Multi-pose Volume Reconstruction Across Arbitrary Trajectory from Multiple Fisheye Cameras

Konstantina Kottari and Konstantinos Delibasis

Dept. of Computer Science and Biomedical Informatics, Univ. of Thessaly, Greece
{kottarikonstantina,kdelibasis}@gmail.com

Abstract. Volume reconstruction from silhouettes is a known subject in the case of projective cameras. The use of dioptric omni-directional cameras (fisheye) that exhibit semispheric Field of View (FoV) allows simultaneous imaging of the whole available space. In this work we employ two fisheye cameras in order to acquire silhouettes of humans that move within the imaged indoor space without any kind of restriction. We present the basic algorithm for reconstructing the volumetric model of a human in the case of using two images of its silhouette, acquired by the fisheye cameras. Then we extend this algorithm in the case of a human moving along any trajectory a) assuming no pose change and b) considering pose change during motion. Quantitative results from synthetic data indicate that volumes can be reconstructed with high accuracy (error of few cm), from a small number of positions using only two fisheye cameras. The proposed algorithm achieves the same level of accuracy in the case of recovering the volume of multiple poses under same conditions.

Keywords: Silhouette to shape - S2S, Fisheye camera, Volumetric model, Optic Hull, Multi-pose volume recovery, Computer Vision, Smart Graphics.

1 Introduction

Given a single silhouette image of an object, we know that the 3D object lies inside the volume generated by backprojecting the silhouette area using the parameterized camera calibration (subsection 3.1). This volume is shaped as a generalized cone. The intersection of the generalized cones generated by each available binary silhouette, builds a volume, which is guaranteed to contain the object. This volume is called visual (optic) hull of the object.

A number of cameras is required for sufficiently accurate volume reconstruction, thus introducing difficulty in accurate positioning and calibration of the cameras. In this work we exploit the main characteristic of the fisheye lens that exhibits a 360^0 azimuthial and 180^0 elevation FoV, in order to obtain simultaneous multiple sihlouettes of an object that moves inside the imaged indoor space. Fisheye cameras are dioptric omni-directional cameras, whose use in computer vision application is becoming popular [1], [2]. In [3], [4], [5] the calibration of fisheye camera is reported to emulate the strong deformation introduced by the fisheye lens. In [6] the authors present a methodology for correcting the distortions induced by the fisheye lens.

R. Chbeir et al. (Eds.): AIAI 2015, IFIP AICT 458, pp. 29–44, 2015.
DOI: 10.1007/978-3-319-23868-5_3

2 Related Work

Recovering 3D structure from multiple binary (2D) shapes (silhouette to shape – S2S, commonly referred as shape from silhouette) by projective cameras is a well-known technique [7], [8]. The limited field of view (FoV) of the projective cameras requires the object to be imaged in a specific location, rather than anywhere in the FoV. Little work has been reported in shape-from-silhouette using fisheye cameras. In [9] the use of an omni-directional camera is proposed for industrial object recognition, employing a simple 2D hull intersection. Epipolar geometry between pairs of catadioptric omni-directional cameras has been established in [10]. A combination of perspective and para-catadioptric cameras is used to define epipolar geometry and is subsequently applied for camera self-calibration [11]. In the two aforementioned works no application of S2S is reported.

In this work, a volume carving algorithm is presented for recovering the shape of the object using multiple silhouettes acquired by two fisheye cameras, at different times and positions along any path. Initially, single pose reconstruction is performed by a still synthetic human, utilizing two fisheye cameras. This algorithm is further refined to reconstruct the single pose by the two fisheye cameras, whereas the object of interest (synthetic human) moves in an arbitrary path in the imaged room. The proposed S2S algorithm is then extended to reconstruct multiple distinct poses of the imaged person that alternate randomly along the path. Reconstruction of different poses is performed by establishing correspondance between silhouettes and previously reconstructed poses, using similarity measures.

In this study we concentrate in synthetic data, obtained from a number of available three dimentional (3D) human models, in order to quantitavely access the accuracy of the algorithm. The main assumptions of the algorithm are: accurate calibration of the fisheye cameras and orientation estimation using the motion of the object.

3 Methodology

3.1 Calibration of the Fisheye Cameras

The two fisheye cameras were both calibrated using a set of manually provided points, as described in detail in [12]. The achieved calibration compares favorably to other state of the art [13] in terms of accuracy. Let B_1, B_2 be the binary frames containing the silhouette as imaged by the two fisheye cameras. After completion of the calibration process for both fisheye cameras, functions F_1, F_2 are defined. These functions map a real world point $\mathbf{x_{real}} = (x_{real}, y_{real}, z_{real})$ to coordinates $(i_1, j_1), (i_2, j_2)$ of binary frames B_1, B_2 of the two cameras c=1,2 , respectively:

$$F_c(\mathbf{x_{real}}) = (i_c, j_c), c = 1,2 \tag{1}$$

The resulting calibration for one of the cameras is visualized in Fig.1.

