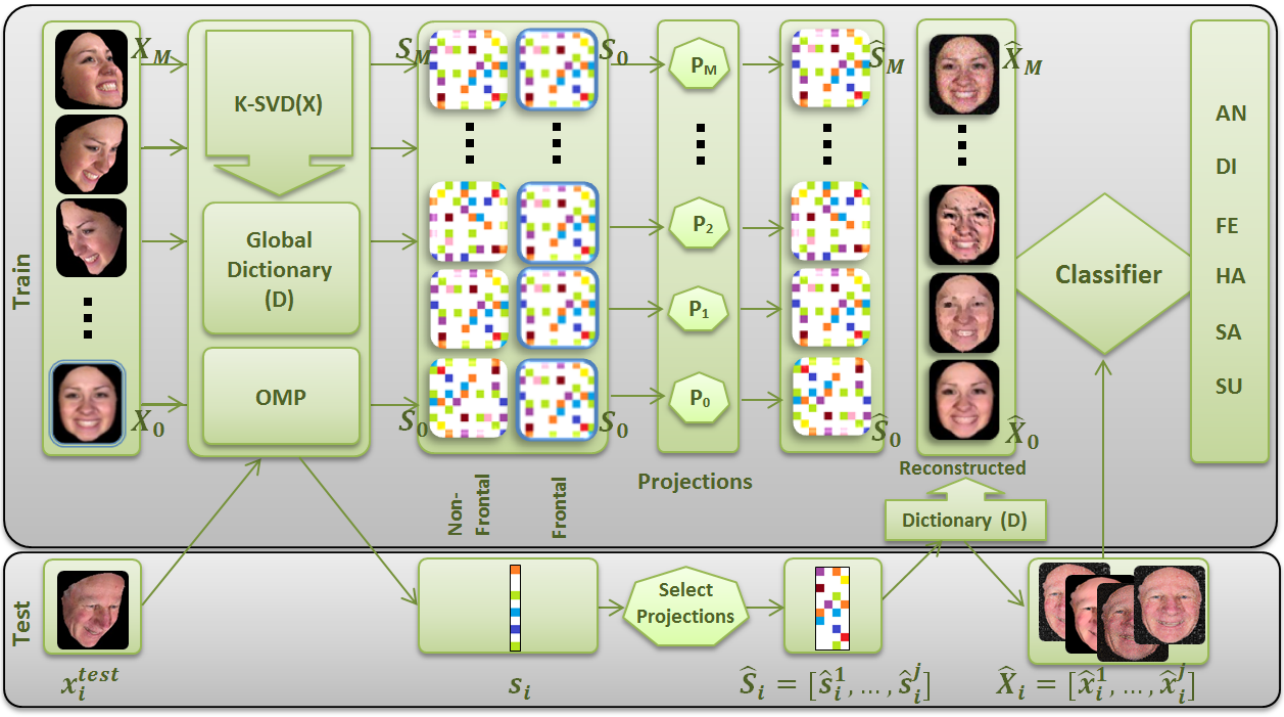


Figure 1: The overall structure of Local Linear Regression of Sparse Features (LLRSF). Train: A global dictionary is trained with K-SVD and facial features are encoded. Projections are estimated using local collections of frontal and non-frontal sparse features, and finally, features are reconstructed using the global dictionary. Test: Input sample is encoded via OMP and encoded vector projected to frontal using appropriate projection. Projected vector is then reconstructed using dictionary and finally it is classified for final expression recognition. Training step benefits from j nearest neighbors to improve our learning model as described in Section 3.5.

$\hat{S}_\theta$ defines the projected sparse codes, and the approximated features of the projected views can be reconstructed using the global dictionary D with:

$$\hat{X}_\theta = D\hat{S}_\theta. \tag{7}$$

To improve our projections, we introduce the locality idea as a local linear regression approach in the following section.

