Computer Vision and Image Understanding 128 (2014) 65–83

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Refinement of human silhouette segmentation in omni-directional indoor videos ☆

K.K. Delibasis [a], V.P. Plagianakos [a], I. Maglogiannis [b,*]

[a] Department of Computer Science and Biomedical Informatics, University of Thessaly, Lamia 35100, Greece
[b] Department of Digital Systems, University of Piraeus, Piraeus, Greece

ARTICLE INFO

Article history:
Received 6 December 2013
Accepted 17 June 2014
Available online 1 July 2014

Keywords:
Video segmentation
Human activity detection
Mathematical model of fisheye camera
Geometric reasoning

ABSTRACT

In this paper, we present a methodology for refining the segmentation of human silhouettes in indoor videos acquired by fisheye cameras. This methodology is based on a fisheye camera model that employs a spherical optical element and central projection. The parameters of the camera model are determined only once (during calibration), using the correspondence of a number of user-defined landmarks, both in real world coordinates and on a captured video frame. Subsequently, each pixel of the video frame is inversely mapped to the direction of view in the real world and the relevant data are stored in look-up tables for fast utilization in real-time video processing. The proposed fisheye camera model enables the inference of possible real world positions and conditionally the height and width of a segmented cluster of pixels in the video frame. In this work we utilize the proposed calibrated camera model to achieve a simple geometric reasoning that corrects gaps and mistakes of the human figure segmentation, detects segmented human silhouettes inside and outside the room and rejects segmentation that corresponds to non-human activity. Unique labels are assigned to each refined silhouette, according to their estimated real world position and appearance and the trajectory of each silhouette in real world coordinates is estimated. Experimental results are presented for a number of video sequences, in which the number of false positive pixels (regarding human silhouette segmentation) is substantially reduced as a result of the application of the proposed geometry-based segmentation refinement.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

The field of human activity monitoring based on cameras has gained significant interest during the last years in the context of developing ambient assisted living environments. A number of approaches exist in the research literature, based either on 3D human models, or on local image descriptors, exploiting both spatial and temporal information. Detailed reviews of this field exist in [1–4]. The majority of the approaches to the problem of vision-based recognition of human motion and action, utilize descriptors from segmented human silhouettes. For instance in [3] 14 publications are listed that use silhouettes as abstraction level for vision based human motion capture. In the survey presented in [1], it is stated that segmented silhouettes are used for human motion detection using 3D human models of image descriptors, but segmentation artifacts limit the performance of the corresponding methods. In [2] the use of silhouettes for human action recognition using image descriptors, as well as space volumes, is surveyed through a number of listed works. These algorithms perform well with datasets that include very short videos of single humans performing a single task at each time. Examples of these datasets include the INRIA XMAS [5], the Weizmann [6], the KTH [7], the CMU MoBo database [8] and the Human EVA [9]. In these video segments, no other action is usually visible in the background, thus the segmentation of the human silhouettes is rather easy, defect free and unambiguous. Only few databases exist that contain challenging video sequences (changing illumination in background, existence of multiple persons close to each other or interacting), such as the HOHA database [10].

It is obvious from the above discussion that the improvement of the human silhouettes segmentation will result in improving the quality of human motion/action recognition. The contribution of this work is a methodology that isolates the segmented human silhouettes from fisheye video sequences inside a designated area, from other irrelevant segmentation and eliminates artifacts and outliers. The video data used in this research are acquired indoors

☆ This paper has been recommended for acceptance by Nikos Paragios.

* Corresponding author. Address: University of Piraeus, Department of Digital Systems, Grigoriou Lampraki 126, PC 18532 Piraeus, Greece. Fax: +30 2104142517.
E-mail addresses: kdelibasis@yahoo.com (K.K. Delibasis), vpp@ucg.gr (V.P. Plagianakos), imaglo@unipi.gr (I. Maglogiannis).

http://dx.doi.org/10.1016/j.cviu.2014.06.011
1077-3142/© 2014 Elsevier Inc. All rights reserved.

from a fixed fisheye camera, installed at the ceiling of the living environment. The proposed algorithm uses a novel fisheye camera model that enables reasoning based on real world geometry to correct and enhance the segmentation output of the human figures. The proposed algorithm does not make use of models of the required object (human silhouette), or local image features.

The input of the proposed algorithm is the result of the initial video segmentation based on background modeling and subtraction. Several methodologies for background modeling exist in literature. For instance in [11] the background model is simply defined as the previous frame and global thresholding is employed to extract the foreground. This method is very simple to implement, but it is prone to a number of segmentation errors. Background can be modeled by median filtering [12] of a predefined number of last frames that are held in a buffer. This approach requires significant computational and memory resources and cannot be executed in real time. In order to alleviate the increased computational requirements, a class of recursive background modeling algorithms is proposed in literature. These algorithms use an incremental update of the background. Simple and efficient members of this class of algorithms are the approximation of the median filtering method [13] and the running Gaussian average method [14]. A comprehensive review of background modeling algorithms for foreground detection is presented in [15]. A popular methodology is the Mixture of Gaussians, initially described for video sequences by Stauffer and Grimson [16], according to which, the values of each pixel are modeled as a lineal combination of weighted Gaussian probability distributions. However, this method is also computationally expensive. In this work we have performed video segmentation using the illumination-sensitive background modeling approximation, as originally described in [17] and modified in [18], although any other video segmentation algorithm can be employed. This algorithm was selected due to its simplicity and its efficient handling of illumination changes.

The video sequences used in this work are captured by a hemispheric camera, also known as omni-directional, or fisheye camera with 180° field of view (FoV). The use of this type of cameras is increasing in robotic and in video surveillance applications [19,20], due to the fact that they allow constant monitoring of all directions with a single camera. In [21–23] the calibration of fisheye camera is reported using high degree polynomials to emulate the strong deformation introduced by the fisheye lens, radial and/or tangential. In [24] the authors present a methodology for correcting the distortions induced by the fisheye lens. In [25,26] a well established calibration for omni-directional cameras is proposed, which utilizes a standard chess pattern imaged at arbitrary orientations, without requiring point input by the user. In this paper we present a very efficient camera model that extends our previously proposed fisheye model that used only 3 parameters [27] with exhaustive search calibration. Subsequently, we utilize the proposed inverse fisheye model to refine the segmentation of moving humans and eliminate non-human activity, such as window reflections, sudden changes of illumination, small objects, moving doors, etc., as well as human silhouette outside the designated room. The position of a human silhouette (or any other segmented foreground object) is estimated accurately under the assumption that its base (area of the surface touching the floor) is small, such as a standing, walking or sitting the human. Finally, the refined segmented silhouettes are uniquely labeled using spatiotemporal information concerning both real world position and RGB appearance and the trajectory of each silhouette in real world coordinates is also estimated.

The rest of the paper is organized as follows. In Section 2, the overall architecture is presented, while the forward and inverse modeling of the fisheye camera is also described. The proposed methodology for the refinement of the human silhouette segmentation using reasoning based on the geometric relation between the binary connected components is presented in Section 2.4. Initial experimental results are presented in Section 3, whereas the proposed algorithm and the future work are discussed in the last concluding Section 4.

2. Materials and method

2.1. Overall description and block diagram of the proposed methodology

The main characteristic of the fisheye camera is the ability to cover a field of view of 180°. The proposed methodology is based on a parametric model of image formation, so that any real-world point (x,y,z) can be associated with a frame pixel (i,j). Furthermore a pixel (i,j) in the video frame can be associated with the direction of view, defined by two angles: the azimuth θ and the elevation φ. The parameters of the fisheye camera model are determined only once (calibration), using manually inserted reference points. The resulting association between pixels and azimuth θ and the elevation φ is stored in look-up tables for quick utilization, as shown in the right side of the block diagram of Fig. 1. The part of the proposed algorithm that is executed in real time includes the segmentation of moving objects (black and white thumbnail image at the left of Fig. 1), its refinement based on the reasoning presented in Section 2.4 and the unique labeling of the silhouettes (in case of multiple silhouette segmentation) as described in Section 2.5. The refined human silhouette segmentation is encoded in color (red for rejected segmentation, other colors for human silhouette segmentation – see thumbnail image at the left part of Fig. 1). Fig. 1 illustrates the modules of the proposed methodology.

2.2. Forward fisheye camera model

The fisheye model M can be written in the general form

$$(j,i) = M(x,y,z) \tag{1}$$

where (j,i) are pixel coordinates in the video frame and (x,y,z) are real world coordinates of the imaged point. In the following subsection, we will describe the inverse fish-eye camera model, i.e. given a pixel (j,i) in the video frame, the direction of view is obtained, defined in spherical coordinates by two angles: the azimuth θ and the elevation φ (Fig. 2):

$$(\theta,\varphi) = M_1(i,j). \tag{2}$$

The definition of a model for the fisheye camera is based on the physics of image formation, as described in [28,29] and demonstrated in [30]. The model consists of:

- A spherical element of arbitrary radius R_0 with its center at $K(0,0,z_{sph})$.
- The plane of the CMOS sensor defined as passing through $(0,0,z_{plane})$ and by its unit length normal vector $\mathbf{n}$.

For any point P with real world coordinates (x,y,z), we determine the intersection Q of the line KP with the spherical optical element of the fisheye lens. The point P is imaged at the central projection (x_{im}, y_{im}) of Q on the image plane, using the O $(0,0,0)$ as center of projection, assuming that the installation of the camera is such that the imaging plane (i.e. the image sensor) is horizontal and the axis of the spherical lens is not misaligned. Thus, it becomes obvious that all real world points that lie on the KP line are imaged at the same point (x_{im}, y_{im}) in the image plane. The KP line is uniquely defined by its azimuth and elevation angles, θ and φ, respectively. The concept of the fisheye geometric model is shown in Fig. 2.

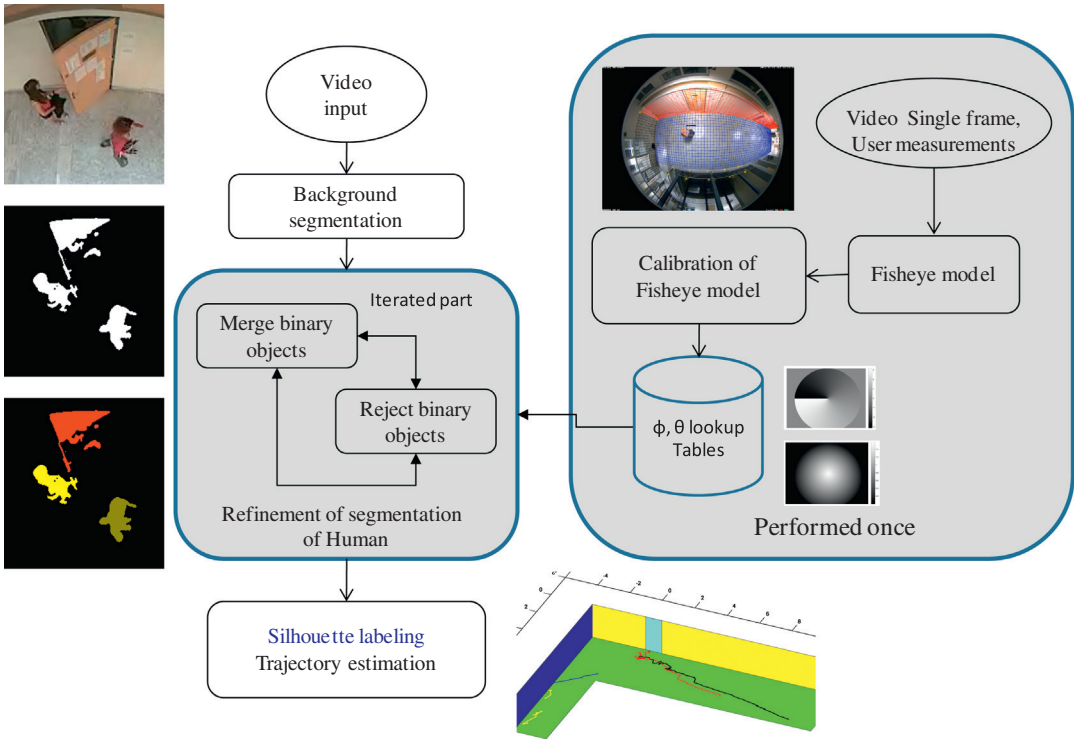


Fig. 1. The overall architecture of the proposed algorithm for the refinement of Human Silhouette segmentation from omni-directional video sequences. Note the segmented video frame at the left (black and white image) and the refinement of the segmentation, encoded in color as following red: rejected segmentation, other than red colors corresponding to different human silhouettes inside the room. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

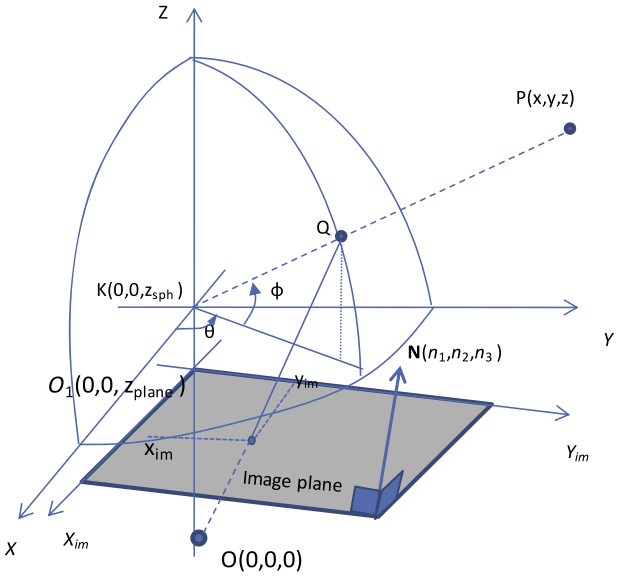


Fig. 2. The geometry of the proposed fisheye camera model.

