

3D Reconstruction of Background and Objects Moving on Ground Plane Viewed from a Moving Camera

Chang Yuan and Gérard Medioni

Institute of Robotics and Intelligent Systems

University of Southern California, Los Angeles, CA 90089, USA

{cyuan, medioni}@usc.edu

Abstract

We present a novel method to obtain a 3D Euclidean reconstruction of both the background and moving objects in a video sequence. We assume that, multiple objects are moving rigidly on a ground plane observed by a moving camera. The video sequence is first segmented into static background and motion blobs by a homography-based motion segmentation method. Then classical “Structure from Motion” (SfM) techniques are applied to obtain a Euclidean reconstruction of the static background. The motion blob corresponding to each moving object is treated as if there were a static object observed by a hypothetical moving camera, called a “virtual camera”. This virtual camera shares the same intrinsic parameters with the real camera but moves differently due to object motion. The same SfM techniques are applied to estimate the 3D shape of each moving object and the pose of the virtual camera. We show that the unknown scale of moving objects can be approximately determined by the ground plane, which is a key contribution of this paper. Another key contribution is that we prove that the 3D motion of moving objects can be solved from the virtual camera motion with a linear constraint imposed on the object translation. In our approach, a planar-translation constraint is formulated: “the 3D instantaneous translation of moving objects must be parallel to the ground plane”. Results on real-world video sequences demonstrate the effectiveness and robustness of our approach.

1 Introduction

We study the video sequences shot by a perspective camera undergoing general motion. Objects are moving rigidly on a ground plane. There might also exist some static structure out of the ground plane. A typical scenario is that of vehicles moving on the ground observed by an airborne camera, with buildings and other 3D structures present. Our goal is to obtain a 3D Euclidean reconstruction of both the

static background and moving objects.

The problem of reconstructing a static rigid scene observed by a moving camera, usually called “Structure from Motion (SfM)”, has been thoroughly studied in the past decades [10, 15, 5, 18, 8]. Recently, reconstruction of multiple moving objects observed by a moving camera has also gained much attention [2, 6, 7, 22]. There are also approaches for aligning moving objects to the static background [17]. However, a pixel-level motion segmentation step before 3D reconstruction was not integrated into the previous systems. Furthermore, as far as we know, there is no closed-form solution to 3D object motion estimation in the case of moving perspective cameras and non-linear trajectories.

We aim to build a practical system to integrate the merits of previous approaches and address the problems. The system performs the following tasks on video sequences:

- **Motion segmentation:** to segment images into motion blobs and static background
- **3D reconstruction of static background:** to estimate camera poses and structure of the static background
- **3D reconstruction of moving objects:** to infer shape and motion of moving objects relative to the background

At the current stage, object shape refers to sparse 3D points. All the structure and motion are obtained in the Euclidean space, up to a global scale.

The ground plane, which exists naturally in real-world sequences, plays an important role in our approach. It induces 2D homographies between consecutive frames which serve as the basis for motion segmentation. Camera self-calibration is done by exploiting the coplanarity of points on the plane. The plane also helps in approximating the unknown scale of 3D object shape and uniquely solving for the 3D object motion. This ground plane is found by manually selecting at least four pairs of matched points in the first two frames and automatically tracking them across the frames. More feature points consistent with the homography are added and tracked as well.

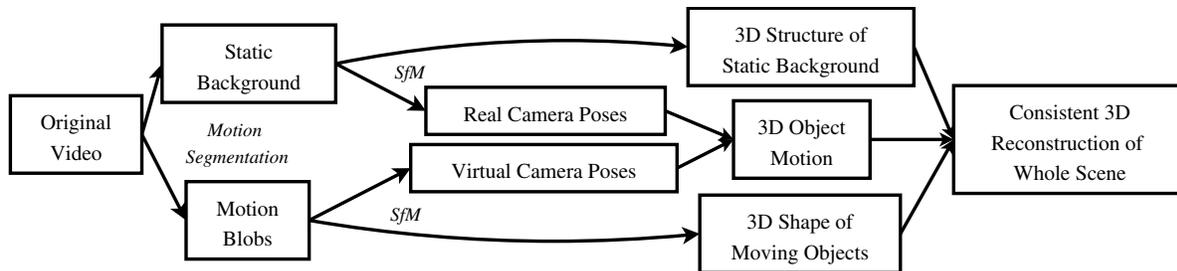


Figure 1. The flowchart of our approach

The flowchart of our approach, as shown in Figure 1, starts with homography-based motion segmentation [12]. Pixels inconsistent with the homographies induced by the ground plane are defined as residual pixels. The residual pixels include those belonging to static out-of-plane structure, termed “motion parallax”. After removing the parallax pixels [9, 19, 12], we obtain a number of motion blobs as well as static background in each frame.

Classical SfM techniques are then applied to obtain a Euclidean reconstruction of the static background. A plane-based self-calibration method [14] is applied to estimate both intrinsic and extrinsic parameters of the camera. With the calibrated cameras, the structure of static background is computed by efficient triangulation methods [16, 18]. The whole set of camera poses and scene structure are further refined by a “Bundle Adjustment” process [21].

