Global Motion Registration via Long-term Photometric Memory

P. Favaro H. Jin S. Soatto

‡ Department of Computer Science, University of California, Los Angeles – CA 90095
† Department of Electrical Engineering, Washington University, St.Louis – MO 63130
Tel: (310)825-4840, Fax: (310)794-5057, email soatto@ucla.edu

Abstract

We introduce an algorithm for causally estimating three-dimensional shape and motion of an object represented locally as a rigid collection of planes supporting projectively deforming texture patches. Due to occlusions and the local nature of any causal algorithm, a drift in the estimates accumulates over time. We describe a method to perform global registration of local estimates of motion and structure by matching the appearance of feature regions stored over long time periods. The irradiance is matched using a score function that takes into account contrast and scaling. We show results on real image sequences that confirm extensive simulation experiments.

1 Introduction

When driving through an unfamiliar town, it is easy to lose one's bearings and "get lost", despite the fact that ego-motion is estimated accurately enough to control a vehicle. However, if a familiar landmark comes into sight (for instance a building seen previously), then one can re-adjust the global reference and correct the perceived attitude. This phenomenon is the natural consequence of the fact that visual information can be integrated over time only to the extent that visual features remain visible. In face of a changing landscape, a drift in the estimate of ego-motion is unavoidable; however, if a lost feature becomes visible again, ego-motion can be globally registered, thus effectively annihilating the bias (see Figure 1). In this paper we describe a method for representing and estimating the geometry, photometry and relative motion between the viewer and the environment that allows integrating visual information globally, when possible.

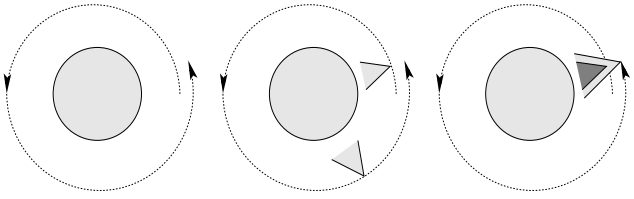

Figure 1: *Motion around an object (left): visual information can only be integrated to the extent that visual features remain visible. After a full turn, a bias will have accumulated due to features disappearing (middle). However, if visual features that were part of the original set become visible (right), one can reset the drift and globally register the estimated shape and motion.*

Of course if one could collect all data ahead of time and process them globally as a batch, this problem would not exist. However, in the applications that we are interested in – for instance virtual architectural walk-through or vision-based navigation – one must produce an estimate of ego-motion and scene structure at the current time, and therefore data must be processed *causally*. Data can be processed in batches (for instance on a window of past data up to the current time) or recursively (maintaining and updating an estimate by processing only the current image). Even when *global* batch processing is possible (for instance in video post-processing and animation), one may not necessarily want to do so since the amount of data would be colossal (we are targeting one hour of video for driving or walking sequences).

We would like to stress that the drift in the estimate is fundamental and unavoidable even if the data are processed in batches, as long as processing is *local in time*. The only case when drift is not an issue is when the sequence of images is processed as a whole. However, in the presence of occlusions, this is rarely possible (consider walking around a block: at some point all features visible at the initial time will disappear, as illustrated in Figure 1). When a long sequence is broken down into subsequences of a certain time length and each subsequence is processed as a batch, a drift will arise when merging estimates from different subsequences, and one is faced with the problem of choosing the length of the batches and merging the corresponding estimates. We choose to work in a recursive estimation framework to avoid the heuristics involved in partitioning the sequence and merging the estimates from different batches; however, all considerations in this paper apply to (causal) batch processing just as well.

1.1 Relation to previous work

This paper describes a way to globally integrate results of local structure from motion (SFM) algorithms. In fact, any SFM algorithm can be used as a starting point. As such, it relates to a vast portion of the literature of Computational Vision that cannot be reviewed within the space allotted. A few references that we find to be most closely related to our approach are [1, 3, 4, 5, 11, 7, 8, 12, 10, 13, 16, 29, 19, 20, 21, 23, 22, 26, 30] but the list is by no means complete. The reader can refer to new and upcoming textbooks for more detailed references on general SFM.

