Foreground Regions Extraction and Characterization towards Real-Time Object Tracking

José Luis Landabaso and Montse Pardàs

Technical University of Catalunya, Barcelona, Spain
jl@gps.tsc.upc.edu, montse@gps.tsc.upc.edu

Abstract. Object localization and tracking are key issues in the analysis of scenes for video surveillance or scene understanding applications. This paper presents a contribution to the object tracking task in indoor environments surveyed by multiple fixed cameras. The method proposed uses a foreground separation process at each camera view. Then, a 3D-foreground scene is modeled and discretized into voxels making use of all the segmented views, preventing the difficulties of inter-object occlusions in 2D trackers, and increasing the robustness for not having to rely only in one view. The voxels are grouped into meaningful blobs, whose colors are modeled for tracking purposes, using a novel voxel-coloring technique that considers possible inter/intra-object occlusions. Finally, color information together with other characteristic features of 3D object appearances are temporally tracked using a template-based technique which takes into account all the features simultaneously in accordance with their respective variances. Extensive experiments dealing with several hours of video sequences in real-world scenarios have been conducted, showing a very promising performance.

1 Introduction

One of the important objectives of image and video analysis is the development of accurate and robust tracking techniques for multiple moving objects in dynamic and cluttered visual scenes. It is particularly desirable in the video surveillance field where an automated system allows fast and efficient access to unforeseen events that need to be attended by security guards or law enforcement officers. It also enables tagging and indexing interesting scene activities / statistics in a video database for future retrieval on demand. In addition, such systems are the building blocks of higher-level intelligent vision-based or assisted information analysis and management systems with a view to understanding the complex actions, interactions, and abnormal behaviors of objects in the scene.

Vision-based surveillance systems can be classified in several different ways, considering the environment in which they are designed to operate. In this paper our focus is on processing videos captured by multiple fixed camera overlooking indoor areas in visual monitoring scenarios.

Multiple camera surveillance has two key advantages over single camera systems. First, the occlusion problem automatically vanishes when using enough cameras. And second, the process gains robustness for not having to rely only in one camera.

There have been several attempts to fuse video information from different cameras. Some approaches start with the assumption that the scene develops in flat areas with large distances between objects and cameras. The tracking becomes then only a problem of 2D localization in a plane. In such situations it is commonplace to project tracking regions from one camera view to another [1]. The process, known as homography between images, can be employed to select which projection is used based on the localization of the object, in order to avoid occlusions present in a camera but not in others.

Although homographic transformations between images have proved to solve some problems, they fail when the assumptions of large distances and flat areas do not hold, such as indoor scenarios. To overcome this limitation, there have been some works which try to do a 2D-based tracking and then fuse the results into a 3D space; and others which try to fuse 3D features first, to use them later in a single tracker.

1.1 Our Approach

We focus on the second approach. In particular, we propose using the camera views to extract foreground voxels, i.e., the smallest distinguishable box-shaped part of a three-dimensional image. Indeed, foreground voxels provide enough information for precise object detection and tracking. Furthermore, there are several alternatives for the voxel extraction process, such as laser range scanners that although providing very precise volumetric information, suffer from very low scanning rates, making them unsuitable for our application. Other non-invasive reconstruction methods use intensity-based techniques [2] that compute correspondences across images and then recover the 3D structure by triangulation and surface fitting. Unfortunately, for effective operation of these techniques the camera views must be close so that the correspondence is effective. Besides, a huge number of points have to be usually matched and fused into a consistent model, making it a slow and difficult task.

Instead, we propose using shape from silhouette, which is another non-invasive and faster technique. A calibrated [3] set of cameras must be placed around the scene of interest, and the camera pixels must be provided as either part of the shape (foreground) or background. Each of the foreground camera point defines a ray in the scene space that intersects the object at some unknown depth along this ray; the union of these visual rays for all points in the silhouette defines a generalized cone within which the 3D object must lie. Finally, the object is guaranteed to lie in the volume defined by the intersection of all the cones. The main drawback of the method is that it doesn't always capture the true shape of the object, as concave shape regions are not expressed in the silhouettes. However, this is not a severe problem in a tracking application as the aim is not to reconstruct photorealistic scenes.