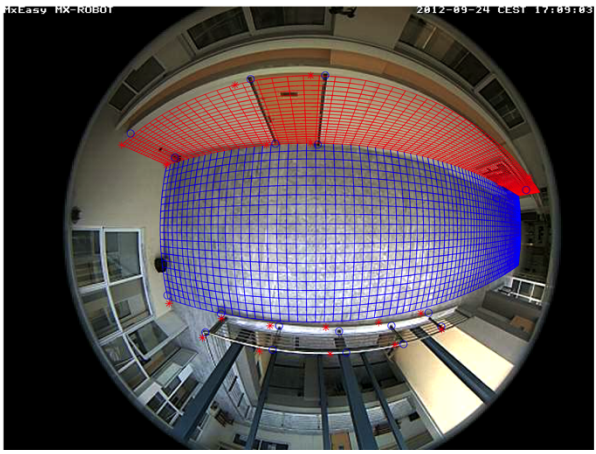
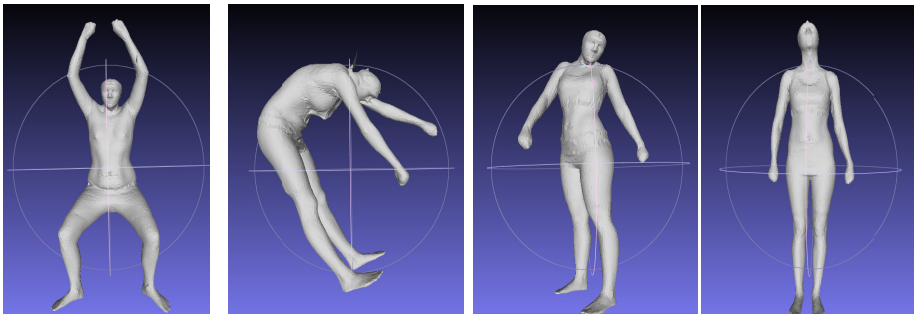


Fig. 1. Visualization of the resulting fisheye model calibration, on the FoV of the indoor environment in which experiments have taken place. The landmark points defined by the user are shown as circles and their rendered position on the frame marked by stars.

3.2 Synthetic Data Using 3D Human Model

In this work we utilized a number of 3D models (Fig. 2) with known real-world coordinates obtained from [14], [15]. These models are in the form of triangulated surfaces. However, only the coordinates of the vertices are used for rendering the binary silhouette frames, as described in subsection 3.3. Every model was placed in room along a trajectory at different orientations along a path of arbitrary shape shown in Fig. 3 as described in 3.4. Moreover, models (a) and (b) were placed along the same path alternating each other at varying order, to create the synthetic data used for conducting the multi-pose reconstruction described in 3.5.



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

Fig. 2. The 3D human model poses used this work

3.3 3D Volume Reconstruction Using One Projection from Each Fisheye Camera

In order to utilize the available fisheye cameras c=1,2, we modified the classic space carving algorithm [1], using the parameterized camera calibration (subsection 3.1). This volume is shaped as a generalized cone, which is guaranteed to contain the object (optic hull). Simple polyhedron-polyhedron intersection algorithms [16 Sub.3.3] are inefficient and produce polyhedral rather than volumetric representations. Therefore, we perform paired intersection of volumes and transformation of pixels of the binary silhouette to voxels through indexing as described below.

Each point of the 3D human model, $\mathbf{v_k}$, with k=1,2,…, N_m (N_m is the number of points of the model) is rendered on a pixel of frames B_1 and B_2, by cameras 1 and 2, respectively, according to the calibration of each camera (subsection 3.1): $B_1\left(F_1\left(\mathbf{v_k}\right)\right)=1, B_2\left(F_2\left(\mathbf{v_k}\right)\right)=1$. Furthermore, a cube C, large enough to include a sufficient portion of the real world space, is initialized and divided into elementary volumes δV of dimensions $\delta l \times \delta l \times \delta l$, with δl equal to 1 cm. In order to minimize memory requirements we define cube C sufficiently large around model's position $\mathbf{p}$, rather than covering the whole room, which is imaged by the available fisheye cameras. The employed silhouette-to-shape algorithm returns a binary volume $\mathbf{V}$, which is the discretized cube C and contains the optic hull, as following:

Function $\mathbf{V}$ **=Create_Hull(** B_1, B_2 **,p)**

Initialize the cube C around position (**p**)
FOR each camera c
  FOR each δV in C
    //Determine the optic hull for the fisheye camera
    IF $B_c\left(F_c\left(\delta V\right)\right)=1$
    THEN
        Obtain the corresponding indexes (i,j,k) of δV
  with respect to C
      Set $V^c_{i,j,k}=1$
  END
  IF $c>1$
    $V^c = V^c \cap V^{c-1}$
  END
END
$\mathbf{V} = V^c$

Fig. 3 visualizes the experimental setup and the input to the above algorithm. In (a) the geometry of the imaged indoor environment (room) is shown, including the two fisheye cameras 1 and 2 ("Cam 1" and "Cam 2" of figure) at their real positions (0,0,0) and (4,1,0), respectively. One instance of the synthetic 3D model of Fig. 2(a) placed at position **p** = (2.6480, -0.0515) is included in the imaged room (a). (All posi-

tions are given in meters, with respect to camera 1). The rendered frames B_1, B_2 containing the binary silhouettes are shown in (b), (c). Similarly, cube *C* is rendered for cameras 1 and 2 in (d) and (e). The strong deformation that maps straight lines to conic curves is evident.