3.3. Local Linear Regression

As mentioned before, a huge number of data with a lot of different properties affect on the mapping function in the global model due to the different viewpoints, expressions, gender, age, skin color, etc. Intuitively, the projection error could be decreased when we increase the number of projections reasonably; this means, splitting data into several meaningful parts and making correspondent projections leads to reduction of the overall projection error compared to using one global projection. Therefore, we used a supervised learning classification to split data using logistic regression SVM where we learn our model with training data based on the viewpoints and classify data into M smaller subsets. Subsequently, linear regressions between specific non-frontal sets X_{θ_i} and the frontal set X_0 estimated for all subsets by:

$$P_i = X_0(X_{\theta_i}^T X_{\theta_i})^{-1} X_{\theta_i}^T \quad i = 1, 2, \ldots, M \tag{8}$$

$$\hat{X}_{\theta_i} = P_i X_{\theta_i} \tag{9}$$

Where $\hat{X}_{\theta_i}$ refers to approximation of frontal features by i^{th} linear regression. Therefore, local linear regression workflow is summarized as:
Step 1: Classifying facial features to the M subsets.
Step 2: Approximating linear regression from non-frontal to frontal subset.
Step 3: Estimating projected facial features by approximated projections: $\hat{X}_{\theta_i} = P_i X_{\theta_i}$
Step 4: Train a global classifier using projected features $\hat{X} = [\hat{X}_0, \hat{X}_{\theta_1}, \hat{X}_{\theta_2}, \ldots, \hat{X}_{\theta_M}]$.
The above workflow is the overall structure for our local linear regression (LLR). Next, we describe how sparse coding could be beneficial for our approach.

3.4. Local Linear Regression of Sparse Features (LLRSF)

GLRSF is an efficient approach for MFER which is also almost stable with outliers but as it uses basic features, it is expensive in terms of memory usage due to the large feature vectors. Therefore, sparse representation is a successful alternative that could help us to improve our solution. We are interested in finding a reconstructive dictionary given the training features X similar to the Eq. 4 and again apply K-SVD [1] and OMP [26] to solve the coding of new test samples, given a fixed dictionary. Similar to LLR, we define M local projections which approximate linear regression for each viewpoint; thus let S_0 be a set of sparse features of frontal facial expressions and $S_{\theta_1}, S_{\theta_2}, \ldots, S_{\theta_M}$ are M sets of sparse features of non-frontal facial expressions where all sets have the same number of samples and provided by OMP. Eq. 8 and 9 could be rewritten for sparse representation as:

$$P_i = S_0(S_{\theta_i}^T S_{\theta_i})^{-1} S_{\theta_i}^T \quad i = 1, 2, \ldots, M \tag{10}$$

$$\hat{S}_{\theta_i} = P_i S_{\theta_i} \tag{11}$$

where P_i is i^{th} projection which has been estimated using correspondent sparse features. $\hat{S}_{\theta_i}$ defines the projected sparse codes, and the approximated features of the projected frontal view can be reconstructed using the global dictionary D with:

$$\hat{X}_{\theta_i} = D\hat{S_{\theta_i}}. \tag{12}$$

The overall structure of our local linear regression of sparse features (LLRSF) is illustrated in Figure 1.

3.5. Soft Learning using Nearest Neighbors

Local linear regression (LLR) or in general, analyzing with subsets of data is almost an efficient solution if we perform meaningful constraint S regarding to splitting data. Nevertheless, supervised learning is sensitive to the number of training samples, therefore, while splitting data into the small collections is useful for regression approximation, it has disadvantage on the supervised classification. To this end, we propose an idea to make a contribution using each cluster neighborhood. In other words, while we classify all data into the M subsets (clusters), there are M cluster centers that they are basically useful to present as a similarity measurement; which means we compute and exploit N-nearest neighbors for each cluster based on the cluster centers similarity. Neighbors are the best candidates for contributing as training data in our work because while we assumed that clusters are splitted based on the viewpoints, there are only small changes within sequences of head poses. Therefore, we profit from the neighbors to train each specific subset. The results in the following show that it can improve our overall rate.