Since we integrate tracking and motion estimation, our work also relates to the large literature on image (2D) motion. However, most tracking schemes rely on point features and do not exploit feedback from higher levels (e.g. global motion estimates). If the scene is a rigid collection of features that undergo the same rigid motion, this global constraint can be enforced by a feature tracker for robustness and precision. A small body of literature on so-called direct methods addresses this issue, for example [27, 9, 18, 6]. Most work in direct SFM ends up representing shape as a collection of points whose projections are subject to brightness constancy and undergo the same rigid motion. Integrating motion information over the whole image, however, is computationally expensive. This suggests representing the scene as a collection of simple shapes. Of all possible shape models, planes occupy a special place in that the projection of a plane undergoing rigid motion evolves according to a projective transformation. It is therefore natural to represent a scene as a collection of planes, which has been done often in the past, as for instance in [28, 25, 24, 2].

1.2 Contributions of this paper

This paper presents a novel causal model for estimating structure and motion of an object represented as a collection of planar regions supporting projectively deforming texture patches. We pose the problem of global registration of *causally estimated* motion and shape. To the best of our knowledge there is no prior work in this area. In order to integrate visual information, feature appearance needs to be stored, which forces a representation of the environment via a collection of geometric-photometric features, in our case planar patches that support a Lambertian reflection distribution. Even if the use of piecewise planar representations is ubiquitous in the literature, and so are so-called "direct" methods, their use in a causal framework is novel. Also new is the attempt to perform global registration within a causal framework. In order to improve the computational efficiency of our algorithm, we develop a heuristic matching strategy to avoid matching feature patches that are not visible.

2 From local photometry to global dynamics

Let S be a rigid, piecewise smooth surface in space, and $\mathbf{X} \in S$ the coordinates of a generic point on it. When seen from a moving frame, the coordinates change in time according to a rigid motion $\mathbf{X}_t = R_t\mathbf{X}_0 + T_t$, where $R_t \in SO(3)$[1] and $T_t \in \mathbb{R}^3$ describe the rigid change of coordinates between the inertial (at time 0) and the moving frame (at time t). We assume to be able to measure, at each instant t, the irradiance $I(\mathbf{x}, t)$ at the point $\mathbf{x}_t = \pi(\mathbf{X}_t)$[2]. As a consequence of motion, the image undergoes a deformation that can be described by a nonlinear time-varying function of the surface S, $g_t^S(\cdot)$, such that

$$I(\mathbf{x}_0, 0) = I(g_t^S(\mathbf{x}_0), t) \tag{1}$$

[1] $SO(3)$ stands for the space of rotation matrices (orthogonal with determinant 1).

[2] π denotes a camera projection, for instance $\pi(\mathbf{X}) = \frac{\mathbf{X}}{\|\mathbf{X}\|}$ in the case of central projection. We do not make distinction between the image coordinates and the homogeneous coordinates (with 1 appended).

when the surface is Lambertian. In general g is nonlinear and depends on an infinite number of parameters (a representation of the surface S):

$$g_t^S(\mathbf{x}_0) = \pi(R_t \mathbf{x}_0 \rho + T_t) \;\; \text{with } \rho \mid \mathbf{x}_0 \rho = \mathbf{X}_0 \in S. \tag{2}$$

However, one can restrict the class of functions g to depend upon a finite number of parameters (corresponding to a finite-dimensional parameterization of S), and therefore represent image deformations as a parametric class. We actually consider a more general deformation model of the form $\alpha I + \beta$ in order to account for local contrast and brightness offset.

