Multicamera tracking of multiple humans based on colored visual hulls

Pashalis Padeleris, Xenophon Zabulis
Institute of Computer Science, FORTH, Heraklion, Crete, Greece
{padeler,zabulis}@ics.forth.gr

Antonis A. Argyros
Computer Science Department, University of Crete, Greece
Institute of Computer Science, FORTH, Heraklion, Crete, Greece,
argyros@ics.forth.gr

Abstract

Detecting, localizing and tracking humans within an industrial environment are three tasks which are of central importance towards achieving automation in workplaces and intelligent environments. This is because unobtrusive, real-time and reliable person tracking provides valuable input to solving problems such as workplace surveillance and event/activity recognition and, also, contributes to safety and optimized use of resources. This paper presents a passive approach to the problem of person tracking that is based on a network of conventional color cameras. The proposed approach exhibits robustness to challenging conditions that are encountered in industrial environments due to illumination artifacts, occlusions and the highly dynamic nature of the observed scenes. The multiple views of the environment that the system employs are used to obtain a volumetric representation of the humans within it, in real-time. Although human tracking can be achieved based solely on such a volumetric representation, in demanding scenes, this information is not enough to recover from tracking failures. Thus, in this work, we collect and update a representation of the color appearance of the persons in the environment. The combination of volumetric and color information reinforces tracking robustness, even when a person is not visible by any of the cameras for extended time intervals. The proposed approach has been extensively evaluated in comparison with an existing state of the art method and pertinent results are reported.

1. Introduction

In the current manufacturing industry, robots execute their tasks within the boundaries of strictly defined production processes and rely on a strong technological infrastructure, which may include the installation of obtrusive and specialized sensors. In future factories, human workers and robots will frequently share common spaces. The co-existence of mobile robots and humans leads to the necessity for robust robot motion coordination. At the same time, algorithms that allow robots to navigate in the environment and accomplish the desired tasks, require information about the locations of humans in order to achieve robot/human collision prediction and avoidance. In this context, it is important to be able to detect, localize and track humans in an industrial environment.

The tracking of multiple humans is a fundamental and challenging problem whose solution has a wide range of applications, including personal and service robots, intelligent cars, crowd control and surveillance. People may interact with each other, merge to form groups of various sizes, or separate from groups. Tracking individual humans and objects within such formations can be difficult due to the unavoidable occlusions. This causes the number of persons-to-data association hypotheses to increase at a level that becomes intractable. The problem becomes even more difficult in cases where the environment contains windows or outdoor regions, as this hinders the operation of depth cameras and laser sensors which could otherwise provide reliable 3D structure information.

The proposed approach tracks the humans in an environment unobtrusively, robustly and in real-time, even in crowded scenes and challenging human configurations. The proposed method relies on input provided by a camera network. The acquired views are used to volumetrically reconstruct the persons at a frequent rate, so as to accurately estimate the location and walk-through trajectory of each person in the environment. In order to reinforce the robustness of the approach, a representation of the color appearance of these persons is acquired. This avoids shortcomings of a tracking approach based solely on volumetric information. Figure 1 illustrates such a case. The figure shows three frames from the same camera at different points in time, together with person localization estimates that are visualized as circles drawn on the ground plane. The colors of these circles code the tracking id. At the moment corresponding to the middle frame, the persons are in contact and, as a result, their vol-

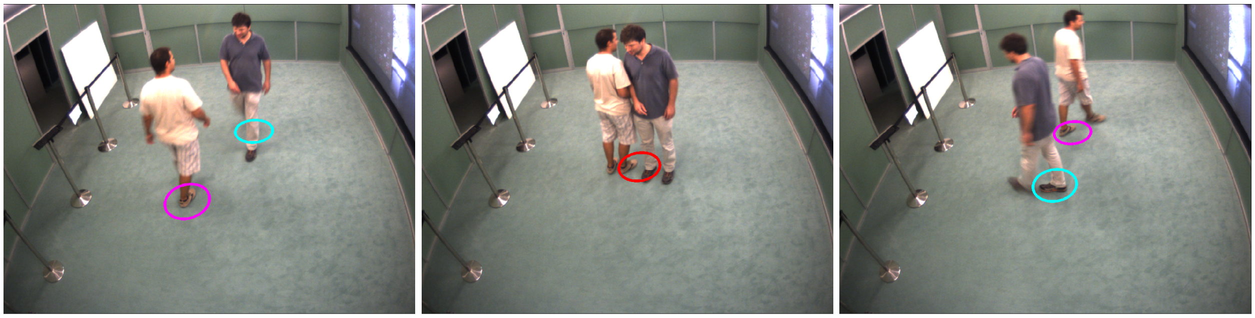

Figure 1. The proposed approach utilizes color appearance to assist tracking in cases where geometric information is insufficient. In this example, at some point in time (middle frame), the volumetric representations of the two persons are merged. Superimposed circles indicate the estimated person positions. These coincide for the middle frame because of the merged volumetric representations, but retain correct values as soon as the persons separate again.

umetric reconstructions are merged. By tracking only the volumes of these persons it is not clear how to propagate correctly the original tracking ids after the merging event. The proposed approach models the color appearance of each person before the merging event so as to use it for disambiguating the situation after the humans are split.

