MOVING OBJECT SEGMENTATION AND DYNAMIC SCENE RECONSTRUCTION USING TWO FRAMES

Amit K Agrawal and Rama Chellappa

University of Maryland
Dept. of Electrical and Computer Engineering
College Park, MD USA

ABSTRACT

We present an iterative algorithm for segmenting independently moving objects and refining and updating a coarse depth map of the scene under unconstrained camera motion (translation and rotation) with the assumption that the independently moving objects undergoes pure translation. Given a coarse depth map acquired by a range-finder or extracted from a Digital Elevation Map (DEM), the ego-motion is estimated by combining a global ego-motion constraint and a local brightness constancy constraint using least median of squares (LMedS) which treats independently moving objects as outliers. Using the estimated camera motion and the available depth estimate, motion of the 3D points is compensated. We utilize the fact that the resulting surface parallax field is an epipolar field and use a corresponding parametric model to estimate the parallax vectors for all pixels. We use the previous motion estimate to get the epipolar direction and hence pixels where the parallax direction is not aligned towards the epipolar direction are segmented out as moving points. The depth map for static pixels is refined using the estimated parallax vectors. All segmented regions are removed for robustly estimating the ego-motion in subsequent iterations. A parametric flow model is fitted to the segmented regions and their 3D motion is estimated using subspace analysis. We present experimental results using both synthetic and real data to validate the effectiveness of the proposed algorithm.

1. INTRODUCTION

The classical structure from motion (SfM) problem deals with a static scene and requires estimation of the relative motion between the camera, scene and the 3D scene structure in the form of a depth map. More interesting problem is the analysis of *dynamic* scenes consisting of a number of objects moving independently. Consider a camera moving in an unconstrained manner (both rotation and translation) viewing a dynamic scene consisting of independently moving objects and assume that each independently moving object is undergoing pure translation motion. Given two views of the scene along with a coarse, noisy and partial depth map (from DEM or range finder) we wish to (a) Estimate the camera motion between two views (b) Refine and update the 3D structure of static scene points (c) Segment independently moving objects and estimate the 3D motion of each moving object. Even though we assume that each independently moving object is undergoing pure translation, the relative motion between the object and camera consist of both translation and rotation because of the camera rotation. Also this assumption is less restrictive than the usually made assumption of objects moving along a straight line [1] [2] [3] or along a conic section [2]. Thus we allow the 3D translation of the object to be different at each frame. In addition, the available depth map usually may not have any information about moving objects. For e.g., a DEM of an urban environment may have coarse information about the buildings but not about any moving vehicle.

Several researchers have worked on moving object segmentation in images. Classical approaches attempt to segment the scene by segmenting 2D optical flow in different regions using flow discontinuities [4] or fit a mixture of probabilistic models [5]. Costeira and Kanade [6] proposed a multi-body factorization algorithm for segmenting multiple moving objects under an orthographic camera. The algorithm relies on block diagonal structure of *shape interaction matrix* to segment the moving objects. But the camera model used (orthographic) restricts its applications. In addition, the shape interaction matrix is block diagonal only if the individual motions are independent [7] which is not true here (both the camera and object have same rotation). These algorithms are multi-frame algorithms and can not be directly applied to two frames. Recently, Vidal et. al. [8] have proposed two frame algorithms based on purely algebraic constraints to segment multiple moving objects in images. They formulate the problem as of clustering feature points on a mixture of subspaces of lower dimensions using the Generalized Principle Component Analysis (GPCA).

The feature based algorithms treats all features equally in the sense that static scene points are treated as moving with zero velocity (for e.g. [1]) or constraints satisfied by all points are used whether they are moving or static [9]. However, in practical scenarios, the number of pixels on static scene are usually larger than those on all the moving objects. Thus, there is the notion of dominant motion corresponding to camera motion. In this paper, we also use a dominant motion approach for camera motion estimation. In [10], we have proposed an algorithm for refining coarse 3D models and ego-motion estimation for *static* environments. Here we show how 3D modeling can be integrated with scene segmentation and how the information from the 3D structure and camera motion (for e.g. negative depths and parallax constraints) can be used to identify moving objects. Thus, we address the problem of using coarse and incomplete depth information along with intensity images to estimate the ego-motion, refine the depth map of the scene, along with detecting moving objects and recovering their motion.