The fisheye camera has no moving parts. Therefore, the relation between z_{sph} and z_{plane} defines the formation of the image. Let us set z_{plane} to an arbitrary value, less than R_0 and define $z_{sph} = pz_{plane}$, where p is the primary parameter of the fisheye model. To account for possible lens misalignments with respect to the camera sensor that could introduce imaging deformations on the imaged frame [23], we introduce two extra model parameters: the X and Y position of the center of spherical lens K $(x_{sph}, y_{sph}, z_{sph})$ with respect to the optical axis of the camera. Now the camera model parameters consist of $\mathbf{n}$, p, x_{sph}, and y_{sph}. Fig. 2 shows the geometry of the fish-eye camera model for $x_{sph} = 0$ and $y_{sph} = 0$.

The position of any point of the line segment KP, thus Q as well, is given by

$$(Q_x, Q_y, Q_z) = (\lambda(x - x_{sph}), \lambda(y - y_{sph}), \lambda(z - z_{sph})), \tag{3}$$

where λ is a parameter in range $[0,1]$. If we insert (3) into the equation of the spherical optical element, we obtain:

$$\begin{aligned}&(\lambda(x - x_{sph}) - x_{sph})^2 + (\lambda(y - y_{sph}) - y_{sph})^2 \\ &+ (\lambda(z - z_{sph}) - z_{sph})^2 - R_0^2 \\ &= 0.\end{aligned} \tag{4}$$

The parameter λ that defines the position of Q may be obtained by solving Eq. (4) and accepting the solution in the allowed range $[0,1]$. In order to generalize the fish-eye model, we assume that the plane of the sensor is not the XY plane, but it is defined as passing through point $(0,0,z_{plane})$ and being normal to vector $\mathbf{n} = (n_1, n_2, n_3)$. The components of $\mathbf{n}$ are included as new parameters to the model of the fish-eye camera that was originally described in [27].

Then, we calculate the central projection (x_{im}, y_{im}, z_{im}) of Q on the image plane, given by:

$$(mQ_x, mQ_y, mQ_z), \tag{5}$$

where the parameter m is given by: $m = \frac{z_{plane} n_3}{Q_x n_1 + Q_y n_2 + Q_z n_3}$.

Finally, the coordinates of the projection point (x_{im}, y_{im}, z_{im}) can be obtained by geometrically transforming the central projection point, in homogeneous coordinates, so that the vector $\mathbf{n}$ normal to the sensor, becomes parallel to the Z axis:

$$(x_{im}, y_{im}, z_{im}, 1)^{\mathrm{T}} = \mathbf{A}(mQ_x, mQ_y, mQ_z, 1)^{\mathrm{T}}, \tag{6}$$

where $\mathbf{A}$ is the matrix that transforms vector (n_1, n_2, n_3) onto the Z axis:

$$\mathbf{A}=\begin{pmatrix} \frac{\lambda}{|\mathbf{n}|} & \frac{-n_1 n_2}{\lambda|\mathbf{n}|} & \frac{-n_1 n_2}{\lambda|\mathbf{n}|} & 0 \\ 0 & \frac{\lambda}{n_3} & \frac{-n_2}{n_3} & 0 \\ \frac{n_1}{|\mathbf{n}|} & \frac{n_2}{|\mathbf{n}|} & \frac{n_3}{|\mathbf{n}|} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \lambda=\sqrt{n_2^2+n_3^2}.$$

when $x \to \pm\infty$ then $Q_x \to R_0 \pm x_{sph}$ (the same holds for the y coordinate as well). Thus, any point P with real world coordinate $z > z_{sph}$, will be imaged on the image plane at position (x_{im}, y_{im}), which is bounded as following:

$$x_{im\_\min} \leqslant x_{im} \leqslant x_{im\_\max} \text{ and } y_{im\_\min} \leqslant y_{im} \leqslant y_{im\_\max} \tag{7}$$

where

$$\begin{aligned} x_{im\_\max} &= \lambda_+(x_{sph}+R_0) \\ x_{im\_\min} &= \lambda_-(x_{sph}-R_0) \end{aligned}, \quad \lambda_\pm = \frac{n_3 z_{pl}}{(x_{sph}\pm R_0)n_1+(y_{sph}\pm R_0)n_2+z_{sph}n_3}$$

The image pixel position (i,j) that corresponds to the projection on the image plane (x_{im}, y_{im}) is calculated by a simple linear transform:

$$\begin{aligned} j &= x_{im}\tfrac{R_{FoV}}{f_1}+CoD_x, f_1 = \frac{x_{im\_\max}-x_{im\_\min}}{2} \\ i &= y_{im}\tfrac{R_{FoV}}{f_2}+CoD_y, f_2 = \frac{y_{im\_\max}-y_{im\_\min}}{2} \end{aligned}, \tag{8}$$

where (CoD_x, CoD_y) is the center of distortion pixel that corresponds to elevation $\varphi = \pi/2$ and R_{FoV} is the radius of the circular field of view (FoV), as it is explained in the next paragraph.

In the case of fisheye lens, [31] suggests that the CoD is located as the center of the circular field-of-view. In our case this is a very practical approach, since almost the whole field-of-view is imaged (full frame imaging). We therefore apply the Canny edge detector [32], using a standard deviation equal to 2 to detect the stronger edges in the image, which are the edges of the circular field of view. Then, a simple least squares optimization to obtain the CoD and the radius of the FoV is employed. The resulting CoD and radius of FoV are shown in Fig. 3. This process is applied only once after the installation of the camera.

2.2.1. Calibration of fisheye camera model

In order to the proposed fisheye camera model in our algorithm, we need to determine the values of the unknown parameters $(p, x_{sph}, y_{sph}, \mathbf{n})$. This is done during calibration. Initially, we provide the position of N_p landmark points $\{(X_{im}^k, Y_{im}^k)\}, k = 1, 2, \ldots, N_p$ on one video frame. Although a minimum required number of points is not defined, in this work we used $N_p = 18$ prominent landmarks that are distributed over the field of view of the camera. The real world coordinates of these landmark points $\{(x_{real}^k, y_{real}^k, z_{real}^k)\}$ were also measured, where $k = 1, 2, \ldots, N_p$, with respect to the reference system, (superscripts do not indicate powers). According to Eq. (1), the expected pixel position of the landmark points in the video frame, given the current model parameters are:

$$(x_{im}^k, y_{im}^k) = M(x_{real}^k, y_{real}^k, z_{real}^k; p, x_{sph}, y_{sph}, \mathbf{n}). \tag{9}$$

The values of the model parameters are obtained by minimizing the error between the expected and the observed pixel coordinates of the landmark points:

$$(p, x_{sph}, y_{sph}, \mathbf{n}) = \arg\min_{p, x_{sph}, y_{sph}, \mathbf{n}} \left(\sum_{k=1}^{N_p} \left((X_{im}^k - x_{im}^k)^2 + (Y_{im}^k - y_{im}^k)^2 \right) \right). \tag{10}$$

Due to the large number of parameters and the complexity of the objective function, we employed a recently proposed variant of the Differential Evolution (DE) presented in [33]. DE has been designed as a stochastic parallel direct search method that typically requires few, easily chosen, control parameters. Experimental results have shown that DE has good convergence properties and outperforms other well-known evolutionary algorithms [34,35]. A population of individuals is randomly initialized in the optimization domain. Subsequently, the individuals are iteratively evolved in order to explore the search space and locate the optima of the objective function. Note that the number of the individuals in the population remains constant throughout the evolution.

At each iteration, which is called *generation*, new vectors (*offsprings*) are produced by a combination of randomly chosen vectors. This operation is referred to as *mutation*. At the next step, the *recombination* operation mixes the offspring vectors with another predetermined vector – the *target* vector. This yields the so-called *trial* vector. The trial vector replaces the target vector if and only if it yields a reduction in the value of the objective function. This last operation can be referred to as *selection*. Performing the mutation, recombination and selection operations for all the population members constitutes a single iteration of the DE algorithm.

We allow p to vary from 0.5 to 1.5, x_{sph}, y_{sph} to vary from in the range of $[-R_0/4, R_0/4]$ and each coordinate of the normal vector $\mathbf{n}$ in the range $[0,1]$. Despite $\mathbf{n}$ being a unit vector ($|\mathbf{n}| = 1$), its components n_1, n_2 and n_3 are treated as independent parameters by the optimizer and vector normalization is applied before calculating Eqs. (5)–(7). In this manner, we avoid the alternative of using only two vector components as independent parameters and calculating the third component using the unit-length, since this approach requires imposing penalties to the objective function in case of vector non-computability. The model parameters are obtained in just a few seconds using the Matlab programming environment in an average laptop computer. The resulting calibration of the fisheye model is shown in Fig. 4, where a virtual grid of points is laid on the floor and on the two walls of the imaged room where the camera is installed.

The proposed fisheye calibration approach is compared to a well established toolbox for omnidirectional image calibration [25,26], which works without point input by the user, using a standard chess pattern, imaged at arbitrary orientations. Fig. 4 shows the resulting calibration using the proposed model and the method in [26]. The landmark points defined by the user are shown as circles and stars mark their expected position on the frame. A selected region of the frame is magnified in (b) and (d) for comparison. The mean displacement error for the proposed

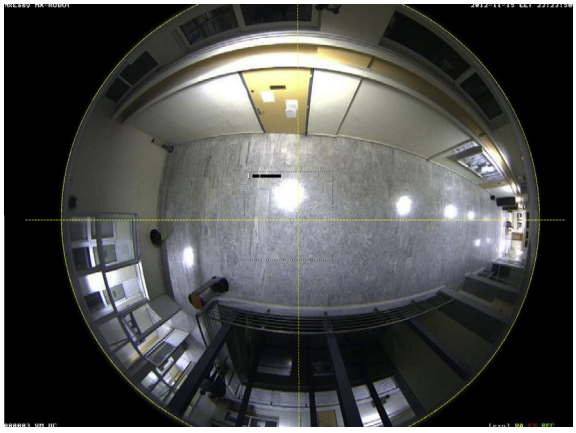


Fig. 3. A typical frame of the fisheye camera. The center of distortion CoD and the radius of the FoV have been marked, to be used in subsequent calculations.

