Modeling Dynamic Scenes Recorded with Freely Moving Cameras

Aparna Taneja, Luca Ballan, and Marc Pollefeys

ETH Zurich

Abstract. Dynamic scene modeling is a challenging problem in computer vision. Many techniques have been developed in the past to address such a problem but most of them focus on achieving accurate reconstructions in controlled environments, where the background and the lighting are known and the cameras are fixed and calibrated. Recent approaches have relaxed these requirements by applying these techniques to outdoor scenarios. The problem however becomes even harder when the cameras are allowed to move during the recording since no background color model can be easily inferred.

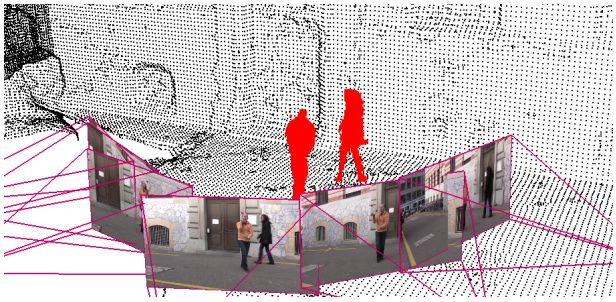
In this paper we propose a new approach to model dynamic scenes captured in outdoor environments with moving cameras. A probabilistic framework is proposed to deal with such a scenario and to provide a volumetric reconstruction of all the dynamic elements of the scene.

The proposed algorithm was tested on a publicly available dataset filmed outdoors with six moving cameras. A quantitative evaluation of the method was also performed on synthetic data. The obtained results demonstrated the effectiveness of the approach considering the complexity of the problem.

1 Introduction

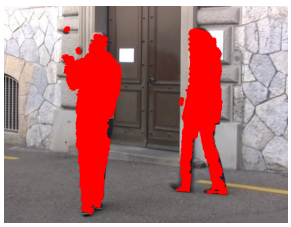
Passive modeling of dynamic scenes is a challenging problem in computer vision. The aim is to recover a mathematical time-varying description of the scene using only videos recorded by some cameras. A considerable number of approaches have been developed in the past to address such a problem. Typically these techniques exploit the use of silhouette [1–3], color/stereo [4–7], shading [8, 9] and motion [10] extracted from the videos in order to infer the geometry of the dynamic elements of the scene. In the case of silhouette based techniques, the geometry of the dynamic objects is recovered using either deterministic [1, 11] or probabilistic [3, 2] visual hull. Color information can be exploited by using either multi-view stereo [6, 7] or narrow baseline stereo [4], or combining both together as proposed in [5]. Silhouette and color information can also be combined to improve the reconstruction results [12–14].

However, most of these works focus on controlled environments where the background is known or can be estimated and the cameras are fixed and calibrated. Only few approaches have tried to deal with outdoor scenarios mainly

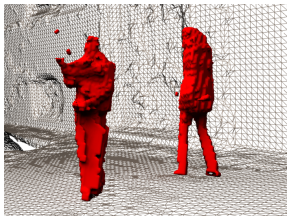


(a)

(b)



(c)



(d)

Fig. 1. Results obtained using our approach to model a dynamic scene with two people: one walking and the other juggling. (a) Volumetric reconstruction. (b) One frame of the videos used for the reconstruction. (c) Reconstructed volume projected back to the previous image. (d) Reconstruction rendered from another viewpoint.

resorting to some scene priors [15] or limiting the reconstruction quality at billboard level [16]. In particular, [15] performs on the assumption that a person is the only one dynamic element in the scene.

Unlike the above approaches, in this paper, we propose a technique to achieve full 3D reconstruction of the scene dynamics in outdoor uncontrolled environments filmed with moving cameras and without making any assumptions on the shape or the motion of elements to be reconstructed. In a sense, our approach can be considered similar, at least in principle, to a silhouette based approach.

Silhouette based approaches rely on the possibility of performing background subtraction on the entire video sequence. This is an easy task in a controlled environment but becomes hard in the more generic case of an outdoor scenario. In [2], for instance, the authors addressed such a scenario but assuming stationary cameras. The problem indeed, becomes even more challenging in the case of moving cameras since a per-pixel color model for the background cannot be recovered anymore. Some relevant works focusing on background subtraction have been developed in the recent past to address this kind of situations. However, these techniques resort to some priors on both the background and the foreground elements of the scene such as shape priors [17, 18], color priors [19, 16] and motion priors [20]. The first class assumes that the foreground objects

can only have specific shapes, for instance, human shapes. The second class assumes that the color models of the foreground and the background objects are known a priori. The last class instead makes assumptions on the type of motion of the dynamic elements. As an example, [20] assumes that the elements that are moving rigidly, with respect to camera, are background while the others are foreground.