The remainder of this paper is organized as follows. In Sec. 2 related work is reviewed. In Sec. 3 the proposed approach to the tracking of multiple persons is presented. The method is experimentally evaluated in Sec. 4. In Sec. 5, conclusions are drawn and directions for future work are provided.

2. Related work

The capability to track persons over long periods is an important and challenging problem and, as such, it has been studied extensively. Monocular approaches [16, 19] are based on image cues such as color and silhouette shape and employ sophisticated tracking methods to cope with scene complexity. The method in [3] utilizes a binocular camera system and combines stereo, color, shape and face detection to improve tracking performance. In general, all single-view or short-baseline approaches suffer from visibility limitations due to the observation of the scene from a specific viewpoint.

A qualitatively different monocular approach to the human tracking problem is based on input provided by depth or RGBD cameras. Undoubtedly, if depth information is available, several cases that are difficult to solve using a conventional monocular tracking approach are disambiguated. Such a case arises when two persons occur merged in the RGB image but appear at different depths in the depth image, see i.e. [15]. Methods relying on 2D laser scans exploit similar information but are constrained to the plane that the scanning takes place. As an example, the method in [8] tracks multiple persons in real-time by maintaining the state of groups of people over time, considering possible splits and merges. A major disadvantage of these types of sensing is that the pertinent system is prone to the effects of outdoor illumination, a fact that limits their potential application domains.

Multiview approaches simplify localization because they acquire information from diverse perspectives and treat occlusions systematically. On the other hand, the large amount of data to be processed induces large computational demands, which are typically addressed through parallel and distributed methods. Also, communication bandwidth issues are raised because the input of more than a handful of cameras needs to be distributed to multiple computers or bus channels.

Multiview human localization methods perform 3D reconstruction of the imaged persons to register them to a map of the workspace. The method in [17] fuses the results obtained by existing single-view tracking methods that are applied individually to each of the views. However, the limitations of single view methods in handling occlusions still affect the fused results. The methods in [6, 12, 13, 4, 10], employ multiple views and a planar homography constraint to map imaged persons to the ground plane. In [7], a voxel grid is utilized to represent the 3D reconstruction and computation is distributed in the GPUs of four computers. For each voxel, a partial estimate of its occupancy is obtained, transmitted centrally, and fused with the rest of estimates for this voxel. Communication cost is significant as the amount of data to be communicated is proportional to the number of voxels. The system in [14] eliminates the communication cost by mounting all cameras to a single computer and centralizing computation. This approach does not scale with the number of views which, in this case, is limited to four.

This work follows a multiview approach to person tracking and utilizes a volumetric 3D reconstruction of persons to increase localization robustness, but it does not require that the number of tracked persons is a priori known, as in [4]. The proposed work capitalizes on computational efficiency in 3D reconstruction to facilitate person tracking, because as person motion becomes more densely sampled in time, it also becomes less ambiguous to track. Thereby, we adopt the work in [20] which provides person tracking based on efficient volumetric reconstruction achieved with the software platform presented in [21, 18]. In the current work, we additionally employ color information to extend the above approach in order to disambiguate challenging tracking cases. In addition, by recording the color appearance of the tracked targets, the tolerance to severe occlusions is increased. As a result, the requirement for a large number of cameras is relaxed.

3 Person localization and tracking

A set of calibrated cameras is employed, which image the scene from multiple views. One computer with programmable GPU hardware (with the option of sharing computational effort in more computers, if required [18]), is responsible for image acquisition, processing and extraction of a spatial representation of the persons in the room. In a typical setup, the cameras are placed evenly and high on the walls of the room, to overlook the scene. Henceforth, the images acquired synchronously at a given time instant are denoted is I_i, where i enumerates the cameras. The corresponding projection matrices of the cameras are denoted as P_i. In all experiments in this paper eight cameras were employed.

Two key components of the proposed approach are the volumetric reconstruction of the humans and the representation of their color appearance. The detection and localization of humans is based solely on volumetric information. In contrast, the detected persons are tracked based on both geometric and color information. The following sections describe these components in more detail.

3.1 Scene reconstruction

The persons in the scene are reconstructed in 3D, based on an estimation of their visual hull [9], which is represented as a mesh of triangle. The method computes the total visual hull of all persons, employing a voxel occupancy grid V.