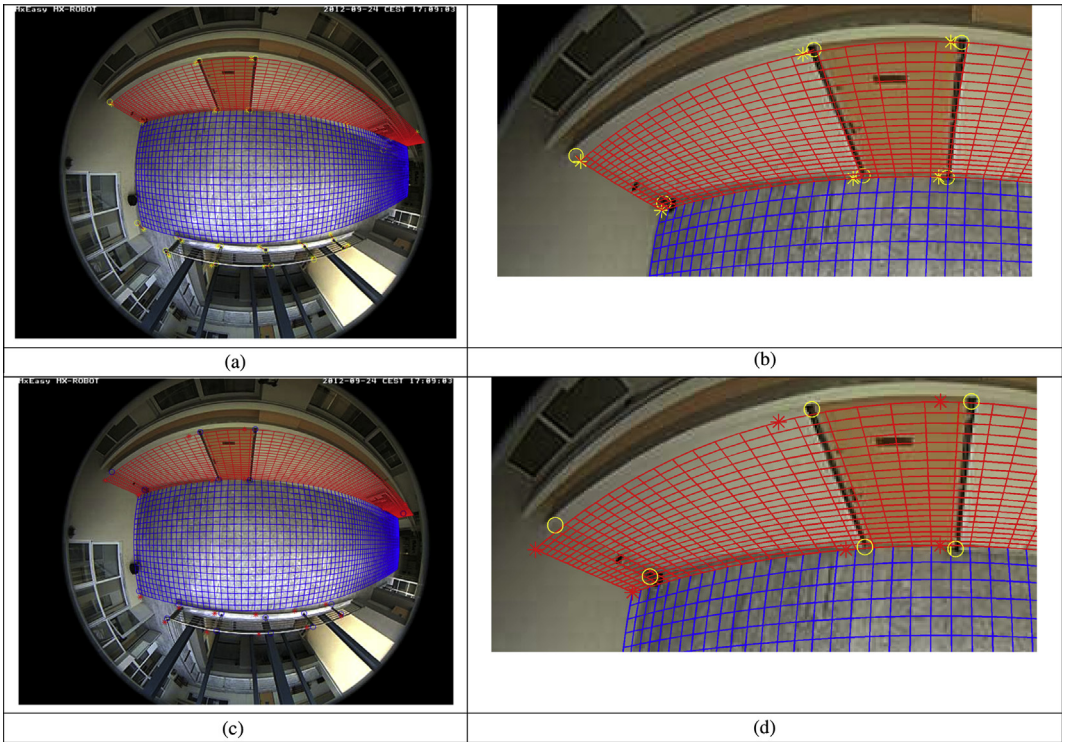


Fig. 4. Visualization of the resulting fisheye model calibration using the proposed fisheye model (a), (b) and the calibration method described in [25,26], (c) and (d). The landmark points defined by the user are shown as circles and their rendered position on the frame marked by stars.

Fig. 5. Evaluation of calibration accuracy using a number of points with known 3D positions identified by the user on the video frame as "+". The determined pixel locations using the proposed calibration are shown as "•".

model is lower than the mean displacement error achieved by [26], considering the N_p landmark points (7.1 versus 9.2 pixels). It should be noted that this operation is only performed once after the initial installation of the fisheye camera and it does not need to be repeated in real time.

The proposed calibration was further assessed by calculating the displacement error (pixels) for a new set of 18 points on the floor of the imaged room, as shown in Fig. 5, different than the N_p (=18) points used for calibration in Eqs. (9) and (10). The resulting error using the proposed calibration and the calibration in [25,26] are given in Table 1. The maximum positional error is approx. 10 pixels, slightly lower than the maximum error of 11 pixels achieved by the calibration of [25,26].

Table 1
The minimum, maximum, median and mean positional error in pixels.

Calibration method	Max	Median	Min	Mean
Proposed	10.25	5.94	1.03	5.80
[25,26]	11.02	6.83	1.19	6.46

2.3. Inverse model of the fisheye camera – azimuth and elevation look-up tables

As with any kind of projective transformation, only the line of view can be recovered by the inverse transform, since any point on this line of view will be projected on the same image pixel. To use the model of the fisheye camera for refinement of video segmentation of human silhouettes, we need to utilize the elevation θ and azimuth φ of the line of view for each segmented pixel. In this subsection, we describe the inverse fish-eye camera model, i.e. given a pixel (j,i) in the video frame, the direction of view is obtained, defined by two angles: the azimuth θ and the elevation φ (Fig. 2), as described in (2). Using Eq. (5), the position of the pixel on the camera sensor is calculated:

$$(x_{im}, y_{im}) = \frac{(f_1, f_2)}{R_{FOV}}((j,i) - (CoD_x, CoD_y)). \tag{11a}$$

The Z-coordinate of the image pixel on the sensor plane is given by:

$$z_{im} = z_{plane} - \frac{x_{im}n_1 + y_{im}n_2}{n_3}. \tag{11b}$$

The intersection Q of the spherical optical element with the line defined by $O(0,0,0)$ and (x_{im}, y_{im}) is determined, as

$$(Q_x, Q_y, Q_z)^{\mathrm{T}} = m\mathbf{A}^{-1}(x_{im}, y_{im}, z_{plane})^{\mathrm{T}}, \tag{12}$$

where the parameter m is determined by requiring that Q lies on the spherical optical element:

$$m^2(x_{im}^2 + y_{im}^2 + z_{plane}^2) - 2(x_{im}x_{sph} + y_{im}y_{sph} + z_{plane}z_{sph})m + x_{sph}^2 + y_{sph}^2 + z_{sph}^2 - R_0^2 = 0. \quad (13)$$

The required θ and φ are obtained by converting the Cartesian (Q_x, Q_y, Q_z) to spherical coordinates:

$$\varphi = \cos^{-1}\left(\frac{Q_z - z_{sph}}{R_0}\right), \qquad \theta = \sin^{-1}\left(\frac{Q_y - y_{sph}}{\sqrt{(Q_x - x_{sph})^2 + (Q_x - x_{sph})^2}}\right). \quad (14)$$

The above process is executed only once, after the calibration of the fisheye camera model and the resulting values for the θ, and φ parameters for each frame pixel are stored in two look-up tables, of size equal to a single video frame. The look-up tables for the azimuth θ and the elevation φ are shown in Fig. 6(a) and (b), respectively. As expected, the azimuth obtains values in $[-\pi, \pi]$, whereas the elevation obtains values in $[0, \pi]$, with the maximum value at the CoD pixel of the frame.

2.4. Refining and optimizing human silhouette segmentation with geometric reasoning

In this subsection we discuss the height and width calculation of segmented objects and describe the employed metric of proximity of different segmented binary objects in the fish-eye frame. The detection of non-plausible segmentation and the proposed algorithm of geometric reasoning is also outlined.

2.4.1. Calculation of height and position

Let us suppose that an object in real world, shown in Fig. 7(a) as observed in a view from above, is segmented in a video frame. Using the azimuth look up table, its minimum, maximum and average azimuth angle can be easily retrieved. Similarly, its minimum and maximum elevation angle can be easily estimated, using the elevation look-up table, as shown in Fig. 7(b).

Assuming that the segmented object is the silhouette of a person touching the floor with a surface of limited area (such as a standing, walking or sitting human), the position on the floor (x_{real}, y_{real}) can be calculated, as follows. Let φ_{max} and θ_{avg} be the maximum elevation and average azimuth of the pixel of the binary object in the segmented frame that corresponds to the person (Fig. 7(a and b)). Since $z_{\max}$ is the z coordinate of the floor with respect to the system of reference, the intersection of the line originating from the camera, defined by $(\theta_{avg}, \varphi_{\max})$ with the floor plane $(z = z_{\max})$ can be easily obtained:

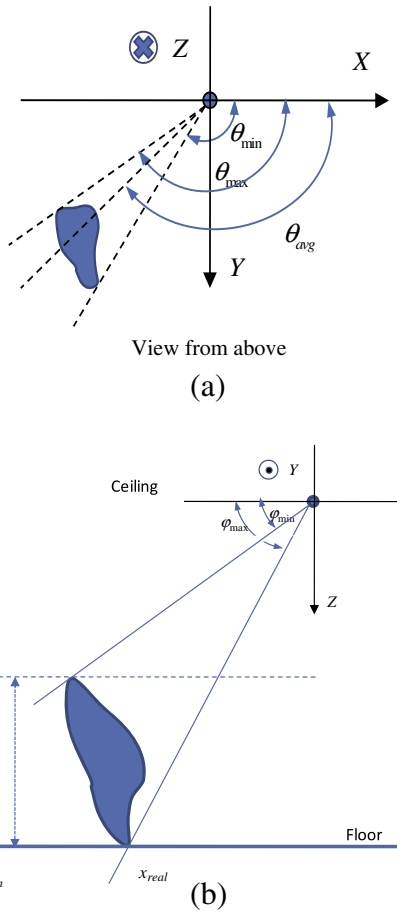


Fig. 7. An object in the real world being imaged by the fisheye camera. In (a) the view from above is shown to clarify the definition of the minimum and maximum azimuth. In (b) a side view is provided for the definition of minimum and maximum elevation.

$$x_{real} = \frac{z_{\max}}{\sin\varphi_{\max}}\cos\varphi_{\max}\cos\theta_{avg}, \qquad y_{real} = \frac{z_{\max}}{\sin\varphi_{\max}}\cos\varphi_{\max}\sin\theta_{avg}. \quad (15)$$

Let us assume further that the person is standing and its position is not directly below the fisheye camera, (equivalently $\varphi_{\min} < \frac{\pi}{2}$). Then the height and width of the object/person are estimated, as following:

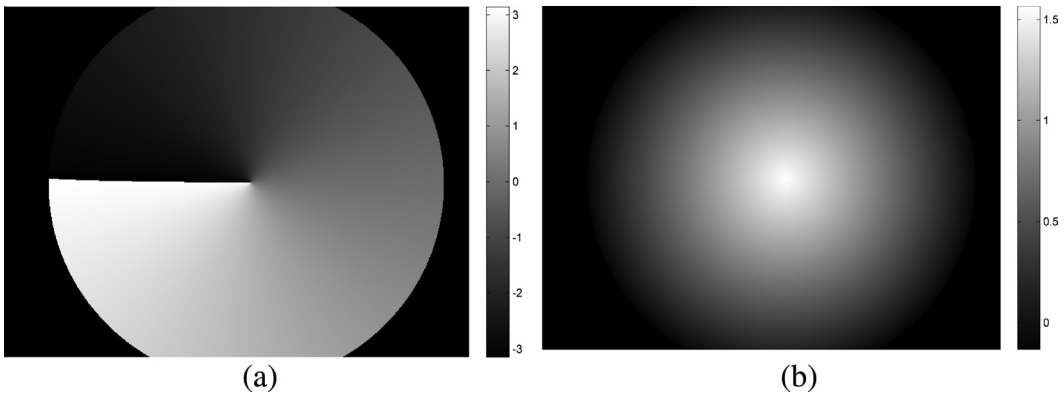


Fig. 6. Graphical representation of the azimuth (a) and elevation (b) look-up tables.

$$\begin{aligned} \text{height} &= z_{\max} - \tan(\varphi_{\min})\sqrt{x_{real}^2 + y_{real}^2}, \\ \text{width} &= 2\sqrt{x_{real}^2 + y_{real}^2}\tan\left(\frac{|\theta_{\max} - \theta_{\min}|}{2}\right). \end{aligned} \tag{16}$$

By requiring that $\varphi_{\min}$ is sufficiently below $\pi/2$ (e.g. $\varphi_{\min} < \frac{9}{10}\frac{\pi}{2}$) and assuming that the imaged object has no strong concavities, then $\frac{|\theta_{\max} - \theta_{\min}|}{2} < \frac{\pi}{2}$ should hold, therefore the maximum width of the object can also be estimated using (17). Otherwise, the height and the width are given a label value of -1.