In this paper we propose a technique to infer the geometry of the dynamic elements of a scene by exploiting the structural information of the static parts of the scene, which is inferred in a preprocessing stage, and the color information from the acquired video sequences.

To avoid building a per-pixel background color model from temporal video data for segmentation, we instead use the precomputed geometry of the static parts of the scene to transfer the current background appearance across multiple views. Given some images captured at the same time instant, our approach is based on projecting each image onto the other images and exploiting their differences. Something similar was partially exploited by [21] to achieve a deterministic and fast background subtraction of a person using three static cameras. These projections however generate some false detections which in our text will be referred as the *ghost* of the foreground (occlusion shadows in their paper). While in [21], the method simply eliminates these artifacts by intersecting all the reprojections, we instead exploit this information as well in a probabilistic framework. As described later in the paper, the ghost may help recover in some situations where no information can be obtained from the actual location of the dynamic object.

Since only the images captured at the same time instant are used to model the current scene we do not suffer from some issues that are common in background subtraction techniques such as changes in illumination or shadows.

Compared to the approach proposed in [16], where the authors suggest to retrieve the background color by exploring the temporal domain of each video independently, our approach exploits the spatial domain, retrieving this information from the other cameras at the same time instant. Moreover, in this approach we do not need an initial color model for the foreground which had to be specified by a user in [16].

This paper is organized in three parts. Section 2 describes the proposed reconstruction algorithm. Section 3 shows the experimental results. In the end, Section 4 draws the conclusions.

2 Reconstruction Procedure

The captured videos are first pre-processed in order to retrieve information about the cameras and the static elements of the scene. Subsequently, the geometries of all the dynamic elements of the scene are reconstructed. This section describes how the pre-processing stage is performed while the next section covers the reconstruction of the dynamic elements. For the sake of simplicity we refer to the static part of the scene as background and the dynamic part as foreground.

Structure-from-Motion (SfM) [22, 23] and multi-view stereo [6] can be applied to some images of the scene, captured in absence of any dynamic elements, in order to recover the background geometry. In our implementation we used the pipeline provided by Zach et al. [24] which generates a continuous mesh model for the background.

Each video camera is then calibrated both spatially and photometrically, and the video streams synchronized. To do so, we follow the approach described in [16]. More specifically, intrinsic parameters are recovered using [25] and they are assumed to be constant throughout the recording. Subsequently the pose, i.e., the extrinsic parameters, for each camera at each time instant, are computed with respect to the background geometry by matching the SIFT descriptors [26] extracted from the current frame and the SIFT descriptors previously extracted during the SfM procedure. These matches generate correspondences between 2D points in the current frame and 3D points in the background geometry. The pose of that camera at that specific time is recovered by applying the three points algorithm [27]. Temporal synchronization of the videos stream is performed using the corresponding audio streams as in [15].

Finally, the video streams are calibrated photometrically with respect to each other using the method proposed in [28]. More specifically a color transfer function mapping the color space of one camera into the color space of another is recovered for each pair of cameras. This is necessary to account for different settings in the cameras like different exposure time, gain and white balancing.

Our formulation is designed to estimate the 3D reconstruction of a single frame. For sake of simplicity, from here on, the analysis will focus only on a specific time instant t and the text will refer to images captured by the cameras as the images captured at that specific time t.

Let I_i denote the image captured by camera $i \in [1, .., n]$, and let π_i be the projection function mapping 3D points in the world coordinate system to 2D points in the image coordinate system of camera i according to both the intrinsic and the extrinsic parameters recovered during the previous stage.

Since both the background geometry and the projection function π_i are known, the depth map of the background geometry seen by camera i can be computed. Let's denote this depth map with Z_i. The value stored in each of its pixels represents the depth of the closest 3D point of the background geometry that projects to that pixel using π_i. In practice, Z_i can be easily computed in GPU by rendering the background geometry from the point of view of camera i and by extracting the resulting Z-buffer.