Inferring the 3D shape of moving objects is done in a similar way. Indeed, the 2D motion blob is generated by a moving object observed by a moving camera. Alternatively, the blob can be treated as if a static object were observed by a virtual camera. The virtual camera shares the same intrinsic parameters with the real camera but moves differently to take into account object motion. The same SfM techniques are applied to estimate the shape of the moving objects as well as the poses of virtual camera.

The scale of 3D object shape is approximately determined by the ground plane. If one object point is known to be on the ground, the object scale is uniquely solved as a minimum scale. Otherwise, we assume that at least one object point is close to the ground plane. In either case, the object scale is estimated, which can be considered as stretching the object until one point falls onto the ground plane.

We propose a novel method to infer 3D object motion from the poses of both real and virtual cameras. It is known that the scale ambiguity in virtual camera motion brings difficulty to object motion estimation [17]. However, we prove that if an additional linear constraint on object translation is introduced, the object motion can be solved uniquely and linearly. In our approach, this constraint is formulated as a planar-translation constraint: *the 3D instantaneous translation of moving objects must be parallel to the ground plane.*

This constraint is more flexible than the assumption of linear trajectories proposed in previous methods [2, 7].

The rest of the paper is organized as follows. In Section 2, we relate our work to the previous methods. Section 3 presents the methods for segmenting moving objects from the static background. Section 4 and Section 5 present similar techniques for reconstructing the static background and moving objects. The problem of object motion estimation is addressed in Section 6. Experimental results are given in Section 7. We conclude the paper and discuss the directions for future work in Section 8.

2 Related work

Classical “SfM” techniques can be divided into two categories: batch(global) or recursive (causal) approaches. The batch process assumes that all the images and feature correspondences are known. Multi-view geometric constraints are estimated from feature correspondences, using the epipolar constraint [18] and trifocal tensor [15]. Then the scene structure and camera poses are estimated in projective and Euclidean space by various methods, as summarized in [8]. On the other hand, recursive approaches [10, 5, 4] assume that image measurements are available only up to the current frame. The camera motion and static structure are recursively estimated by non-linear filters. Our method is essentially a batch process since it performs a global optimization. However, our method can be used as the initialization step of a recursive process.

SfM of moving objects has been given much attention recently. A seminal work was proposed in [2] as “trajectory triangulation”. It is shown that if a point moving on a 3D line is viewed in at least five images, the 3D line can be estimated without ambiguity. The linear trajectory is extended to conics [2] and more general curves [11]. Similarly, [7] assumes that object points are moving linearly in constant speeds and observed by a weak-perspective camera. Structure and motion are obtained by factorizing the measurement matrix based on rank constraints. In contrast, we assume a more general case: the object moves on a general planar trajectory being observed by a perspective camera.

[6] proposes a method to use multiple independently

moving objects for camera self-calibration. Since the moving objects share the same camera, the fundamental matrices estimated from each object provides more constraints to the camera parameters. Similarly, a multi-body fundamental matrix [22] encodes the epipolar constraints from each object and can be efficiently decomposed to obtain multiple object motions. However, it is not clear how the object motion can be related to camera motion. Instead, our method explicitly estimates camera poses and then relate the moving objects to the cameras.

Several constraints were proposed in [17] to resolve the ambiguity in object motion observed by moving cameras. The relative scale is solved by evaluating statistical independence and rank constraint from a planar object trajectory measurements across a large number of frames. In contrast, our method works on a small number of frames and provides a closed-form solution to object motion.

3 Motion Segmentation

To identify the moving blobs in the video frames, we apply a method similar to the one described in [12]. However, since there exists a constant ground plane in the scene, homographies are used as 2D motion models instead of the affine transform. The ground plane is found by manually selecting at least 4 pairs of matched points in the first two frames and automatically tracking them in the following frames [3]. A homography \mathbf{H}_t^{t+1} between frame t and $t+1$ is robustly estimated by fitting to at least 4 pairs of feature points within a RANSAC-based scheme [20]. More feature points consistent with the homography are included in the estimation and tracked across the frames as well.

Any pixels inconsistent with the inter-frame homographies are defined as residual pixels. If there exists static structure which is away from the ground plane, the corresponding pixels are erroneously identified as residual pixels as well, termed as “motion parallax”. A number of geometric constraints can be applied to remove these parallax pixels [9, 19, 12]. For example, pixel correspondences consistent with the 2-view epipolar constraint and 3-view “structure consistency constraint” are classified as belong to the static background [12]. Then each frame in the video is segmented into a number of motion blobs and static background.

Moreover, video frames with large inter-frame camera motion are preferred for a reliable reconstruction. Therefore, a number of distinct frames (typically 5-10), called “key frames”, are selected from the whole sequence [18]. Hereafter, the reconstruction processes only deal with these key frames.

4 Reconstruction of the Static Background

The reconstruction of the static background is based on the feature correspondences located in the background area. The camera self-calibration step uses the ground plane to estimate the camera intrinsic and extrinsic parameters [14]. With the calibrated cameras, 3D point positions are obtained by triangulation. The structure and motion are finally refined by Bundle Adjustment.