Each time a synchronized set of images is acquired, a 3D reconstruction of the scene is performed. This computation has two stages. The first, concerns the volumetric reconstruction of persons. In the second stage, this reconstruction is enriched with color information, resulting in the colored visual hulls of these persons. Both stages are parallelized on the GPU.

3.1.1 Volumetric reconstruction

The first step towards scene reconstruction is to find which voxels of the environment are occupied by persons. The result is stored in the volumetric occupancy grid V, which represents this information.

The image I_i from camera i is read into the GPU and rectified to cancel lens distortions, based on the available calibration information. Foreground detection is performed based on a GPU implementation [18] of [23], which parallelizes computation at a pixel level. The outcome of foreground detection is binary image B_i. In contrast to [7, 14] and aiming at efficiency, segmentation errors are not smoothed (i.e. through morphological filtering), but taken into account at later stages.

The next step is the computation of the occupancy grid V. The value of each voxel of V is independently computed as follows. We refer to the 3D point $\vec{x}$ as being *potentially visible from view* i, if its projection $P_i(\vec{x})$, occurs within the field of view of camera i. If, ideally, no errors would occur during foreground segmentation, an occupied voxel $\vec{x}$ would project only at foreground regions of the views, let i', that it is potentially visible from. In this case, it would hold that:

$$s(\vec{x}) = \sum_{i'} \left(B_{i'}(P_{i'}(\vec{x}))\right) = \max(i'), \qquad (1)$$

for all voxel centers $\vec{x}$ within the visual hull of the person, while for any other location, $s(\vec{x})$ would have a smaller value. To compensate for errors in foreground segmentation we adopt the strategy in [20] and we consider a voxel as occupied if it projects to a foreground region in all but μ views that it is potentially visible from. Thus, $V(\vec{x})$ is set to 1 if it holds that $s(\vec{x}) \geq max(i') - \mu$ and to -1 otherwise. By employing this rule, up to μ views may have a foreground detection error at pixels $B_{i'}(P_{i'}(\vec{x}))$. In some cases this might dilate the visual hull by a voxel. Considering the intended use of the visual hull computation, we found this inaccuracy to be acceptable in terms of person localization. Conversely, if this relaxed constraint is not employed, a failure to segment a person from the background in a single view could annihilate its 3D reconstruction.

3.1.2 Surface and color reconstruction

In order to increase tracking robustness, the color appearance of humans is represented and utilized (see Sec. 3.3). To obtain this representation, triangles of the visual hull that are visible by the cameras are found and assigned color information, by means of texture mapping.

The surface of the visual hull is obtained as the 0-isosurface in V [22]. The isosurface is computed by a parallel implementation [18] of the “Marching Cubes” algorithm [11]. The 0-isosurface is encoded as a mesh M of triangles. Notice that M contains the surface of all persons in the scene, despite that these persons may not be in proximate locations with each other.

The next step is to compute the texture for each triangle in M. This is achieved by determining the texture that occurs on each triangle in M, from images I_i. Initially, the views from which a triangle j is visible are identified, by employing a depth buffer Z_i for each view i. Each

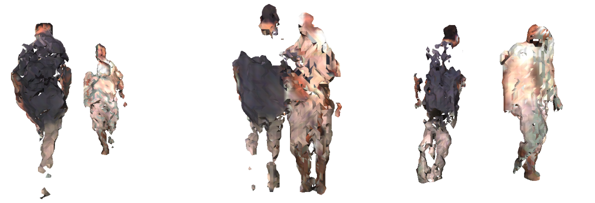

Figure 2. The textured visual hulls obtained for the frames in Fig. 1.

pixel in Z_i encodes the distance from the camera center $\vec{\kappa}_i$ to the surface that is imaged at that pixel. Buffer Z_i is computed by calculating the distance $\delta_{ij} = |\vec{\tau}_j - \vec{\kappa}_i|$ for each triangle j, where $\vec{\tau}_j$ is the triangle's 3D centroid. Triangles are projected on Z_i and the minimum distance that is imaged in each pixel of Z_i is assigned to that pixel. Let Δ be the length of a voxel's side. Then,

$$|\delta_{ij} - Z_i(P_i \cdot [\vec{\tau}_j; 1]^T)| < \Delta, \tag{2}$$

is a criterion that indicates if triangle j is indeed imaged at location of view i. The condition in Eq.(2) is false if triangle j is occluded in view i. Threshold Δ is sufficient as M's triangles are contained within voxel size. This criterion also facilitates parallel execution since, otherwise, the maintenance of the list of triangles imaged at $Z_i(P_i \cdot [\vec{\tau}_j; 1]^T)$ would be required to cope with pixels imaging multiple triangles. For efficiency, the number of considered triangles is reduced by disregarding those whose normal forms an angle greater than $\pi/2$ with the optical axis of view i. Texture coordinates of triangle nodes, $P_i \cdot [\vec{\tau}_j; 1]^T$, have been already computed during the evaluation of Eq.(2) and are retrieved.