Given two cameras i and j, let R_{ij} denote the image obtained by projecting the image I_j into camera i, i.e., by rendering the background geometry from the point of view of camera i using the color information of camera j and taking into account the color transfer function between i and j. More formally, for each pixel p in R_{ij}, we know that $\pi_i^{-1}([p, Z_i^p]^T)$ represents the coordinates of the closest 3D point in the background geometry projecting in p. Note that π_i^{-1} is the inverse of the projection function π_i where the depth is assumed to be known and equal

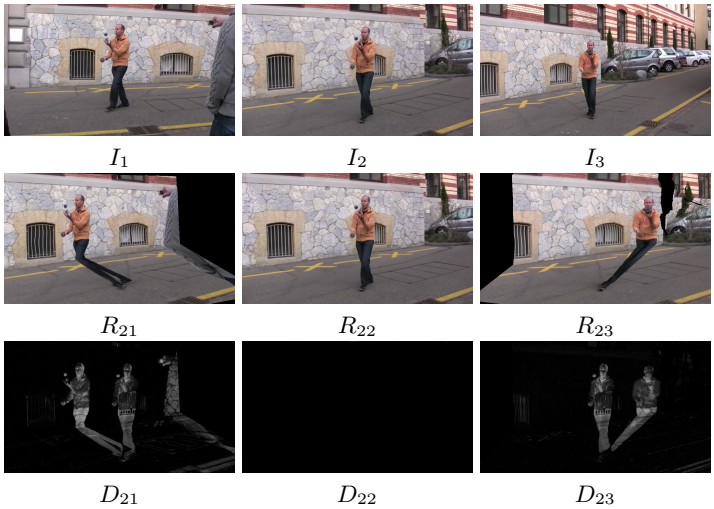


Fig. 2. (Top row) Source images acquired respectively by camera 1, 2 and 3. (Middle row) Images R_{ij} computed by projecting the previous images into camera 2 (black pixels indicate missing color information, i.e., $\alpha = 0$). (Bottom row) Difference images D_{ij}. (Best viewed in color)

to Z_i^p. Therefore, the coordinates of pixel p in the image j are equal to

$$\pi_j(\pi_i^{-1}([p, Z_i^p]^T)) \tag{1}$$

In the end, the color of the pixel p in R_{ij} is defined as follows

$$R_{ij}^p = I_j(\pi_j(\pi_i^{-1}([p, Z_i^p]^T))) \tag{2}$$

Let us note that no color information can be retrieved for pixels of R_{ij} that map outside the field of view of camera j and also for those which have no depth information in Z_i, e.g., for those projecting onto regions not modeled by the background geometry. We keep track of such pixels by defining a binary mask α_{ij} such that, $\alpha_{ij}^p = 0$ indicates the absence of color information at pixel p in R_{ij}. The procedure of computing R_{ij} is performed in GPU using shaders.

Figure 2 shows some example images R_{ij} obtained by projecting the images captured by three different cameras, namely #1, #2 and #3, into the camera #2. The background geometry, in this case, models both the building and the street but it does not include the juggler. The reader can notice that, when the background geometry matches the current scene geometry the captured image I_i and the image R_{ij} look alike in all the pixels with α_{ij}^p equal to one. On the contrary, if the current scene geometry includes an additional object which was not present in the background geometry, this gets projected into the background points behind it. We refer to this reprojection as the *ghost* of the foreground object in the image R_{ij}. Figure 3 explains this concept visually. In Figure 2, the

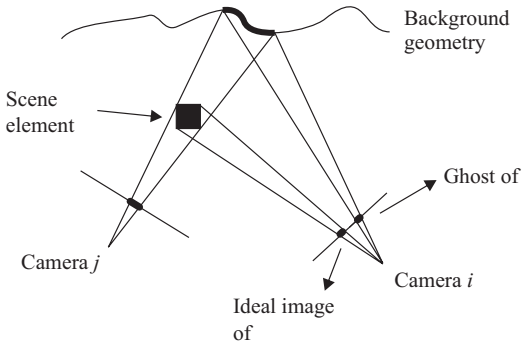


Fig. 3. Image formation process for a reprojection image R_{ij}. Since the scene element γ is not a part of the background geometry, it generates a ghost image on camera i which is far away from the region it should ideally project to if it were a part of the background geometry.

ghost of the juggler can be observed in both images R_{21} and R_{23} while it's not visible in R_{22} since the image is projected on itself.

By visually comparing R_{ij} and I_i, one can observe differences in the pixels belonging to foreground elements as well as in the pixels belonging to their ghosts. Let's call D_{ij} the image obtained by a per-pixel comparison between image I_i and image R_{ij}. In order to make our comparison method robust to errors that may be present in either the calibration or in the background geometry, the similarity measure used to compare these two images takes into account for local affine transformations in the image space. We propose to compute D_{ij} as

$$D^p_{ij} = \min_{q \in W_p} (\left\| I^p_i - R^q_{ij} \right\|) \tag{3}$$

where W_p is a window around p and $\|\cdot\|$ is the L^1 norm in the RGB color space. This similarity measure proved to be more robust but, unfortunately, some details around the ghost borders are lost. This can be seen in Figure 4(a) where the ghost of the foreground object gets shrunk by half the window size used. In order to avoid these artifacts, the same approach is repeated by comparing, this time, the pixel p in R_{ij} to a corresponding window W_p in I_i. A result obtained by using this second approach is shown in Figure 4(b) where, this time instead, the silhouette of the foreground object gets shrunk by half the window size. In the end we chose to use the following metric which combines the advantages of the both the previous metrics:

$$D^p_{ij} = \max(\min_{q \in W_p} (\left\| I^p_i - R^q_{ij} \right\|), \min_{q \in W_p} (\left\| R^p_{ij} - I^q_i \right\|)) \tag{4}$$

A result obtained by applying this new metric can be seen in Figure 4(c).