4.1 Preliminaries

Let us introduce some preliminary notations and definitions. The camera intrinsic parameters are assumed to be constant across the frames and simplified as $\mathbf{K} = \text{diag}(f, f, 1)$, where the focal length f is the only unknown parameter. The extrinsic parameters for frame k ($k = 1, 2, \dots$) are the rotation matrix \mathbf{R}_k and the camera center \mathbf{C}_k relative to a world coordinate. Then the translation vector between the world and camera coordinate systems is $\mathbf{T}_k = -\mathbf{R}_k \mathbf{C}_k$.

Let $\mathbf{X}(x, y, z)^T$ denote 3D points and $\mathbf{x}(u, v)^T$ 2D points. Here, we do not differentiate between Euclidean and homogeneous coordinates of the same point. The projection from 3D points to 2D points is decomposed into two steps. First the 3D points are transformed from world into camera coordinate system,

$$\mathbf{X}^c = \mathbf{R}_k \mathbf{X}^w + \mathbf{T}_k \quad (1)$$

where the superscripts c (w) refers to camera(world) coordinate system. Then the 2D points are obtained by perspective projection,

$$\mathbf{x} \sim \mathbf{K} \mathbf{X}^c \quad (2)$$

where \sim means equivalence up to a scale factor. The camera projection matrix for frame k is then obtained by combining intrinsic and extrinsic parameters as $\mathbf{P}_k = \mathbf{K} [\mathbf{R}_k \quad \mathbf{T}_k]$.

If the world coordinates of a certain point are scaled to be $s\mathbf{X}^w$ and accordingly the camera translation is scaled to be $s\mathbf{T}_k$, the perspective projection of this point remains the same despite the unknown non-zero scale s . This so called “scale ambiguity” is presented as follows,

$$\mathbf{x} \sim \mathbf{K} (s\mathbf{R}_k \mathbf{X}^w + s\mathbf{T}_k) = s\mathbf{K} \mathbf{X}^c \sim \mathbf{K} \mathbf{X}^c \quad (3)$$

Suppose the camera rotation and translation for frame 1 and 2 are respectively $(\mathbf{R}_1, \mathbf{T}_1)$ and $(\mathbf{R}_2, \mathbf{T}_2)$. After simple derivation, the relative rotation $\delta\mathbf{R}$ and translation $\delta\mathbf{T}$ between the two frames are obtained as follows,

$$\delta\mathbf{R} = \mathbf{R}_2 \mathbf{R}_1^T, \delta\mathbf{T} = \mathbf{T}_2 - (\mathbf{R}_2 \mathbf{R}_1^T) \mathbf{T}_1 \quad (4)$$

Inversely, given $(\mathbf{R}_1, \mathbf{T}_1)$ and $(\delta\mathbf{R}, \delta\mathbf{T})$, the pose for frame 2 is solved as

$$\mathbf{R}_2 = \mathbf{R}_1 \delta\mathbf{R}, \mathbf{T}_2 = \delta\mathbf{R} \mathbf{T}_1 + \delta\mathbf{T} \quad (5)$$

4.2 Recovery of Structure and Motion

We apply a plane-based self-calibration method [14] to obtain camera calibration information. The method requires inter-frame homographies corresponding to a plane as input, which are already estimated in the motion segmentation step. By exploiting the consistent set of collineations between images, the method iteratively estimates camera intrinsic and extrinsic parameters as well as the normal vector of the ground plane.

For simplicity, we convert the camera projection matrices into their canonical forms [8] such that the camera pose in frame 1 becomes $\mathbf{R}_1 = \mathbf{I}$ and $\mathbf{C}_1 = \mathbf{0}$. The 3D positions of static points are estimated in a process similar to that in [18]. Starting from frame 1 and 2, static point positions are estimated by an efficient triangulation method [16]. Then with each new frame added, the positions of existing points are refined and new points are added into the 3D point set.

Let $\mathbf{N}(n_1, n_2, n_3)^T$ denote the normal vector of the ground plane and d the distance of the world origin from the plane. Any point in the plane satisfies $\mathbf{N}^T \mathbf{X} = d$. Therefore, the 3D position of any in-plane point is uniquely determined by its 2D projection and the plane position. Furthermore, since all the object points and camera centers are on the same side of the ground plane, the following inequality holds¹,

$$\mathbf{N}^T \mathbf{X} \leq d \quad (6)$$

Indeed, this inequality can be used to verify the visibility of 3D points. If $\mathbf{N}^T \mathbf{X} > d$, the point \mathbf{X} is below the ground plane and therefore invisible.

Finally, both the cameras and points are refined by a ‘‘Bundle Adjustment’’ process [21]. The parameters to be refined include the focal length f (1 unknown), camera rotation \mathbf{R}_k (3 unknowns), camera center \mathbf{C}_k (3 unknowns), point coordinates \mathbf{X}_i^w (3 unknowns for off-plane points; 0 for in-plane points) and plane parameters (\mathbf{N} , d). The parameters are estimated by non-linear Newton methods to minimize the total reprojection errors as follows,

$$\epsilon_{static} = \sum_i \sum_k \|\mathbf{x}_{ik} - \mathbf{K} \mathbf{R}_k [\mathbf{I} \quad -\mathbf{C}_k] \mathbf{X}_i^w\| \quad (7)$$

The scene structure and camera motion are recovered up to a global scale ambiguity as shown in (3). For the in-plane points, this ambiguity is easily solved as the points are intersections of optical rays with the ground plane. For the off-plane points, we need additional physical information to fit the point to the real world.