To resolve multi-texturing conflicts in triangles visible from multiple views, a single view is selected to provide the texture for each triangle. More specifically, the view at which triangle j appears with the greatest area is selected. In this way, color samples originating from close and frontal views of a surface portion, imaging it in higher resolution, are preferred. Texture mapping is encoded through the texture coordinates of each mesh vertex. For simplicity, all images I_i are concatenated in a single image and texture coordinates refer to this image. Texture mapping is parallelized for each triangle j of M, in the GPU. In Fig. 2, the visual hulls obtained for the three frames of Fig. 1 are shown. Note that, due to errors in background subtraction and lack of visibility, these hulls can be far from complete. However, the reconstructed volume and corresponding textured surface provide sufficient information for person detection and tracking to be achieved.

3.2 Person detection and localization

Persons are localized based on the information provided in V. As in [6, 12, 13, 4, 10], a 2D image F is formed from V that is aligned to the ground plane and V. Essentially, F is a 2D histogram. A pixel (or bin) in F counts the occupied voxels in V along the direction perpendicular to the ground plane. Persons appear in F as intense and size-dominant blobs, with their intensities and areas proportional to the volume they occupy. The sum of intensities within a blob is proportional to the occupied volume by the visual hull that gives rise to the blob. Localizing a blob in F is equivalent to registering the location of each human in the ground floor reference frame.

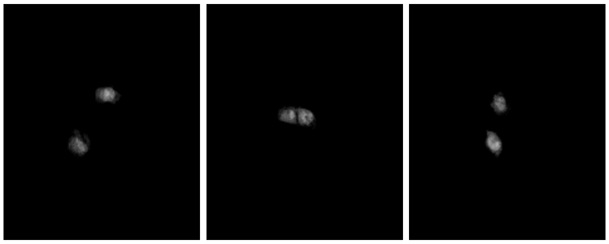

Figure 3. Visualization of data structure F_n for the 3 time instances shown in Fig. 1. In the images, intensity is mapped in proportion to the number occupied voxels (see text).

Depending on the number and the placement of the cameras, voxels of V are visible from a different number of cameras. In the corresponding regions, the signal in F becomes weaker as voxels are summarized along a smaller volume. To compensate for such variations, the values in F are normalized by the number of potentially visible voxels along height, which are potentially visible for that pixel of F. This process is efficiently implemented as follows. A normalization map S of equal dimensions to F is computed at initialization time. Each pixel in S summarizes the number of voxels accounted for the normalization. At run time, the value of a pixel $\vec{q}$ in F, for which $S(\vec{q}) \neq 0$, is normalized as $F_n(\vec{q}) = F(\vec{q})/S(\vec{q})$. Noise is initially suppressed in F_n by thresholding small values, followed by Gaussian smoothing. Figure 3 visualizes the content of F_n for the 3 time instances of the example in Fig. 1. In the resulting image, conventional connected component labelling is performed to detect blobs. Blobs that correspond to very small volumes are filtered out, as they are typically due to reconstruction errors. The detected blobs are directly localized as persons on the ground plane of the scene and their centroids and silhouettes are the measurements feeding the tracking module.

3.3 Person tracking

A blob tracker is applied to F_n in order to identify blobs and associate them with individual persons. More specifically, the tracker in [1] is modified to track intensity blobs in F_n, rather than skin-colored blobs in color images for which it was originally developed. This tracker may track a potentially varying number of targets and is robust to transient localization failures. Even more importantly, it is designed to retain the tracking of blobs even if they occur merged for long temporal intervals; in other words, the method keeps track of the number of persons, even if those appear as merged in a single blob in F_n. In this way, person tracking is becomes possible in cases where the persons come quite close and give rise to a single connected component in V and F_n (see Fig. 5).

Nevertheless, there exist scenarios that result in tracking failures. This happens, for example, when persons move very close to each other and interact strongly before they split again. Tracking failures are reinforced at areas covered by a few (or none) of the cameras. To enhance the robustness of the above method, the representation of the color appearance of each person is employed. The color appearance of each person is represented by a color histogram. The samples for this histogram are collected from the texture that is mapped on the portion of the visual hull that corresponds to the tracked blob in F_n (see Sec. 3.1.2). As the visual hull refers to the entirety of persons within the scene, the portion of the visual hull corresponding to the particular blob is isolated. This is achieved by selecting the triangles of M that are above the tracked blob.