Given the input images I_i, all the possible images D_{ij} for each $i > j$ are computed. This leads to a set of $(n^2 - n)/2$ difference images D_{ij} that we will

(a) (b) (c)

Fig. 4. Results obtained by applying different color similarity measures to compare the two images I_j and R_{ij} in order to build the image D_{ij}. (a) Result obtained by applying the Equation 3. (b) Result obtained by applying the Equation 3 with I_j and R_{ij} swapped. (c) Result obtained by applying the Equation 4.

refer to as D. In the next paragraph the problem of recovering the 3D geometry of the foreground object is formulated in a probabilistic way using as observation the computed set of images D.

The scene to be reconstructed is discretized as a voxel grid. Let V be the random vector representing the occupancy state of all the voxels inside this grid where $V_k = 1$ indicates the voxel k is full and empty otherwise. The aim of our algorithm is to find a labeling L^* for V which maximizes the posterior probability $P(V = L|D)$, i.e.,

$$L^* = \arg\max_L P(V = L|D) \tag{5}$$

By the Bayes' rule, this is equivalent to

$$L^* = \arg\max_L (\log(P(D|V = L)) + \log(P(V = L))) \tag{6}$$

We first describe how the probability $P(D|V = L)$ is computed for a given labeling of the voxel grid, while $P(V = L)$ is described later.

Let ϕ_i^k denote the footprint of the voxel k in camera i, i.e., the projection of all the 3D points belonging to k onto the image plane of camera i. Furthermore, denote with χ_{ij}^k the set of the ghost pixels of voxel k in the image R_{ij}. Since these pixels are the ones corresponding to the background geometry points occluded by the foreground object in camera j, i.e. $\pi_j^{-1}([\phi_j^k, Z_j^{\phi_j^k}]^T)$, χ_{ij}^k can be computed as follows

$$\chi_{ij}^k = \pi_i(\pi_j^{-1}([\phi_j^k, Z_j^{\phi_j^k}]^T)) \tag{7}$$

i.e., by projecting those background points into camera i (See Figure 3).

We make three conditional independence assumptions for computing the probability $P(D|V = L)$: first, the state of the voxels are assumed to be conditionally independent; second, the image formation process is assumed to be independent for the all images and third, the color of a pixel in an image is independent from the others. Using these assumptions, the probability $P(D|V = L)$ can be expressed as

$$P(D|V = L) = \prod_k P(D|V_k = L_k) \tag{8}$$

where

$$P(D|V_k) = \prod_{i,j,p} P(D_{ij}^p|V_k) \quad \forall p \in \phi_i^k \cup \chi_{ij}^k \tag{9}$$

Let us now introduce another random variable C_{ij} representing the consensus between the pixels in image I_i and the ones in image R_{ij}. $C_{ij}^p = 1$ indicates that the color information at pixel p in I_i agrees with the color information at p in R_{ij}. Clearly, this variable strongly depends on the image D_{ij}.

Specifically, $P(D_{ij}^p|V_k)$ is modeled using a formulation similar to the one proposed by Franco and Boyer in [3], i.e.,

$$\begin{aligned} P(D_{ij}^p|V_k) = P(D_{ij}^p|C_{ij}^p = 1)P(C_{ij}^p = 1|V_k) + \\ P(D_{ij}^p|C_{ij}^p = 0)P(C_{ij}^p = 0|V_k) \end{aligned} \tag{10}$$

While in their work they used background images to determine $P(D_{ij}^p|C_{ij}^p)$ we assume the following: in case of consensus ($C_{ij}^p = 1$) the probability of D_{ij}^p being high is low and vice versa. Therefore $P(D_{ij}^p|C_{ij}^p = 1)$ is chosen to be a Gaussian distribution truncated for values smaller than 0. Concerning the pixels with no color information, i.e., the ones with $\alpha_{ij}^p = 0$, we assume this probability to be uniform and therefore,

$$P(D_{ij}^p|C_{ij}^p = 1) = \begin{cases} TG(D_{ij}^p) & \alpha_{ij}^p = 1 \\ U & \alpha_{ij}^p = 0 \end{cases} \tag{11}$$

where $TG(d)$ is the truncated Gaussian function and U the uniform distribution.