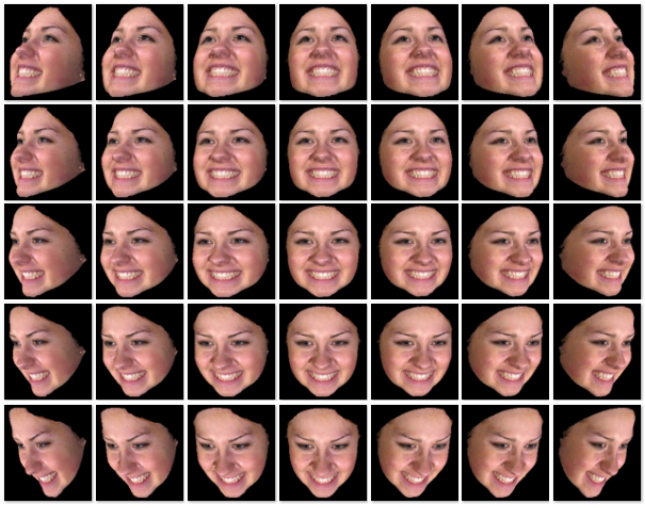


Figure 2: Multi-view rendered faces of a subject from BU3DFE-P1 (35 viewpoints)

Figure 3: Multi-view rendered faces of a subject from BU3DFE-P2 (5 viewpoints)

4. Experimental Results

In order to demonstrate the performance of our model we evaluate on the BU3DFE which is most the popular dataset for multi-view facial expression recognition. We follow on the standard evaluation scheme and apply a 5-fold cross validation over the highest level of expression intensity. The details are given in the following.

4.1. BU3DFE dataset

BU3DFE is a publicly available dataset containing 3D scanned faces of 100 subjects with six basic expressions. More details can be found in [29]. We rendered multiple views from the 3D faces in seven pan angles (0°, ±15°, ±30°, ±45°) and five tilt angles (0°, ±15°, ±30°) which means 35 view-

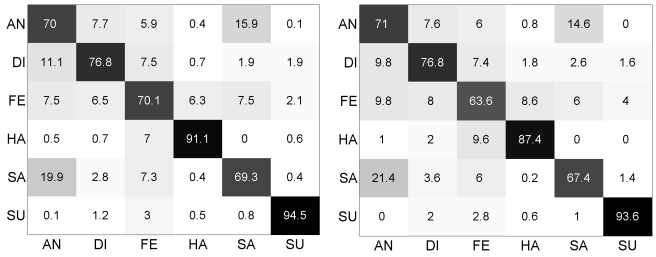
Table 1: Multi-view facial expression recognition comparison between proposed approaches

Method	Dataset	Accuracy
SF	BU3DFE-P1	68.14
GLRSF	BU3DFE-P1	69.87
LLRSF	BU3DFE-P1	74.42
Soft-LLRSF	BU3DFE-P1	**78.64**
SF	BU3DFE-P2	70.33
GLRSF	BU3DFE-P2	75.10
LLRSF	BU3DFE-P2	75.07
Soft-LLRSF	BU3DFE-P2	**76.64**

points to compare our results with the related works [21, 22, 23, 20, 32], shown in Figure 2. In addition we generated views for 0°, 30°, 45°, 60° and 90° which means 5 viewpoints as second protocol to compare our model with papers that applied this protocol [9, 8], as shown in Figure 3. Therefore, as there are 6 expressions for 100 subjects over the highest level of expression intensity in 35 viewpoints, we have 21000 samples in the first protocol and 3000 samples in the second protocol of BU3DFE.

4.2. Evaluation of proposed approaches

We proposed two regression based methods namely Global Linear Regression of Sparse Features (GLRSF) introduced in Section 3.2 and Local Linear Regression of Sparse Features (LLRSF) proposed in Section 3.4. Another reasonable comparison is provided by SF, defining the baseline results of classifying direct on the sparse features, without neither global nor local projection. This highlights the impact of our original idea to use linear regression for projecting non-frontal to frontal views. Parameters like dictionary size and sparsity in K-SVD and number of nearest neighbors in soft learning are evaluated where the best result is achieved by dictionary size of 200 with sparsity 150 and N=4 for nearest neighbors. Moreover, the local linear regression makes projections much more accurate than global regression because local distributions are more compact, smaller and almost easier than global feature space, it can bee seen in the results where Table 1 shows that LLRSF has best overall recognition rate among of SF, GLRSF and LLRSF. It has also found that our local regression projection approach using sparse features has about 6% improvement compared to non-projection baseline SF which shows the importance of proposed idea. Moreover, Soft-LLRSF which is