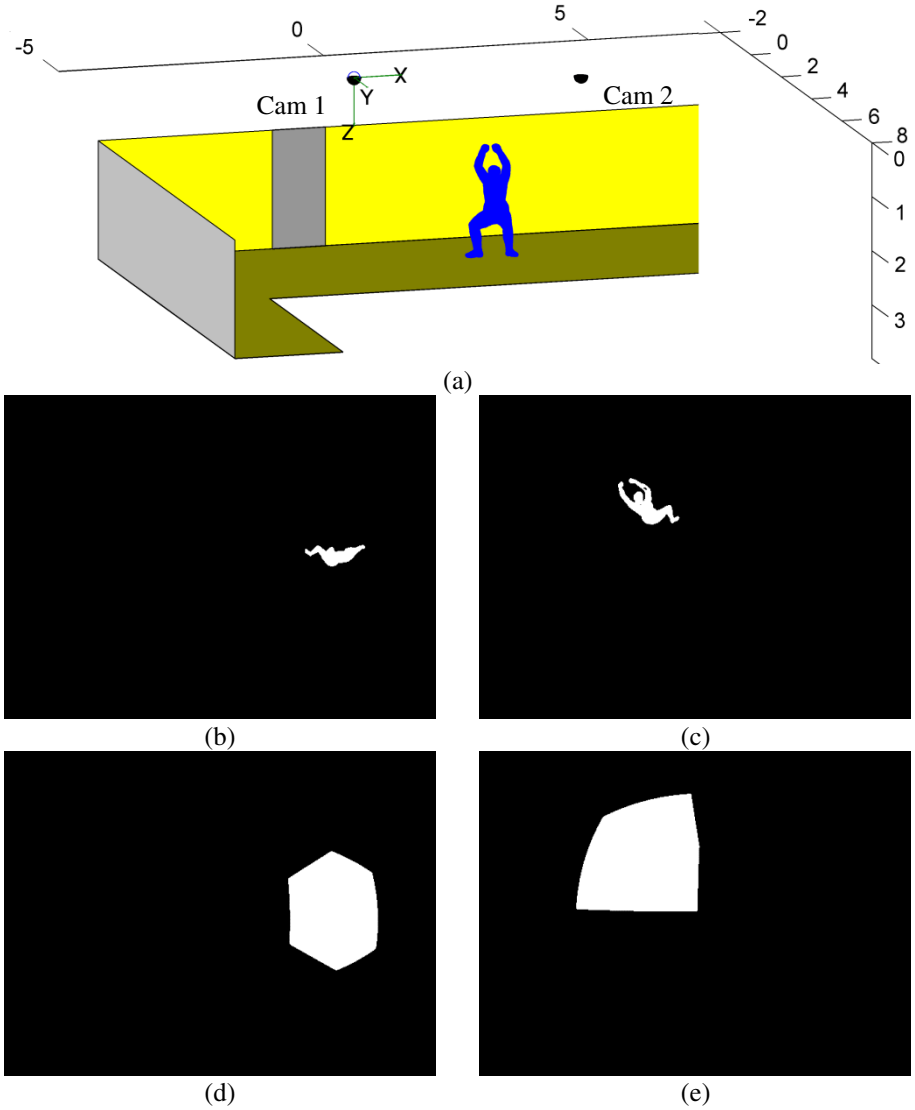


Fig. 3. The geometry of the imaged room including the two fisheye cameras and one instance of the synthetic 3D model of Fig. 2(a) is shown in (a). In (b), (c), the binary silhouettes of the model are rendered (frames B_1, B_2) for cameras 1 and 2, respectively, using the calibration of the cameras. Similarly, cube *C* is rendered for cameras 1 and 2 in (d) and (e), respectively.

3.4 Utilization of Multiple Projections at Different Positions Along a Trajectory

The larger the number of intersections used, the more accurate becomes the recovered visual hull of the imaged object. The use of the fisheye cameras, instead of perspective ones, allows the simultaneous silhouette acquisition at a large number of different positions ($\mathbf{p}_i$) and orientations. Thus, while the human silhouette is walking along a trajectory, many frames can be acquired by both cameras. With multiple views of the same object, we can intersect the generalized cones generated by each image (as discussed in 3.2), to build a more accurate visual hull of the object.