On the contrary, when there is no consensus ($C_{ij}^p = 0$) no information can be stated for D_{ij}^p and therefore $P(D_{ij}^p|C_{ij}^p = 0)$ is set to uniform distribution.

$P(C_{ij}^p = 1|V_k)$ and $P(C_{ij}^p = 0|V_k)$ are defined in a similar way as in [3] but while in their formulation, the state of the voxel k is influenced only by the background state of the pixels in ϕ_i^k, in our formulation its state is also influenced by the pixels in χ_{ij}^k. While this property adds additional dependence between the voxels, it provides more information on the state of each voxel. In fact, we not only rely on the consensus observed in the voxel's footprint ϕ_i^k but also on the consensus observed in χ_{ij}^k.

This allows us to recover from two kinds of situations, namely: when the colors of the foreground object are similar to the colors of the actual background points behind it, and when the information corresponding to the foreground object in the image R_{ij} is missing. However, our approach will not help if the colors of the actual background points in χ_{ij}^k are also similar to the colors of the foreground element.

Concerning $P(V = L)$ we assume dependency only between neighboring voxels. In this way, Equation 6 can be entirely solved using graph cuts [29–31]. More precisely, the pairwise potential $\log(P(V_a = L_a, V_b = L_b))$ between two neighboring voxels a and b is defined considering that if these voxels project to pixels lying on edges of the original images I_i there should be a low cost for cutting across these voxels and viceversa. To account for this, in our implementation, we compute the projection of the centers of each pair of neighboring voxels a

and b on each image I_i. Subsequently we check all the pixels on the line connecting these two projections looking for an edge. If an edge is not found then the pairwise potential is increased.

To account for temporal continuity in the final mesh the voxel state prior takes into account its labeling computed in the previous frame according to $P(V_a = 1) = 0.3 + \xi(L_a^{*,t-1})$ where ξ defines the temporal smoothness. Once graph cuts provides a grid labeling L^* as a solution for Equation 6, marching cubes [32] is applied to obtain a continuous mesh of the dynamic object.

3 Results

The algorithm was tested on both real and synthetic data. For the real data test, we used a publicly available dataset provided by [16] where a juggler was filmed outdoors by six people holding cameras while some other people were walking by. Video streams have a resolution of 960×544 pixels at 25 fps. Background geometry was obtained using SfM+MVS on the available images of the dataset while the cameras were calibrated both photometrically and spatially using the techniques described in Section 2. About 300 frames of this sequence were processed by our method using a voxel grid of resolution $140 \times 140 \times 140$ covering the entire extent of the scene where the action took place.

Figures 1(a,d) show one reconstructed frame of this sequence where two persons are present in the scene. Figure 1(c) shows the reconstructed volume projected onto one of the cameras superimposed with the corresponding captured image. As the reader may notice, our system is also able to recover the shape of the balls being juggled by the performer. This however happens only in half of the reconstructed frames since motion blur is explicitly present in such parts of the image. Some more results are shown in Figures 5. Figures 5(e) and 5(f) show two situations where the algorithm does not work properly. In these two cases, the person walking behind is not visible in one of the views and is also occluded completely by the juggler in some of the other views. This leads to a noisy reconstruction.

This sequence was processed on a 2.93GHz Intel i7 computer with a NVIDIA GTX 285. For the chosen grid resolution, each frame took around 45 seconds to process. The current implementation however does not have any major optimizations, in fact only some parts of the code were implemented on GPU.

The results obtained for the juggler sequence were also compared with the ones obtained by applying standard background subtraction on the videos and then applying deterministic visual hull. The texture of the background geometry was estimated from the images used during the preprocessing stage. However, even small changes in the illumination or shadows in the scene did not allow us to infer accurate silhouettes for the performer. This is not an issue in our approach since only images taken at the same time instant are used for comparison.