Prepared through collaborative participation in the Advanced Decision Architectures Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0009.

0-7803-8874-7/05/$20.00 ©2005 IEEE

2. MOTION MODELS

We assume a perspective camera with known calibration parameters. Let $P^s = (X^s, Y^s, Z^s)$ be a static 3D point. Let T^c denotes the translational component of camera motion and Ω^c denotes the rotational part. Thus for static points, the relative motion can be written as [11] $U^s = -T^c - \Omega^c \times P^s$. The image of a scene point P is the point p given by $p = f\frac{P}{Z}$. Thus the motion field for static points is given by

$$\mathbf{u}^s = f\frac{Z^s U^s - U_z^s P^s}{(Z^s)^2} = Ah^s T^c + B\Omega^c \tag{1}$$

where $B = \begin{bmatrix} \frac{xy}{f} & -(f+\frac{x^2}{f}) & y \\ (f+\frac{y^2}{f}) & -\frac{xy}{f} & -x \end{bmatrix}$, $h^s = \frac{1}{Z^s}$ and $A = \begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix}$. Now consider $P^o = (X^o, Y^o, Z^o)$, a 3D point on any moving object. Each moving object i is assumed to be translating with velocity t_i^o. Then the relative motion for moving points can be written as $U^o = -T^c - \Omega^c \times (P^o + t_i^o) + t_i^o = -T^o - \Omega^c \times P^o$, where $T^o = T^c + \Omega^c \times t_i^o - t_i^o$. Thus the motion field for moving points is given by

$$\mathbf{u}^o = Ah^o T^o + B\Omega^c \tag{2}$$

where $h^o = \frac{1}{Z^o}$ denotes the inverse depth for the moving point.

Notice that there are two scale ambiguities: one in determining T^s and Z^s and other in determining T^o and Z^o. Even if the scale between T^s and Z^s is fixed, it does not uniquely determine t_i^o due to the scale factor between T^o and Z^o. Thus in the rest of the paper, we focus on estimating the *total* translational motion T^o, instead of the independent (camera subtracted) motion of object t_i^o.

3. ALGORITHM

The algorithm uses two intensity images (referred to as *key* and *offset* frames) and an initial coarse, and incomplete depth map (referred to as *reference depth map*) to estimate the ego-motion and the depth map along with segmenting the image into regions corresponding to independently moving objects in an iterative fashion (we call these iterations *global iterations*). We start with estimating the ego-motion using the available depth map and LMedS. Using the estimated camera motion, the available depth map is refined and the image is segmented using parallax constraints. This is done iteratively until the ego-motion estimates converge or a specified number of iterations have been reached.

Let $\mathbf{r} = (x, y)$ denote an image pixel and t denote the time index. Assuming brightness constancy, we have

$$I(\mathbf{r}, t) = I(\mathbf{r} - \mathbf{u}, t-1) \tag{3}$$

where $I(\mathbf{r}, t)$ and $I(\mathbf{r}, t-1)$ denote the key and offset frames respectively and $\mathbf{u}$ denotes the flow for the corresponding pixel. As in [10], let i denote the global iteration index, $\mathbf{u}_i$ denote the current estimate of the flow field during the i^{th} global iteration (obtained from current depth and ego-motion estimates using (1)) and $\delta\mathbf{u}_i$ denote the incremental 2D motion for a local iteration due to motion refinement or depth refinement. The appropriate motion (or depth) refinement can be estimated by minimizing

$$E(\delta\mathbf{u}_i) = \sum_R (\nabla I^T \delta\mathbf{u}_i + \Delta I)^2 \tag{4}$$

with respect to $\delta\mathbf{u}_i$ over suitable regions R, where $\nabla I = [I_x, I_y]^T$ denotes the spatial image derivatives and $\Delta I = I(\mathbf{r}, t) - I(\mathbf{r} - \mathbf{u}_i, t-1)$. We now describe the ego-motion estimation and depth refinement and object segmentation steps in detail.