Generating Binary Silhouettes Along a Trajectory. To simulate the human silhouette trajectory, we place the synthetic model, shown in Fig. 2 (a), at several positions along an arbitrary path shown in Fig. 4. At each one of these positions $\mathbf{p}_i$, the tangent vector **t** indicates the orientation (angle θ_i) of the human silhouette by transforming Cartesian to Polar frame of reference. The placement of the model at the different positions and orientations is illustrated in Fig. 5. The algorithm assumes that the orientation of the silhouette can be obtained by using the tangent vector of its motion. More specifically, we defined N=21 different positions $\mathbf{p}_i$ in the room and we employed B-splines to simulate smooth motion and to generate a large number of intermediate points (x, y). In order to calculate orientation θ_i of the tangent vector linear convolution (denoted by "*") with kernel M is performed.

$$M = \frac{1}{2}[1, 0, -1] \tag{2}$$

$$\theta_i = \tan^{-1}\left(\frac{(M * y)_{(i)}}{(M * x)_{(i)}}\right), i = 1, 2, \ldots, N \tag{3}$$

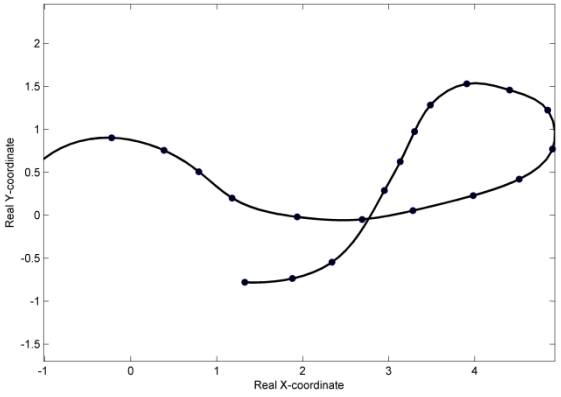


Fig. 4. The path of arbitrary shape used in this study.

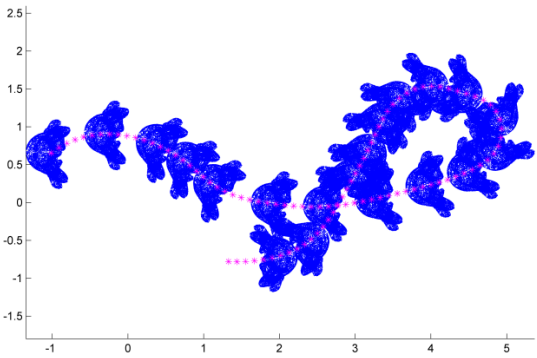


Fig. 5. Synthetic human model at different positions and orientations of the arbitrary shape.

Refining Visual Hull by Multiple Intersections. This algorithm is an extension of the algorithm presented in the previous paragraph, to implement volume reconstruction of a silhouette captured at different times and positions along any path in the imaged room. At each one of the positions i =1,2, …,N we utilized the calibration of the two fisheye cameras to obtain a pair of views (binary silhouettes in frames B_1^i, B_2^i). For each pair of views, function `Create_Hull` is executed. The orientation of resulting volume V_i is restored to the initial one (of the synthetic model) by rotating it round the Z-axis by $-\theta_i$ (Eq.3). The rotated volume is intersected with the current visual hull. Explicitly, let us define the sequences of the binary silhouettes for fisheye cameras c=1,2 for each one of the available positions $\mathbf{p}_i$, with i =1,2,…,N : $\mathbf{B_1} = \{B_1^i\}, \mathbf{B_2} = \{B_2^i\}$.

We present the pseudocode that accepts as input $\mathbf{B_1}, \mathbf{B_2}$ and the sequence of positions **p** and returns as output the refined visual hull.

Function V=Create_Hull_MultiPositions($\mathbf{B_1}, \mathbf{B_2}$,p)
FOR each position index *i*
  p=$\mathbf{p}_i$
  Obtain the tangent vector $\mathbf{t}_i$ and its orientation θ_i
  $\mathbf{V}_i$=Create_Hull(B_1^i, B_2^i **,p**)
  Rotate **V** round the *Z*-axis by angle $-\theta_i$
  IF *i*>1
    $V_i = V_i \cap V_{i-1}$
  END
END

3.5 Multi-pose Reconstruction Using Multiple Synthetic Models

In this subsection, the previously described algorithm is extended so that different poses of human model are recovered. According to our experimental setup, the model

of Fig. 2(a) and (b) are imaged in frames B_1^i, B_2^i alternatively in each $\mathbf{p}_i$ position, where i =1,2, … ,N. Let $\mathbf{P} = \{P_j\}, j = 1,2,...,N_p$ be the set of volumes that have been identified as distinct poses (P_j) by the proposed algorithm, and Bp_c^j the rendered frames of P_j of each fisheye camera c=1,2.

For each pair of views B_1^i, B_2^i acquired at time i, function `Create_Hull` is executed. The orientation of resulting volume V_i is restored to the initial one (of the synthetic model) by rotating it round the Z-axis by $-\theta_i$ (Eq.3).