The results were also compared with two state of the art techniques namely [20] and [16] but they were not convincing from a reconstruction point of view. In fact, [16] focuses on segmentation rather than reconstruction since that would

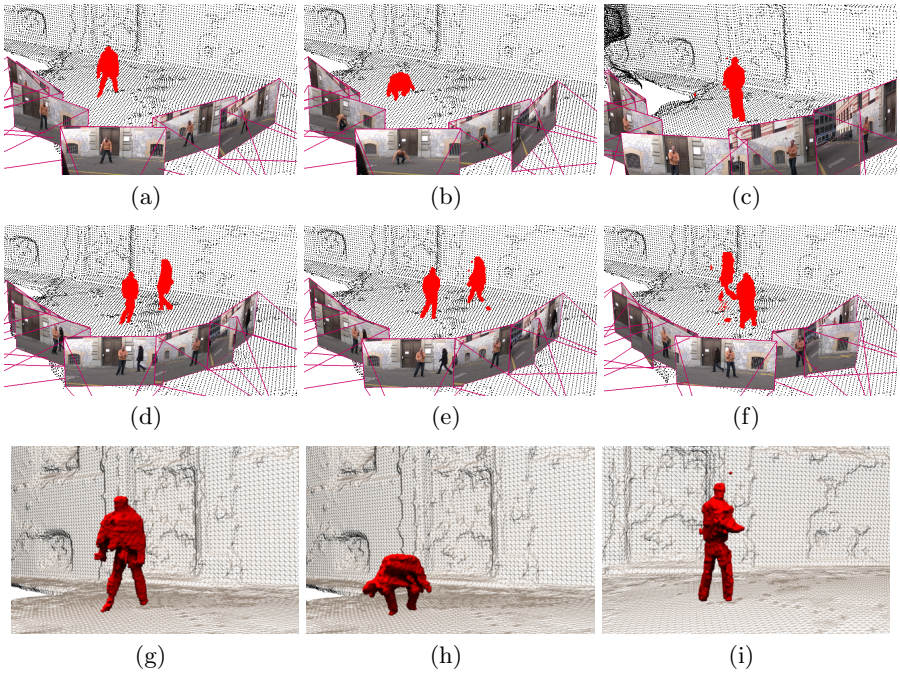


Fig. 5. (a-d) Results obtained using our approach. (e) and (f) show two cases where the algorithm does not behave properly due to strong occlusions between the two persons. (g,h,i) Reconstruction rendered from different viewpoints for the results in the top row. (Best viewed in color)

be too sensitive to segmentation errors. A user interaction is also needed to label both foreground and background in some video frames. [20] assumes that the foreground is moving relative to the background, i.e., it is not moving rigidly with respect to the camera. However, this approach may fail in detecting objects or body parts moving slowly. This occurred frequently in the juggler sequence since, while the performer was juggling fast there was not much movement around his legs, and therefore they were often misclassified as background in the output obtained using [20]. Compared to the manually segmented silhouettes of the foreground objects [20] misclassified 25% of the pixels on an average while by projecting the volume computed with our method only 1% of the pixels were misclassified.

Some tests were performed on synthetic data to provide a quantitative evaluation of the algorithm. Using a commercial software, we rendered a scene with two balls bouncing in the center of a room filmed by 7 cameras moving in circle at a distance of 3m from the center of the action. The field of view of the cameras was 42° and the resolution of the video streams was 800×600. At first, the dataset reveals to be very simple and the algorithm performed an almost

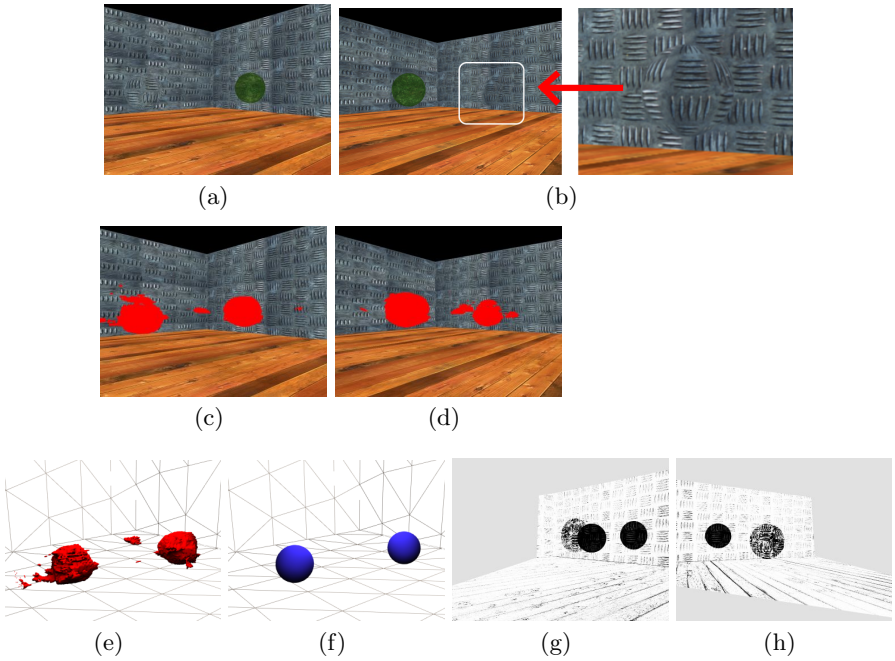


Fig. 6. (a,b) Two images from the synthetic sequence rendered from different cameras at the same time. The gray ball is barely distinguishable in these images. (c,d) Reconstructed volume projected back to the corresponding images in the above row. (e) Reconstructed volume. (f) Ground truth. (g,h) Two images D_{ij} computed during the shape estimation (the colors are inverted for visibility). (Best viewed in color)