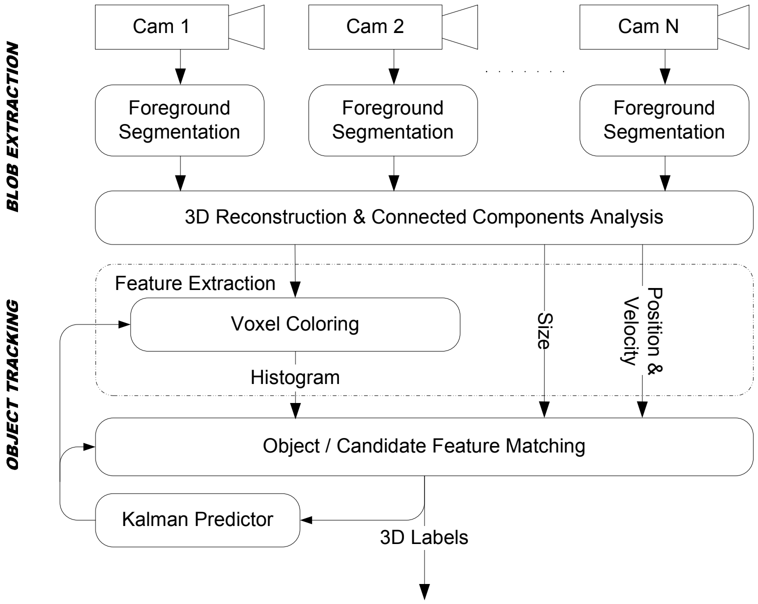

Fig. 1. The system block diagram showing the chain of functional modules

After the voxelization process (see figure 1), a connected component analysis *CCA* is followed to cluster and label the voxels into meaningful 3D-blobs, from which some representative features are extracted. Finally, there is a template-based matching process aiming to find persistent blob correspondences between consecutive frames.

The paper is structured as follows. In the next section the techniques for pixel-domain analysis leading to the segmented foreground views are described. Section 3 is devoted to discussion on issues concerning 3D-blob extraction, including the voxelization process and the voxel coloring. Section 4 describes the object tracking approach adopted. Section 5 illustrates the experimental evaluations of the system. And, finally the paper concludes in Section 6 .

2 2D Foreground Segmentation

The 2D foreground extraction technique that we have used [4–6] is based on the adaptive background subtraction method proposed by Stauffer and Grimson [7]. A mixture of K Gaussian distributions is used to model RGB color changes, at each pixel location, in the imaging scene over the time. With each incoming frame the Gaussian distributions are updated, and then used to determine which pixels are most likely to result from a background process. This model allows a proper representation of the background scene undergoing slow lighting and scene changes as well as momentary variations.

The foreground pixels thus obtained, however, are not exempt from false detections due to noise in the background and camera jitters. A false-foreground pixels suppression procedure is introduced to alleviate this problem. Basically, when a pixel is initially classified as a foreground pixel, its 8-connected neighboring pixels' models are examined. If the majority of these models, when applied to this pixel, agree that it's a background pixel, then it's considered as a false detection and removed from foreground.

Once the foreground objects pixels have been identified, an additional scheme [5] is applied to find out if some of these foreground pixels correspond to areas likely to be cast shadows or specular reflections. The working mechanism of this novel scheme is the following:

As the first step, we evaluate the variability in both brightness and color distortion [8] between the foreground pixels and the adaptive background, and possible shadows and highlights are detected. It was observed though that this procedure is less effective in cases that the objects of interest have similar colors to those of presumed shadows. To correct this, an assertion process comparing the gradient / textures similarities of the foreground pixels and corresponding background is incorporated. These processing steps, effectively removing cast shadows, also invariably delete some object pixels and distort object shapes. Therefore, a morphology-based conditional region growing algorithm is employed to reconstruct the object's shapes. This novel approach gives favorable results compared to the current state-of-the-art to suppress shadows / highlights.

3 3D Blob Extraction

Once the foreground region has been extracted in each camera view, the blobs in the 3D space are constructed. In our implementation, the bounding volume (the room) is discretized into voxels. Each of the foreground camera points defines a ray in the scene. Then, the voxels are marked as *occupied* when there are intersecting rays from enough cameras $MINC$ over the total N.