After the triangles corresponding to each blob are selected, color values are sampled from them. To gather samples from a particular triangle of M, the pixels from the view that provides texture to this triangle are accessed, through their texture coordinates. The procedure is optimized using an OpenGL renderer which parallelizes the computation for each triangle in the GPU. Each pixel is considered as a sample, however its impact to the histogram is normalized by the area of the 3D triangle that it was acquired from. In this way, triangles contribute to the representation proportionally to their 3D surface area. In the histogram, the RGB samples are converted to the HSV colorspace with only the hue and saturation components being considered. Thus, the employed color histogram is two dimensional and only chromaticity plays a role in color representation. The value/intensity component is not considered in order to cope with illumination variations of the same physical point due to artifacts such as shadows, inconsistencies in brightness response among different cameras, etc. In Fig. 4, the acquired histograms for the two persons of the example of Fig. 1 are shown.

The obtained color information is employed in person tracking, for the disambiguation of proximate blobs. In particular, the tracker described above has been modified to prioritize the temporal correspondence of blobs with similar color appearance. Thus, when tracking blobs in F_n, a temporal correspondence of two proximate blobs will be established primarily based on the similarity of their associated color histograms. Nevertheless, if the color similarity does not provide sufficiently disambiguating information (i.e. if color similarity is equivalently high for all candidates) spatial continuity of blob motion is also considered to establish the temporal correspondence, as originally performed by the employed tracker.

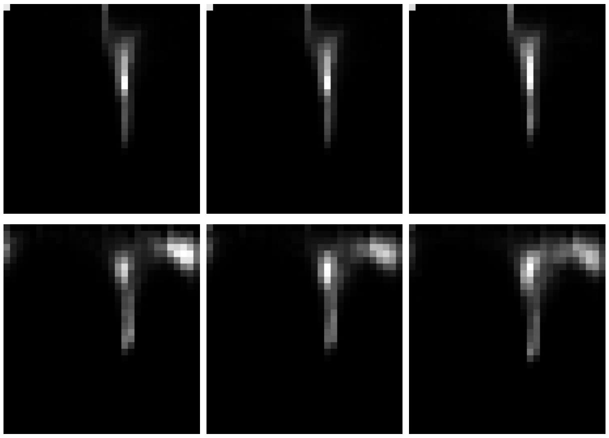

Figure 4. Color histograms for the two tracked persons of the example in Fig. 1, for the corresponding time instances (columns). The top row corresponds to the person with the light-colored clothing and the bottom row to the other.

The similarity of two color histograms h_1 and h_2 is quantified by a correlation metric, as

$$d(h_1, h_2) = \frac{\sum_j \left((h_1(j) - \alpha_1) \cdot (h_2(j) - \alpha_2)\right)}{\sqrt{\sum_j (h_1(j) - \alpha_1)^2 \sum_j (h_2(j) - \alpha_2)^2}}, \tag{3}$$

where

$$\alpha_k = \sum_j h_k(j) \tag{4}$$

and j enumerates the bins of the 2D histograms.

As persons are tracked through their blob representations, the associated histograms are continuously updated. The goal is to better capture the color appearance of each person, as the first frame at which a person appears might be insufficient for this purpose, i.e. due to restricted visibility, noise, or occlusions. At each frame that a person is tracked, the associated histogram is updated through a weighted averaging of two histograms: the one that has been computed up to the current time instant and the one computed at the current frame. The weights are in proportion to the confidence assigned to each histogram. As the former histogram has been maintained for a longer time interval it is assigned with a weight of n, equal to the number of frames that has been maintained. The latter histogram is then assigned with a weight of 1. Note, however, that the above update takes place only if the similarity of the two histograms is above the same threshold employed for person matching. The reason for this is to avoid a drift of the histogram representation in case of inaccurate reconstruction or tracking failure.

Figure 5 illustrates the tracker's result for the example of Fig. 1. In the left image, the two distinct blobs in F_n

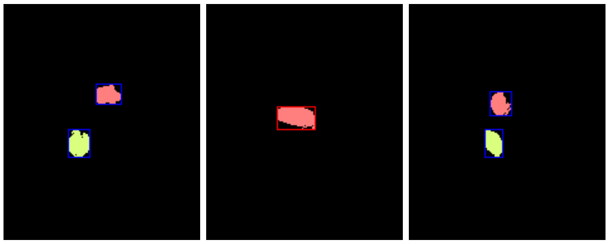

Figure 5. The internal representation of the blob tracker for the three time instances of the example of Fig. 1 (see text).

give rise to two tracking hypotheses. The middle image, corresponds to a frame acquired approximately $1.5\,sec$ later. In this frame, the two blobs are merged into one. This is acknowledged by the tracker which, nevertheless, maintains the two tracking hypotheses and their associated histograms (see Fig 4). The time instant corresponding to the right image occurred approximately $3.5\,sec$ later than the initial (left) image. In that image, the blob is observed to split into two smaller blobs. The maintained hypotheses are associated correctly to blobs based on color histogram similarity, as indicated by the color coding of blobs in the figure.