perfect reconstruction of the scene dynamics, obviously up to the chosen grid resolution. Therefore we rendered the dataset again introducing some ambiguities, more precisely, we textured the walls of the room with the same texture as one of the balls. Two frames of this new sequence are shown in Figures 6(a,b). As the reader can notice, even for a human it is difficult to visually distinguish between the gray ball and the gray wall.

A color based segmentation/visual hull technique will, in this case, either consider the entire wall as foreground or completely background, in both cases resulting in a bad reconstruction.

On the contrary, our reprojection based approach together with the robustness of the probabilistic framework was able to recover a reasonable reconstruction of the scene, as can be seen in Figure 6(e). For a visual comparison, the ground truth is shown in Figure 6(f).

The main reason why such a reconstruction can be achieved can be explained by looking at one of the D_{ij} images shown in Figures 6(g,h). While for a color based approach it is not feasible to distinguish between foreground and background in the case of the gray ball, if the texture of the background is provided

by another camera some discrepancies between this texture and the observed image can be measured. A similar result can also be obtained if a per pixel color model of the background is available for each camera. However, since the cameras are moving this model cannot be easily retrieved. Figures 6(c,d) show the reconstructed volume projected back to the respective original images.

We ran our algorithm on the full sequence consisting of 15 frames and the computed reconstructions were compared with the ground truth. At each frame, the error between the two models was evaluated numerically by measuring the average euclidean distance between the two surfaces. The average error for the whole sequence was 2cm, which corresponds to the used voxel size. The standard deviation for this error was 1.8cm. Note that, by definition the metric that we are using does not account for the sparse blobs in the reconstruction.

This error increases if we introduce inaccuracies in the background geometry and in the camera calibration. We ran the test again after adding Gaussian noise to the camera position with a standard deviation of 1.6cm and a uniformly distributed noise of $\pm$8cm to the background geometry. The average reconstruction error increased to 3.6cm, where the majority of the error was induced from the errors in calibration and not from errors in the geometry.

4 Conclusions

In this paper we proposed a new technique to model dynamic scenes in outdoor uncontrolled environments filmed with freely moving cameras. A probabilistic framework is proposed to deal with such a scenario and to provide a volumetric reconstruction of all the dynamic elements. The method exploits the structural information of the static parts of the scene, inferred in a preprocessing stage, to transfer the current background appearance across multiple cameras. Hence, it avoids the need to build a per-pixel background color model from temporal video data for segmentation, which is very challenging for scenes recorded with moving cameras.

Tests on synthetic data revealed a reconstruction accuracy of 2cm for footage filmed by 0.5MPixels cameras placed at a distance of 3m from the objects to be reconstructed. This error is relatively low considering the challenges present in the used dataset such as multiple occlusions and similar background/foreground colors (see Figure 6). Our approach reveals to be robust enough to deal with such ambiguities and also with calibration and geometry inaccuracies to an extent.

Experiments on real data proved the ability of our approach to recover the geometries of multiple dynamic objects filmed outdoors with freely moving cameras (see Figure 1). The reconstruction accuracy is not comparable with the one that other techniques can obtain for indoor controlled environments with static cameras. However, it must be noted that the scenario we used for our tests is much more challenging.

There are three main limitations of our approach. First, the algorithm depends on the possibility of estimating the color transfer function between the cameras which, in our case, was performed using a rather simple technique [28].

This works well in the tested sequences but, in the future, for a more generic scenario we should resort to a more complex calibration technique, like [33].

The second limitation is the resolution of the voxel grid which cannot be increased indefinitely without considering calibration and background geometry errors. This limitation however, does not prevent us from recovering the small balls being juggled by the performer in half of the frames of the real data sequence.

The method inevitably inherits the limitations of the visual hull techniques on the class of reconstructible objects, i.e., it is not able to recover concave parts of the object if these concavities are not visible in at least one camera.

As a future extension, we plan to consider inside the proposed probabilistic framework other kinds of depth cues, like multiview stereo and narrow baseline stereo. A synergical fusion of these information will help overcome the last two limitations as well as increase the reconstruction accuracy.

Acknowledgements

We would like to thank Christopher Zach and David Gallup for their valuable help. The research leading to these results has received funding from the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013)/ERC grant#210806.