The relaxation in the number of intersecting rays at a voxel prevents typical missing-foreground errors at the pixel level in a certain view, consisting in foreground pixels incorrectly classified as background. Besides, camera redundancy also prevents analog false-foreground errors, since a wrongly defined ray in a view will unlikely intersect with at least $MINC - 1$ rays from the rest of the cameras at any voxel.

3.1 Voxel Connectivity Analysis

After marking all the *occupied* voxels, with the process described above, a connectivity analysis is performed to detect clouds of connected voxels, i.e. 3D-blobs, corresponding to tracking targets. We choose to group the voxels with 26-connectivity which means that any possible contact between voxels (vertices, edges, and surfaces) makes them form a group. Then, from all the possible blobs, we consider only the ones with a number of connected voxels greater than a certain threshold *B_SIZE*, to avoid spurious detections.

3.2 Voxel Coloring

After voxel grouping, the blobs are characterized with their color (dominant color, histogram, histogram at different heights, etc.), among other features. This characterization is employed later for tracking purposes. However, a trustworthy and fast voxel coloring technique has to be employed before any color extraction method is applied to the blob.

We need to note that during the voxelization and labeling process, inter/intra-object occlusions are not considered, as it is irrelevant whether the ray came from the occluded or the occluding object. However, in order to guarantee correct pixel-color mapping to visible voxels in a certain view, occlusions have to be previously determined.

We discard slow exhaustive search techniques, which project back all the *occupied* voxels to all the camera views to check intersecting voxels along the projection ray. Instead, for the sake of computational efficiency, we propose a faster technique, making use of target localization, which can be obtained from the tracking system.

As photorealistic coloring is not required in our application, intra-object occlusions are simply determined by examining if the voxel is more distant to the camera than the centroid of the blob the voxel belongs to. On the other hand, inter-object occlusions in a voxel are simply determined by finding objects (represented by their centroid) in between the camera and the voxel. This is achieved by computing the closest distance between the segment voxel-to-camera and the objects' centroids ($dist(\underline{\mathbf{vc}}, \mathbf{o_c})$). The process is schematized in the Voxel-Blob level in figure 2.

To reduce even further the computational complexity, the voxels can be approximated by the position of the centroid of the blob they belong to, as it's shown in the Blob level in figure 2, and intra-object occlusions are not examined.

Finally, the color of the voxels is calculated as an average of the projected colors from all the non-occluding views.

4 Object Tracking

After labeling and voxel coloring, the blobs are temporally tracked throughout their movements within the scene by means of temporal templates.

Each object of interest in the scene is modeled by a temporal template of persistent features. In the current studies, a set of three significant features are used for describing them: the velocity at its centroid, the volume, and the histogram. Therefore at time t, we have, for each object l centered at (p_{lx}, p_{ly}, p_{lz}), a template of features $M_l(t)$. Prior to matching the template l with a candidate

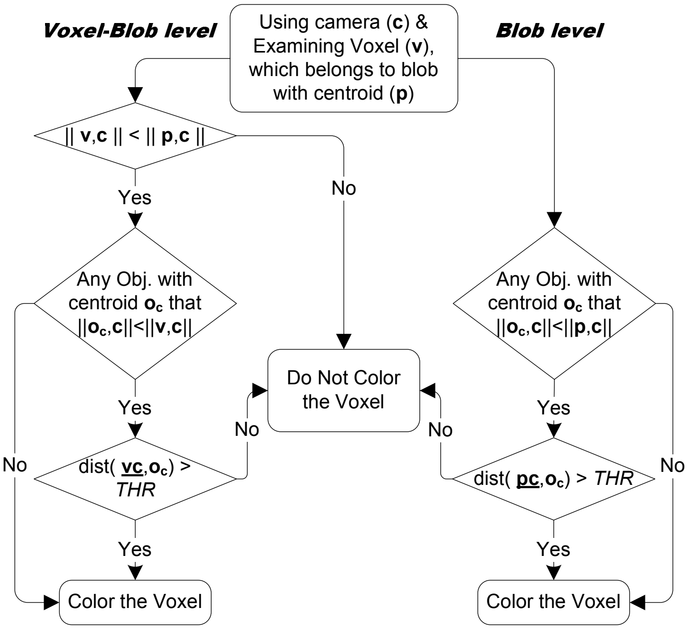
Voxel-Blob level
Using camera (c) & Examining Voxel (v), which belongs to blob with centroid (p)
Blob level
|| v,c || < || p,c ||
Yes
No
Any Obj. with centroid o_c that $||o_c,c||<||v,c||$
Yes
No
dist(vc,o_c) > THR
No
Yes
Color the Voxel
Do Not Color the Voxel
Any Obj. with centroid o_c that $||o_c,c||<||p,c||$
Yes
No
dist(pc,o_c) > THR
No
Yes
Color the Voxel