5 Shape Inference of Moving Objects

Now that the Euclidean reconstruction of static background is obtained, we turn to inferring the 3D shape (in

¹If $\mathbf{N}^T \mathbf{X} \geq d$, then let $d \leftarrow -d$ and $\mathbf{N} \leftarrow -\mathbf{N}$.

this section) and motion (in next section) of moving objects. Since each moving object is treated independently with the same method, we will focus on only one object for simplicity.

5.1 Estimating 3D object shape

Since the motion region is the image of an moving object observed by a moving camera, it is impossible to apply the standard SfM methods to obtain the object shape. However, this problem can be converted to another configuration where the object is static while being observed by a hypothetical moving camera, called the ‘‘virtual’’ camera. The camera which shots the actual sequence is then called the ‘‘real camera’’. The motion of virtual camera is a combination of motions of both the real camera and the moving object. Indeed, the virtual camera shares the same intrinsic parameters with the real camera, but undergoes different motion. Standard SfM methods are then applied to obtain the object shape and virtual camera motion.

Let $\mathbf{X}_{j,1}^b$ denote the world coordinate of the j -th 3D point on the surface of a moving object in frame 1, where superscript b indicates that the point belongs to the moving object. Let $(\mathbf{R}_k^v, \mathbf{T}_k^v)$ denote the pose of the virtual camera at frame k , where the superscript v means virtual. The 2D object points \mathbf{x}_{jk} are generated as follows,

$$\mathbf{x}_{jk} \sim \mathbf{K} (\mathbf{R}_k^v \mathbf{X}_{j,1}^b + \mathbf{T}_k^v) \quad (8)$$

We assume that there exist at least five pairs of matched feature points from each moving object in consecutive frames. The feature points are extracted by the SIFT algorithm from the detected motion blobs and matched across the frames [13]. In the calibrated case, the relative pose between frame $k-1$ and k is encoded in the essential matrix \mathbf{E}_k as follows,

$$\mathbf{E}_k = \delta \mathbf{T}_k^v \times \delta \mathbf{R}_k^v \quad (9)$$

where $\delta \mathbf{R}_k^v$ and $\delta \mathbf{T}_k^v$ denote the relative translation and rotation of the virtual camera from frame $k-1$ to k . The essential matrix is efficiently estimated by a five-point algorithm [16]. Then the rotation matrix $\delta \mathbf{R}_k^v$ can be uniquely recovered from \mathbf{E}_k and yet translation $\delta \mathbf{T}_k^v$ can be only recovered up to a scale factor, which is usually normalized such that $\|\delta \mathbf{T}_k^v\|=1$.

Based on the relative pose $(\delta \mathbf{R}_k^v, \delta \mathbf{T}_k^v)$ ($k = 2, \dots, K$) and the assumption that $\mathbf{R}_1^v = \mathbf{I}$ and $\mathbf{T}_1^v = \mathbf{0}$, the 3D poses of virtual camera $(\mathbf{R}_k^v, \mathbf{T}_k^v)$ are recovered by (5). The same technique for reconstructing the static scene in Section 4 is applied to obtain a Euclidean reconstruction. The object points and poses of the virtual camera are finally refined by the ‘‘Bundle Adjustment’’ to minimize the following reprojection errors:

$$\epsilon_{virtual} = \sum_j \sum_k \|\mathbf{x}_{jk} - \mathbf{K} [\mathbf{R}_k^v \quad \mathbf{T}_k^v] \mathbf{X}_{j,1}^b\| \quad (10)$$

5.2 Estimating the unknown scale

Similar to the case of static background, the set of object points $\mathbf{X}_{j,1}^b$ are obtained up to a unknown scale factor s . However, we show that s can be approximately determined by the ground plane.

Since any object point is on or above the ground plane, the linear inequalities in (6) still hold despite the varying scale s :

$$\mathbf{N}(s\mathbf{X}_{j,1}^b) \leq d \quad (11)$$

For simplicity, we assume $d \geq 0$ and $s > 0$, while other conditions are addressed similarly.

Let s_0 be the maximal solution to (11). One can see that s_0 corresponds to the case where one object point falls on the ground plane. If this is not true, we assume that there is at least one object point close to the ground plane. In either case, $s \approx s_0$ provides a good estimate of object scale. An intuitive explanation is that the 3D object shape is stretched by s_0 such that one object point falls onto the ground plane, which happens quite often in reality. Then the object points and translations of virtual cameras are re-scaled as: $\mathbf{X}_{j,1}^b \leftarrow s_0\mathbf{X}_{j,1}^b$, $\mathbf{T}_k^v \leftarrow s_0\mathbf{T}_k^v$.

However, when the object points are not close to the ground, $s \approx s_0$ is not valid anymore. Still, we can estimate the scale if some *a priori* of the object shape is known. For instance, if the object points are known to be on a sphere, the ground plane becomes a tangent plane of the sphere. In other words, the distance of the sphere center $\bar{\mathbf{C}}$ from the ground plane is exactly the radius \bar{r} . Indeed, if the object points are scaled by s , the center and radius of the sphere are also scaled by s . Therefore, an exact solution to s can be obtained as follows,

$$\mathbf{N}^T(s\bar{\mathbf{C}}) - d = s\bar{r}\|\mathbf{N}\| \Rightarrow s = d/(\mathbf{N}^T\bar{\mathbf{C}} - r\|\mathbf{N}\|) \quad (12)$$

where $\bar{\mathbf{C}}$ and \bar{r} are estimated by non-linear fitting to the object points. Moreover, a dense shape of the sphere can be obtained, even for the invisible part.