2.1 A generative model

There is a very simple instance when image deformations are captured by a finite-dimensional deformation model, that is when we restrict the class of surfaces to planes with unknown normal vector $\frac{\nu}{\|\nu\|} \in \mathbb{S}^2$ and intercept $\|\nu\|$. In fact, it is well known that a plane not passing through the origin (the optical center) can be described as $\Pi = \{\mathbf{X} \mid \nu^T \mathbf{X} = 1\}$, and therefore

$$g_t^\Pi(\mathbf{x}_0) = (R_t + T_t \nu^T)\mathbf{x}_0. \tag{3}$$

Given any matrix $M_t \in \mathbb{R}^{3\times 3}/\mathbb{R}$ with rank at least 2, it can be shown that it is in one to one correspondence with matrices of the form $R_t + T_t \nu^T$. Therefore, if the scene consists of a single planar surface, we can integrate photometric information on the entire surface by finding the matrix M that minimizes a discrepancy measure between $I(\mathbf{x}_0, 0)$ and $I(M_t \mathbf{x}_0, t)$ integrated over $\mathbf{x}_0$ on the entire image domain D; for instance

$$\hat{T}_t, \hat{R}_t, \hat{\nu} = \arg\min_{T_t, R_t, \nu} \int_D \|I(\mathbf{x}, 0) - I(M_t \mathbf{x}, t)\| d\mathbf{x} \tag{4}$$

for some choice of norm $\|\cdot\|$. Notice that the residual to be minimized is computed in the space of irradiance functions, and that the current model M_t, together with the first image $I(\mathbf{x}_0, 0)$, can be used to predict the next image $I(\mathbf{x}_{t+1}, t+1)$. In this sense this model is generative.

Of course, planes are quite a restrictive class of surfaces. However, we can use the above residual to test the hypothesis that a region of the image corresponds to (is well approximate by) a plane in space. Away from discontinuities, the larger the curvature, the smaller the region that will pass the test. By running the test all over the image (or on the portion of it that corresponds to high gradient of the irradiance, so as to eliminate at the outset regions with little or no texture), we can segment the image into a number of patches that correspond to planar approximations of the surface S. Obviously discontinuities and occluding boundaries will fail the test and therefore be rejected as outliers.

As a result of the procedure thus described, we are left with describing a surface with a certain number K of planar patches with normals $\nu^1, \dots \nu^K$, all undergoing the same rigid motion T_t, R_t. Photometric information is integrated within each patch, while geometric and dynamic information is integrated across patches. In this sense, this model describes the scene using *local photometry and global dynamics*. A model of the time evolution of all the unknown quantities is therefore

$$\begin{cases} \nu_{t+1}^i = \nu_t^i \quad i = 1 \dots K \\ T_{t+1} = \exp(\widehat{\omega}_t) T_t + V_t \\ R_{t+1} = \exp(\widehat{\omega}_t) R_t \\ V_{t+1} = V_t + n_V(t) \\ \omega_{t+1} = \omega_t + n_\omega(t) \\ I(\mathbf{x}_0^i, 0) = I(\pi((R_t + T_t {\nu^i}_t^T)\mathbf{x}_0^i), t) + w_t \quad \forall\, \mathbf{x}_0^i \in D^i \end{cases} \tag{5}$$

where $n_V(t)$ denotes the unknown linear acceleration, $n_\omega(t)$ the rotational acceleration, and D^i is the region of the image that corresponds to the approximation of the surface S by the i-th planar patch with normal ν^i. The noise term w_t is modeled as an independent sequence identically distributed in such a way as to guarantee that the measured image I is positive.

3 Causal estimation of a photo-geometric model

Having agreed to represent a surface as a rigid collection of planar patches supporting a radiance function that can deform according to a projective model, we can describe the unknown parameters (plane normals, rigid motion and velocity) as the state and input of a nonlinear dynamical system (5). Causally inferring a model of the scene then corresponds to estimating the state of the model (5) from its output (measured images). In order to arrive at a computationally simple solution to this problem, we will make a number of assumptions on the initial conditions and driving noises of the model (5).

3.1 Nonlinear filter and implementation

The first step towards implementation is to choose a local coordinate system for the model (5). To this end, we represent $SO(3)$ locally in canonical exponential coordinates: let Ω be a vector in $\mathbb{R}^3$, then a rotation matrix can be represented by $\widehat{\Omega} \in so(3)$ such that $R = \exp(\widehat{\Omega})$[3]. It is clear from the measurement equation in (5) that a scale factor between ν and T has to be fixed as they appear only as a product. Since we know that for a plane to be visible the z component of its normal vector has to be strictly positive, we choose to fix the z component of one normal to a positive constant, say 1.