4. Experiments

The proposed method for 3D reconstruction and its implementation result in an efficient and scalable system. The achieved rate of updating V ($10\ - 30\,Hz$, depending on hardware configuration) allows for the assumption of motion continuity during tracking. At the same time, the system can support reconstructions of relatively high resolution, thereby supporting the disambiguation of proximate persons. This is important as the robustness of tracking also improves with increasing the density in temporal sampling, as well as with increasing the spatial resolution of the reconstruction. Observing the scene at a high framerate ($> 10\,Hz$) casts blob motion in F_n smooth and continuous and, thus, supports the unambiguous tracking of blobs. Fine granularity in occupancy estimation is important as proximate blobs will merge in F_n only if they occur closer than voxel size. In the experiments, a volume of $6 \times 6 \times 2m$ was covered with voxels of $1\,cm^3$, yielding a framerate of approximately $20Hz$. This volume was imaged by 8 cameras (*Dragonfly*, *Point Gray Research*) connected on a single computer, evenly distributed in 2 FireWire network cards. The computer is equipped with an *Intel i920* quad-core CPU, and a *NVIDIA GTX 580* programmable GPU. The cameras are mounted at the corners and at the in-between mid-wall points of the room viewing it in yaw-steps of $\simeq 45°$. The cameras are pointing at the floor center in a relative pitch of $-43°$, on average. The height of the mounting points is $\approx 2.6\,m$ from the ground.

The proposed method has been qualitatively and quantitatively evaluated on a number of datasets. During the experiments, the system was storing the acquired images which enabled the subsequent annotation of these data with ground truth. Using these annotations, the computation of the MOTA and MOTP tracking evaluation metrics [2], which measure tracking accuracy and precision respectively, was made possible.

System precision was quantified using the MOTP metric on datasets where persons were instructed to reach predefined locations that were marked on the floor, whose locations were a priori measured. For three and five persons, the localization error was in the order of $2 - 4\,cm$ respectively. The result was the same for both the proposed method and that of [20], as the motion of subjects did not include challenging person configurations. It is thus deducted that the achieved localization accuracy is sufficient with respect to localization requirements and to the spatial granularity of V.

Tracking accuracy was quantified with the MOTA metric on four datasets of increasing complexity. A baseline dataset $D0$ was recorded where a single person enters the setup, visits practically all of its locations, and exits. In the $D1$ dataset (662 frames, 1181 tracked objects), two persons walk together and then one of them walks at a corner of the room that constitutes a “blind” spot (not sufficiently visible to be reconstructed). The other person repeatedly walks towards and away from him. In the $D2$ dataset (904 frames, 1909 tracked objects), four persons perform walkthroughs in the imaged area. Often, a person stands at a corner of a room while another passes close by or is in contact, thus increasing the possibilities of a tracking mismatch. The $D3$ dataset (1011 frames, 4945 tracked objects) is overly challenging as seven persons move rapidly, getting in contact sometimes altogether and constantly occluding each other. We included this difficult dataset in order to observe the performance of the proposed tracking approach in very challenging situations. In Fig. 6, characteristic snapshots from these datasets are shown.

Based on the ground truth annotations, we measured the tracker’s misses, false positives and mismatches in all frames of datasets $D0$-$D3$. A distance threshold of $0.5\,m$ indicated whether a tracking error would be considered as a tracking miss rather than tracking imprecision. The proposed method was compared using the MOTA metric against the person tracking method in [20]. The latter method is similar to the proposed, but utilizes only the volumetric information availed by the visual hull of persons as represented in F_n. The results are summarized in Table 1. From left to right, data columns present the MOTA score, the percentage of “Misses”, “Mismatches”, and “False Positives” (see [2] for a definition of these quantities).