In order to compare volume V_i with the existing poses $\mathbf{P}$, the pair of binary silhouettes B_1^i, B_2^i is compared to Bp_1^j and Bp_2^j respectively, for all values of j, by calculating their pairwise similarity S as following:

$$S\left(B_c^i, Bp_c^j\right) = \frac{\#\left\{B_c^i \cap Bp_c^j\right\}}{\max\left(\#\left\{B_c^i\right\}, \#\left\{Bp_c^j\right\}\right)} \tag{4}$$

where # indicates the number of pixels of the image in { }. Let J be the set of values of j for which $S\left(B_c^i, Bp_c^j\right)$ is greater than a predefined threshold T, for both cameras (c=1,2):

$$J = \left\{j : S\left(B_c^i, Bp_c^j\right) \geq T, c = 1,2\right\} \tag{5}$$

If no value of j satisfies condition (5) ($J = \varnothing$), a new pose is introduced: $P_{N_p+1} = V_i$. Otherwise, j_{best} is set as the value that maximizes $\sum_c S\left(B_c^i, Bp_c^j\right)$:

$$j_{best} = \arg\max_{j \in J}\left(\sum_c S\left(B_c^i, Bp_c^j\right)\right) \tag{6}$$

and $P_{j_{best}}$ is updated as the intersection of the rotated volume V_i with $P_{j_{best}}$.

Function P=Create_Hull_MultiPositions_MultiPoses($\mathbf{B_1}, \mathbf{B_2}$ **,p)**

```
j=1;
FOR each position index i
  p=p_i
  Obtain the tangent vector t_i with orientation θ_i
  V_i=Create_Hull( B_1^i, B_2^i ,p)
  Rotate V round the Z-axis by angle -θ_i
  IF i==1
    Create new pose P_j
  ELSE
```

```
    Calculate S  //Find similarity P_j , V_i  according to (4)
    IF  S  < T , with c=1,2
      j=j+1
      P_j = V_i  //Create new pose
    ELSE
      P_j = P_j ∩ V_i
      num_j=num_j+1;
    END
  END
END
```

P= P_j

The value of the threshold of Eq.(5) is experimentally determined equal to 0.55. A higher threshold value could affect the result of the method by generating new non-existing poses. On the other hand, a lower value for *T* would cause the intersection of the wrong selected pose (P_j) with the current optic hull (V_i), thus producing inaccurate 3D model. In general, the introduction of a non-existing pose (*j*) is not crucial since it can be overcome by neglecting it if too few (num_j) optic hulls (V_i) have been assigned to it. However, merging a pose with a non-corresponding hull produces a wrong pose, which cannot be recovered a-posteriori.

4 Results

4.1 Calculation of Reconstruction Error

The use of synthetic data allows the accurate quantification of the error in volume reconstruction. As described above, the optic hull is generated in the form of a volumetric model **V**. In order to calulate the displacement error with respect to the 3D human model, which is in the form of point cloud, we apply a simple linear transformation from cloud space to voxel space to the points of the 3D human model. Thus, volume $\mathbf{V_T}$ (of equal dimensions to **V**) is generated. Subsequently, the distance transform (DT) of $\mathbf{V_T}$ is computed in 3 dimensions. The average error (*err*) is computed as the average distance of the non-zero voxels of V from the non-zero voxels of $\mathbf{V_T}$ as following:

$$err = \frac{1}{N} DT\left(V_{\mathrm{T}}\right)_{ijk}, N = \#\left\{(i, j, k) : V_{ijk} > 0\right\} \tag{7}$$

The error derived is expressed in units of δl, as described in 3.3 (which is set equal to 1 cm for the results presented in this work).

4.2 Quantitative Results

The Mobotix Q24 hemispheric camera was used as the fisheye camera, which was installed on the ceiling of the imaged university room. The pixilation of each frame is 480x640.

Single Pose Volume Reconstruction Along Trajectory. The single poses of Fig.2 were placed at different positions and orientations along the path shown in Fig. 4 and 5. The algorithm presented in 3.4 was applied and the resulting optic hulls of Fig.2(a) are presented in Fig. 6 (blue dots) superimposed on the ground truth (3D human model) shown in magenta dots. The reconstruction errors (described in the previous subsection), for all four single poses, are plotted against the number of positions in Fig. 7. As it can be observed, the reconstruction error, in the majority of the executions, approached 2 cm after a few positions.