2.4.2. Definition of proximity in the space of the calibrated fisheye frame

Very often the human silhouette is not segmented as a single binary object, but as a collection of separate adjacent binary objects. In addition, two or more segmented silhouettes, each one consisting of a number of binary components, may be close to each other in the video frame. In most of the cases where a perspective camera is used, the neighborhood of a pixel is defined as a square/parallelogram of constant size, irrespectively of the position of the pixel in the image. In the case of fisheye video frames, this approach is not satisfactory. In this work, the proximity of two segmented binary objects is inferred by their elevation and azimuth angles, as following. Let $\theta_{i,\min}$, $\theta_{i,\max}$, $\varphi_{i,\min}$, $\varphi_{i,\max}$ be the minimum and maximum azimuth and elevation for objects $i = 1,2$. Also, let $\delta\theta$ and $\delta\varphi$ be two thresholds for the azimuth and elevation, respectively. If the two binary objects in the segmented frame do not lie close to the CoD (equivalently they are not directly below the camera, or $\varphi_{i,\min} < \pi/2$), then they are proximal in the space of the fisheye frame, if the following holds:

$$\begin{aligned} &((\theta_{1,\min} < \theta_{2,\min}\ AND\ \theta_{1,\max} > \theta_{2,\max})\ OR\ |\theta_{1,\max} - \theta_{2,\max}| \\ &\quad < \delta\theta\ OR\ |\theta_{1,\min} - \theta_{2,\min}| < \delta\theta)\ AND \\ &((\varphi_{1,\min} < \varphi_{2,\min}\ AND\ \varphi_{1,\max} > \varphi_{2,\max})\ OR\ |\varphi_{1,\max} - \varphi_{2,\max}| \\ &\quad < \delta\theta\ OR\ |\varphi_{1,\min} - \varphi_{2,\min}| < \delta\varphi). \end{aligned} \tag{17a}$$

If any of the segmented binary objects lies directly below the camera ($\varphi_{i,\max} = \pi/2$) then the concept of $\delta\theta$ is not applicable and proximity in the fisheye frame is defined if the following holds

$$\begin{aligned} &(\theta_{1,\min} < \theta_{2,\min}\ AND\ \theta_{1,\max} > \theta_{2,\max})\ AND \\ &((\varphi_{1,\min} < \varphi_{2,\min}\ AND\ \varphi_{1,\max} > \varphi_{2,\max})\ OR\ |\varphi_{1,\max} - \varphi_{2,\max}| \\ &\quad < \delta\theta\ OR\ |\varphi_{1,\min} - \varphi_{2,\min}| < \delta\varphi). \end{aligned} \tag{17b}$$

Recent research has been reported in feature detection in fisheye images using the geodesic distance between pixels on a sphere to define pixel neighborhoods in omni-directional images [36,37]. However, the geodesic distance results in neighborhoods with size that does not reduce as the pixel moves towards the edge of the field of view of the fisheye image. In this work we used the following heuristic: the $\delta\varphi$ threshold is predefined and kept constant ($\delta\varphi = \pi/32$). The definition of $\delta\theta$ is slightly more complicated however: $\delta\theta = \delta\theta_0\sin(\varphi_{i,\max})$, with $\delta\theta_0 = \pi/32$. The use of thresholds $\delta\theta$ and $\delta\varphi$ provides a more accurate metric of proximity in the case of fisheye images, than a rectangular region of pixels, commonly used in images. Fig. 8 shows the corresponding regions using the constant $\delta\varphi = \pi/32$ and $\delta\theta = \delta\theta_0\sin(\varphi_{i,\max})$, with $\delta\theta_0 = \pi/32$ for a number of points every 50 pixels, along the horizontal line passing through the CoD.

Fig. 9(a) shows the two main binary objects from the segmentation of a human silhouette: the main body (object 1) and the head (object 2). Using the azimuth and elevation look-up tables, we locate the pixels with minimum and maximum angle values: for object 1 points C and D with values $\varphi_{1,\min}$, $\varphi_{1,\max}$, points G and D with $\theta_{1,\min}$, $\theta_{1,\max}$. The corresponding points for object 2 are: points A and B with values $\varphi_{2,\min}$, $\varphi_{2,\max}$, points E and F with $\theta_{2,\min}$, $\theta_{2,\max}$. The $\varphi_{1,\min}$, $\varphi_{1,\max}$, $\varphi_{2,\min}$, $\varphi_{2,\max}$ and $\delta\varphi$ are shown graphically in Fig. 9(b). The criterion of Eq. (18) is satisfied, therefore, objects 1 and 2 are in proximity in the space of the fisheye frame.

Fig. 8. The corresponding regions using the constant $\delta\varphi = \pi/32$ and $\delta\theta = \delta\theta_0\sin(\varphi_{i,\max})$, with $\delta\theta_0 = \pi/32$ for a number of points every 50 pixels, along the horizontal line passing through the CoD of the fisheye video frame.

2.4.3. Detecting non-plausible segmentation

Change of illumination from a window, displacement of small objects on walls or inside the room, reflections of moving objects, or automatic changes of camera exposure settings, often results in foreground segmentation and false human silhouette detection. The calculation of possible object height, width and its position coordinates in the real world (x_{real} and y_{real}), combined with the known dimensions of the room allows the identification of non-plausible segmentation (segmentation that does not correspond to human silhouette). If a binary object in the segmented frame is not in proximity to any other object according to Eq. (17), then its real world position (x_{real}, y_{real}, $z_{\max}$) is calculated as if the object was touching the floor. If the estimated real position, or its height is outside the allowed range (i.e. outside the room), the segmentation is flagged as non-plausible. Object B is an artificial example of such a case in Fig. 10(a). If the estimated position (x_{real}, y_{real}, $z_{\max}$) falls within the allowed range, then the estimated height and width according to (17) are used to detect non-plausible segmentation, by imposing lower and upper thresholds. By relaxing the height threshold, the proposed algorithm can correctly identify as human silhouette, persons at different poses. The results shown in this work have been produced with a low height threshold of 1 m, which allows identification of bending or sitting humans, while rejecting small moving objects (handbags, stools, etc.). The high threshold was set to 2.2 m. Low and high thresholds for the estimated width were set to 0.2 m and 1.5 m respectively (the high threshold should not exclude silhouettes of single humans with activity such as hand gestures).

Object C in Fig. 10(a) is a graphic example of such a case. This method allows detection of segmented objects that are not silhouettes of standing humans. An example of the applicability of detecting non-plausible segmentation is shown in Fig. 10(b). In this specific frame three humans are inside the room, the room borders are also superimposed. The moving door also causes foreground segmentation, as well as the reflection of one of the humans. As it will be shown in the result section, the proposed segmentation refinement algorithm rejects the segmentation caused by the moving door and the moving human reflection and identifies the binary components that correspond to each one of the three persons, despite the fact that two of them appear close to each other. Additionally, since the dimensions of the room are known, the proposed

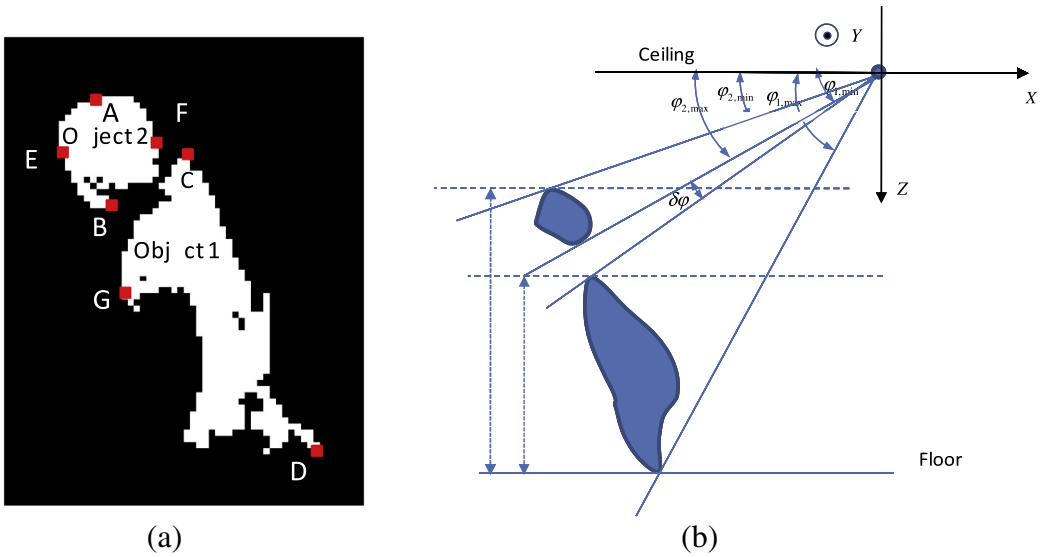


Fig. 9. (a) A typical segmented frame (zoomed) containing two binary objects in the segmented frame and (b) the explanation of the determination of their proximity. For simplicity reasons only the elevation angle is shown.

algorithm will also detect if the humans move outside the room border (e.g. through one of the doors).

2.4.4. The algorithm for geometric reasoning

The proposed methodology for geometric reasoning initially utilizes a standard algorithm for labeling the binary objects (connected components) in the current segmented frame. This algorithm produces the labeled image L. The term "connected component" and "binary object" is used interchangeably in this section. In order to avoid excessive numbers of binary objects due to segmentation noise, each pixel of the segmented image is set to 1 if it has at least 5 non-zero neighbors (including itself), otherwise it is set to 0. We denote by L_i the binary image containing the pixels of the segmented frame that belong to the ith binary connected component (i.e. $L_i(p) = 1, p{:}L(p) = i$). Therefore, we will also use L_i for denoting the ith binary object. We will use

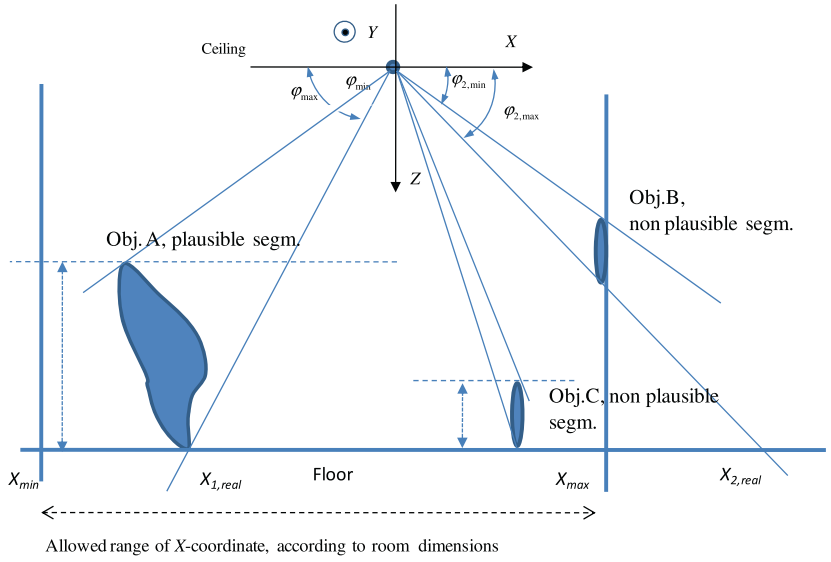
- The function OK(B) that returns TRUE if the binary object in B is plausible and FALSE otherwise, as described in Section 2.4.3.
- An array `gp` that holds a flag indicating that the ith binary object has been processed.
- The `proximity` function which, given 2 binary segmented objects from a fisheye video frame, returns TRUE if the objects are in proximity according to Section 2.4.1 and FALSE otherwise.

Initially, the largest plausible, non-processed binary object is selected as the required object L_0 and appropriately flagged as processed. Subsequently, the first non-processed binary object L_i is selected that is proximal to the required object, according to (17). The height and the width of the union of L_0 and L_i is calculated using (17). If the combined height and width is acceptable (`OK(`$L_0 \cup L_i$`) == 1`), then the required object L_0 is being updated as the union of L_0 and L_i ($L_0 = L_0 \cup L_i$) and the minimum and maximum values of the azimuth and elevation of the required object ($\theta_{0,\min}$, $\theta_{0,\min}$, $\varphi_{0,\min}$, $\varphi_{0,\min}$) are updated according the look-up tables. The ith binary object in the segmented frame is flagged as processed. The process is repeated until no other segmented object may be appended to the current object. If more binary objects exist in the segmented images that have not been processed, the steps are repeated for the detection of a different person in the room. Notice that the plausible segmentation is checked for the binary object L_0 (the first time that the object is selected as the largest, non-processed, plausible object) and it is also checked for each time another object is appended to L_0.