6 3D Object Motion Estimation

Starting from the shape of the moving object recovered in Section 5, we want to estimate the 3D object motion in the world coordinate system.

6.1 Derivation of virtual camera motion

Let us first derive the virtual camera motion based on the object motion and real camera motion. The j -th object point $\mathbf{X}_{j,k}^b$ in frame k is transformed from its original position $\mathbf{X}_{j,1}^b$ in frame 1 by rigid object motion as follows,

$$\mathbf{X}_{j,k}^b = \mathbf{R}_k^b\mathbf{X}_{j,1}^b + \mathbf{T}_k^b \quad (13)$$

where \mathbf{R}_k^b and \mathbf{T}_k^b are respectively the 3D rotation and translation of the whole object. Obviously, $\mathbf{R}_1^b = \mathbf{I}$ and $\mathbf{T}_1^b = \mathbf{0}$. Indeed, the translation vector \mathbf{T}_k^b is a 3D displacement of the unknown object centroid between different frames in the world coordinate system.

Combining (1) and (13), we have the 3D coordinates of object points relative to the k -th camera with the pose $(\mathbf{R}_k^c, \mathbf{T}_k^c)$ as

$$\mathbf{X}_{j,k}^c = (\mathbf{R}_k^c\mathbf{R}_k^b)\mathbf{X}_{j,1}^b + (\mathbf{R}_k^c\mathbf{T}_k^b + \mathbf{T}_k^c) \quad (14)$$

Note that static objects could be considered as a special case with $\mathbf{R}_k^b = \mathbf{I}$ and $\mathbf{T}_k^b = \mathbf{0}$ which simplifies (14) to be (1).

Then, the pose of the virtual camera $(\mathbf{R}_k^v, \mathbf{T}_k^v)$ is related to the pose of the real camera $(\mathbf{R}_k^c, \mathbf{T}_k^c)$ as follows,

$$\mathbf{R}_k^v = \mathbf{R}_k^c\mathbf{R}_k^b, \mathbf{T}_k^v = \mathbf{R}_k^c\mathbf{T}_k^b + \mathbf{T}_k^c \quad (15)$$

In frame 1, we have $\mathbf{R}_1^v = \mathbf{R}_1^c = \mathbf{I}$ and $\mathbf{T}_1^v = \mathbf{T}_1^c = \mathbf{0}$. In the following frames, the virtual camera has different relative pose than the real camera due to object motion.

Ideally, given the pose of real camera $(\mathbf{R}_k^c, \mathbf{T}_k^c)$ and virtual camera $(\mathbf{R}_k^v, \mathbf{T}_k^v)$, we should be able to solve the object pose $(\mathbf{R}_k^b, \mathbf{T}_k^b)$ relative to its original position in frame 1 from (15). This is true for the object rotation \mathbf{R}_k^b which is solved without any ambiguity,

$$\mathbf{R}_k^b = (\mathbf{R}_k^c)^{-1}\mathbf{R}_k^v \quad (16)$$

However, solving the object translation \mathbf{T}_k^b is not straightforward. The main reason is that the translation of virtual camera \mathbf{T}_k^v is not uniquely determined, during the relative pose estimation. As pointed out in [17], there exists a family of translation parameters with different scales which produce the same image projection. Let us further investigate this ambiguity by revisiting the relative pose problem.

Let us assume that the poses of real camera in frame $k-1$ and k to be $(\mathbf{R}_{k-1}^c, \mathbf{T}_{k-1}^c)$ and $(\mathbf{R}_k^c, \mathbf{T}_k^c)$, the pose of moving object in frame $k-1$ $(\mathbf{R}_{k-1}^b, \mathbf{T}_{k-1}^b)$ and the relative pose of virtual camera $(\delta\mathbf{R}_k^v, \delta\mathbf{T}_k^v)$ between frame $k-1$ and k . Combining (15) and (4), we have

$$\delta\mathbf{T}_k^v \sim \mathbf{R}_k^c\mathbf{T}_k^b + \mathbf{T}_k^c - \delta\mathbf{R}_k^v(\mathbf{R}_{k-1}^c\mathbf{T}_{k-1}^b + \mathbf{T}_{k-1}^c) \quad (17)$$

which is re-written as

$$\mathbf{R}_k^c\mathbf{T}_k^b + \mathbf{Q}_k \sim \delta\mathbf{T}_k^v \quad (18)$$

where $\mathbf{Q}_k = \mathbf{T}_k^c - \delta\mathbf{R}_k^v(\mathbf{R}_{k-1}^c\mathbf{T}_{k-1}^b + \mathbf{T}_{k-1}^c)$.

Generally speaking, the three unknowns in the translation vector \mathbf{T}_k^b cannot be uniquely solved from (18), because three equations up to an arbitrary scale factor generate only two linear equations.