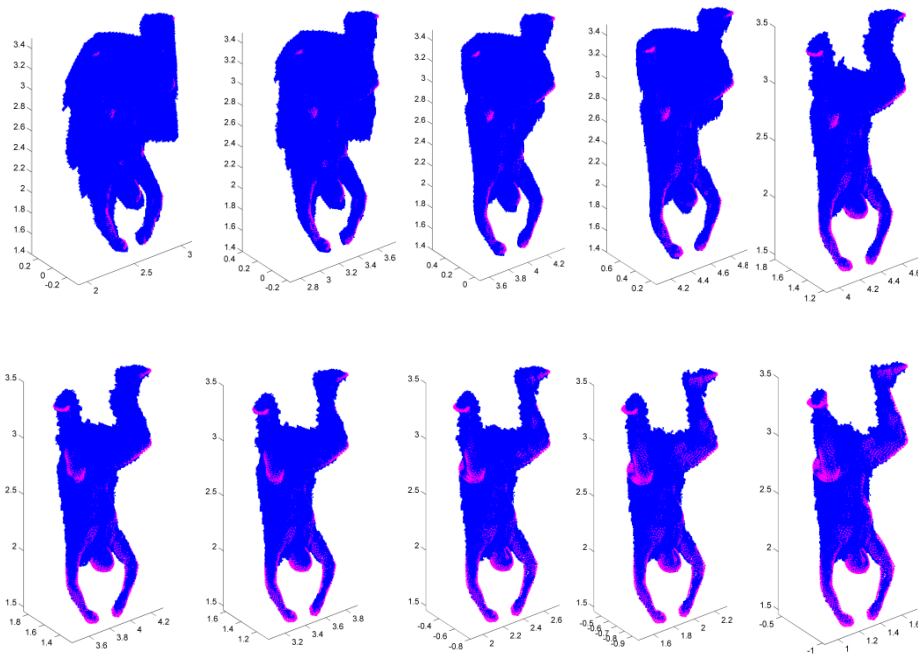


Fig. 6. Evolution of the volumetric model (optic hull) of a single pose, as a function of different positions used. Blue dots indicate reconstructed volume and magenta indicates the ground truth (3D human model).

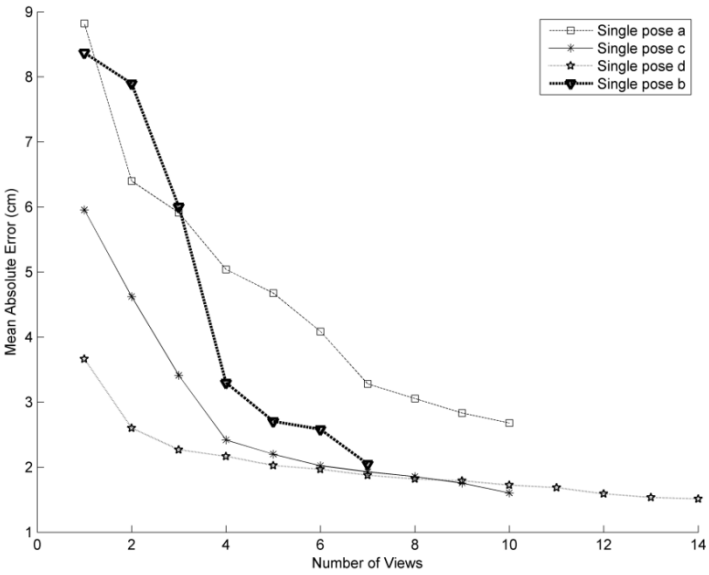


Fig. 7. The evolution of the single pose reconstruction error with the number of positions, for four independent executions using the poses (a), (b), (c) and (d) shown in Fig. 2.

Volume Reconstruction Along Trajectory for Multiple Poses. Two 3D human models, representing two different poses (first line of Fig. 8), were placed at different positions and orientations along the path shown in Fig. 4 and 5. The algorithm presented in 3.5 was applied. The resulting optic hulls for the two poses are presented in the two columns of Fig. 8, respectively. Blue dots indicate the reconstructed volume for each position, superimposed on the ground truth (3D human model) shown in magenta dots. The reconstruction error in cm, calculated as described in the subsection 3.1, is plotted against the number of positions in Fig. 9. As it can be seen, the reconstructed volume obtaines the correct shape after only few positions. In that figure, the error of four different poses is shown, instead of the two we used in this experiment. That result is due to the behavior of algorithm `Create_Hull_MultiPositions_MultiPoses`, described in subsection 3.5. Threshold T used for volume similarity was set to a high value in order to prevent intersection between different poses. The falsely identified new poses do not persist for many positions (iterations of subsection 3.5 algorithm), thus they are easily discarded. Furthermore, they do not affect the reconstruction error and the resulting volumes of the correctly identified poses. The error approached 2 cm for the reconstructed volume of the two poses, after a few positions, despite the increased complexity of the task.