Once the model (5) is written in local coordinates it is immediate to use an extended Kalman filter to estimate the state. For the filter to work in practice, one has to take occlusions into consideration. During the camera motion, objects in the scene may occlude each other and hence cause some image patches to disappear. On the other hand, some new image patches can become visible. When a patch disappears, we simply remove the corresponding normal vector from the state. When a new patch appears, we first estimate its normal with a reduced filter and once its estimate is stable (the innovation of the filter has reached steady-state) we insert it into the state.

3.2 Occlusions and drift

As we discussed in the previous section, in order to solve the scale factor ambiguity we choose to fix the z component of one normal. As long as the patch corresponding to the selected normal is visible, all the parameters will be estimated according to it. However, during a long sequence, visual features are bound to disappear due to occlusions or falling out of the field of view; hence, when the selected feature disappears, another normal has to be chosen and its third component has to be fixed. Since we do not have the exact value of the new fixed component with respect to the previous one, using its current estimate necessarily introduces a error that propagates to all the other estimated parameters. It is global in the sense that it also affects the global motion estimates: R and T. Therefore, our observation of motion and structure accumulates a drift which is bound through a constant by the number of times the patches of the fixed normals are lost. Notice that it does not make a difference whether the scale factor is associated to one particular feature or to a collective property of all points (e.g. the depth of their centroid), or to the norm of the translation vector: *every time any new state is associated with the scale factor, a drift occurs.*

As we said, this drift does not occur if at least one visual feature is visible from the beginning to the end of the sequence (and the scale factor happens to be associated with it). While this is unlikely in any real sequence, it is often the case that features that disappear become visible again. This can be because they become unoccluded, or due to the relative motion between the camera and the object (e.g. the viewer returns to a previously visited position). In order to exploit this information one must be able to match features that were visible at previous times during the sequence. This can be done since our features are represented as planar patches that support a Lambertian radiance or "texture", as we discuss in the next subsection.

3.3 Global registration

Features may disappear for a number of reasons as discussed above. Every time this happens, we store a geometric representation of the feature (coordinates of a point and the normal to the feature plane in an inertial reference frame) as well as the texture patch it supports. As the sequence progresses, more patches are added; in the case of the sequence portrayed in Section 4, as the camera navigates around the object, more and more features are added. However, because no single feature survives for the duration of the experiment, a small drift is accumulated (see Figure 3). In order to avoid this, the basic idea is the following: whenever a new feature is selected, the pose of its supporting plane is first estimated relative to the inertial frame, and then its texture is matched with all neighboring features. If a high score is achieved, we seek local support from neighboring features, to eventually conclude that the old and the new patch have overlap. Once this decision is made, the drift is compensated and the trajectory is adjusted, weighted by the covariance of the current estimate of the pose of that feature.

Let $\mathbf{x}$ and ν be respectively the center and the normal of the patch we are matching, and let $\tilde{\mathbf{x}}$ be the matched position. For each patch, we denote R^i and T^i the relative motion between the frame at time τ, when $\mathbf{x}$ is stored, to the current frame. Thus, we have the following set of equations:

$$\begin{cases} (R^1 + \lambda T^1 {\nu^1}^T)\mathbf{x}^1 = \rho^1 \tilde{\mathbf{x}}^1 \\ \vdots \\ (R^N + \lambda T^N {\nu^N}^T)\mathbf{x}^N = \rho^N \tilde{\mathbf{x}}^N \end{cases} \tag{6}$$

[3] Rodrigues' formula is a convenient way to compute the exponential.

