 DOI: 10.3217/978-3-85125-603-1-22

Large Area 3D Human Pose Detection Via Stereo Reconstruction in Panoramic Cameras

Christoph Heindl[1] ,Thomas Pönitz[1], Andreas Pichler[1] and Josef Scharinger[2]

***Abstract*— We propose a novel 3D human pose detector using two panoramic cameras. We show that transforming fisheye perspectives to rectilinear views allows a direct application of two-dimensional deep-learning pose estimation methods, without the explicit need for a costly re-training step to compensate for fisheye image distortions. By utilizing panoramic cameras, our method is capable of accurately estimating human poses over a large field of view. This renders our method suitable for ergonomic analyses and other pose based assessments.**

I. INTRODUCTION AND RELATED WORK

Human pose estimation, characterized as the problem of localizing specific anatomic keypoints, has enjoyed substantial attention in recent years due to the large number of potential applications. It has been shown that keypoint based pose descriptions provide important cues for a variety of tasks such as activity recognition [1] and biomechanical analysis [2].

Inferring pose from a highly articulated, potentially self-occluding, non-rigid body is, in general, a hard and ill-posed problem. Non-optical approaches encompass electromechanical [3] or inertial sensor [4] based suits. Optical methods traditionally applied intrusive active or passive markers [5], [6] for keypoint detection. Early marker-free methods detected body parts in single images [7], [8], [9], [10]. 3D stereo imaging was used to infer human poses from sparse depth-maps [11], [12]. Real-time dense depth cameras greatly simplified the reconstruction task [13], [14], [15], [16] by providing additional metric constraints.

Recently, single and multi-person pose estimation in monocular images made significant progress [17], [18], [19]. Especially the existence of large-scale human annotated datasets [20], [21] accelerated deep learning based approaches [22], [23], [24].

Fisheye lenses have, despite their large field of view, received little attention mostly due to their inherent image distortions which significantly alter the appearance of objects as they move through its line of sight. Among the methods published, researchers have considered single person [25] detection, safe human-robot interactions [26] and head pose tracking [27]. As most of the optical solutions mentioned above assume a pinhole lens model, their results cannot be directly applied to fisheye images.

In this work we propose a 3D pose detector using two fisheye cameras in general position. We apply a deep convolutional network based 2D pose estimator to the input images and reconstruct the corresponding 3D joint coordinates via stereoscopic constraints. We show that proper rectilinear view generation from raw fisheye input images allows us to avoid tedious dataset generation and network training steps. We demonstrate the usefulness of our approach in a challenging 6×6 meter working area, and consider the applicability to ergonomic analysis with respect to accuracy and robustness. To our knowledge we are the first to propose a practical large-scale 3D human pose estimation system based on fisheye lenses.

[1]Profactor GmbH, Im Stadtgut A2, 4407 Steyr-Gleink, Austria
[2]JKU Department of Computational Perception, Altenbergerstr. 69, 4040 Linz, Austria

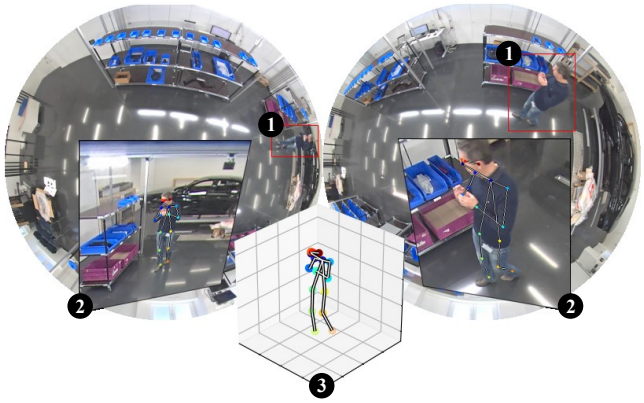


Fig. 1: Overview. From two highly distorted 180° fisheye images coarse human location cues are inferred (1). Regions of interest are transformed into rectilinear views and articulated 2D human poses are then predicted via deeply learned architectures (2). The corresponding 2D joints are then triangulated via stereoscopic constraints to yield accurate 3D body part locations (3) even in the outer edges of a fisheye lens.

II. NOTATION

Throughout this work we use lower-case non-bold characters x to denote scalars, bold-faced lower-case characters $\mathbf{x}$ represent column-vectors and upper-case bold characters $\mathbf{A}$ for matrices. $\mathbf{x}_i$ denotes the i-th element of x, $\mathbf{A}_{ij}$ the i-th row and j-th column of $\mathbf{A}$. For low dimensional vectors we also write $\mathbf{q}_x$ instead of $\mathbf{q}_0$ when it seems practical.

III. LENS MODELS AND PROJECTION FUNCTIONS

A majority of pose estimation methods mentioned in Section I assume that the camera model is sufficiently described by the pinhole camera model. Significant attention has therefore been paid in describing and correcting distortions

that usually appear in ordinary lenses with moderate radial distortion [28], [29]. However, these models are incompatible with wide angle lenses as their projective properties are not well captured.

We provide an brief overview of common lens models and projection functions in the next subsections. For an in depth discussion see [30], [31], [32]. We consider only rotational symmetric lenses and assume that the principal point and the focal length is known. Both parameters can be determined by a number of methods [33], [34], [35].

