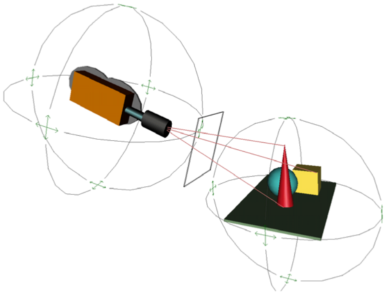
3D Structure from 2D Motion

Tony Jebara, Ali Azarbayejani and Alex Pentland
MIT Media Laboratory, Cambridge MA, 02139
{ jebara, ali, sandy }@media.mit.edu



I. Introduction

In their day-to-day lives, people naturally understand and operate in a three dimensional world. Curiously, though, they only sense 2D projections of it. The seemingly effortless act of inferring 3D from 2D observations is the result of complex mechanisms that are still quite far from being resolved. For many years, this task has been considered the primary role of visual processing. Pioneers in the fields of artificial intelligence and computer vision set out to recover a 3D representation of visible scenes which could then be used to recognize objects and reason about the world.

However, the general problem of recovering 3D from 2D imagery and the many steps involved [1] require a significant understanding of how the mind works, from issues of learning to intelligent behavior. Thus, the field is plagued by several of the same hurdles that have occupied AI researchers for many years. A tractable and more theoretically well-posed problem is the specific computation of 3D geometry from 2D geometry or Structure-from-Motion (*SfM*).

A. The Structure from Motion Task

Several simplifying assumptions are made to the general problem of 3D models from 2D imagery to formulate the Structure from Motion task. The figure above shows a standard SfM setup where a camera is viewing a scene. One key assumption is that objects in the scene are moving rigidly or, equivalently, only the camera is allowed to move in the environment.

An additional simplification is that there exists a module which pre-processes the camera's images to consistently extract, locate and label 2D features in the scene. Such 2D features could include salient points in the image, corners of objects, lines along their edges or curves around their contours. In each frame, the features are detected and associated to their corresponding instantiations in the other frames. These (usually noisy and error-prone) 2D measurements are the inputs to the SfM problem. The availability of corresponded features restricts the SfM problem to the so-called *Corresponded Structure from Motion* geometric task which will be the focus herein. It should be noted, however, that matching and detecting feature points is a fundamental and decidedly difficult computer vision problem which can not be dismissed so easily in practical implementations.

The locations of the 2D features in the images depend on 1) their coordinates in 3D space, 2) the relative 3D motion between the camera and the scene and 3) the camera's internal geometry. We assume that we have no prior knowledge of these three causes and wish to recover their parameters only from 2D point coordinate measurements over several frames or views. Of course, there exist many alternative problem statements in the SfM community with various twists ranging from the types of input features (i.e. curves or line features are alternatives), to the algorithm's required output, and so on. We shall focus primarily on the task stated above. Other SfM overviews can be seen in [31] [14] [41] [18].

The paper motivates the SfM approaches by describing some current practical applications. This is followed by a brief discussion of the background of the field. Then, several techniques are outlined that show various important approaches and paradigms to the SfM problem. Critical issues, advantages and disadvantages are pointed out. Subsequently, we present our SfM approach for recursive estimation of motion, structure and camera geometry in a nonlinear dynamic system framework. Results are given for synthetic and real imagery. These are used to assess the accuracy and stability of the technique. We then discuss some practical and real-time applications we have encountered and the reliability and flexibility of the approach in those settings. Finally, we conclude with results from an independent evaluation study conducted by industry where the proposed SfM algorithm compared favorably to alternative approaches. Our SfM software is available for public ftp at:

```
ftp whitechapel.media.mit.edu /pub/sfm
```

[1] These steps include difficult problems such as segmentation, recognition, correspondence, etc.

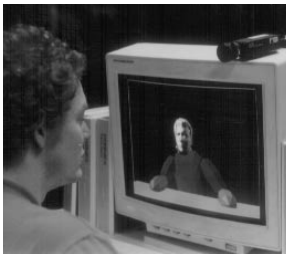


Fig. 1. Motion Matching: Camera-based head tracking for animating virtual 3D face models

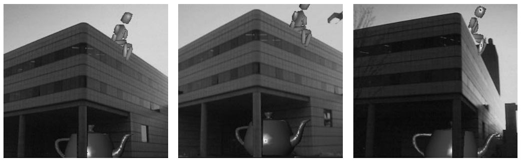


Fig. 2. Motion Matching: Adding 3D graphics to video.