6.2 What kind of object translation can be solved?

One may ask a question naturally: in what cases, the object translation could be linearly and uniquely solved? Before answering this question, let us prove a lemma.

Lemma 1. *There exists a linear and unique solution to (x_1, x_2, x_3) in a set of 4 linear equations with one arbitrary non-zero scale s as follows,*

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} = s \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ 0 \end{bmatrix} \quad (19)$$

Proof. Let $\mathbf{a}_i = (a_{i1}, a_{i2}, a_{i3})^T$ ($i = 1, \dots, 4$) and $\mathbf{x} = (x_1, x_2, x_3)^T$. Any pair of c_i and c_j ($c_i c_j \neq 0, i \neq j$) contribute a linear equation as follows,

$$c_j (\mathbf{a}_i^T \mathbf{x} + b_i) - c_i (\mathbf{a}_j^T \mathbf{x} + b_j) = 0 \quad (20)$$

Otherwise, suppose $c_i=0$, a linear equation is generated without c_i ,

$$\mathbf{a}_i^T \mathbf{x} + b_i = 0 \quad (21)$$

Let n_0 denote the number of zero values in (c_1, c_2, c_3) . If $n_0=0$, we have at least two equations in the form of (20); If $n_0=1$, one in (20) and one in (21); if $n_0=2$, two in (21); if $n_0=3$, three in (21). Furthermore, there is an additional linear equation,

$$\mathbf{a}_4^T \mathbf{x} + b_4 = 0 \quad (22)$$

Therefore, in any case, there always exist at least three linear equations. \mathbf{x} is then uniquely solved by LU or SVD decomposition. \square

Then the question is answered in the following theorem,

Theorem 1. *The sufficient condition for finding a linear and unique solution to object translation \mathbf{T}_k^b is that an additional linear constraint exists as follows,*

$$\mathbf{a}_k^T \mathbf{T}_k^b + b_k = 0 \quad (23)$$

where \mathbf{a}_k is a 3×1 vector and b_k is a scalar.

Proof. The three equations up to a scale factor in (18) and the additional constraint in (23) consist of a set of four equations with three unknowns in \mathbf{T}_k^b and one arbitrary scale factor. According to Lemma 1, there exists a unique solution to \mathbf{T}_k^b . Therefore, (18) and (23) are sufficient for finding a linear and unique solution to \mathbf{T}_k^b . \square

Theorem 1 requires only one additional linear constraint on object translation, which is more general than those in previous work. For example, the linear trajectory constraint [2, 7] is represented by $\mathbf{T}_k^b - \mathbf{T}_{k-1}^b \sim \mathbf{T}_0$ where \mathbf{T}_0 is

the direction vector of the line. This constraint introduces two additional linear equations. Moreover, Theorem 1 does not require \mathbf{a}_k and b_k to be constant in different frames. Therefore, \mathbf{a}_k and b_k can be derived from all the information in previous frames, such as camera poses, object motion and shape.

In some video sequences, moving objects may appear static in the image when they are tracked by the camera at the same direction and speed. This is a degenerate case for object relative pose estimation in Section 5 since $\delta \mathbf{T}_k^v = \mathbf{0}$. Surprisingly, although it is impossible to obtain the 3D object shape, object motion can still be solved uniquely.

6.3 The planar-translation constraint

In our approach, the ground plane is used to impose the addition linear constraint onto object translation. It is called the planar-translation constraint: *the 3D instantaneous translation of the moving object must be parallel to the ground plane.* The constraint is consistent with our real-life experience that humans can easily tell the depth and motion of an object with the aid of the ground plane.

The planar-translation constraint implies that the distance of 3D object centroid from the ground plane is constant. It also implies that the translation vector is perpendicular to the normal vector of the ground plane:

$$\mathbf{N}^T (\mathbf{T}_k^b - \mathbf{T}_{k-1}^b) = 0 \quad (24)$$

where \mathbf{N} is the plane normal vector and $\mathbf{T}_{k-1}^b, \mathbf{T}_k^b$ are the object translation in frame $k-1$ and k . We have $\mathbf{a}_k = \mathbf{N}$ and $b_k = -\mathbf{N}^T \mathbf{T}_{k-1}^b$ in the form of Theorem 1. According to Theorem 1, \mathbf{T}_k^b can be solved without ambiguity.

Finally, object shape $\mathbf{X}_{j,1}^b$ and motion $(\mathbf{R}_k^b, \mathbf{T}_k^b)$ are further refined by Bundle Adjustment to minimize the reprojection errors as follows,

$$\epsilon_{moving} = \sum_j \sum_k \|\mathbf{x}_{jk} - \mathbf{P}_k^c (\mathbf{R}_k^b \mathbf{X}_{j,1}^b + \mathbf{T}_k^b)\| \quad (25)$$

where \mathbf{P}_k^c is the real camera projection matrix in frame k .

7 Experimental Results

Our method was tested over a number of real-world video sequences. We show two examples shot by handheld cameras. Both videos were taken in office environments, with camera tilt angles from 45 to 60 degrees. Cardboard boxes were placed on the floor as off-plane structure. Both qualitative and quantitative results are presented to demonstrate the effectiveness and robustness of our method.