We consider the characteristics of a lens to be captured in functional relationship between distorted image points on the focal plane and corresponding object points. As illustrated in Figure 2, the radial distance r_d of the optical center to the distorted image point is a function of the object's inclination angle with the z-axis θ, the plane-polar angle around the optical axis ϕ and the focal length f. We consider radially symmetric lenses and therefore assume that the angle ϕ, in contrast to θ, remains unchanged by the lens (see Figure 2).

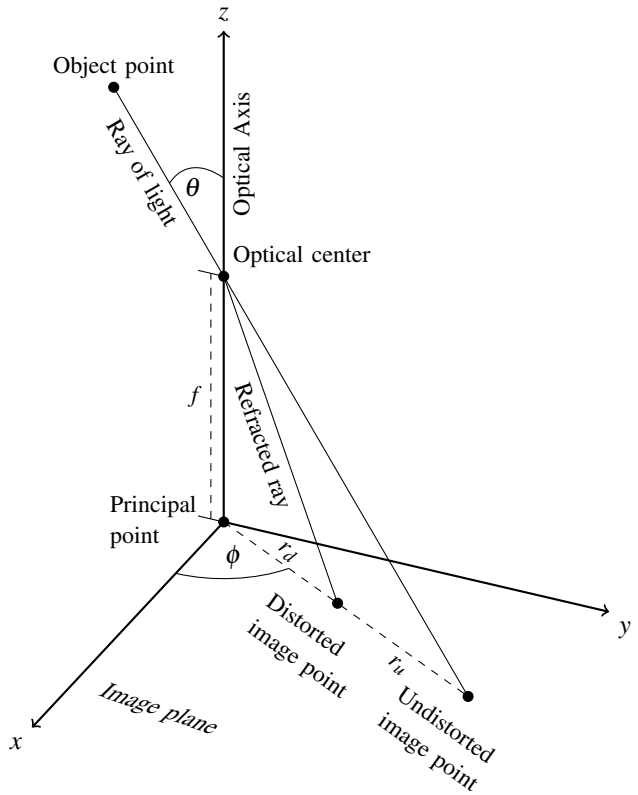


Fig. 2: Illustration of the general projection model and its related parameters. A ray of light emitted from a 3D object passes through the optical center and is potentially refracted due to lens characteristics. The measures r_u and r_d correspond to the ideal (rectilinear) and distorted (actual) radial distances from the principal point. The inclination angle θ measures the angular difference between the ray of light and the optical axis. The angle ϕ denotes the plane-polar angle around the optical axis. The focal length f represents the distance between the optical center and the image plane.

A. Rectilinear projection

The most frequently found projection in computer vision is the pinhole projection. Due to the property that this projection preserves straight lines it is also termed the rectilinear projection. The projection function is given by

$$r_d = f\tan(\theta). \tag{1}$$

For this particular model $r_d = r_u$ and for large field of views the projected image becomes increasingly large and finally infinite when the field of view reaches 180° degrees.

B. Fisheye projections

Similar to rectilinear lenses, fisheye lenses have been manufactured to adhere to optical-engineered projection behavior. The projection functions governing these designs are also known as the classic projection functions and are listed below

$$\text{Equidistant: } r_d = f\theta \tag{2}$$
$$\text{Stereographic: } r_d = 2f\tan(\tfrac{\theta}{2}) \tag{3}$$
$$\text{Equisolid: } r_d = 2f\sin(\tfrac{\theta}{2}) \tag{4}$$
$$\text{Orthographic: } r_d = f\sin(\theta). \tag{5}$$

In Figure 3 we compare radial distorted distances, r_d, as a function of the inclination angle, θ, for all projection models. Note, the monotonicity of the functions that ensures that all angles are mapped to different radial distances.

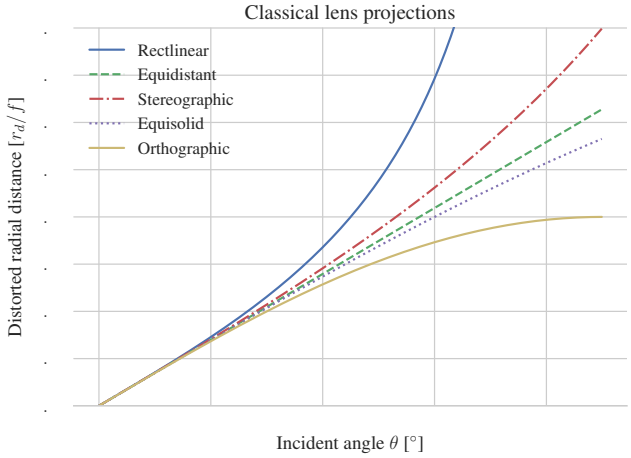


Fig. 3: Plots of classical fisheye projection equations showing normalized radial distorted distances r_d/f as a function of the inclination angle θ. Note the monotonicity of the functions that ensures that all angles are mapped to different radial distances.

Besides the classic projection functions, various other models have been proposed. Most notably are polynomial models [30], a summation of sine terms model [36] and a universal model [35]. Unlike the classical optical-engineered models, these models try to capture a variety of different lenses in a single formula. While the classic projection formulas can be inverted algebraically (i.e determining the θ from r_d) this may not be as easily true for the alternatives.