where N is the number of matched points, ρ^i is the ratio between the depth of the i^{th} patch at time τ and the depth of the same patch at the current time, and λ is the scale factor drift. Due to noise in determining the matching position, the above equations do not hold in general. Therefore, we look for λ, ρ^i that minimize the distance between the estimated positions and the matching positions:

$$\hat{\lambda}, \hat{\rho}^1, \dots \hat{\rho}^N = \arg \min_{\lambda, \rho^1, \dots \rho^N} \sum_i \left\| (R^i + T^i \nu^{i^T}) \mathbf{x}^i - \tilde{\mathbf{x}}^i \right\| \tag{7}$$

where $\|\cdot\|$ is some norm. If we chose the L_2 norm as our distance function, $\lambda, \rho^1, \dots \rho^N$ can be computed using least squares. Rearranging the equations (6), we have

$$\begin{pmatrix} -T^1 \nu^{1^T} \mathbf{x}^1 & \tilde{\mathbf{x}}^i & \dots & 0 \\ \vdots & \vdots & & \vdots \\ -T^N \nu^{N^T} \mathbf{x}^N & 0 & \dots & \tilde{\mathbf{x}}^i \end{pmatrix} \begin{pmatrix} \lambda \\ \rho^1 \\ \vdots \\ \rho^N \end{pmatrix} = \begin{pmatrix} R\mathbf{x}^1 \\ \vdots \\ R\mathbf{x}^N \end{pmatrix} \tag{8}$$

Therefore, the optimal λ and ρ^i can be computed as follows:

$$\begin{pmatrix} \hat{\lambda} \\ \hat{\rho}^1 \\ \vdots \\ \hat{\rho}^N \end{pmatrix} = \begin{pmatrix} -T^1 \nu^{1^T} \mathbf{x}^1 & \tilde{\mathbf{x}}^i & \dots & 0 \\ \vdots & \vdots & & \vdots \\ -T^N \nu^{N^T} \mathbf{x}^N & 0 & \dots & \tilde{\mathbf{x}}^i \end{pmatrix}^{\dagger} \begin{pmatrix} R^1 \mathbf{x}^1 \\ \vdots \\ R^N \mathbf{x}^N \end{pmatrix} \tag{9}$$

where $A^{\dagger}$ denotes the pseudo-inverse of the matrix A. In our implementation we weigh the least-squares norm with the covariance of the parameters estimated by the extended Kalman filter.

Notice that the global registration performed at a certain instant of time does not affect the entire trajectory, but only the current pose of the camera relative to the inertial frame. This is because – in a causal recursive framework – we are only concerned with the estimate of shape and motion at the present point in time. If off-line operation is allowed, one may want to re-adjust the entire trajectory, but this is not the focus of this paper.

As the length of the experiment grows, maintaining a database of all previously visible features and matching each new feature with the entire database becomes unfeasible. In the next subsection we discuss a few heuristics to avoid a global search.

3.4 Visibility

Since we assume that the sequence is taken with a calibrated camera, at each time instant the field of view of the camera can be computed in the inertial frame, and all features that fall outside the visibility cone can be discarded at the outset. However, in principle one is still faced with having to match each new feature with all features in an infinite cone. While there is no significant penalty in not matching a visible feature, there possibility of imposing a scale correction because of a wrong match has to be minimized. Therefore, we employ conservative heuristics: first, we only consider a limited section of the cone (e.g. within 10 meters), since textures supported by planes at largely different depths cannot be matched due to the difference in scale and resolution[4]. To this end we need to determine the depth of each point $\mathbf{x}^i$ in the current reference frame. Recall the plane constraint $\nu^T \mathbf{X} = 1$, therefore at the time τ when a point is introduced, its depth can be computed as:

$$\rho_0^i = \frac{1}{\nu^{i^T} \mathbf{x}^i}. \tag{10}$$

Therefore, the depth at the current frame becomes the third component of

$$R^i \rho_0^i \mathbf{x}^i + T^i$$

where R^i and T^i again are the relative motion. Second, we consider all visible features at that instant of time, and carve opaque cones around them: if a feature is currently visible, all previously stored features that are occluded by it cannot be

[4]Note that, since the position of previously visible features is stored in a global reference frame, it is possible to predict their resolution in the current frame, and therefore adapt the matching lattice in the global registration.

seen. The visibility of each patch respect to the camera can be computed based on its normal at the initial time. More precisely, if

$$\mathbf{x}^{i^T} R^i \nu^i > 0 \tag{11}$$

we declare the point to be visible, otherwise we declare it occluded. Finally, we restrict our search to a neighborhood of each new feature, 10 pixels around it, to speed up the search and exclude outliers from other heuristics.