In $D0$, no misses, false positives, or mismatches occurred for both methods, even when the person entered and exited blind spots of the environment, for both datasets. This is due to the fact that the tracker copes with

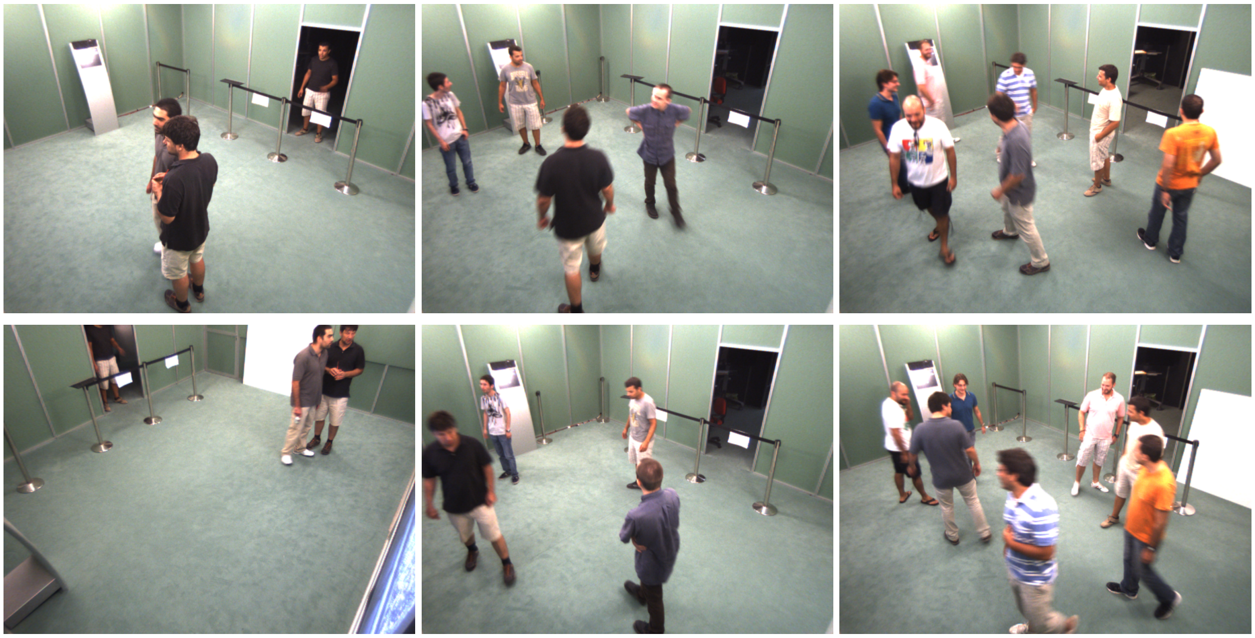

Figure 6. Characteristic snapshots from datasets $D1$, $D2$ and $D3$ (left, middle and right column, respectively).

Table 1. Quantitative evaluation of tracking performance of the proposed method against the method described in [20] (see text).

	Method	MOTA	Miss	MM	FP
D0	[20]	1.000	0.000	0.000	0.000
	This work	1.000	0.000	0.000	0.000
D1	[20]	0.981	0.000	0.007	0.011
	This work	0.987	0.000	0.001	0.011
D2	[20]	0.922	0.036	0.101	0.040
	This work	0.986	0.007	0.020	0.002
D3	[20]	0.681	0.340	0.252	0.310
	This work	0.712	0.149	0.019	0.147

transient reconstruction failures (i.e. as when a person walks through a region that is not sufficiently covered by cameras) and can achieve this even using only volumetric information. Hence MOTA was 1 for this dataset, in both conditions of the experiment. In the remaining conditions, the proposed method consistently outperformed [20], as a result of the additional information that it utilizes. In particular, the MOTA score is consistently higher in all conditions of the experiment, while it also produces as smaller proportion of errors. In $D1$ and $D2$, although the tracker is able to correctly assign the ids, there exist occasions where lack of visibility produces false-positive detections (assignment of a tracking hypothesis to a non-existing person in the scene), for both methods. These false-positives are mainly due to a volumetric artifact occurring at the dynamic occlusion of an area, which has been studied in [5]. The increased complexity of $D3$ results in lower MOTA values for this dataset, for both methods. It is interesting, however, to observe that for all datasets, and particularly $D2$ and $D3$ which are the most complex ones, the number of mismatches (column MM) are fewer for the proposed method. This is a direct result of the disambiguation of persons due to color information. Conversely, the method in [20] which uses only geometric information fails to a larger extent, because mismatches occur when persons are lost by the tracker in a low visibility region of the scene and are mismatched with another person close by.

5. Conclusions

In this work, we presented a method for tracking multiple humans based on a camera network. The proposed approach performs a colored visual hull reconstruction of persons within the imaged environment. Human tracking is then performed based on the obtained volumetric and color information. Color information was availed through the surface reconstruction of the tracked persons. Through quantitative experiments, we have shown that adding color to the information that is utilized to track persons increases tracking robustness compared to an existing, state-of-the-art method that uses geometric information, only. Both methods are characterized by computational efficiency. This is achieved by parallelizing the pertinent computation on graphics hardware (GPU). This efficiency has an impact on tracking robustness, because the temporal sampling of the environment yields minute motions of the tracked persons. As a result, temporal continuity is preserved and the data association problem becomes easier to solve.