IV. RECTILINEAR VIEW GENERATION

Generating rectilinear views from fisheye images, as shown in Figure 4, is an important technique in our approach, as our 2D pose detectors assume upright images taken by a pinhole camera model.

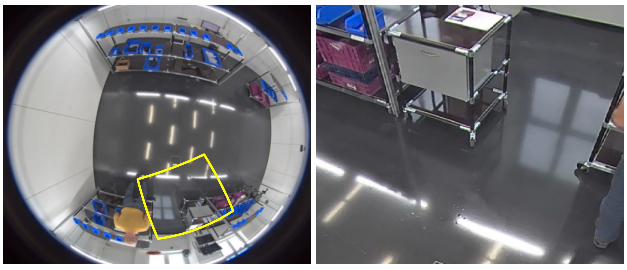


Fig. 4: Upright rectilinear view generation. Fisheye input image on the left, rectilinear view on the right. The bounds of the rectilinear view are shown in yellow in the left image.

Our approach to generate rectilinear views is based on virtual pinhole cameras that share the same origin as the fisheye cameras but are arbitrarily rotated with respect to their physical counterpart. In order to map image points between the artificial pinhole and physical fisheye view the following steps are applied in order.

1) *Un-project* - computes object points from distorted image points using the destination camera model.
2) *Rotate* - rotates object points into the source camera space.
3) *Project* - computes distorted image points from object points using the source camera model.

When applied to all pixels of a destination view, this method leads to efficient lookup maps of corresponding locations with sub-pixel accuracy. The destination image is then formed by interpolating pixels via lookup coordinates. While we are usually interested in mapping fisheye image and rotated pinhole image coordinates, our method works for any pair of camera models.

A. Projection from object space

To compute distorted homogeneous image coordinates $\mathbf{i} = [\mathbf{i}_x, \mathbf{i}_y, 1]^T$ for a Cartesian point $\mathbf{o} = [\mathbf{o}_x, \mathbf{o}_y, \mathbf{o}_z]^T$ in object space, we first compute its spherical coordinates with respect to the camera intrinsic frame

$$\begin{bmatrix} r \\ \theta \\ \phi \end{bmatrix} = \begin{bmatrix} \|\mathbf{o}\| \\ \arccos\left(\mathbf{o}_z / \|\mathbf{o}\|\right) \\ \arctan 2(\mathbf{o}_y, \mathbf{o}_x) \end{bmatrix}. \tag{6}$$

Next, we compute r_d from θ according to the lens projection model (see III-B). The vector $[r_d \ \phi]^T$ then denotes the polar coordinates of the distorted image point. The Cartesian coordinates are given by

$$\begin{bmatrix} \mathbf{i}_x \\ \mathbf{i}_y \\ 1 \end{bmatrix} = \begin{bmatrix} r_d \cos\phi \\ r_d \sin\phi \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{c}_x \\ \mathbf{c}_y \\ 1 \end{bmatrix} \tag{7}$$

where $[\mathbf{c}_x \ \mathbf{c}_y \ 1]^T$ denotes the camera principal point. Henceforth, we denote the projection operation $\text{project}(\mathbf{o}; M)\colon \mathbb{R}^3 \to \mathbb{R}^3$ as a functional mapping that takes three-dimensional object points, $\mathbf{o}$, to homogeneous two-dimensional image points $[\mathbf{i}_x, \mathbf{i}_y, 1]^T$ using the lens model M.

B. Reverse projection from image space

Reversing the process outlined in Section IV-A is ambiguous, as depth is lost during projection and all locations along a ray project to the same image coordinates. For our purposes it suffices to un-project image coordinates to points on the unit sphere, as our consideration mainly involves purely rotated cameras. First, we compute the polar coordinate representation for the image point i

$$\mathbf{n} = \mathbf{i} - \mathbf{c} \tag{8}$$

$$\begin{bmatrix} r_d \\ \phi \end{bmatrix} = \begin{bmatrix} \|\mathbf{n}\| \\ \arctan 2(\mathbf{n}_y, \mathbf{n}_x) \end{bmatrix}. \tag{9}$$

We then apply the reverse projection function to obtain θ from r_d according to the lens model. The vector $[r \ \theta \ \phi]^T$ describes a ray from the origin into object space through i in spherical coordinates. Setting $r = 1$, constrains the point to the unit sphere. Converting back to Cartesian coordinates gives

$$\begin{bmatrix} \mathbf{o}_x \\ \mathbf{o}_y \\ \mathbf{o}_z \end{bmatrix} = \begin{bmatrix} r\sin(\theta)\cos\phi \\ r\sin(\theta)\sin\phi \\ r\cos(\theta) \end{bmatrix}. \tag{10}$$

We define the reverse projection operation $\text{unproject}(\mathbf{o}; M)\colon \mathbb{R}^3 \to \mathbb{R}^3$ to be a functional mapping from homogeneous image points $[\mathbf{i}_x, \mathbf{i}_y, 1]^T$, to three-dimensional object points $\mathbf{o}$ using the lens model M.