Although this global registation can be made more and more sophisticate by considering robust statistics, soft-matching and a number of other statistical techniques, we found that the procedure described in this section is a good compromise between accuracy and computational efficiency. Our implementation is just short of real-time at the moment, and we expect to be able to operate at frame-rate within a year, with the help of faster processor and code optimization.

4 Experiments

In Figure 2 we show a few images of a sequence obtained by moving a camera around an object (the actual motion is performed by rotating the object on a turntable, which is equivalent to the camera moving around it). This motion is designed in such a way that no feature remains visible throughout the course of the experiment. Therefore, as expected, drift will accumulate, as it can be seen in Figure 3 (top). The actual trajectory of the camera is a perfect circle that passes through the

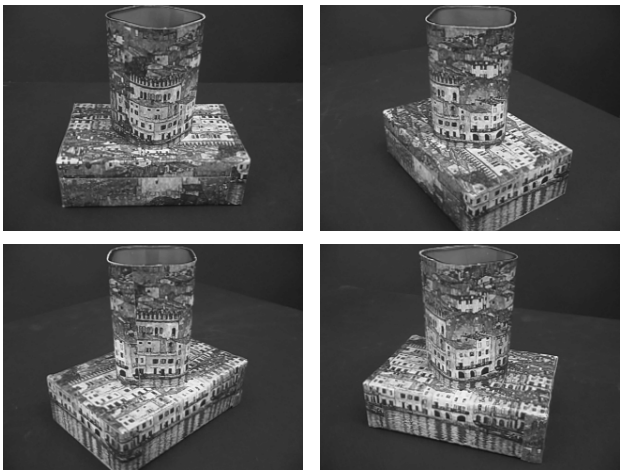

Figure 2: *Original dataset: the camera moves around the object so that no feature remains visible throughout the course of the sequence.*

origin, but the estimated trajectory misses the origin due to the drift. Even though the drift may seem small when visualized in terms of the estimates of motion, it severely affects the estimates of shape, since it results in photometric patches being misaligned and therefore spoils the meaningful merging of estimates from multiple passes around the object. By matching visible features, however, the drift can be compensated for, as shown in the solid line on the bottom. Failure to perform global registration results in a significant drift during the second pass around the object, shown as a dotted line. Once registered, different sequences around the object can be merged and the shape (position of orientation of planar patches) and photometry (texture supported on such planes) can be reconstructed. In Figure 4 we overlay the estimates to a set of images of the object, to show that the texture patches nicely align to the appearance of the object.

5 Discussion

We have introduced a scheme for handling occlusions and global registration by storing the appearance of previously visible features, and discussed heuristics to avoid a global search of all stored features. We also outlined a direct method to estimate motion and structure causally from image sequences. Instead of using point features we use planar patches, while non-planar patches and outliers are rejected using a simple hypothesis test. An extended Kalman filter is used to implement the algorithm