One key direction for future work concerns the ability of the system to operate in less constrained environments. In this respect, we are aiming towards the possibility of coping with changes in the background, by dynamically updating its model. Another research direction regards the inclusion of mobile robots in the environment. Extending the proposed framework to benefit from the moving cameras mounted on the robots constitutes a challenging research goal. In this context, we intend to study the reliability and bandwidth of the required wireless communication interface, as well as, its potential for synchronized image acquisition and real-time operation.

Acknowledgments

This work was partially supported by the FORTH-ICS internal RTD Programme "Ambient Intelligence and Smart Environments", by the EU-IST-FP7-288917 project DALi, as well as, by 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-DISFER.

References

[1] A. Argyros and M. Lourakis. Real time tracking of multiple skin-colored objects with a possibly moving camera. In *European Conference on Computer Vision*, pages 368–379, 2004.

[2] K. Bernardin and R. Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal of Image and Video Processing*, 2008, 2008.

[3] T. Darrell, G. Gordon, M. Harville, and J. Woodfill. Integrated person tracking using stereo, color, and pattern detection. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 601–609, 1998.

[4] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):267–282, 2008.

[5] G. Haro and M. Pardàs. Shape from incomplete silhouettes based on the reprojection error. *Image and Vision Computing*, 28(9):1354–1368, 2010.

[6] S. Khan and M. Shah. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *European Conference on Computer Vision*, pages 133–146, 2006.

[7] A. Ladikos, S. Benhimane, and N. Navab. Efficient visual hull computation for real-time 3D reconstruction using CUDA. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8, 2008.

[8] B. Lau, K. O. Arras, and W. Burgard. Multi-model hypothesis group tracking and group size estimation. *International Journal of Social Robotics*, 2(1):19–30, 2010.

[9] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994.

[10] M. Liem and D. Gavrila. Multi-person tracking with overlapping cameras in complex, dynamic environments. In *British Machine Vision Conference*, 2009.

[11] W. Lorensen and H. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Special Interest Group on GRAPHics and Interactive Techniques*, pages 163–169, 1987.

[12] A. Mittal and L. Davis. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. In *International Journal of Computer Vision*, pages 189–203, 2003.

[13] D. Reddy, A. Sankaranarayanan, V. Cevher, and R. Chellappa. Compressed sensing for multi-view tracking and 3-D voxel reconstruction. In *IEEE International Conference on Image Processing*, pages 221–224, 2008.

[14] A. Schick and R. Stiefelhagen. Real-time GPU-based voxel carving with systematic occlusion handling. In *DAGM Symposium on Pattern Recognition*, pages 372–81, 2009.

[15] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[16] S. Tran, Z. Lin, D. Harwood, and L. Davis. UMD VDT, an integration of detection and tracking methods for multiple human tracking. In *Multimodal Technologies for Perception of Humans*, volume 4625/2008 of *Lecture Notes in Computer Science*, pages 179–190. Springer, 2008.

[17] A. Tyagi, M. Keck, J. Davis, and G. Potamianos. Kernel-based 3D tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[18] K. Tzevanidis, X. Zabulis, T. Sarmis, P. Koutlemanis, N. Kyriazis, and A. Argyros. From multiple views to textured 3D meshes: a GPU-powered approach. In *European Conference on Computer Vision Workshops*, pages 5–11, 2010.

[19] B. Wu, V. Singh, C. Kuo, L. Zhang, S. Lee, and R. Nevatia. CLEAR'07 evaluation of usc human tracking system for surveillance videos. In *Multimodal Technologies for Perception of Humans*, volume 4625/2008 of *Lecture Notes in Computer Science*, pages 191–196. Springer, 2008.

[20] X. Zabulis, D. Grammenos, T. Sarmis, K. Tzevanidis, P. Padeleris, P. Koutlemanis, and A. A. Argyros. Multicamera human detection and tracking supporting natural interaction with large-scale displays. *Machine Vision Applications*, 24(2):319–336, 2013.

[21] X. Zabulis, T. Sarmis, K. Tzevanidis, P. Koutlemanis, D. Grammenos, and A. A. Argyros. A platform for monitoring aspects of human presence in real-time. In *International Symposium on Visual Computing*, 2010.

[22] X. Zabulis, T. Sarmis, K. Tzevanidis, P. Koutlemanis, D. Grammenos, and A. A. Argyros. A platform for monitoring aspects of human presence in real-time. In *International Symposium in Visual Computing*, pages 584–595, 2010.

[23] Z. Zivkovic. Improved adaptive Gaussian mixture model for background subtraction. In *International Conference on Pattern Recognition*, pages 28–31, 2004.