3.1. Robust ego-motion estimation given a depth map

There is a need for robust ego-motion estimation because of the presence of independently moving objects. Also, the reference depth map usually does not have information about the depths of moving objects. As in [10], one could do a least square optimization for estimating the ego-motion given a depth map. However, a least square solution would assume all points as static and hence would give incorrect estimate. As the number of pixels on the static background is usually larger than those on all the moving objects combined together, we consider all the pixels on the moving objects as outliers in ego-motion estimation. Thus a LMedS solution[1] is obtained which is found to give satisfactory results. Also, the region R is decided on the basis of following two inputs. Firstly, only pixels with high confidence value are chosen (the confidence measures are provided by the depth refinement phase as described in section 3.2) and secondly, segmented regions using parallax constraint are not included. Thus, even in the presence of moving objects, the dominant motion corresponding to camera motion can be obtained.

3.2. Depth refinement and moving object detection

Let T_i^c, Ω_i^c denote the current ego-motion estimate and Z_i denote the available depth map estimate. Let δZ_i be the incremental depth map estimate for the i^{th} global iteration and $Z_{i+1} = Z_i + \delta Z_i$ be the refined depth map. Using (1), the incremental 2D motion for static scene points can be written as $\delta\mathbf{u}_i = A(h_{i+1} - h_i)T_i^c = (T_i^c)_z(h_{i+1} - h_i)\begin{bmatrix} x - x_f \\ y - y_f \end{bmatrix}$, where $h_{i+1} = \frac{1}{Z_{i+1}}$, $h_i = \frac{1}{Z_i}$, $(T_i^c)_z$ denotes the Z component of camera motion and (x_f, y_f) denotes the focus of expansion. Thus, the incremental motion due to depth refinement (*surface parallax field*) is in the epipolar direction. However, for moving points, the incremental motion is $\delta\mathbf{u}_i = A(h_{i+1} - h_i)T_i^o$. Thus moving points do not have parallax vectors aligned along the epipolar direction[2]. This fact can be used to estimate independent moving objects. We estimate the parallax vectors (both magnitude and direction) as described below for all the pixels. Pixels where parallax vectors are not aligned along the epipolar direction are classified as belonging to moving objects.

The form of $\delta\mathbf{u}_i$ from above allows us to use the following parametric model: $\delta\mathbf{u}_i = a_0 * \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}$ where a_0, a_1 and a_2 denote the parameters. Substituting in (4), we get

$$E(\delta\mathbf{u}) = \sum_{N \times N} ((I_x x + I_y y)a_0 + I_x a_1 + I_y a_2 + \Delta I)^2 \tag{5}$$

Here for each pixel (x, y), the region R is defined to be neighborhood of $N \times N$ pixels and the parameters are assumed to be constant over the neighborhood. Thus for each pixel, a least squares

[1] See http://www-sop.inria.fr/robotvis/personnel/zzhang/Publis/Tutorial-Estim/node25.html for LMedS algorithm

[2] Unless camera rotation is zero and both object and camera move in the same or opposite directions. In such a case, certain circumstances such as object moving faster than the camera can be identified using negative depths.

(LS) solution can be obtained for the parameters and the parallax vector $\delta\mathbf{u}_i$ can be obtained using estimated a_0, a_1 and a_2. We then estimate the angle between the estimated parallax vector and the epipolar direction **e** (after normalizing both to unit magnitude) as $\theta = \cos^{-1}(\delta\mathbf{u}_i^T\mathbf{e})$. If the angle is greater than a pre-specified threshold, the pixel is segmented as belonging to the independently moving object.

After segmenting the image, depths can be refined for the static pixels using the magnitude of parallax vectors estimated previously as in [10].