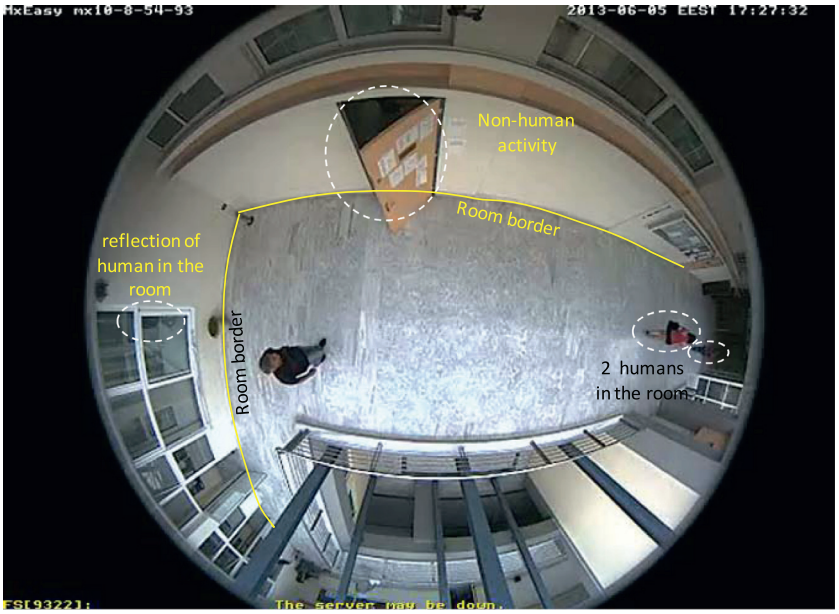
The binary objects that are appended to L_0 are not checked independently for plausible segmentation. The overall algorithm for geometric reasoning is presented in pseudocode below in Algorithm 1, where we assume for simplicity that the 1st binary object is the largest one. The corresponding workflow diagram is also provided in Fig. 11.

Algorithm 1: The geometric reasoning algorithm in pseudocode.

```
i=0 // Set the binary object index i=0
num=max(L) // Set the maximum number of binary
  objects in L,
while i<num
  i=1+1
  LO=(L==i) // binary image of the ith connected
  component (object)
  il=0 // secondary index of connected components
  to be amended to ith object
  If OK(Li) AND gp(i)=0
    gp(i)=1
    il=1
    while (il<num)
      Ll=(L==il)
      if gp(il)=0 AND proximity(Li,Ll)=1 AND
  OK(LO∪Li)==1
        LO=LO∪Li
        gp(il)=1
        il=0
        update θ0,min, θ0,max, φ0,min, φ0,max
      end
      il=il+1
    end
  end
end
```



(a)



(b)

Fig. 10. (a) Using the elevation angle φ to determine non-plausible segmentation. (b) An exemplar frame showing instances of non-plausible segmentation activity (reflection in the window, moving door), as well as the border of the room.

The proposed algorithm merges binary objects into a segmented silhouette, rejects binary objects that cannot be merged to any silhouette and it is capable of handling multiple silhouettes. The algorithm also calculates the trajectory of each silhouette. Typical segmentation of a human silhouette consisting of 12 binary objects from a single frame is shown in Fig. 12(a)–(c). The largest connected component is shown in (a), with the elevation φ values of each pixel encoded as color. In (b) an intermediate step of the geometry-based silhouette refinement algorithm is shown, where the binary object corresponding to the human head has been added. In (c) the final result is shown, with the right arm and foot also added to the segmented human. Fig. 13 shows examples of the application of the algorithm in cases of multiple activities. The output of the algorithm is color-encoded for visualization purposes: red color indicates segmentation that has been rejected, while other colors indicate activity segmented as human silhouette (each color corresponds to a single silhouette). In Fig. 13(a) the human silhouette that consists of 4 binary objects has been indentified (yellow color), whereas the moving door is discriminated against the silhouette and rejected (red color). Fig. 13(b) shows a similar case with two human silhouettes discriminated against each other and against the moving door. In Fig. 13(c) the two human silhouettes have been correctly discriminated against each other (yellow and green color), despite the fact that they appear very close in the frame. In Fig. 13(d) a human silhouette is correctly identified against a smaller object (moving stool). In Fig. 12(f)–(h) and (l), multiple human silhouettes are correctly identified despite their proximity and the difficulty in segmenting the shirt in one of them, whereas in (k) incorrect merging of two silhouettes is shown. In figure (e) a segmented silhouette is rejected as being outside the room. Finally in Fig. 12(i) and (j) a sitting human silhouette is also correctly identified by merging a large number of binary segments.

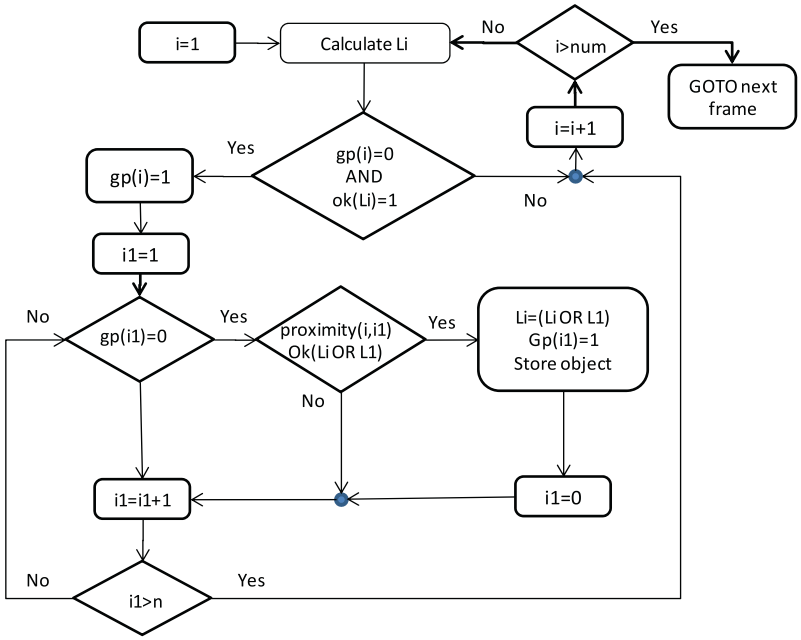


Fig. 11. Schematic representation of the geometric reasoning algorithm.

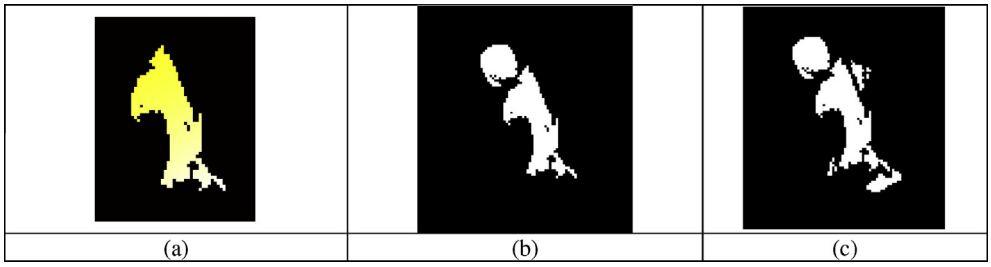


Fig. 12. Merging binary objects to a segmented human silhouette. (a) The largest binary component with the elevation angle encoded in color. In (b) the head has been appended, using the proposed algorithm, (c) The final segmentation that consists of a number of binary connected components. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

It has to be mentioned that if one silhouette is partially occluded by another, then the two silhouettes will be counted as one. A long as this occlusion does not last for long, the unique labeling of the silhouettes (described in the next subsection) will still assign the correct label, using position, temporal information and appearance.

2.5. Silhouette labeling

The proposed algorithm identifies the individual human silhouettes in each video frame and rejects irrelevant segmentation. The algorithm that is proposed in this subsection uses appearance information as well as spatio-temporal information to assign a unique label to each silhouette, valid for the whole video. In this way, the trajectory of each silhouette may be determined and clues about the behavior of each person may be drawn.

The problem may be formulated as following: assuming that N binary silhouettes have been segmented in the current frame k and M silhouettes have been uniquely labeled by the proposed algorithm so far, we need to map the N current silhouettes to the existing M labels according to a distance metric and possibly introduce a new label if necessary.

The following are maintained for each unique label: the previous positions in 3D space (determined by the proposed geometry-based algorithm) and its mean normalized values of each of the Red, Green, Blue channels. Each new segmented silhouette i of the current frame is compared with each of the existing labels j, using a distance metric r_{ij} defined as the product of the Euclidean distance d_{ij} between their known real positions on the floor and their difference in appearance in terms of their mean normalized RGB values.

The minimum Euclidean distance d_{ij} is calculated between i and the last K_0 positions of label j:

$$d_{ij} = \min_m((x_{i,real} - x^m_{j,real})^2 + (y_{i,real} - y^m_{j,real})^2), \quad m = k - K_0 + 1, \ldots k \tag{18}$$

Similarly, the minimum distance in the normalized RGB color space d_{ij} is calculated between i and the last K_0 positions of label j where

$$c_{ij} = \min(\max(|R_i - R^m_j|, |G_i - G^m_j|, |B_i - B^m_j|)), \quad m = k - K_0 + 1, \ldots, k \tag{19}$$

The second term is the infinity norm of the mean normalized RGB values of the pixels that belong to silhouette i and unique label j (at m_0 frame). RGB normalization is achieved by $(R_i, G_i, B_i) = \frac{(r_i, g_i, b_i)}{r_i + g_i + b_i} \in [0, 1]$, where r_i, g_i, b_i are the mean R, G, B values of the video frame pixels that belong to the binary segmented silhouette i. This normalization allows independence from the image intensity, a feature necessary in

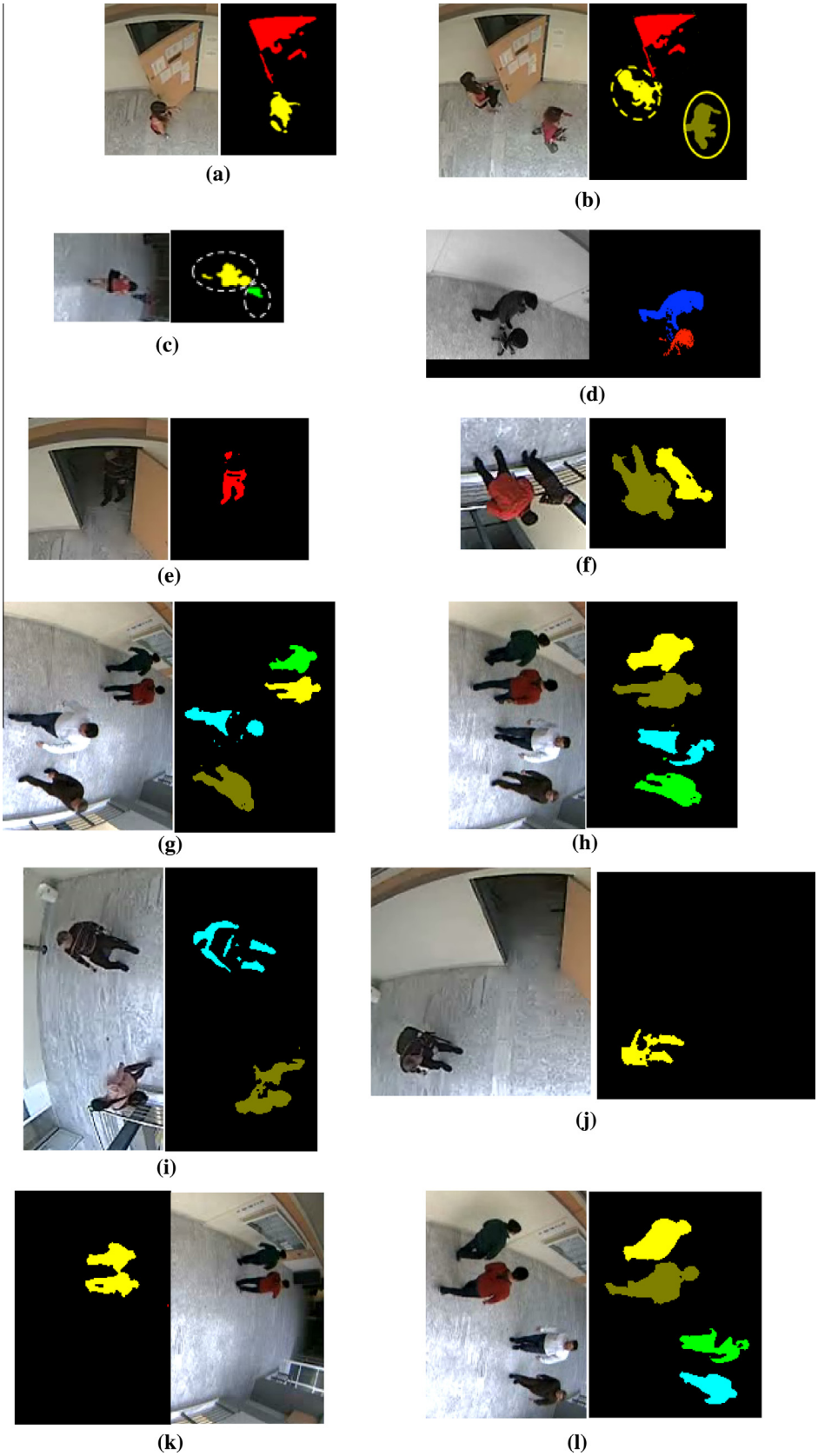


Fig. 13. Application the geometric reasoning algorithm to detect non-plausible segmentation (shown in red) as opposed to the plausible segmentation. Individual silhouettes are identified by unique colors (other than red). Silhouettes are identified against a moving door (a), as well as against each other (b), even in cases of small apparent distance (c). In (d) the human silhouette is correctly identified against a smaller moving object and in (e) a segmented silhouette is rejected as being outside the room. In (i) and (j) correct identification is shown for the silhouette of a sitting human (despite the many binary segmented. Finally, in (g), (h) and (l) silhouettes are correctly discriminated against each other despite being apparently close, whereas in (k) incorrect merging of two silhouettes is shown. The corresponding portions of the original frames are shown at the left. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

our problem, since the surveillanced space extents over 12 m, thus significant illumination changes are expected.