The general mapping of image points between two purely rotated cameras can now be written as

$$\mathbf{o} = \text{unproject}([\mathbf{i}_x \ \mathbf{i}_y \ 1]^T; M) \tag{11}$$

$$\mathbf{o}' = \mathbf{R}'^T \mathbf{R}\mathbf{o} \tag{12}$$

$$\mathbf{i}' = \text{project}(\mathbf{o}'; M') \tag{13}$$

where M, M' are the respective camera lens models and $\mathbf{R}$, $\mathbf{R}'$ are camera orientations. The mapping is simplified when both lens models are rectilinear, in which case the mapping can be conveniently described by a homography of the following form

$$\mathbf{i}' = \mathbf{K}'\mathbf{R}'^T\mathbf{R}\mathbf{K}^{-1}[\mathbf{i}_x, \mathbf{i}_y, 1]^T \tag{14}$$

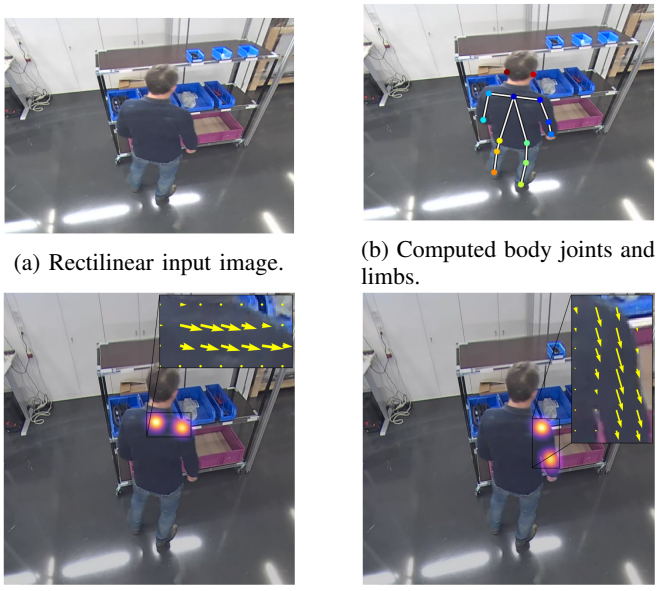
where $\mathbf{K}$ and $\mathbf{K}'$ contain camera intrinsics.

V. HUMAN POSE ESTIMATION

Human body part estimation consists of two steps. First, two-dimensional human poses are estimated in rectilinear views formed from both fisheye cameras. Then, a three-dimensional reconstruction is computed based on stereographic constraints. Both steps are detailed below.

A. 2D Human Pose Estimation

Our 2D pose detection is based on the method of Cao et al. [24], which takes as input a rectilinear image view and outputs anatomic keypoint locations. A neural network is used to predict body joint confidence maps and a set of two dimensional vector fields that encode so called body limb affinities. The basic building block of neural network is a set of convolutional filters that are iteratively applied to previous results in order to refine confidence and affinity maps. In the predicted confidence maps, peak localization is performed to identify potential joint candidates. Then, a graph algorithm guided by affinity maps is used to determine the connectivity. Figure 5 illustrates various stages of the algorithm.



(a) Rectilinear input image.

(b) Computed body joints and limbs.

(c) Joint confidences and part affinities between neck and right shoulder.

(d) Joint confidences and part affinities between right shoulder and right elbow.

Fig. 5: Various stages of the 2D pose estimation algorithm. From a rectilinear view (top-left) a convolutional neural network predicts for every joint and limb a confidence and affinity vector field (bottom-right and-bottom left). A graph based algorithm the constructs a skeleton body model based on these inputs (top-right).

As the detector is directly applied to upright rectilinear views we can use pre-trained network weights and avoid time consuming manual annotation of fisheye images. In order to bootstrap the rectilinear view generation an initial guess of people positions in fisheye images is needed. This can be solved in a number of ways, such as using a people detector [37] or performing foreground segmentation [38]. Once an initial view orientation is known, subsequent rectilinear views can be computed automatically by re-focusing the view on detected 2D human poses.

B. 3D Human Pose Reconstruction

Computing 3D joint coordinates requires at least 2 fisheye images with projections of the same world space joint. Given a rigid transformation between two capture devices, the 3D location can be obtained via triangulation. For static camera setups the rigid transformation can be estimated [39] in a preprocessing step. For non-rigid setups, camera position and scene geometry needs to be inferred simultaneously. This is considered a bundle adjustment problem for which numerous iterative non-linear solutions have been proposed [40], [41].

In either case, the usual linear epipolar constraints for stereo setups do not hold, because fisheye cameras exhibit non-linear projection functions. Therefore, we perform triangulation directly in rectilinear views instead of fisheye images, for which the constraints hold. Without loss of generality, let $\mathbf{P} \in \mathbb{R}^{3\times4}$ be the camera projection matrix of the first rectilinear camera given by

$$\mathbf{P} = \mathbf{K}\begin{bmatrix}\mathbf{I} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{R}^T & \mathbf{0} \\ 0 & 1\end{bmatrix}\mathbf{W}^{-1} \tag{15}$$

where $\mathbf{K} \in \mathbb{R}^{3\times3}$ is the rectilinear projection matrix, $\mathbf{R} \in \mathbb{R}^{3\times3}$ the orientation of rectilinear view with respect to the parental fisheye camera, $\mathbf{W} \in \mathbb{R}^{4\times4}$ is the position and orientation of the fisheye camera in world space and $\mathbf{I} \in \mathbb{R}^{3\times3}$ is the identity matrix. Similarily we define $\mathbf{P}'$ for the second rectilinear view. The simultaneous projection of a homogeneous world point $\mathbf{x}$ in either camera focal plane is given by

$$w\left[\mathbf{i}_x, \mathbf{i}_y, 1\right]^T = \mathbf{P}\mathbf{x} \tag{16}$$

$$w'\left[\mathbf{i}'_x, \mathbf{i}'_y, 1\right]^T = \mathbf{P}'\mathbf{x}. \tag{17}$$