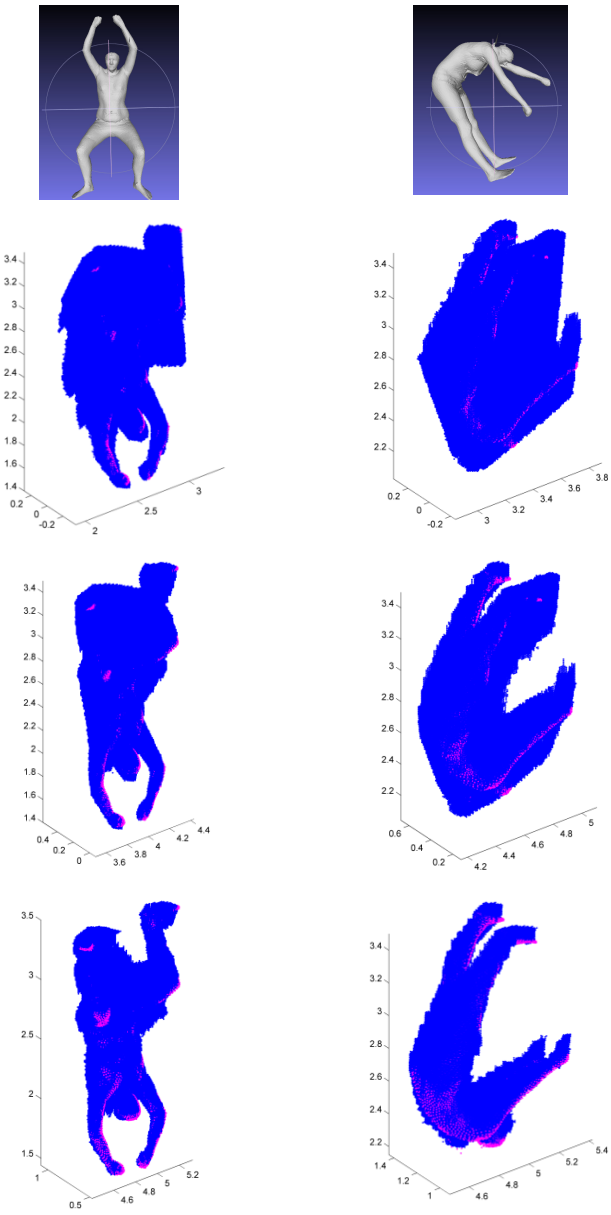


Fig. 8. Evolution of the volumetric model (optic hull) of two distinct poses as a function of different positions used. Blue dots indicate reconstructed volume and magenta indicates the ground truth (3D human model).

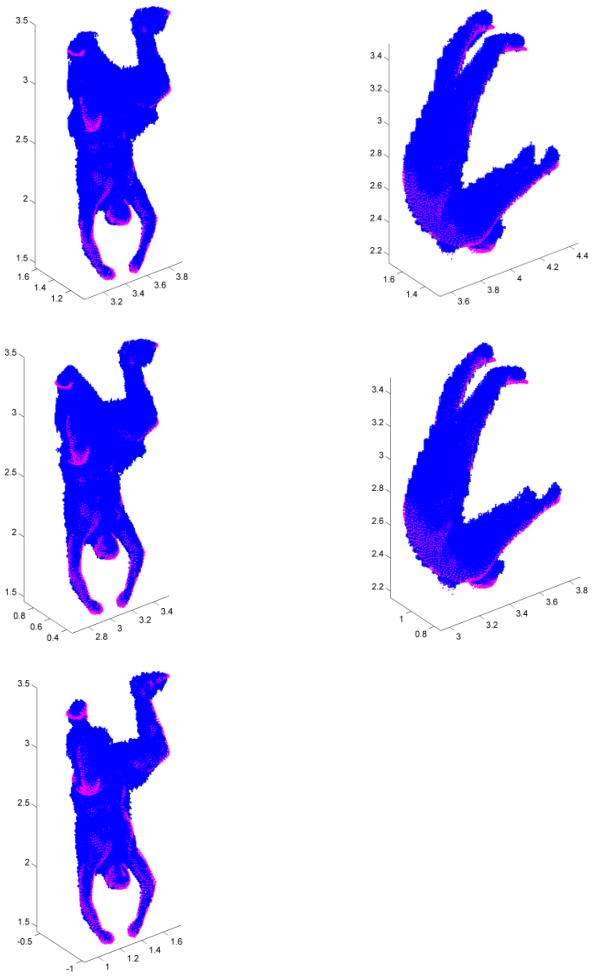


Fig. 8. (*Continued*)

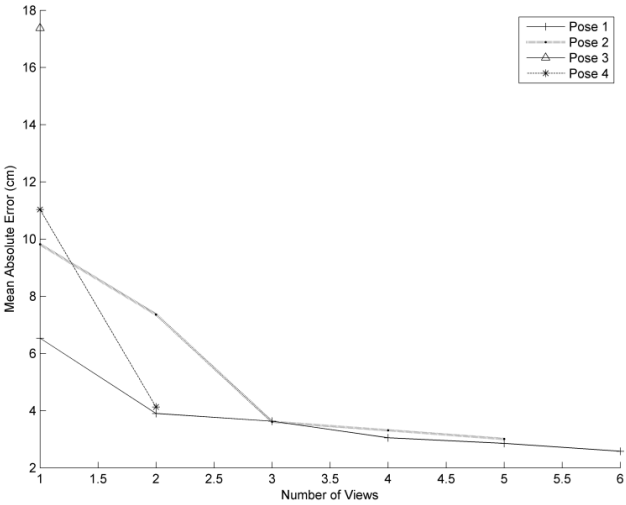


Fig. 9. The evolution of the reconstruction error with the number of positions, in the case of two poses.