3.3. Motion estimation of moving regions

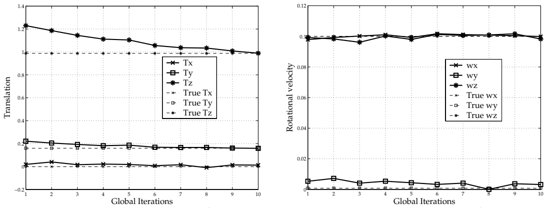
The segmented image (after few iterations of ego-motion refinement and depth refinement) is divided into different regions corresponding to different moving objects as follows. First a connected component analysis of the segmentation map is performed to get the connected regions. This will give potential candidate regions. Regions with sizes less than some threshold (typically 0.5% of image size) are discarded. Finally morphological operations (hole filling) are done to obtain blobs where each blob correspond to a different moving object. Each blob is processed separately for estimating its relative 3D translation motion T^o.

Consider (2). Let $\mathbf{u}^o = \mathbf{u}^o_{tr} + \mathbf{u}^o_{rot}$. The rotational flow $\mathbf{u}^o_{rot}$ does not depend on object depth and the rotational velocity for the object is equal to that of the camera. The rotational flow can be obtained using the pixel coordinates of the object blob and estimated Ω^c. Thus we need to estimate only the translational flow $\mathbf{u}^o_{tr}$ for the object which is much easier. In practical scenarios, the object can be assumed to have smooth depths and it is reasonable to assume that the *depth variations* on any particular object are much smaller than the *mean depth* of the object from the scene even though the entire scene may have large depth variations. Thus we can assume constant depth for the object for estimating the translational flow. Let $\mathbf{u}^o_{tr} = [u_{tr}, v_{tr}]$. Using (2), we have $u_{tr} = \frac{-fT^o_x + xT^o_z}{Z}$,$v_{tr} = \frac{-fT^o_y + yT^o_z}{Z}$ Thus we use $\mathbf{u}^o_{tr} = \begin{bmatrix} a_1 + a_3 x \\ a_2 + a_3 y \end{bmatrix}$ as the parametric model over the *entire* object region, where $a_1 \ldots a_3$ are the parameters. These parameters can be obtained by an iterative approach as in [12]. Using the estimated parameters, $\mathbf{u}^o_{tr}$ can be obtained. Eliminating Z between u_{tr} and v_{tr}, we get $\begin{bmatrix} fv_{tr} & -fu_{tr} & yu_{tr} - xv_{tr} \end{bmatrix} \begin{bmatrix} T^o_x \\ T^o_y \\ T^o_z \end{bmatrix} = 0$. Stacking flow values from all the points on the object, an over-constrained system $Ax = 0$ can be build with x corresponding to translational direction. This can be solved using SVD. Notice that this is similar to subspace analysis [13] but here the problem is much simpler since we know the rotational flow. Also only the translation direction can be estimated, thus reflecting the scale ambiguity in estimating translation and depth.

4. EXPERIMENTS

4.1. Synthetic Example

A semi-synthetic 3D model (with real textures) of an urban environment was rendered in OpenGL. We simulate a sequence of images by moving a virtual camera in the scene. The depth maps were obtained from the OpenGL Z buffer. Figs. 2(a) shows the key image. The dominant camera motion consists of translation along the Z direction (≈ 1 unit per frame) with rotational camera velocity $\Omega^c = [0.1, 0.003, 0.1]^T$. The independent motion consist of a sphere on the left moving along the X axis (towards right). The true total translation direction for the sphere is $[0.706, 0.119, 0.698]^T$. Figs. 2(b) and 2(c) show the true depth maps for the key image and the initial coarse and reference depth map respectively. The depth map is color coded (darker regions are farther from the camera). Note that the reference depth map contains only the information about the ground plane in the scene and does *not* contain any information about buildings or the spheres (both static and moving). The reference depth map as shown is made coarse by smoothing with a constant window of size 25×25 pixels. In addition, the moving sphere overlaps with the ground plane and hence all pixels belonging to the reference depth map do not belong to the static scene.