The first video contains a toy-car moving along a winding curve on the floor, as shown in Fig. 2(a). The segmented motion blob is labeled red within a black bounding box in

Fig. 2(b). The reconstructed shape of the toy-car and the background are visualized with simple texture mapping in Fig. 2(c) and (d), where the 3D mesh faces are manually selected from the reconstructed points. As one can observe, the 3D shape of the toy-car is visually satisfactory. Yet the cardboard box is reconstructed as a parallelepiped instead of cuboid. One reason to this is that the camera is viewing it at such a high tilt angle that the perspective projection can be well approximated by an affine one.

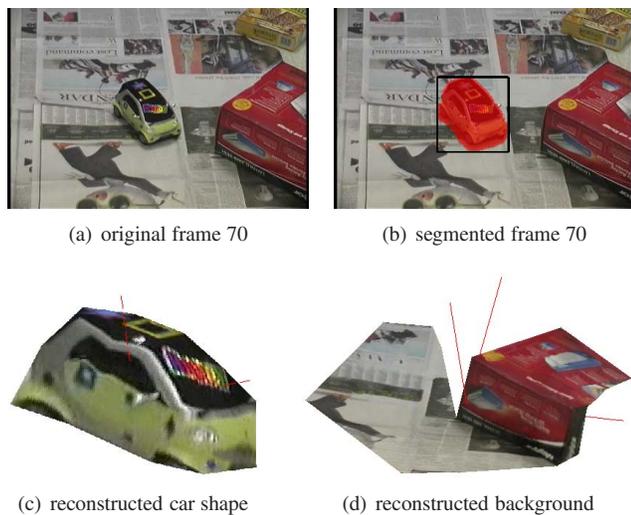


Figure 2. Results for the toy-car sequence

The estimated motion of the toy-car in the key frames is shown as a group of 3D point trajectories in Fig. 3. A mosaic image is generated from the consecutive frames to provide an expanded view of the ground plane. The scale of the toy-car is approximated determined as described in Section 5.2 by assuming that at least one point is close to the ground. The object points run along different trajectories while the 3D centroid undergoes the planar-translation motion. As one can observe from the two views, the car is rotating and translating in a planar curve parallel to the ground plane, which truly happens in the real world.

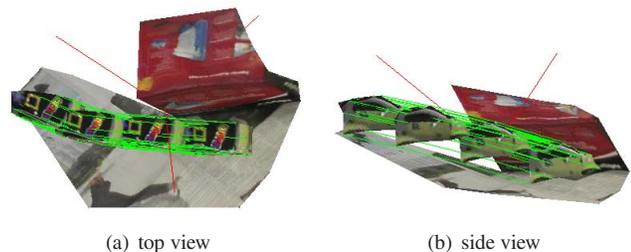


Figure 3. 3D Trajectory of the toy-car

In the second video, a volleyball is rolling on the floor, as

shown in Fig. 4(a). Note that due to the color similarity with the floor, the white regions on the ball are not segmented, as shown in Fig. 4(b). However, this does not affect finding good feature correspondences from the patterns on the ball surface. As stated in Section 5.2, knowing that the ball is a sphere, its parameters (center and radius) and scale can be uniquely estimated. Then any point on the sphere, even the invisible ones, can be reconstructed. In Fig. 4(c), one can observe that the centers of virtual cameras are far away from those of the real cameras. This is because the distance between camera centers and the ball center is much larger than the ball radius such that a small rotation of the ball points generates a large displacement for the virtual camera. Although the points on the ball follow a complex 3D trajectory, the 3D trajectory of the sphere center lies in a plane parallel to the ground plane, as shown in Fig. 4(d).

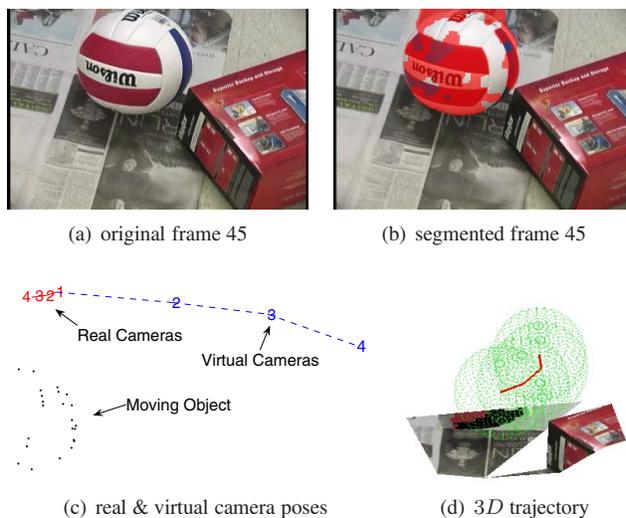


Figure 4. Results for the ball sequence

Below are some quantitative aspects of the experimental results. Six key frames are extracted for the reconstruction of both video sequences. For both sequences, there are more than 100 feature points tracked across all the key frames and yet only around 60 of them with accurate correspondences are used in the final reconstruction.

The reconstruction results are evaluated by measuring the average 2D reprojection errors of 3D points. Indeed, there are three reprojection errors defined in our paper: one generated by real cameras and static background in (7), one by virtual cameras and moving objects in (10) and one by real cameras and moving objects in (25). The reprojection errors are minimized by a commercial Bundle Adjustment package PhotoModeler [1]. The average errors (in pixels) for both sequences as well as the numbers of reconstructed points are shown in Table 1.