Fig. 2. Voxel Coloring block diagram, showing the two proposed methods. On the left, the Voxel-Blob level, which addresses voxel coloring individually. On the right, a faster approach using only the centroids of the blobs.

blob k in frame $t+1$, centered at $(p'_{kx}, p'_{ky}, p'_{kz})$ with a feature vector $B_k(t+1)$, Kalman filters are used to update the template by predicting its new velocity and size in $\hat{M}_l(t+1)$. The mean $\overline{M_l}(t)$ and variance $V_l(t)$ vector of the templates are updated when a candidate blob k in frame $t+1$ is found to match with it. The updates are computed using the latest corresponding L blobs that the object has matched.

For the matching procedure we choose to use a parallel matching strategy. The main issue is the use of a proper distance metric that best suits the problem under study. The template for each object being tracked has a set of associated Kalman filters that predict the expected value for each feature (except for the histogram) in the next frame. Obviously, some features are more persistent for an object while others may be more susceptible to noise. Also, different features normally assume values in different ranges with different variances. Euclidean distance does not account for these factors as it will allow dimensions with larger scales and variances to dominate the distance measure.

One way to tackle this problem is to use the Mahalanobis distance metric, which takes into account not only the scaling and variance of a feature, but also the variation of other features based on the covariance matrix. Thus, if there are correlated features, their contribution is weighted appropriately.

However, with high-dimensional data, the covariance matrix can become non-invertible. Furthermore, matrix inversion is a computationally expensive process, not suitable for real-time operation. So, in the current work a weighted Euclidean distance between the template l and a candidate blob k is adopted, assuming a diagonal co-variance matrix. For a heterogeneous data set, this is a reasonable distance definition. Further details of the technique have been presented in the past [4].

5 Results

The voxelization and tracking methods have been evaluated extensively using, among others, our own recordings at the *UPC* smart-room and the benchmarking video sequences provided by the *CHIL* project [9]. The *CHIL* sequences are provided with manually labeled tags of the tracking target corresponding to thousands of frames of seminar presentations in a smart room.

The room discretization was done using $5 \times 5 \times 5$ cm^3 cubes. During the voxelization process we used 4 cameras, accepting voxel reconstruction with at least $MINC = 3$ intersecting rays. Blobs with B_SIZE lower to 700 were filtered out and voxel coloring was performed with the Blob-level faster approach, setting $THR = 40$ cms.

Under the above mentioned conditions, the voxelization and tracking process performs at 5 fps; with an average tracking error under 20 cms (see the complete results in Table 1).

30 minutes of video	Error	Results with Error > 30 cms
Results	148.2 mms	3.8%

Table 1. First column shows the mean of Euclidian distance between the estimated position of the centroid, and the ground truth of the head center. Note that for this evaluation, not 3D distances are used, but rather the 2D distance between the projection on the ground of the estimated head centre and that of the ground truth labels. The second column expresses the percentage of frames where the distance between the estimated distance and the ground truth was worse than 30 cms.