Rewriting Equation 16 line by line and denoting by $\mathbf{p}_i$ the i-th row of $\mathbf{P}$ we get

$$w\mathbf{i}_x = \mathbf{p}_0\mathbf{x} \tag{18}$$

$$w\mathbf{i}_y = \mathbf{p}_1\mathbf{x} \tag{19}$$

$$w = \mathbf{p}_2\mathbf{x}. \tag{20}$$

Then, the Direct Linear Transform (DLT)[42] algorithm is given by eliminating w from Equations 18, 19 via substitution using Equation 20. Rewriting leads to two linear equations in four unknowns of $\mathbf{x} = \begin{bmatrix}x & y & z & w\end{bmatrix}^T$

$$(x'\mathbf{p}_2 - \mathbf{p}_0)\mathbf{x} = 0 \tag{21}$$

$$(y'\mathbf{p}_2 - \mathbf{p}_1)\mathbf{x} = 0. \tag{22}$$

Two views then yield the four equations. The system of equations may be written in matrix form as $\mathbf{A}\mathbf{x} = \mathbf{0}$ and solved for $\mathbf{x}$ in multiple ways [43]. The DLT algorithm allows us to compute 3D point coordinates for corresponding image coordinates in two rectilinear views. Applying it to two-dimensional joint correspondences leads to three-dimensional reconstruction of body parts. Figure 6 shows several successful reconstructions.

VI. EVALUATION AND RESULTS

The setup we use for evaluation covers a 6×6 meter working area with two fisheye cameras of type Axis M3007-PV mounted to the ceiling at height of 3 meters. The baseline between the cameras is roughly 1.5 meters. The simulated working area consists of several shelves that often