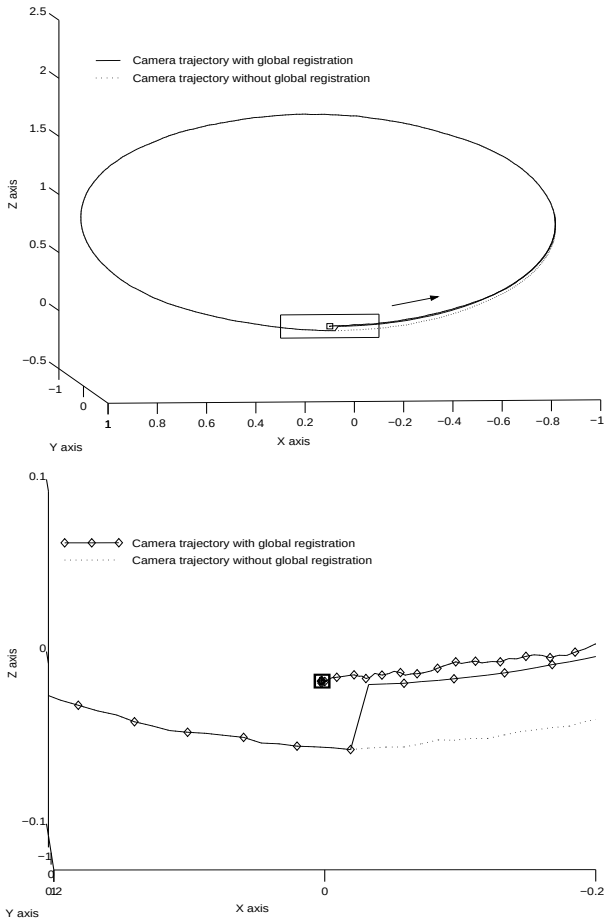

Figure 3: *Causally estimated spatial trajectory for a sequence of images (samples of which are shown in Figure 4). The trajectory of the camera surrounds the object so that no features survive from the beginning to the end of the experiment. Despite the fact that the camera goes back to the original configuration, the estimated trajectory does not reach the origin (top). This can be seen in the detail image on the bottom. This is unavoidable since no visual features are present from the beginning to the end of the sequence. However, starting from frame* 524*, several features that were visible at some point become visible again. Our filter stores both the pose and orientation of the planar patches that become occluded, as well as the texture patch that they support. Matching the current field of view with stored features allows to globally register the trajectory and effectively eliminate the drift. Not imposing global registration results in a drift, shown in the dotted line, during a second pass around the object.*

in a causal fashion. We perform experiments on real scenes. Even though the model we describe uses planes as primitives, the algorithm can be readily extended to any parametric representation of non-planar surfaces.

Acknowledgements

This research is supported in part by NSF grant IIS-9876145, ARO grant DAAD19-99-1-0139 and Intel grant 8029.

References

[1] G. Adiv. Determining 3-d motion and structure from optical flow generated by several moving objects. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 7(4):384–401, July 1985.

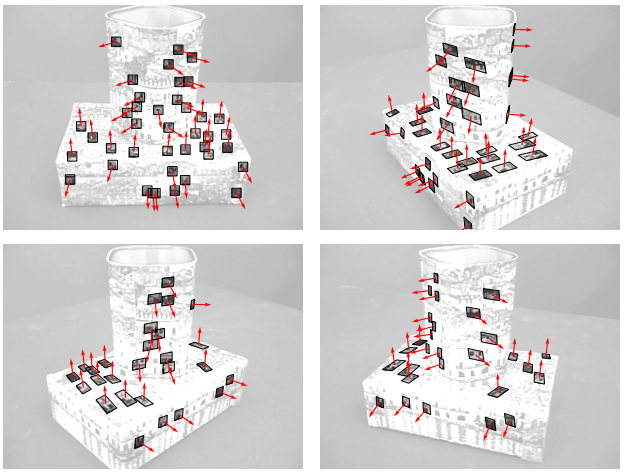

Figure 4: *Estimated representation of the scene: each feature corresponds to a planar patch represented by a point and a normal vector. The filter described estimates the geometric parameters and stores the texture patch that is supported on the planar feature. A few views of the reconstructed geometry (normal vectors) and texture (texture patches registered to the estimated pose of the corresponding planes) are superimposed to contrast-reduced views of the original scene to show that the texture patches capture the local appearance of the object. It is these planar patches that are matched in the current field of view in order to eliminate the bias seen in Figure 3.*

[2] J. Alon and S. Sclaroff. Recursive estimation of motion and planar structure. In *IEEE Computer Vision and Pattern Recognition*, pages II:550–556, 2000.

[3] A. Azarbayejani and A.P. Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(6):562–575, June 1995.

[4] T.J. Broida and R. Chellappa. Estimation of object motion parameters from noisy images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(1):90–99, January 1986.

[5] S. Christy and R. Horaud. Euclidean shape and motion from multiple perspective views by affine iterations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(11):1098–1104, November 1996.