The algorithm performs extremely well except in object grouping situations, not being able to segment them. In spite of that, the tracker is able to recover the correct tags after the objects ungroup. Some videos are available in our web at: http://gps-tsc.upc.es/imatge/_jl/Tracking.html

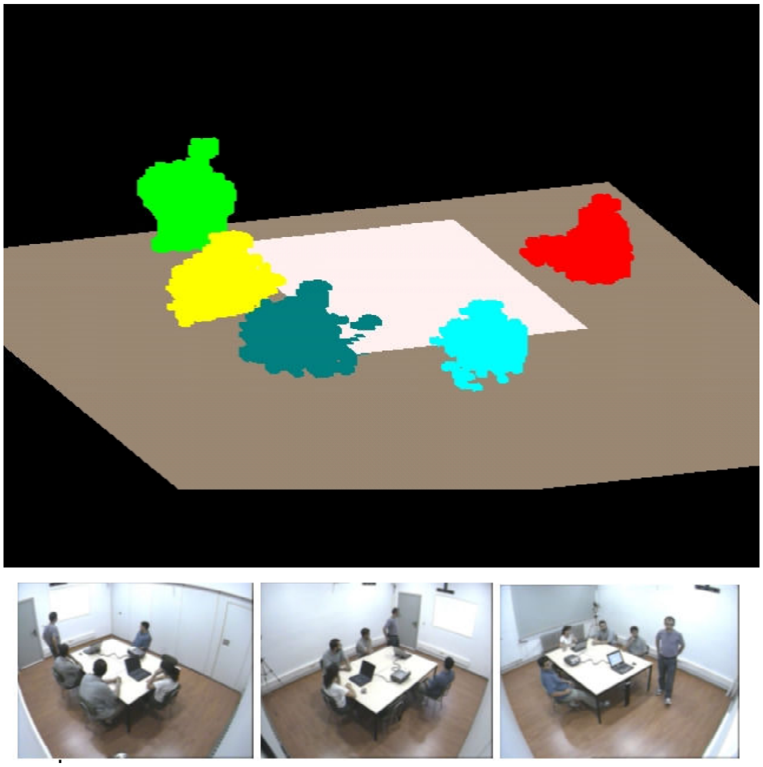

Fig. 3. Voxel reconstruction and labeling of a video sequence recorded at the *UPC* smart-room

6 Conclusions and Future Work

In this paper, we have presented a system able to create a 3D-foreground scene, characterize objects with 3D-blobs and track them, preventing the difficulties of inter-object occlusions in 2D trackers, and increasing the robustness for not having to rely only in one view. The system uses a novel voxel coloring scheme which allows fast object histogram retrieval used later with other features in a parallel matching technique during the tracking.

Some of the directions to take to improve results include projecting back the 3D-blobs to assist the foreground segmentation technique. Also, dynamic adjustment of the required number of intersecting rays at a voxel *MINC* will be investigated. The parameter may be set depending on the position of the tracking target, allowing tracking in areas where only fewer cameras have visibility.

Acknowledgments

This material is based upon work partially supported by the European Union under the Integrated Project *CHIL IST-2004-506909* and by the Spanish Ministry of Education under actions *TIN2004-0171-E* and *TEC2004-01914*

References

1. Black, J., Ellis, T., Rosin, P.: Multi view image surveillance and tracking. Proceedings of the Workshop on Motion and Video Computing 2002
2. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press 2000
3. Zhang, Z.: A flexible new technique for camera calibration. Technical report, Microsoft Research, August 2002
4. Landabaso, J.L., Xu, L-Q., Pardàs, M.: Robust Tracking and Object Classification Towards Automated Video Surveillance. Proceedings of ICIAR **2** 2004 463–470
5. Xu, L-Q., Landabaso, J.L. Pardàs, M.: Shadow removal with blob-based morphological reconstruction for error correction. Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on, **Vol.2**, Iss., March 18-23, 2005 729–732
6. Landabaso, J.L., Pardàs, M., Xu, L-Q.: Hierarchical representation of scenes using activity information. Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on, **Vol.2**, Iss., March 18-23, 2005 677–680
7. Stauffer, C., Grimson, W.E.L.: Learning patterns of activity using real-time tracking. IEEE trans. on Pattern Analysis and Machine Intelligence, **22(8)**, August 2000
8. Horpraset, T., Harwood, D., Davis, L.: A statistical approach for real-time robust background subtraction and shadow detection. Proceedings of International Conference on Computer Vision, 1999
9. *CHIL* project home page: http://chil.server.de