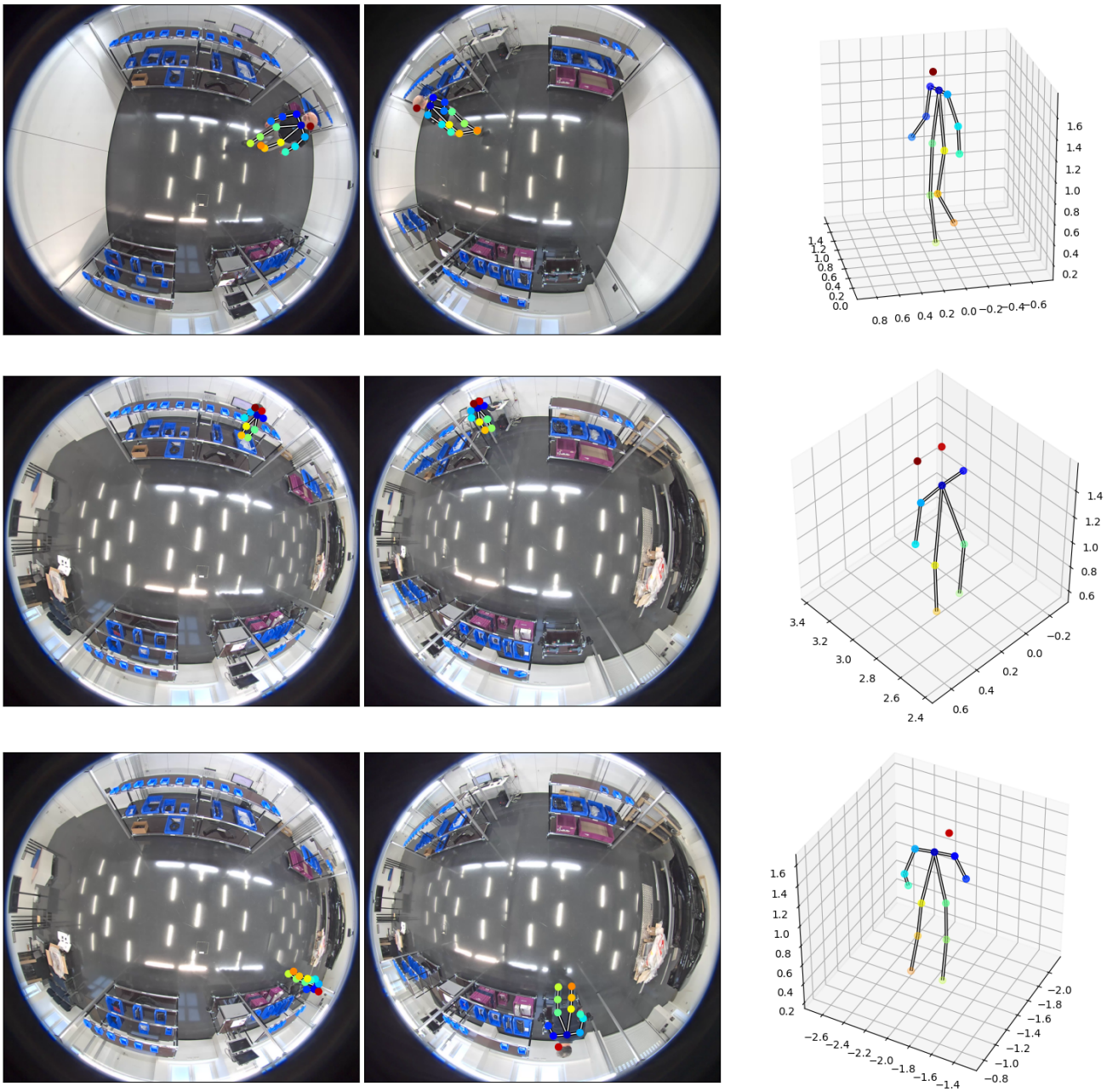


Fig. 6: Examples of 3D stereo reconstruction of human body joints in fisheye images. Left/middle: input images and superimposed detected two dimensional body joints. Right: Three dimensional metric body model.

cause partial body occlusions. We captured raw RGB video from both cameras at rate of 12 FPS at a resolution of 2592×1944 over a period 4 weeks producing a total of 20 hours material. The video data contains 4 different people performing common assembly tasks.

The fisheye cameras have been intrinsically calibrated using the method described in [33]. We obtained the extrinsic calibration for each camera separately by using an external tracking device[1], whose tracking targets can be automatically detected in images. By capturing a set of 3D and corresponding distorted 2D image correspondences in rectilinear views, we solve for the unknown pose $\mathbf{W}$ using a iterative scheme [44].

We assess the accuracy of the calibration and rectilinear view generation by measuring lengths of known objects in reconstructed stereo scenes. For better readability we split the field of view of the fisheye camera into disjoint rings corresponding to increasing radial distortions. The results are shown in Table I.

We trained the 2D pose estimation algorithm on the

[1]HTC Vive `https://www.vive.com/eu/`

	Target length (m)	Measured length (m)
Central area	1.5	1.49 ±0.01
Outer area	1.5	1.52 ±0.018
Lens edge area	1.5	1.56 ±0.035

TABLE I: Measurement errors incurred by the stereo setup inaccuracies. For better comprehensibility we split the fish-eye field of view into three concentric rings that mark central (low-error), outer area (mid-error) and edge area (high-error). Shown are target lengths as well as upper/lower limits over multiple measurements.

COCO[21] dataset, which defines 18 body joints and 17 limbs (see Figure 7). Since the accuracy of the 2D pose estimation has already been studied elsewhere[24], [45], we concentrate on evaluating the quality of the 3D reconstruction. We validate the 3D reconstruction by accumulating 3D limb lengths over video segments grouped by individual persons. Ideally, limb lengths are stationary. However, due to stereo setup imprecision and fluctuations in 2D detection we observe varying limb lengths as shown in Figure 8. Bear in mind that the errors are mostly introduced when people move around in the lens edge area. Figure 9 shows the relative reconstruction frequencies of each limb.

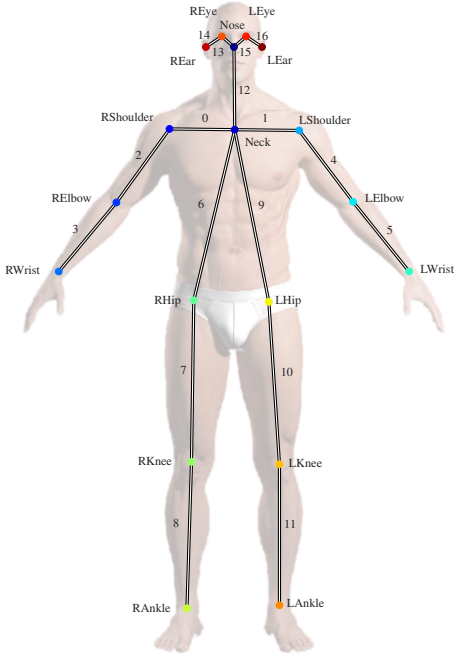


Fig. 7: Joint names and limb indices for the COCO[21] model.

We ran the evaluation on a workstation with an Intel i7-7700 3.6 GHz, 16 GB RAM, and a NVIDIA GeForce GTX1060 graphics card with 6 GB memory. Relevant performance metrics for key stages in our algorithm are given in Table II.