References

1. Kim, H., Sarim, M., Takai, T., Guillemaut, J.Y., Hilton, A.: Dynamic 3d scene reconstruction in outdoor environments. In: 3DPVT. (2010)
2. Guan, L., Franco, J.S., Pollefeys, M.: Multi-object shape estimation and tracking from silhouette cues. In: CVPR. (2008)
3. Franco, J.S., Boyer, E.: Fusion of multi-view silhouette cues using a space occupancy grid. In: ICCV. (2005) 1747–1753
4. Furukawa, Y., Ponce, J.: Dense 3d motion capture for human faces. In: CVPR. (2009) 1674–1681
5. Tung, T., Nobuhara, S., Matsuyama, T.: Complete multi-view reconstruction of dynamic scenes from probabilistic fusion of narrow and wide baseline stereo. In: ICCV. (2009)
6. Seitz, S., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: CVPR. (2006)
7. Seitz, S.M., Dyer, C.R.: Photorealistic scene reconstruction by voxel coloring. In: CVPR. (1997) 1067
8. Vlasic, D., Peers, P., Baran, I., Debevec, P., Popović, J., Rusinkiewicz, S., Matusik, W.: Dynamic shape capture using multi-view photometric stereo. SIGGRAPH Asia (2009)
9. Ahmed, N., Theobalt, C., Dobrev, P., Seidel, H.P., Thrun, S.: Robust fusion of dynamic shape and normal capture for high-quality reconstruction of time-varying geometry. In: CVPR. (2008)
10. Vedula, S., Baker, S., Seitz, S., Kanade, T.: Shape and motion carving in 6d. In: CVPR. (2000)

11. Matusik, W., Buehler, C., Raskar, R., Gortler, S.J., McMillan, L.: Image-based visual hulls. In: SIGGRAPH, New York, NY, USA, ACM Press (2000) 369–374
12. Goldlucke, B., Ihrke, I., Linz, C., Magnor, M.: Weighted minimal hypersurface reconstruction. PAMI (2007) 1194–1208
13. Hilton, A., Starck, J.: Multiple view reconstruction of people. 3DPVT (2004)
14. Sinha, S.N., Pollefeys, M.: Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. ICCV (2005) 349–356
15. Hasler, N., Rosenhahn, B., Thormahlen, T., Wand, M., Gall, J., Seidel, H.P.: Markerless motion capture with unsynchronized moving cameras. In: CVPR. (2009)
16. Ballan, L., Brostow, G.J., Puwein, J., Pollefeys, M.: Unstructured video-based rendering: Interactive exploration of casually captured videos. SIGGRAPH (2010)
17. Baumberg, A., Hogg, D.: An efficient method for contour tracking using active shape models. In: Motion of Non-Rigid and Articulated Objects. (1994) 194–199
18. Leibe, B., Cornelis, N., Cornelis, K., Gool, L.V.: Dynamic 3d scene analysis from a moving vehicle. In: CVPR. (2007)
19. Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.S.: Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proceedings of the IEEE **90** (2002) 1151–1163
20. Sheikh, Y., Javed, O., Kanade, T.: Background subtraction for freely moving cameras. In: ICCV. (2009)
21. Ivanov, Y., Bobick, A., Liu, J.: Fast lighting independent background subtraction. International Journal of Computer Vision **37** (2000) 199–207
22. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. Second edn. Cambridge University Press, ISBN: 0521540518 (2004)
23. Pollefeys, M., Van Gool, L., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. IJCV **59** (2004) 207–232
24. Zach, C., Pock, T., Bischof, H.: A globally optimal algorithm for robust TV-L^1 range image integration. In: ICCV. (2007)
25. Zhang, Z.: A flexible new technique for camera calibration. PAMI **22** (2000)
26. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60** (2004) 91–110
27. Haralick, R.M., Lee, C.N., Ottenberg, K., Nölle, M.: Review and analysis of solutions of the three point perspective pose estimation problem. IJCV **13** (1994)
28. Reinhard, E., Adhikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. Computer Graphics and Applications, IEEE **21** (2001) 34–41
29. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI **23** (2001) 1222–1239
30. Kolmogorov, V., Zabin, R.: What energy functions can be minimized via graph cuts? PAMI **26** (2004) 147–159
31. Boykov, Y., Kolmogorov, V.: An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision. PAMI **26** (2004) 1124–1137
32. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. SIGGRAPH **21** (1987) 163–169
33. Kim, S., Frahm, J., Pollefeys, M.: Radiometric calibration with illumination change for outdoor scene analysis. In: CVPR. (2008) 1–8