The combined distance metric r_{ij} is calculated and stored in matrix $R_{ij} = \{r_{ij}\}$ for all available labels as following:

$$r_{ij} = \begin{cases} d_{ij}c_{ij}, & \text{if } d_{ij} < d_0 \text{ AND } c_{ij} < c_0 \\ \max\_val, & \text{otherwise} \end{cases}$$

If c_{ij} or d_{ij} are greater than a threshold, r_{ij} is set to a large value (max_val) to indicate no match between the silhouette and the existing labels. In this work, d_0 was set to 0.5 m and $c_0 = 20/255$. The silhouette labeling algorithm proceeds as following: the pair (i_0, j_0) is found that holds the minimum value of r_{ij}. If the minimum value is less than the max_val, then silhouette j_0 is assigned the label i_0. Otherwise, the next label $(M + 1)$ is used. In order to avoid assigning the same label to more than one silhouette of the current frame, the j_0th column and label i_0th line of r_{ij} matrix is set to the max_val. The above algorithm may be described in pseudocode as following:

```
Initialize the current number of unique labels, M=0
For each frame
  N: number of segmented silhouettes in current
   frame
  For each binary segmented silhouette in the frame,
  j=1,...,N
    Obtain its position in 3D coordinates (x_{j,real}, y_{j,real})
  from the proposed geometry based algorithm
    Calculate the mean normalized values for each of
  the R, G, B channels
    For all labels i=1,2,...,M
      Calculate d_ij, c_ij and matrix R_ij = {r_ij}
    end
  end
  while R_ij contains values < max_val
  find the position (i_0, j_0) of the minimum value m of R_ij
  If m<max_val (matching label exists)
    label the current segmented silhouette i_0 with j_0
    update the real position and mean RGB values for
  label i_0
    set R_{i_0 j} = max_val, j = 1,2,...,N
    set R_{i j_0} = max_val, i = 1,2,...,M
  else (no matching label exists)
    update the number of unique labels M = M + 1
    label the current segmented silhouette i_0 using
  the next available label j_0 = M
    update the real position and mean RGB values for
  label j_0
    append one line to the R_ij table, with values
  max_val: R_{M j} = max_val, j = 1,2,...,N
    set the (j_0)th column of R_ij equal to max_val:
  r_{i j_0} = max_val, j = 1,2,...,M
  end
end
```

After the completion of the algorithm, unique labels that only appeared for vary limited number of frames, due to incorrect segmentation/geometric reasoning may be easily rejected.

The proposed labeling algorithm has been shown to work well when a small/moderate number of persons are present in the room and the occlusions are relatively short. Exemplar results from the execution of this algorithm are given in Fig. 14 for an environment of 4 discreet silhouettes with occlusions. The red color denotes rejected segmentation (not identified as human silhouette), the rest of the colors denote unique segmented silhouettes. The unique labels are superimposed on each silhouette (color indicates acceptance/rejection, but not unique label). It can be seen that correct unique labeling is achieved, despite two occlusions. Details are given in the figure captions. Trajectory estimation of each silhouette is shown in Fig. 17(c) and (d).

3. Results

During the experiments video sequences were acquired using the Mobotix Q24 hemispheric camera, which was installed on the ceiling of the imaged university room. The pixilation of each frame is 480×640, the frame rate was set to 25 fps and the duration of each video was 45 s to 2 min. Specific details are given below for each video sequence:

- *Video 1:* single person entering the supervised room, exits the room from a different location. Activity: turning the light on, door opening and closing, reflection of the person in glass windows, etc. Segmentation difficulty: low (high contrast between person's clothes and the surrounding).
- *Video 2:* 3–5 different persons entering and leaving the room from various points. Activity: door opening and closing, reflection of the person in glass windows, persons moving close to each other, persons outside the designated room area. Segmentation difficulty: moderate.
- *Video 3:* a single person entering and leaving the room. Activity: leaves a handbag, returns and picks the handbag. Segmentation difficulty: high (person's t-shirt very similar to the floor)
- *Video 4–5:* a single person entering and leaving the room. Activity: opens door, moves a stool, person's reflection. Segmentation difficulty: low.
- *Video 6:* total of 4 persons, one standing almost still throughout the sequence with another person moving round the first one, a third person moving across the corridor and a fourth one entering the room and sitting on a chair. Segmentation difficulty: low.

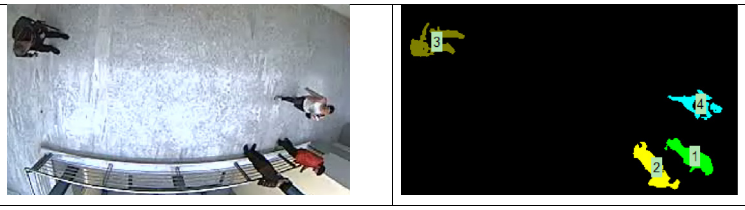
Fig. 15 shows exemplar results of human silhouette segmentation refinement using the proposed geometry-based algorithm from the 2nd video sequence (the one with the largest number of persons simultaneously inside the room). The rejected segmentation (not plausible human silhouettes inside the room) is shown in red. The rest of the colors correspond to segmentation accepted as human silhouettes and are used to indicate the grouping of binary regions into discrete silhouettes. In (a) the binary objects of the segmented opening door are grouped together as a single object, which is subsequently rejected are non-plausible human silhouette segmentation. Two segmented humans are rejected as being outside the room (at the left of the frame, behind the glass window). In (b), (c) and (d), two humans have entered the room. The proposed algorithm rejects the segmented moving door, although it is close to the persons.

In (e) one of the humans previously standing behind the glass window outside the room (see Fig. 13d), has now entered the room (green segmentation). The corresponding segmentation is correctly accepted by the algorithm as silhouette. While most of the segmented pixels of the moving door continue to be rejected, one small vertical disconnected part of the segmented door exists that cannot be rejected unambiguously and appears as incorrectly accepted.

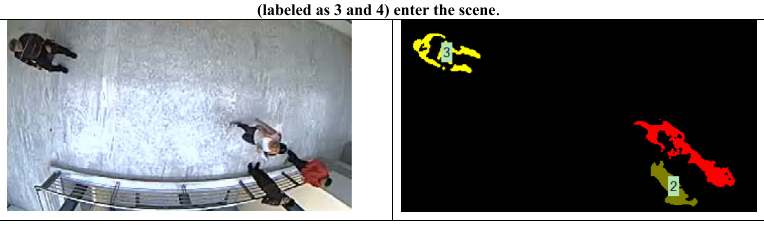
In (f) a number of small, disconnected binary components appear in the initial segmentation results, which have been rejected by the proposed algorithm. Note that the silhouette of the two humans, partially occluded at the lower left corner of the frame is accepted by the proposed algorithm. The human

(a) Two silhouettes with their unique labels

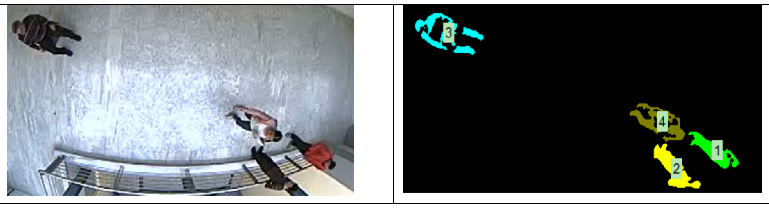
(b) Silhouette with label 1 occludes the label 2 silhouette. One silhouette is detected, labeled as 1.



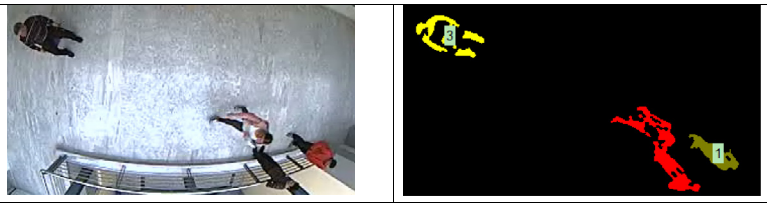
(c) The silhouettes labeled 1 and 2 are both visible again and assigned their unique label. Two moew silhouettes (labeled as 3 and 4) enter the scene.



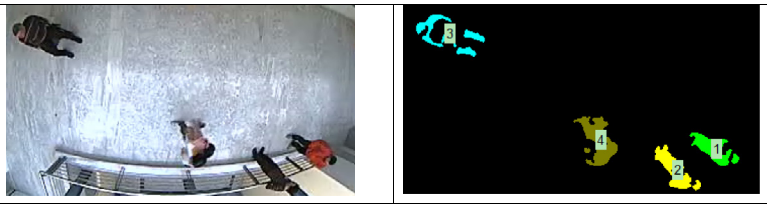
(d) Silhouette 4 partially occludes silhouette 1. Both silhouettes are segmented as a single object, which is rejected by the geometry-based refinement, due to excessive height (thus the red color). Silhouettes 2 and 3 retain their correct labels.



(e) No silhouette is occluded in this frame and all four silhouettes are correctly labeled.



(f) Silhouette 4 partially occludes silhouette 2. Both silhouettes are segmented as a single object, which is rejected by the geometry-based refinement, due to excessive height (thus the red color). Silhouettes 1 and 3 retain their correct labels.



(g) No silhouette is occluded in this frame and all four silhouettes are correctly labeled again.

Fig. 14. The performance of the silhouette labeling algorithm in an environment of 4 discreet silhouettes with occlusions.

(a)

(b)

(c)

(d)

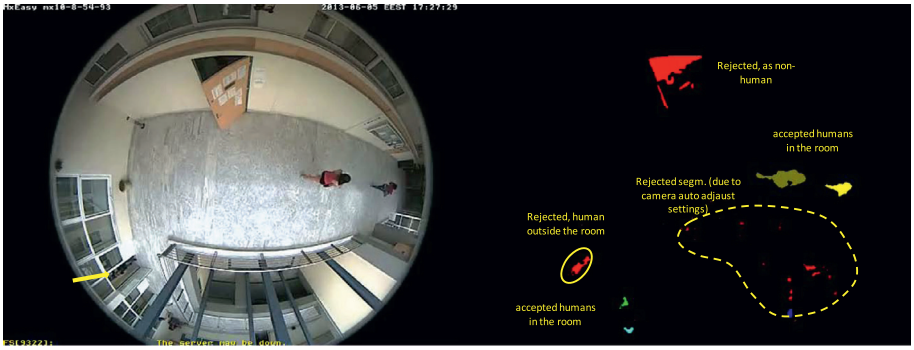
Fig. 15. The output of the proposed algorithm for a number of frames of video sequence 2 (original frames at the left side). Color indicates the output of the algorithm: red for rejected segmentation, olive (128, 128, 0), yellow (255, 255, 0), green, cyan (0, 255, 255), violet, silver and magenta indicate the 1st, 2nd up to sixth object accepted as silhouette. See text for details. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

standing at the door of the room (left side of the frame) is segmented, but subsequently rejected as being outside the room.