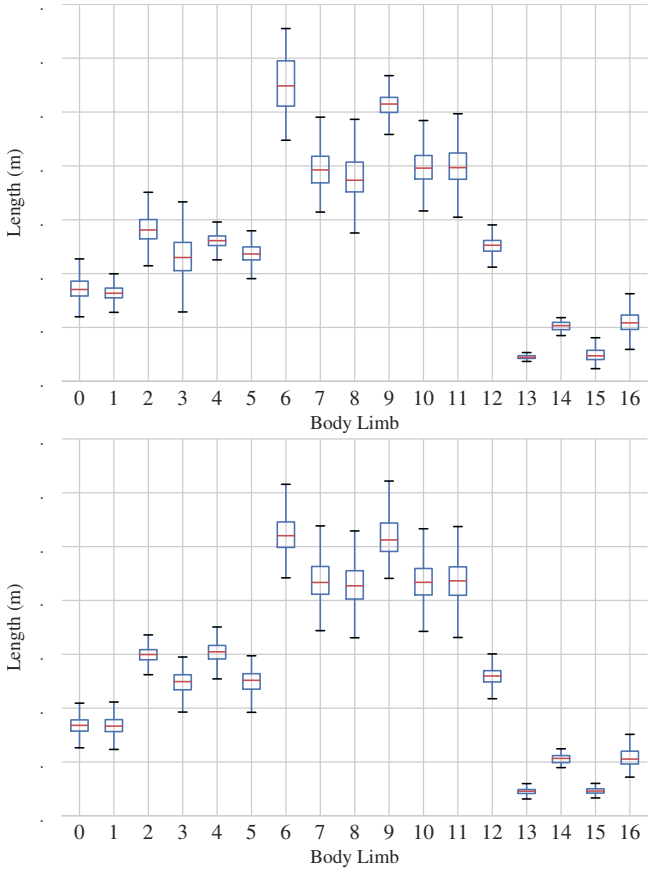


Fig. 8: Metric limb length statistics for two different people over sequence of 10800 frames (15 minutes). Note, the second person has longer legs - he is roughly 5-8 cm taller. The area covered is 6×6 meter and includes serval obstacles that lead to occlusions. Refer to Figure 7 for limb number lookup.

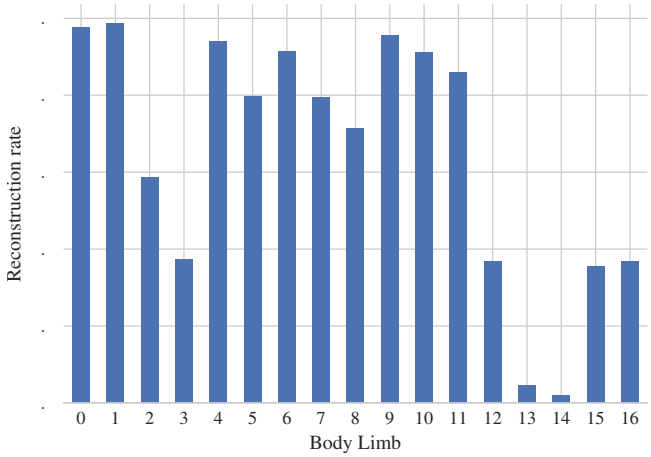


Fig. 9: Reconstruction frequencies of individual limbs over the entire video testing set. Refer to Figure 7 for limb number lookup. Facial features are reconstructed less frequently compared to larger body parts due to visibility constraints.

View Resolution	View Generation (s)	2D Detection (s)	3D Reconstruction (s)
320×320	0.02 ±0.001	0.60 ±0.02	0.01 ±0.001
640×640	0.10 ±0.01	1.80 ±0.05	0.01 ±0.001

TABLE II: Performance timings of key stages in our algorithm. We compare two different resolutions of rectilinear views and note how they affect each stage of the reconstruction pipeline.

VII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel human pose detector that predicts three-dimensional body part locations from two highly distorted fisheye cameras in general position. We demonstrated that the highly enlarged field of view of a fisheye lens is a compelling advantage in reducing hardware complexity. Especially the number of cameras needed to capture the scene can ne reduced and thus many related calibration efforts can be avoided. With regard to pose evaluations, we find that analyses are possible with an accuracy of 2-3 cm over a range of 6x6 meters.

We utilized recent deep-learning based approaches to 2D pose estimation in images and showed that generating artificial rectilinear views avoids the re-training of the neural network. To our knowledge we are the first to consider deep-learning based human pose reconstruction using stereo fisheye lenses. As a matter of fact we observe increasing inaccuracies in 3D reconstruction in the limit of the lens.

In future work we will therefore reconsider the current triangulation method and verify whether additional smoothness constraints can help to reduce the errors. Another point of interest is reduction of runtime complexity, by improving the runtime of the 2D pose detector, in order to achieve real-time performance.

ACKNOWLEDGMENT

This research was supported in part by Lern4MRK (Austrian Ministry for Transport, Innovation and Technology), "FTI Struktur Land Oberoesterreich (2017-2020)", the European Union in cooperation with the State of Upper Austria within the project Investition in Wachstum und Beschäftigung (IWB), as well as AutoScan (FFG, 853416).

REFERENCES

[1] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, p. 16, 2011.

[2] T. B. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *Computer vision and image understanding*, vol. 104, no. 2-3, pp. 90–126, 2006.

[3] M. Motion, "Gypsy motion capture system," 2004.

[4] D. Roetenberg, H. Luinge, and P. Slycke, "Xsens mvn: full 6dof human motion tracking using miniature inertial sensors," *Xsens Motion Technologies BV, Tech. Rep*, vol. 1, 2009.

[5] I. Vicon Motion Systems, "Vicon Motion Systems." http://www.vicon.com.

[6] I. Qualisys, "Qualisys.The Swedish motion capture company." http://www.qualisys.com.

[7] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International journal of computer vision*, vol. 61, no. 1, pp. 55–79, 2005.

[8] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1014–1021, IEEE, 2009.