(a) Protocol1=78.64% (b) Protocol2=76.64%

Figure 4: The confusion matrices of Soft-LLRSF method for two protocols of BU3DFE (a),(b)

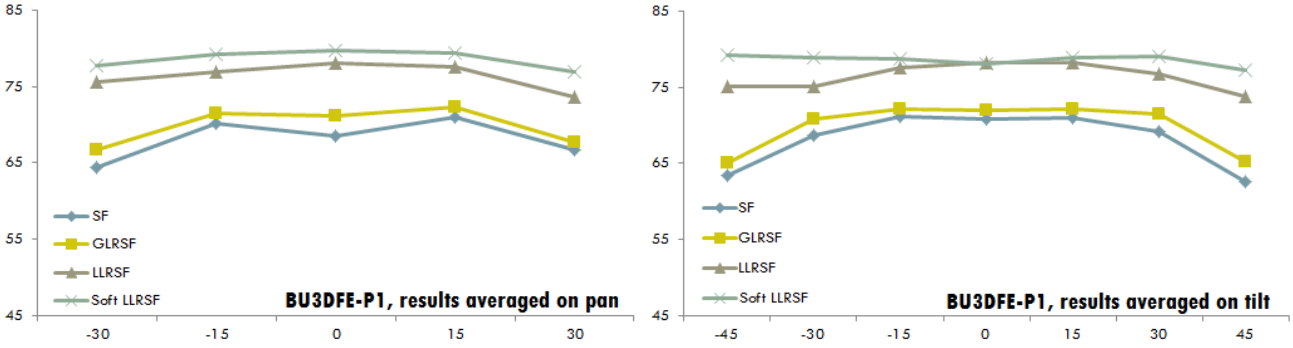
extended form of LLRSF to compensate low number of training samples using N nearest neighbors successfully outperforms all methods in both protocols.

4.3. Results on BU3DFE-P1 (35 viewpoints)

In this section, we propose the performance of our approach on the first protocol of BU3DFE. The confusion matrics between expressions are presented in Figure 4 where the largest confusion is occurred between AN with SA and AN with DI. Our overall accuracy rate is 78.64% when we used 5-fold cross-validation, averaged across all subjects, expressions, poses on highest intensity level of expression on BU3DFE-P1. Performing comparison of our approaches over the variations in pan and tilt, illustrated in Figure 5, note that the results in the Figure 5(a) is averaged across corresponding pan and the Figure 5(b) is averaged across corresponding tilt angels. As can be seen, our regression models (LLRSF and Soft-LLRSF) are obviously better than other methods. This is our expectation that local linear regression outperforms global regression because of the projections accuracy.

4.4. Results on BU3DFE-P2 (5 viewpoints)

Some related works evaluated their results on the protocol 2; we have also applied our model by this protocol. First, we show the performance of our approach based on the viewpoints and expressions where it is demonstrated in Table 1. As can be seen, again Soft-LLRSF outperforms other methods however there is no improvement on LLRSF and GLRSF due to the small problem space (5 viewpoints) whereas it is clearly better than SF (baseline) which shows again proposed regression model over sparse features improves overall recognition rate. Moreover, with attention to the confusion matrix provided by Figure 4 (b) we can find that the



(a) (b)

Figure 5: Proposed methods (SF, GLRSF, LLRSF and Soft-LLRSF) performance (a) over pan (b) and tilt