5 Conclusion

The use of fisheye cameras for volumetric model reconstruction was investigated in this work, using synthetic data. Initially, a 3D model of a human with known real world coordinates was placed in any location withing the FoV of both fisheye cameras, covering the whole room. The proposed algorithm utilizes the calibration of the fisheye cameras to reconstruct the volumetric model. Synthetic models were used in order to measure the error of reconstructed volume numerically (in cm). This process exploited the placement of the 3D model at different positions and orientations. Finally, the recovery of two poses that alternate between positions along an arbitrary trajectory is also investigated.

Quantitative results show the usefulness of the fisheye cameras, since accurate shape (volume) reconstruction of more than one poses of an object is possible with a minimum number of cameras, by using silhouette views at many positions and orientations along any path.

The generation of volumetric models by acquiring silhouettes, utilizing person movements along any path, is a seamless procedure that may find many applications. For instance, it can be further used to recognise an object or to identify the posture of a moving object of interest. Using silhouettes to reconstruct shape is a technic able to be used in smart homes, for surveillance or assisted living, in games, for avatar generation, or even in fashion, for testing design or demonstrating creations.

Volume reconstruction execution time was approximately equal to 1 second, for each position, using a volumetric cube C of 1.5×1.5×2.2 meters, with an elementary cube δV of 1×1×1 cm, using Matlab on an Intel(R) Core i5-2430 CPU @ 2.40 GHz Laptop with 4 GB Ram, under Windows 7 Home Premium.

Further work is required using real data to test the effect of calibration, as well as of the estimation of position and orientation, to the reconstruction of the optic hull of each pose of the object.

Acknowlegments. The authors would like to thank the European Union (European Social Fund ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: \Thalis\ Interdisciplinary Research in Affective Computing for Biological Activity Recognition in Assistive Environments for financially supporting this work.

References

1. Kemmotsu, K., Tomonaka, T., Shiotani, S., Koketsu, Y., Iehara, M.: Recognizing human behaviors with vision sensors in a Network Robot System. In: Proceedings of IEEE Int. Conf. on Robotics and Automation, pp. 274–1279 (2006)
2. Zhou, Z., Chen, X., Chung, Y., He, Z., Han, T.X., Keller, M.: Activity Analysis, Summarization and Visualization for Indoor Human Activity Monitoring. IEEE Trans. on Circuit and systems for Video Technology 18(II), 1489–1498 (2008)
3. Mei, C., Rives, P.: Single view point omnidirectional camera calibration from planar grids. In: IEEE International Conference on Robotics and Automation, pp. 3945–3950. IEEE, Rome (2007)
4. Li, H., Hartley, R.I.: Plane-based calibration and auto-calibration of a fish-eye camera. In: Narayanan, P.J., Nayar, S.K., Shum, H.-Y. (eds.) ACCV 2006. LNCS, vol. 3851, pp. 21–30. Springer, Heidelberg (2006)
5. Shah, S., Aggarwal, J.: Intrinsic parameter calibration procedure for a high distortion fish-eye lens camera with distortion model and accuracy estimation. Pattern Recognition 29(11), 1775–1788 (1996)
6. Wei, J., Li, C.F., Hu, S.M., Martin, R.R., Tai, C.L.: Fisheye Video Correction. IEEE T. on Visualization and Computer Graphics 18(10), 1771–1783 (2012)
7. Kutulakos, K., Seitz, S.: A theory of shape by space carving. International Journal of Computer Vision 38(3), 199–218 (2000)
8. Potmesil, M.: Generating Octree Models of 3D Objects from their Silhouettes in a Sequence of Images. CVGIP 40, 1–29 (1987)
9. Tsuneo, Y., Koeda, M., Fujimoto, H.: Shape recognition and grasping by robotic hands with soft fingers and omnidirectional camera. In: IEEE International Conference on Robotics and Automation, ICRA, pp. 299–304. IEEE (2008)
10. Svoboda, T., Pajdla, T., Hlaváč, V.: Epipolar geometry for panoramic cameras. In: Burkhardt, H.-J., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1406, pp. 218–231. Springer, Heidelberg (1998)
11. Sturm, P.: Mixing catadioptric and perspective cameras. In: OMNIVIS Third Workshop on Omnidirectional Vision, pp. 37–44. IEEE Computer Society, Copenhagen (2002)
12. Delibasis, K., Plagianakos, V., Maglogiannis, I.: Refinement of human silhouette segmentation in omni-directional indoor videos. Computer Vision and Image Understanding 128, 65–83 (2014)

13. Rufli, M., Scaramuzza, D., Siegwart, R.: Automatic detection of checkerboards on blurred and distorted images. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008), Nice, France, pp. 3121–3126 (2008)
14. Hasler, N., Ackermann, H., Rosenhahn, B., Thormahlen, T., Seidel, H.P.: Multilinear pose and body shape estimation of dressed subjects from image sets. In: IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2010), pp. 1823–1830 (2010)
15. http://resources.mpi-inf.mpg.de/scandb/
16. Matusik, W., Buehler, C., Raskar, R., Gortler, S.J., McMillan, L.: Image-based visual hulls. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2000), pp. 369–374. ACM Press/Addison-Wesley Publishing Co., New York (2000)