(a) Camera Translation (b) Camera Rotation

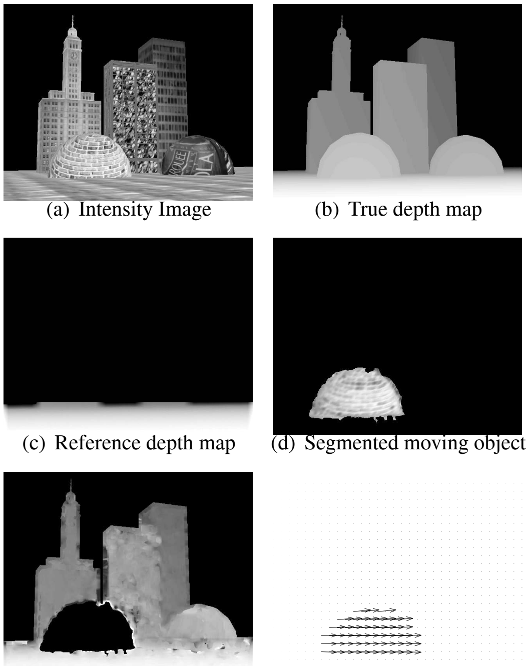
Fig. 1. Ego-Motion for Synthetic Example

A total of ten global iterations were performed. Figs. 1(a) and 1(b) shows the convergence of ego-motion parameters with global iterations which converge to their true values. Fig. 2(d) shows the segmented regions corresponding to moving object (sphere) which is quite accurate. Fig. 2(e) shows the estimated depth map for the static scene points. The root mean square error (RMSE) between the estimated depth map Z_{est} and the true depth map Z_{true} defined as $RMSE = \frac{100}{N}\sum_1^N (\frac{Z_{true} - Z_{est}}{Z_{true}})^2$ where N denotes the total number of pixels was 2.79% for static scene points. The translation optical flow[3]for the segmented object is shown in Fig. 2(f). The total 3D translation direction (including camera motion) for the moving object was estimated as $[0.701, 0.127, 0.701]^T$ which gives an error of 0.55 degrees.

4.2. Real Example

A video sequence of toy objects was taken in a lab. The camera was moved on a planar surface in the X direction. Figs. 3(a) shows the key image from the sequence. The independent motion consist of the hand holding an object (labelled *green tea*) moving vertically. The motion of hand is sufficiently larger than the camera translation. For this sequence, we did not have any prior depth information for the entire image. Also, since this is an indoor lab sequence, the variation in the scene depth is small. Therefore, the reference depth map was chosen to be a constant all over the image. A total of five global iterations were performed. Figs. 3(e) and 3(f) shows the convergence of ego-motion parameters with global iterations. Notice that there are sufficient number of pixels on the moving object. Hence the initial estimate of the camera translation had a predominant Y component due to independent motion of hand. However, the algorithm was able to estimate the correct camera translational direction (along X axis) in few iterations. Fig. 3(b) shows the segmented regions corresponding to

[3] In all experiments, the flow field has been down sampled by 20 for proper viewing.

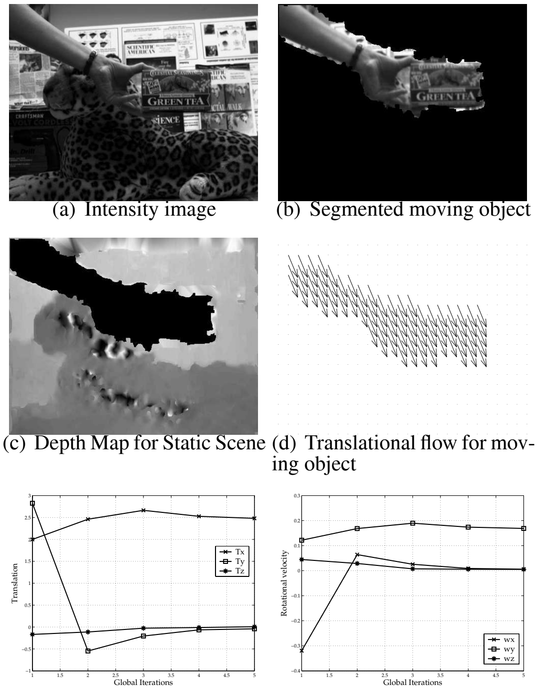


(a) Intensity Image (b) True depth map

(c) Reference depth map (d) Segmented moving object

(e) Depth Map (Static Scene) (f) Translational flow (moving object)