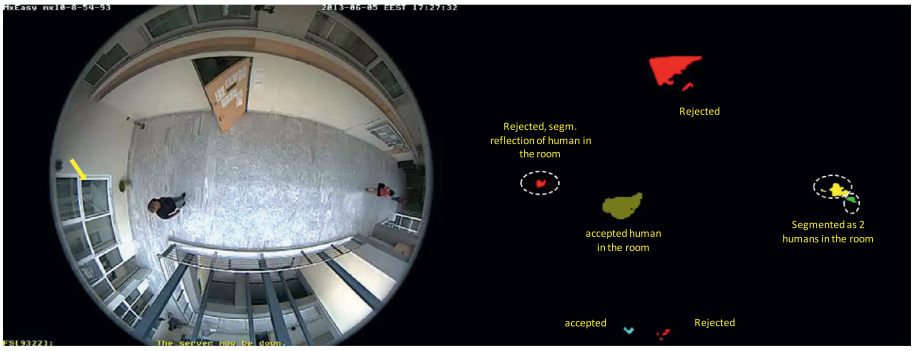
Finally, in (g) the two humans that appear at the far right of the frame, are recognized by the algorithm as two different silhouettes, despite their small apparent distance. The reflection in the glass window of the human standing at the left of the frame is successfully rejected.

Fig. 16 shows combined results from the segmentation of the human silhouette from a number of videos with one human silhouette present and demonstrates the calculation of the trajectory in

(e)



(f)



(g)

Fig. 15 (*continued*)

real world coordinates. The left column contains superimposed the resulting segmentation of the human silhouette, every 1 s, to assist visual validation of the results. Segmented pixels are shown in black. The path of the silhouette has been calculated as described in [18] and it is plotted in real world coordinates. The path is also rendered through the fisheye model and it has been displayed on the frame with the combined segmentation at the left column of Fig. 16. It can be visually observed that the recovered positions of the segmented human are in accordance with the human motion, as seen in the left column. Low quality segmentation, such as in video 3 (shown in Fig. 16a) may result in noisy trajectory estimation.

Fig. 17 shows combined results from the segmentation of the human silhouette from two video sequences (number 2 and 6) with many human silhouettes present and demonstrates the calculation of each silhouette trajectory in real world coordinates. The left column contains superimposed the resulting segmentation of the human silhouettes from selected frames, to assist visual validation of the results. The silhouettes are uniquely labeled using the proposed algorithm and the paths of each unique silhouette have been calculated and plotted in real world coordinates. The paths are also rendered through the fisheye model and have been displayed on the frame with the combined segmentation at the left column of Fig. 18. It can be visually observed that the recovered positions of the segmented human are in accordance with the human motion, as seen in the left column. In Fig. 17(d) the labels have been superimposed, as shown in Fig. 14.

In order to quantify the merit and efficiency of the proposed algorithm, we also performed manual segmentation of the human silhouettes inside the room in one every 50 frames of the acquired video sequences 1–5. Table 2 shows the confusion matrix for the segmentation of the video sequences, with and without the application of the proposed geometric refinement. The confusion matrix was calculated considering 2 classes of pixels: pixels that belong to human silhouettes inside the room (excluding any other kind of segmentation) and the rest of the pixels. The structure of the confusion matrices is as following: first row true positive pixels (TP), false negative pixels (FN), second row: false positive pixels (FP), true negative pixels (TN). Frames with no action were excluded from the calculation of the confusion matrices. As it has been established, the proposed geometry-based algorithm distinguishes between human silhouette segmentation and other irrelevant

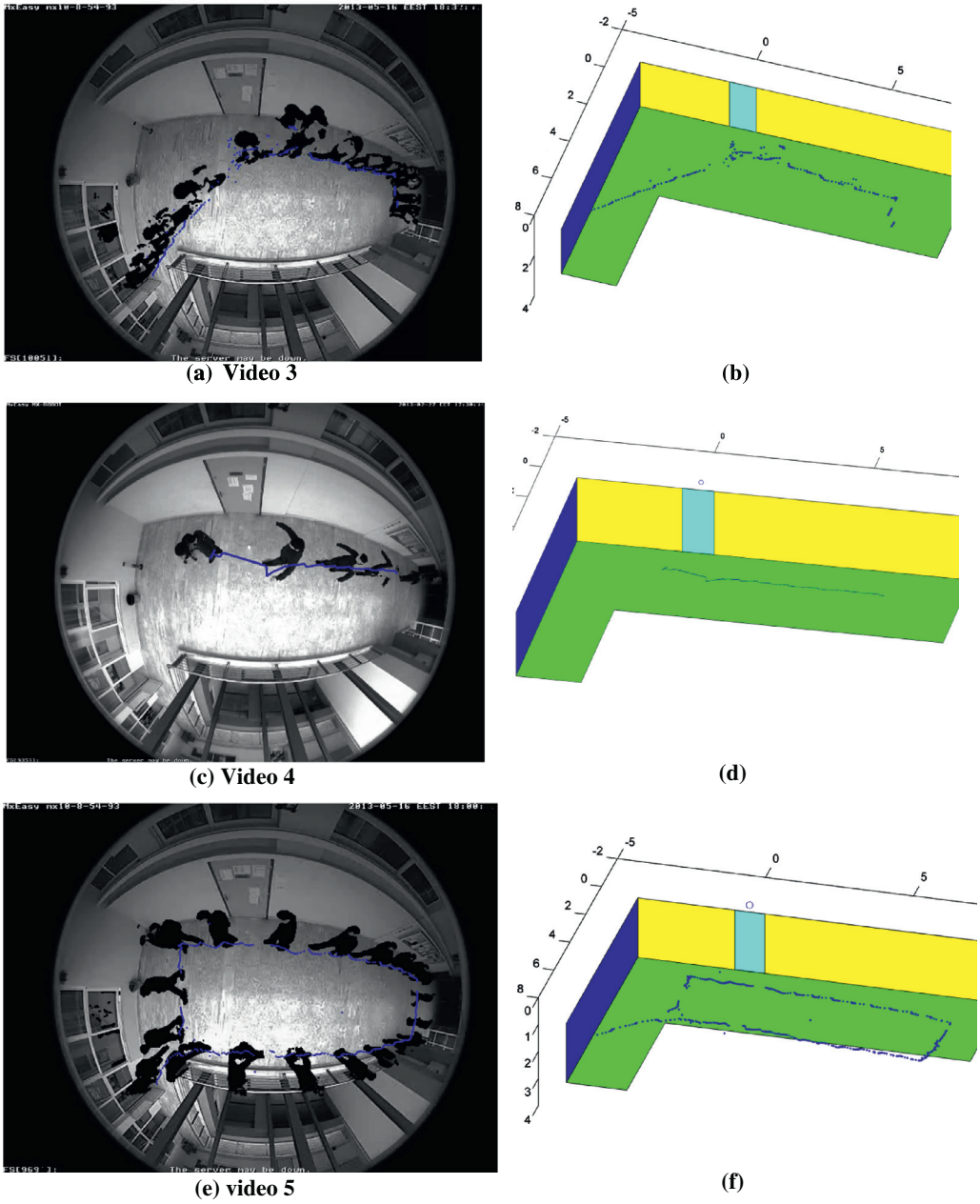


Fig. 16. The (x_{real}, y_{real}) positions of the segmented human on the floor, calculated using (16), are rendered by the fisheye camera model for video sequences 3, 4 and 5 in (a), (b) and (c). An instance of the segmented human from different frames every 1 s is also superimposed. In (b), (d) and (f) the positions of the segmented human on the floor are plotted in the real world frame of reference for the three video sequences.

segmentation in indoor environment. Thus, the number of TP and FN pixels should remain almost the same with and without the application of the geometry-based refinement (a small number of TP pixels may be excluded by the geometric algorithm). The number of FP pixels should decrease significantly with the application of the proposed algorithm (and subsequently the number of TN should increase). This effect is noticeable in Table 2 that exhibits the segmentation results.

Fig. 18 depicts also the efficiency of the silhouette segmentation refinement algorithm. In (a) the number of binary components of the segmented frame is shown for the first 1000 frames of video 2. This video sequence was selected because it is the most complex one, involving up to 5 different persons. After the application of the proposed algorithm, the number of identified human silhouettes inside the room is shown in (b) by the blue curve. The real number of human silhouettes has been counted by the user and displayed as the red curve in Fig. 18(b). It can be observed that there is an agreement between the real and the estimated number of different silhouettes. In certain cases the proposed algorithm presents a small delay in recognizing when a person enters the room (typically 20–30 frames) and may also identify a person exiting the room a few frames earlier than the human observer. Momentary disagreements between the blue and the red curve are mainly caused by more than one silhouettes positioned in such a way in the 360° FoV that they may not be distinguished as distinct silhouettes by the proposed algorithm. However this is corrected after a few frames.

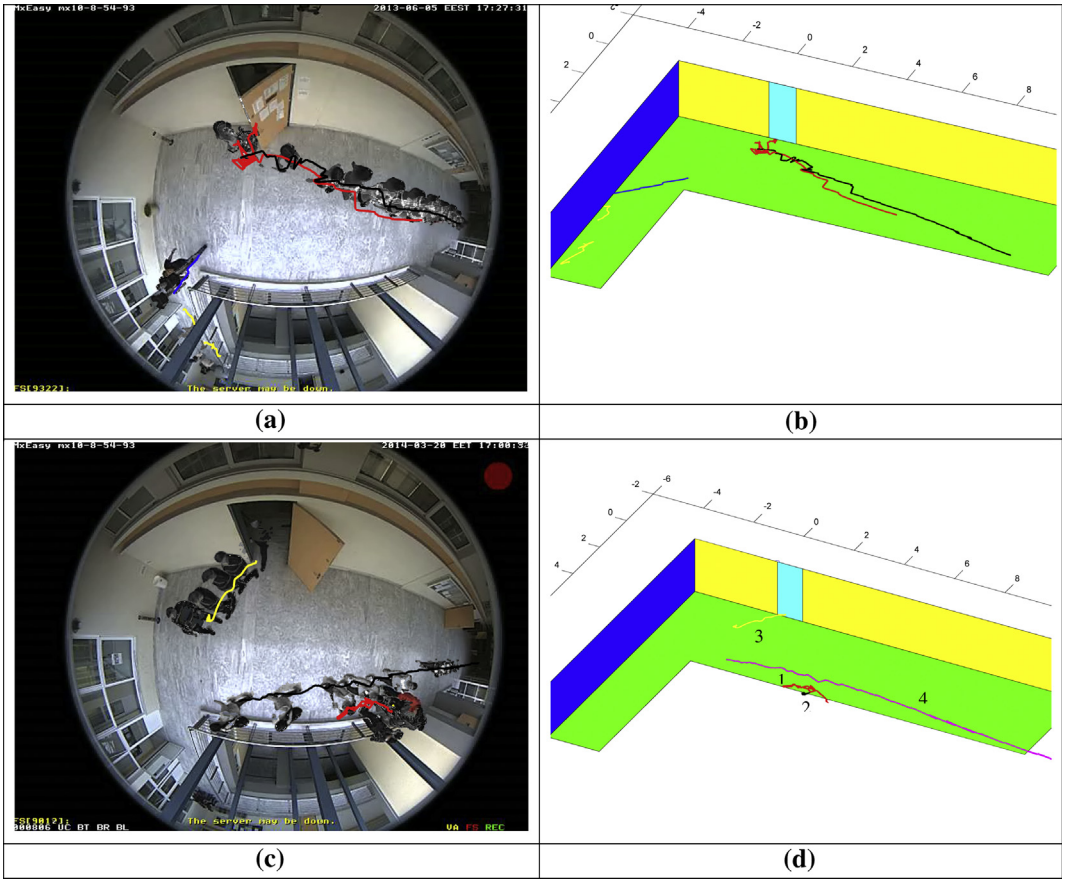


Fig. 17. The result of the application of the proposed silhouette labeling algorithm on two video sequences with multiple silhouettes: video 2 (1st row) and video 6 (2nd row). On the left, the (x_{real}, y_{real}) positions of the segmented humans on the floor are rendered by the fisheye camera model. An instance of the segmented humans from different frames every 1 s is also superimposed. On the right, the positions of the labeled human silhouettes on the floor are plotted in the real world frame of reference.