Table 2: Comparison of proposed model with the state-of-the-art

Method	Dataset	Accuracy
Zheng et al. [32]	BU3DFE-P1	68.20
Tang et al. [20]	BU3DFE-P1	75.30
Tariq et al. [21]	BU3DFE-P1	76.10
Tariq et al. [22]	BU3DFE-P1	76.34
Tariq et al. [23]	BU3DFE-P1	76.60
Soft-LLRSF	BU3DFE-P1	78.64
Huang et al. [9]	BU3DFE-P2	72.47
Hu et al. [8]	BU3DFE-P2	74.46
Soft-LLRSF	BU3DFE-P2	76.64

largest confusion is again between AN and DI which shows that the similarities of these two expressions is usually more than other expressions. The overall facial expression recognition rate in this protocol is 75.07% for LLRSF although using Soft-LLRSF it is 76.64% when we used 5-fold cross-validation, averaged across all subjects, expressions, poses and highest intensity level of expression on BU3DFE. In the following, we compare our approach with the state-of-the-art.

4.5. Comparison with the state-of-the-art

In this section, we provide a comparison of our regression-based approach (Soft-LLRSF) with the state-of-the-art. Table 2 depicts that our proposed approach outperforms the state-of-the-art in both protocols of BU3DFE. A similar sparse coding approach proposed by [21] which achieved 76.10% accuracy on the same dataset whereas our model reasonably outperforms it with 78.64% due to employing local regression projection of sparse features.

5. Conclusion

In this paper, we introduced linear regression projection of sparse features for multi-view facial expression recognition where all facial features first encoded to the sparse codes then they projected to the frontal features and finally facial features reconstructed from projected sparse features. We proposed two methods of global linear regression of sparse features (GLRSF) and local linear regression of sparse features (LLRSF) to solve the problem of multi-view facial expression recognition. Our methods are capable to compensate the facial features of missing parts of the faces. In both methods, the features estimation of non-visible parts of the faces is estimated using regression projection. We have shown that the proposed local regression based model for multi-view facial expression recognition outperforms not only baseline SF but also the stat-of-the-art approaches on both protocols of BU3DFE. Another advantage of our approach is that it does not need landmark detection, therefore, it is more suitable for practical applications. We start from an extremely low baseline (SF, GLRSF) compared to related work. Therefore, examination of alternative facial expression features and investigation of non-linear projections for approximation of frontal-views would be the possible directions for future works.

References

[1] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, 2006. 3, 5

[2] B. B. Amor, H. Drira, S. Berretti, M. Daoudi, and A. Srivastava. 4d facial expression recognition by

learning geometric deformations. *IEEE Transactions on Cybernetics*, 44:1–16, 2014. 2

[3] X. Chai, S. Shan, X. Chen, and W. Gao. Locally linear regression for pose-invariant face recognition. *Image Processing, IEEE Transactions on*, 16(7):1716–1725, 2007. 1

[4] C.-C. Chang and C.-J. Lin. Libsvm: A library for support vector machines. *ACM T. Intell. Syst. Technol.*, 2(3), 2011. 3

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 2

[6] N. Hesse, T. Gehrig, H. Gao, and H. Ekenel. Multi-view facial expression recognition using local appearance features. In *ICPR*, pages 3533–3536, 2012. 2

[7] Y. Hu, Z. Zeng, L. Yin, X. Wei, J. Tu, and T. Huang. A study of non-frontal-view facial expressions recognition. In *ICPR*, pages 1–4, 2008. 2

[8] Y. Hu, Z. Zeng, L. Yin, X. Wei, X. Zhou, and T. Huang. Multi-view facial expression recognition. In *FGR*, pages 1–6, 2008. 6, 7

[9] X. Huang, G. Zhao, and M. Pietikinen. Emotion recognition from facial images with arbitrary views. In *Proc. the British Machine Vision Conference (BMVC 2013), Bristol, UK*, page 11 p, 2013. 2, 6, 7

[10] X. Huang, G. Zhao, M. Pietikinen, and W. Zheng. Dynamic facial expression recognition using boosted component-based spatiotemporal features and multi-classifier fusion. *ACIVS*, pages 312–322, 2010. 2

[11] T. Jabid, M. H. Kabir, and O. Chae. Facial expression recognition using local directional pattern (LDP). In *ICIP*, 2010. 2

[12] I. Kotsia, S. Zafeiriou, and I. Pitas. Texture and shape information fusion for facial expression and facial action unit recognition. *Pattern Recognition*, 41(3), 2008. 2