[6] F. Dellaert, S. Seitz, C. Thorpe and S. Thrun. Structure from motion without correspondence. *Proc. of the Intl. Conf. on Comp. Vis. and Patt. Recog.*, June 2000.

[7] E.D. Dickmanns and V. Graefe. Applications of dynamic monocular machine vision. *Machine Vision and Applications*, 1:241–261, 1988.

[8] O.D. Faugeras. Three-dimensional computer vision: A geometric viewpoint. *MIT Press*, 1993.

[9] K.J. Hanna. Direct multi-resolution estimation of ego-motion and structure from motion. In *Workshop on Visual Motion*, pages 156–162, 1991.

[10] D.J. Heeger and A.D. Jepson. Subspace methods for recovering rigid motion, part ii: Theory. In *RBCV-TR*, 1990.

[11] S. Hsu, S. Samarasekera, R. Kumar, and H.S. Sawhney. Pose estimation, model refinement, and enhanced visualization using video. In *IEEE Computer Vision and Pattern Recognition*, pages I:488–495, 2000.

[12] X.P. Hu and N. Ahuja. Motion and structure estimation using long sequence motion models. *Image and Vision Computing*, 11(9):549–569, November 1993.

[13] J.J. Koenderink and A.J. vanDoorn. Affine structure from motion. *Journal of the Optical Society of America*, 8(2):377–385, 1991.

[14] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Image Understanding Workshop*, pages 121–130, 1981.

[15] Y. Ma, J. Kosecka, and S. Sastry. Linear differential algorithm for motion recovery: A geometric approach. *International Journal of Computer Vision*, 36(1):71–89, January 2000.

[16] L.H. Matthies, R. Szeliski, and T. Kanade. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3(3):209–238, September 1989.

[17] P.F. McLauchlan. A batch/recursive algorithm for 3d scene reconstruction. In *IEEE Computer Vision and Pattern Recognition*, pages II:738–743, 2000.

[18] J. Oliensis and M. Werman. Structure from motion using points, lines, and intensities. In *IEEE Computer Vision and Pattern Recognition*, pages II:599–606, 2000.

[19] J. Philip. Estimation of three-dimensional motion of rigid objects from noisy observations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(1):61–66, January 1991.

[20] C.J. Poelman and T. Kanade. A paraperspective factorization for shape and motion recovery. In *European Conference on Computer Vision*, pages B:97–108, 1997.

[21] H.S. Sawhney. Simplifying motion and structure analysis using planar parallax and image warping. In *International Conference on Pattern Recognition*, pages A:403–408, 1994.

[22] L.S. Shapiro, A. Zisserman, and M. Brady. Motion from point matches using affine epipolar geometry. In *European Conference on Computer Vision*, pages B:73–84, 1994.

[23] M. Spetsakis and Y. Aloimonos. A multi-frame approach to visual motion perception. *International Journal of Computer Vision*, 6(3):245–255, August 1991.

[24] P. Sturm. Algorithms for plane-based pose estimation. In *IEEE Computer Vision and Pattern Recognition*, pages I:706–711, 2000.

[25] P.F. Sturm and S.J. Maybank. A method for interactive 3d reconstruction of piecewise planar objects from single images. In *British Machine Vision Conference*, page Single View Techniques, 1999.

[26] R. Szeliski and S.B. Kang. Recovering 3d shape and motion from image streams using non-linear least squares. *Journal of Visual Communication and Image Representation*, 5(1):10–28, March 1994.

[27] R. Szeliski and S.B. Kang. Direct methods for visual scene reconstruction. In *Representation of Visual Scenes*, pages 26–33, 1995.

[28] R. Szeliski and P.H.S. Torr. Geometrically constrained structure from motion: Points on planes. In *3D Structure from Multiple Images of Large-Scale Environments*, pages 171–86, 1998.

[29] J.I. Thomas and J. Oliensis. Recursive multi-frame structure from motion incorporating motion error. In *Image Understanding Workshop*, pages 507–513, 1992.

[30] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154, November 1992.