Fig. 2. Synthetic Example



(a) Intensity image (b) Segmented moving object

(c) Depth Map for Static Scene (d) Translational flow for moving object

(e) Camera Translation (f) Camera Rotation

Fig. 3. Real Example

the moving object which is quite accurate. Fig. 3(c) shows the estimated depth map for the static scene points. Notice the finely extracted boundaries for the tiger in the scene. The translation optical flow for the hand is shown in Fig. 3(d). The total 3D translation direction (including camera motion) for the moving object was estimated as $[-0.40, 0.92, 0.02]^T$ which shows correctly a predominant motion in Y direction along with a component along X direction corresponding to camera translation.

5. CONCLUSIONS

A two-frame approach has been presented for segmentation of independent moving objects in video along with estimation of egomotion, independent object motion and reconstruction of the dynamic scene using intensity images. The proposed method utilizes LMedS in estimating ego-motion and parallax constraints for segmenting independently moving objects. 3D structure for static scene is also estimated using surface parallax. The motion of moving objects is estimated by first fitting a parametric flow model followed by subspace analysis. The algorithm works well for unconstrained translational motion of moving objects.

6. REFERENCES

[1] A. Shashua and A. Levin, "Multi-frame infinitesimal motion model for the reconstruction of (dynamic) scenes with multiple linearly moving objects," in *Proc. Int'l Conf. Computer Vision*, Vancouver, Canada, July 2001, pp. 592–599.

[2] S. Avidan and A. Shashua, "Trajectory triangulation: 3d reconstruction of moving points from a monocular image sequence," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 348–357, Apr. 2000.

[3] M. Han and T. Kanade, "Reconstruction of a scene with multiple linearly moving objects," in *Proc. Conf. Computer Vision and Pattern Recognition*, Hilton Head Island, SC, USA, June 2000, pp. 542–549.

[4] A. Spoerri and S. Ullman, "The early detection of motion boundaries," in *Proc. Int'l Conf. Computer Vision*, 1987, pp. 209–218.

[5] M. Black and P. Anandan, "Robust dynamic motion estimation over time," in *Proc. Conf. Computer Vision and Pattern Recognition*, Maui, Hawaii, June 1991, pp. 296–302.

[6] J.P. Costeira and T. Kanade, "A multibody factorization method for independently moving-objects," *Int'l J. Computer Vision*, vol. 29, no. 3, pp. 159–179, Sept. 1998.

[7] R. Vidal and R. Hartley, "Motion segmentation with missing data using powerfactorization and GPCA," in *Proc. Conf. Computer Vision and Pattern Recognition*, Washington, D.C., June 2004, vol. TBD, p. TBD.

[8] R. Vidal and Y. Ma, "A unified algebric approach to 2-d and 3-d motion segmentation," in *Proc. European Conf. Computer Vision*, Prague, 2004, p. tbd.

[9] Y. Wexler and A. Shashua, "On the synthesis of dynamic scenes from reference views," in *Proc. Conf. Computer Vision and Pattern Recognition*, Hilton Head Island, SC, USA, June 2000, pp. 576–581.

[10] A. K. Agrawal and R. Chellappa, "Robust ego-motion estimation and 3D model refinement using depth based parallax model," Proc. Int'l Conf. Image Processing, Oct. 2004.

[11] E. Trucco and A. Verri, *Introductory Techniques for 3D Computer Vision*, chapter 8, Prentice Hall, 1998.

[12] J.R. Bergen, P. Anandan, K.J. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," in *Proc. European Conf. Computer Vision*, Santa Margherita Ligure, Italy, May 1992, pp. 237–252.

[13] D.J. Heeger and A.D. Jepson, "Subspace methods for recovering rigid motion i: Algorithm and implementation," *Int'l J. Computer Vision*, vol. 7, pp. 95–117, 1992.