[13] W. Liu, C. Song, and Y. Wang. Facial expression recognition based on discriminative dictionary learning. In *ICPR*, pages 1839–1842, 2012. 1

[14] S. Moore and R. Bowden. Multi-view pose and facial expression recognition. In *Proc. BMVC*, 2010. 2

[15] S. Moore and R. Bowden. Local binary patterns for multi-view facial expression recognition. *Computer Vision and Image Understanding*, 115:541–558, 2011. 2

[16] I. Naseem, R. Togneri, and M. Bennamoun. Linear regression for face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(11):2106–2112, 2010. 1

[17] O. Rudovic, I. Patras, and M. Pantic. Regression-based multi-view facial expression recognition. In *ICPR*, pages 4121–4124, 2010. 2

[18] O. Rudovic, V. Pavlovic, and M. Pantic. Multi-output laplacian dynamic ordinal regression for facial expression recognition and intensity estimation. In *CVPR*, pages 2634–2641, 2012. 1

[19] S. Taheri, P. K. Turaga, and R. Chellappa. Towards view-invariant expression analysis using analytic shape manifolds. In *Automatic Face & Gesture Recognition*, 2011. 2

[20] H. Tang, M. Hasegawa-Johnson, and T. Huang. Non-frontal view facial expression recognition based on ergodic hidden markov model supervectors. In *Multimedia and Expo (ICME), 2010 IEEE International Conference on*, pages 1202–1207, 2010. 6, 7

[21] U. Tariq, J. Yang, and T. Huang. Multi-view facial expression recognition analysis with generic sparse coding feature. *ECCV*, pages 578–588, 2012. 1, 2, 6, 7

[22] U. Tariq, J. Yang, and T. Huang. Maximum margin gmm learning for facial expression recognition. *FG*, pages 1–6, 2013. 6, 7

[23] U. Tariq, J. Yang, and T. Huang. Supervised supervector encoding for facial expression recognition. *Pattern Recognition Letters*, 46:89–95, 2014. 6, 7

[24] Y. Tian, T. Kanade, and J. Cohn. Facial expression recognition. In *Handbook of Face Recognition*, pages 487–519. 2011. 1

[25] R. Timofte, V. De, and L. V. Gool. Anchored neighborhood regression for fast example-based super-resolution. *ICCV*, pages 1920–1927, 2013. 2

[26] J. Tropp and A. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666, 2007. 3, 5

[27] X. Wang, T. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. *ICCV*, pages 32–39, 2009. 3

[28] L. Xu and P. Mordohai. Automatic facial expression recognition using bags of motion words. In *BMVC*, pages 13.1–13.13, 2010. 1

[29] L. Yin, X. Wei, Y. Sun, J. Wang, and M. Rosato. A 3d facial expression database for facial behavior research. In *Automatic Face and Gesture Recognition, 2006. FGR 2006. 7th International Conference on*, pages 211–216, 2006. 5

[30] Z. Zeng, M. Pantic, G. Roisman, and T. Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *PAMI*, 31(1):39–58, 2009. 1

[31] D. Zhang, M. Yang, and X. Feng. Sparse representation or collaborative representation: Which helps face recognition? *ICCV*, pages 471–478, 2011. 2

[32] W. Zheng, H. Tang, Z. Lin, and T. Huang. Emotion recognition from arbitrary view facial images. In *ECCV*, pages 490–503. 2010. 2, 6, 7

Index of Authors

The 20th Computer Vision Winter Workshop (CVWW) was organized by the Institute for Computer Graphics and Vision at Graz University of Technology. It took place from 9th to 11th of February 2015 in Seggau, Austria.

The Computer Vision Winter Workshop is the annual meeting of several computer vision research groups located in Graz, Ljubljana, Prague, and Vienna. The basic goal of this workshop is to communicate new ideas within the groups and to provide conference experience to PhD students.

ISBN 978-3-85125-388-7
9 783851 253887

www.ub.tugraz.at/Verlag
ISBN: 978-3-85125-388-7