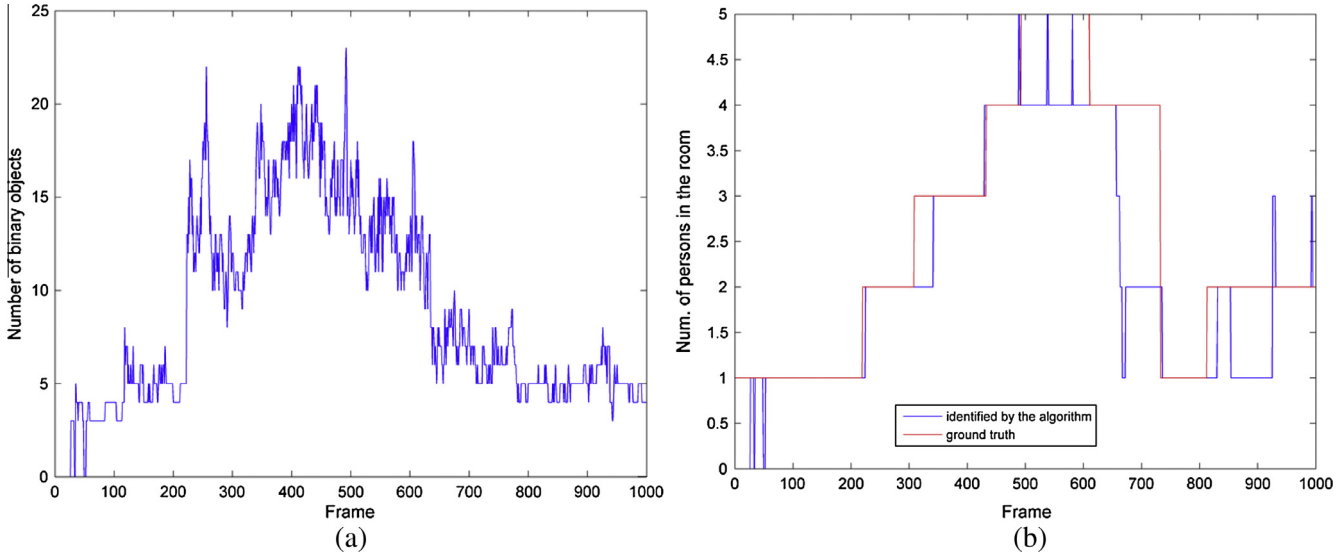


Fig. 18. The number of connected components of the segmentation before and after the application of the proposed geometry-based silhouette refinement algorithm (blue and red curve respectively), in the case of video sequence 2, (a) and (b) respectively. The number of persons that existed in the room is shown as the red curve in (b). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
The confusion matrix of the human silhouette segmentation for videos 1–5, with and without the application of the proposed geometric reasoning algorithm.

	Segmentation only		Segmentation and geometry-based silhouette refinement	
Video 1	22655	594	22654	632
	48980	6071771	5728	6114986
Video 2	49168	8832	49136	8864
	57688	12172312	8208	12221792
Video 3	20315	1812	20305	1823
	1778	7656095	1452	7656420
Video 4	54540	6840	54362	7018
	31860	13730760	6660	13755960
Video 5	25710	3750	25652	3748
	2370	9184170	1677	9184923

4. Conclusions

In this paper we presented a methodology that identifies the segmentation of human silhouettes from fisheye video sequences, against other irrelevant segmentation, or versus segmentation of silhouettes outside a predefined area. The proposed algorithm is based on clues of real world geometry, derived from a calibrated model of a fisheye camera.

Regarding the complexity of the proposed methodology the performed experiments have shown that the number of connected binary objects in the segmented frames, *num*, is of the order of 20. It can be verified that the worst case complexity of geometric reasoning for any binary frame is $O(N^2)$, where N is the number of binary objects in the segmented frame. The segmentation algorithm, as well as the geometric reasoning were implemented using Matlab and executed on an Intel(R) Core i5-2430 CPU @ 2.40 GHz Laptop with 4 GB Ram, under Windows 7 Home Premium. The mean execution time for the segmentation was approximately 50 ms per frame of dimension 480×640. The mean execution time for the proposed geometric reasoning was measured approximately 40 ms per frame. The silhouette labeling algorithm imposes negligible computational load compared to the other algorithmic steps. No special source code optimization or parallelization was used for these time measurements.

Initial results show that the proposed algorithm significantly reduces the false positive pixels regarding human silhouette segmentation, by rejecting a number of instances of segmentation that cannot be easily rejected without any geometric information. Furthermore, the proposed algorithm assigns the remaining binary, segmented objects to the silhouette they belong. The algorithm can handle multiple silhouettes. Finally, the proposed approach employs an algorithm for identifying individual silhouettes using unique labels, thus allowing the estimation of the real world trajectory of each silhouette. This output may be used to enhance human motion and action recognition software that is based on silhouettes, often used in assistive environments. The silhouette labeling algorithm has been shown to work in video sequences with a moderate number of silhouettes with occlusions of short duration.

Future work includes the enhancement of the labeling algorithm to improve the accuracy of the geometry-based segmentation refinement algorithm, by correcting sudden changes of the number of identified silhouettes by the recent history of number, position and trajectory of the silhouettes. Finally, the proposed methodology may be applicable to the traditional type of projective cameras, since they would require much simpler calibration. The geometry based segmentation refinement algorithm and the silhouette labeling can also be applied, after modifications to other (projective) type of cameras. Furthermore, if the camera is installed on the wall (with a horizontal central line of view) rather than on the ceiling, the geometry-based silhouette segmentation refinement would be simpler (in this manuscript, a standing human may have any orientation in the video frame, depending on its location).

Acknowledgments

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] Ronald Poppe, Vision-based human motion analysis: an overview, Comput. Vis. Image Underst. 108 (2007) 4–18.
[2] Ronald Poppe, A survey on vision-based human action recognition, Image Vis. Comput. 28 (2010) 976–990.
[3] Thomas B. Moeslund, Erik Granum, A survey of computer vision-based human motion capture, Comput. Vis. Image Underst. 81 (2001) 231–268.
[4] J.K. Aggarwal, M.S. Ryoo, Human activity analysis: a review, ACM Comput. Surv. 43 (3) (2011).
[5] D. Weinland, R. Ronfard, E. Boyer, Free viewpoint action recognition using motion history volumes, Comput. Vis. Image Underst. 103 (2–3) (2006) 249–257.
[6] M. Blank, L. Gorelick, E. Shechtman, M. Inari, R. Barsi, Actions as space-time shapes, in IEEE International Conf. on Computer Vision, 2005, pp 1395–1402.
[7] C. Schuldt, I. Laptev, B. Caputo, Recognizing human action: a local svm approach, ICPR (2004) 32–36.
[8] R. Gross, J. Shi, The cmu motion of body (mobo) database, Tech. Rep. CMU-RI-TR-01-18, Robotics Instiute, Pittsburgh, PA (June 2001).
[9] L. Sigal, M. Black, Human Eva: Synchronized Video and Motion Capture Dataset for Evaluation of Articulated Human Motion, Tech. Rep. Brown University, 2006.
[10] I. Laptev, M. Marszalek, C. Schmidt, B. Rozenfeld, Learning Realistic Human Actions from Movies, in: IEEE Conf on Computer Vision and Pattern Recognition (2008).
[11] J. Willems, G. Debard, B. Bonroy, B. Vanrumste and T. Goedemé 2009, How to detect human fall in video? An overview, In Proceedings of the Positioning and Contex-Awareness International Conference (Antwerp, Belgium, 28 May, 2009), POCA '09.
[12] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, Detecting moving objects, ghosts, and shadows in video streams, IEEE Trans. Pattern Anal. Mach. Intell. 25 (10) (2003) 1337–1442. 2003.
[13] N. McFarlane, C. Schofield, Segmentation and tracking of piglets in images, Mach. Vision Appl. 8 (3) (1995) 187–193 (May 1995).
[14] C. Wren, A. Azarhayejani, T. Darrell, A.P. Pentland, Pfinder: real-time tracking of the human body, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 780–785 (October 1997).
[15] T. Bouwmans, F. El Baf, B. Vachon, Background modeling using mixture of gaussians for foreground detection – a survey, Recent Pat. Comput. Sci. 1 (3) (2008) 219–237.
[16] C. Stauffer, W. Grimson, Adaptive background mixture models for real-time tracking, in: Proceedings of the conference on computer vision and pattern recognition (Ft. Collins, USA, June 23–25, 1999), CVPR '99. IEEE Computer Society, New York, NY, pp. 246–252.
[17] F.C. Cheng, S.C. Huang, S.J. Ruan, Implementation of illumination-sensitive background modeling approach for accurate moving object detection, IEEE Trans. Boardcasting 57 (4) (2011) 794–801.
[18] A. Christodoulidis, K.K. Delibasis, I. Maglogiannis, Near real-time human silhouette and movement detection in indoor environments using fixed cameras, in the 5th ACM International Conference on PErvasive Technologies Related to Assistive Environments, Heraklion, Crete, Greece, 2012.
[19] K. Kemmotsu, T. Tomonaka, S. Shiotani, Y. Koketsu, M. Iehara, Recognizing human behaviors with vision sensors in a Network Robot System, in Proceedings of IEEE Int. Conf. on Robotics and Automation, 2006, pp. I274–1279.

[20] Z. Zhou, X. Chen, Y. Chung, Z. He, T.X. Han, J.M. Keller, Activity analysis, summarization and visualization for indoor human activity monitoring, IEEE Trans. Circuit Syst. Video Technol. 18 (II) (2008) 1489–1498.
[21] H. Li, R. Hartley, Plane-based calibration and auto-calibration of a fish-eye camera, in: P.J. Narayanan et al. (Eds.): ACCV 2006, LNCS 3851, 2006, pp. 21–30, c Springer-Verlag, Berlin Heidelberg 2006.
[22] A. Basu, S. Licardie, Modeling fish-eye lenses, in: Proceedings of the 1993 IEEWSJ International Conference on Intelligent Robots and Systems, Yokohama, Japan July 2630, 1993.
[23] S. Shah, J. Aggarwal, Intrinsic parameter calibration procedure for a high distortion fish-eye lens camera with distortion model and accuracy estimation, Pattern Recogn. 29 (11) (1996) 1775–1788.
[24] J. Wei, C.F. Li, S.M. Hu, R.R. Martin, C.L. Tai, Fisheye video correction, IEEE Trans. Visual. Comput. Graph. (2012).
[25] https://sites.google.com/site/scarabotix/ocamcalib-toolbox.
[26] M. Rufli, D. Scaramuzza, R. Siegwart, 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, September 2008.
[27] K.K. Delibasis, T. Goudas, V.P. Plagianakos, I. Maglogiannis, Fisheye camera modeling for human segmentation refinement in indoor videos, in: Proceedings of the 6th International Conference on PErvasive Technologies Related to Assistive Environments, 2013.
[28] N. Max, Computer graphics distortion for IMAX and OMNIMAX projection, Proc Nicograph 83 (1983) 137. December 1983.
[29] N. Greene, Environment mapping and other applications of world projections, IEEE Comput. Graphics Appl. 6 (11) (1986) 21.
[30] http://paulbourke.net/dome/fisheye/.
[31] B. Micusik, T. Pajdla, Structure from motion with wide circular field of view cameras, IEEE Trans. Pattern Anal. Machine. Intel., PAMI 28 (7) (2006) 1–15.
[32] J. Canny, A computational approach to edge detection, IEEE Trans. Pattern Anal. Mach. Intel. 8 (6) (1986) 679–698.
[33] M.G. Epitropakis, D.K. Tasoulis, N.G. Pavlidis, V.P. Plagianakos, M.N. Vrahatis, Enhancing differential evolution utilizing proximity-based mutation operators, IEEE Trans. Evol. Comput. 15 (2011) 99–119.
[34] R. Storn, System design by constraint adaptation and differential evolution, IEEE Trans. Evol. Comput. 3 (1999) 22–34.
[35] V. Plagianakos, G. Magoulas, M. Vrahatis, Evolutionary training of hardware realizable multilayer perceptrons, Neural Comput. Appl. 15 (2005) 33–40.
[36] Javier Cruz-Mota, Iva Bogdanova, Benoît Paquier, Michel Bierlaire, Jean-Philippe Thiran, Scale invariant feature transform on the sphere: theory and applications, Int. J. Comput. Vis. 98 (2012) 217–241.
[37] Cédric Demonceauxa, Pascal Vasseurb, Yohan Fougerolle, Central catadioptric image processing with geodesic metric, Image Vis. Comput. 29 (12) (2011) 840–849.