[9] L. Pishchulin, M. Andriluka, P. Gehler, and B. Schiele, "Poselet conditioned pictorial structures," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pp. 588–595, IEEE, 2013.

[10] Y. Yang and D. Ramanan, "Articulated human detection with flexible mixtures of parts," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 12, pp. 2878–2890, 2013.

[11] H.-D. Yang and S.-W. Lee, "Reconstruction of 3d human body pose from stereo image sequences based on top-down learning," *Pattern Recognition*, vol. 40, no. 11, pp. 3120–3131, 2007.

[12] Y. Matsumoto and A. Zelinsky, "An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement," in *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pp. 499–504, IEEE, 2000.

[13] S. Knoop, S. Vacek, and R. Dillmann, "Sensor fusion for 3d human body tracking with an articulated 3d body model," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, pp. 1686–1691, IEEE, 2006.

[14] Y. Zhu and K. Fujimura, "Constrained optimization for human pose estimation from depth sequences," in *Asian Conference on Computer Vision*, pp. 408–418, Springer, 2007.

[15] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun, "Real-time identification and localization of body parts from depth images," in *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pp. 3108–3113, IEEE, 2010.

[16] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1297–1304, Ieee, 2011.

[17] T. Pfister, J. Charles, and A. Zisserman, "Flowing convnets for human pose estimation in videos," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1913–1921, 2015.

[18] U. Iqbal, M. Garbade, and J. Gall, "Pose for action-action for pose," *arXiv preprint arXiv:1603.04037*, 2016.

[19] U. Iqbal, A. Milan, and J. Gall, "Pose-track: Joint multi-person pose estimation and tracking," *CoRR*, vol. abs/1611.07727, 2016.

[20] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*, pp. 3686–3693, 2014.

[21] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," *CoRR*, vol. abs/1405.0312, 2014.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[23] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," *CoRR*, vol. abs/1312.4659, 2013.

[24] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime multi-person 2d pose estimation using part affinity fields," in *CVPR*, vol. 1, p. 7, 2017.

[25] K. Otsuka, S. Araki, K. Ishizuka, M. Fujimoto, M. Heinrich, and J. Yamato, "A realtime multimodal system for analyzing group meetings by combining face pose tracking and speaker diarization," in *Proceedings of the 10th international conference on Multimodal interfaces*, pp. 257–264, ACM, 2008.

[26] E. Cervera, N. Garcia-Aracil, E. Martinez, L. Nomdedeu, and A. P. Del Pobil, "Safety for a robot arm moving amidst humans by using panoramic vision," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 2183–2188, IEEE, 2008.

[27] R. Stiefelhagen, J. Yang, and A. Waibel, "Simultaneous tracking of head poses in a panoramic view," in *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, vol. 3, pp. 722–725, IEEE, 2000.

[28] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[29] D. C. Brown, "Decentering distortion of lenses," *Photogrammetric Engineering and Remote Sensing*, 1966.

[30] A. Basu and S. Licardie, "Alternative models for fish-eye lenses," *Pattern recognition letters*, vol. 16, no. 4, pp. 433–441, 1995.
[31] D. Schneider, E. Schwalbe, and H.-G. Maas, "Validation of geometric models for fisheye lenses," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 64, no. 3, pp. 259–266, 2009.
[32] C. Hughes, P. Denny, E. Jones, and M. Glavin, "Accuracy of fish-eye lens models," *Applied optics*, vol. 49, no. 17, pp. 3338–3347, 2010.
[33] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 8, pp. 1335–1340, 2006.
[34] S. Shah and J. Aggarwal, "Intrinsic parameter calibration procedure for a (high-distortion) fish-eye lens camera with distortion model and accuracy estimation," *Pattern Recognition*, vol. 29, no. 11, pp. 1775–1788, 1996.
[35] D. B. Gennery, "Generalized camera calibration including fish-eye lenses," *International Journal of Computer Vision*, vol. 68, no. 3, pp. 239–266, 2006.
[36] T. J. Herbert, "Calibration of fisheye lenses by inversion of area projections," *Applied optics*, vol. 25, no. 12, pp. 1875–1876, 1986.
[37] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," *arXiv preprint arXiv:1703.07402*, 2017.
[38] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis, "Real-time foreground–background segmentation using codebook model," *Real-time imaging*, vol. 11, no. 3, pp. 172–185, 2005.
[39] S. Abraham and W. Förstner, "Fish-eye-stereo calibration and epipolar rectification," *ISPRS Journal of photogrammetry and remote sensing*, vol. 59, no. 5, pp. 278–288, 2005.
[40] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
[41] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An invitation to 3-d vision: from images to geometric models*, vol. 26. Springer Science & Business Media, 2012.
[42] R. Hartley, R. Gupta, and T. Chang, "Stereo from uncalibrated cameras," in *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, pp. 761–764, IEEE, 1992.
[43] R. I. Hartley and P. Sturm, "Triangulation," *Computer vision and image understanding*, vol. 68, no. 2, pp. 146–157, 1997.
[44] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of applied mathematics*, vol. 2, no. 2, pp. 164–168, 1944.
[45] G. Ning and Z. He, "Dual path networks for multi-person human pose estimation," *arXiv preprint arXiv:1710.10192*, 2017.