As one can observe, the errors for reconstructing the

static background are larger than those of moving objects. This might be caused by the fact that the feature correspondences from a cluttered background are less reliable than those from a single compact object. Nevertheless, reprojection errors around one pixel are achieved in both cases. As shown in the third row, the motion estimation error for the toy-car sequence is not even noticeable. In contrast, the error of the ball motion is large, probably due to the large translation of virtual cameras, so is numerically less stable.

| Errors (Number of Points) | Toy-car | Ball |
|---------------------------|------------|------------|
| Real Cam.+Background | 0.794 (23) | 1.037 (36) |
| Virtual Cam.+Moving Obj. | 0.781 (29) | 0.651 (28) |
| Real Cam.+Moving Obj. | 0.781 (29) | 2.232 (28) |

Table 1. Avg. reprojection errors (in pixels)

8 Conclusion and Future Work

We have presented an approach for building a Euclidean 3D reconstruction of both the static background and objects moving on ground plane. The ground plane is manually initialized in the first two frames and tracked across the frames. One of our contributions is a practical system which performs the following tasks: 2D motion segmentation, 3D reconstruction of the static background and 3D reconstruction of moving objects. Another contribution is approximating the unknown scale of moving objects by assuming at least one object point is on the ground plane. One more contribution is investigating the problem of finding sufficient constraints for uniquely estimating 3D object motion in presence of camera motion, formulated as a planar-translation constraint in our method. Encouraging experimental results are presented to prove the effectiveness and robustness of our approach.

There are several possible directions for the future research. 2D lines and surface patches can be added into the feature set to facilitate the reconstruction. We will test our method against more sequences and discover if there exists any degenerate configuration in which the object motion cannot be uniquely solved. Also, our method can be extended to the dense 3D reconstruction of video scenes.

Acknowledgments

This research was funded, in part, by the Advanced Research and Development Activity of the U.S. Government under contract # MDA-904-03-C-1786.

References

[1] PhotoModeler. <http://www.photomodeler.com>.

- [2] S. Avidan and A. Shashua. Trajectory triangulation: 3D reconstruction of moving points from a monocular image sequence. *IEEE PAMI*, 22(4):348–357, 2000.
- [3] S. Birchfield. An implementation of the Kanade-Lucas-Tomasi feature tracker. <http://www.ces.clemson.edu/~stb/klf/>.
- [4] A. Calway. Recursive estimation of 3D motion and surface structure from local affine flow parameters. *IEEE PAMI*, 27(4):562–574, 2005.
- [5] A. Chiuso, P. Favaro, H. Jin, and S. Soatto. Structure from motion causally integrated over time. *IEEE PAMI*, 24(4):523–535, 2002.
- [6] A. W. Fitzgibbon and A. Zisserman. Multibody structure and motion: 3-D reconstruction of independently moving objects. In *ECCV*, volume 1, pages 891–906, 2000.
- [7] M. Han and T. Kanade. Reconstruction of a scene with multiple linearly moving objects. *IJCV*, 59(3):285–300, 2004.
- [8] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.
- [9] M. Irani and P. Anandan. A unified approach to moving object detection in 2D and 3D scenes. *IEEE PAMI*, 20(6):577–589, June 1998.
- [10] T. Jebara, A. Azarbayejani, and A. Pentland. 3D structure from 2D motion. *IEEE Signal Processing Magazine*, 16(3):66–84, 1999.
- [11] J. Y. Kaminski and M. Teicher. A general framework for trajectory triangulation. *Journal of Mathematical Imaging and Vision*, 21:27–41, 2004.
- [12] J. Kang, I. Cohen, G. Médioni, and C. Yuan. Detection and tracking of moving objects from a moving platform in presence of strong parallax. In *ICCV*, volume 1, pages 10–17, 2005.
- [13] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.
- [14] E. Malis and R. Cipolla. Camera self-calibration from unknown planar structures enforcing the multiview constraints between collineations. *IEEE PAMI*, 24(9):1268–1272, 2002.
- [15] D. Nistér. Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. In *ECCV*, volume 1, pages 649–663, 2000.
- [16] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE PAMI*, 26(6):756–770, 2004.
- [17] K. E. Özden, K. Cornelis, L. Van Eycken, and L. Van Gool. Reconstructing 3D trajectories of independently moving objects using generic constraints. *CVIU*, 96(3):453–471, 2004.
- [18] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, and J. Tops. Visual modeling with a hand-held camera. *IJCV*, 59(3):207–232, 2004.
- [19] H. S. Sawhney, Y. Guo, J. Asmuth, and R. Kumar. Independent motion detection in 3D scenes. *IEEE PAMI*, 22(10):1191–1199, 2000.
- [20] P. Torr and A. Zisserman. Robust computation and parameterization of multiple view relations. In *ICCV*, pages 727–732, 1998.
- [21] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – a modern synthesis. In *Workshop on Vision Algorithms*, pages 298–372, 1999.
- [22] R. Vidal, Y. Ma, S. Soatto, and S. Sastry. Two-view multibody structure from motion. *IJCV*, 2005. (In Press).