II. Applications

The Structure from Motion community is not only motivated by the long term goals of computer vision, AI and 3D visual understanding. It also has many practical applications which presently drive research in SfM. Below, we illustrate example applications. Some of these are still in their early stages of development while others are quickly becoming commercially viable techniques in industry.

- 3D Model Reconstruction
  Many techniques exist for scanning real-world objects to form computer graphic 3D models. These range from 3D laser scanning to depth from defocus estimation. Structure from Motion is an important alternative and has been used to flexibly construct 3D coordinates and 3D models from 2D imagery of real objects. One demonstration, for instance, is Debevec, Taylor and Malik's [13] reconstruction of Berkeley's Campanile clock tower and surrounding campus via photogrammetric techniques.
- 3D Motion Matching
  The recovery of 3D motion parameters in the SfM framework can also be used to drive 3D models for animation purposes. Virtual objects can be affixed to real ones in the scene [6] (Figure 2) or computer graphics animations can be visually controlled [5] (Figure 1). Such techniques are currently being integrated into standard computer graphics software for use in film, video, games, interactive media, industrial design and visualization.
  In addition, motion matching can be used in virtual and augmented reality environments. For example, Kutulakos [35] describes a system with see-through head mounted display where 3D objects are superimposed on the user's scene in real-time.
- Camera Calibration
  Recovering a camera's external and internal parameters is another practical application. The external parameters describe a camera's position in 3D real-world coordinates and its internal parameters include variables such as focal length. In the field of Active Vision where cameras are expected to move around and zoom in autonomously, automatic recalibration is crucial [8].
- 3D Vision
  In many applications in computer vision, SfM paradigm is a useful computational sub-component. The 3D reconstruction that is recovered need not be the final goal of a vision system but an important intermediate step that can be fed back and fed forward to other vision modules. Thus, it can help in tracking, recognition and modeling (for example, see [32]).
- Perceptual Computer Interfaces
  Using vision as an interface for computer human interaction is also a potential application for 3D recovery techniques which can be used to identify user gestures that complement traditional keyboard and mouse paradigms [57].
- Robotics
  In the area of robotics which includes hand-eye coordination tasks, navigation and obstacle detection, 3D scene structure is an important intermediate step. Related work has been done by Wells [59] and Beardsley *et al.* [7].
- 3D Coding of Image Sequences
  The estimation of 3D parameters to describe image sequences is an important way to compactly encode information about the scene. This representation can then be used for low bit-rate communication, compression as well as noise reduction (see "A Review of Object-based Coding of 3D sequences", this issue).
- Mosaics and Rectification
  In photogrammetry, multiple images of a landscape or scene are taken and need to be aligned into a large composite image. SfM estimates the displacements and aligns images to re-project them into a large single image. Similarly, imagery can be mosaiced into a larger scene such as in [53].

III. SfM Origins: Photogrammetry and Early Vision

The roots of the Structure from Motion community can be traced back to two key fields, photogrammetry and computer vision. Although SfM problems account for a large portion of contemporary computer vision work, they have a long history and span several schools of thought.

Photogrammetry is a relatively old technique for measuring and processing lengths and angles in photographs for mapping purposes [47]. Reconstruction efforts were initially attempted using a pair of ground cameras separated by a fixed baseline. Pioneers in the 1840's include Arago, Jordan, Stolze and Laussedat who used cameras for estimating the shape of terrain from ground and aerial photographs, coining the name 'photogrammetrie' [39]. The arrival of airplane and space photography techniques spurred further development in the area. Estimates of motion from 2D photographs were used to rectify images into appropriate coordinates, mosaic multiple frames as well as estimate structure and elevation.

In the vision community, which was traditionally driven more from biology and AI roots, early achievements include the recovery of 3D scene structure from stereo by Marr and Poggio [37] where the correspondence is established automatically from two images via an iterative cooperative algorithm. The algorithm searches for unique matches of points between two images and recovers smooth disparity (an intermediate form of 3D depth) between them. Ullman [58] pioneered work on motion based reconstruction. The approach showed that four point correspondences over three views yield a unique solution [2] to motion and structure which could be solved via a nonlinear algorithm.

The formalism derived in photogrammetry and earlier vision research provided an important foundational theory for the SfM community. However, issues of implementation, stability, accuracy and so on have spurred numerous developments in the field. In addition, the goals and applications confronting the vision community have changed and hence emphasized different ways of thinking about the SfM problem. One of the key issues in the community is the use of linear versus nonlinear techniques. We discuss linear techniques, some of the critical issues in motion based structure estimation and the nonlinear techniques. Along the way, several camera models will be described as they are needed.

IV. Linear Approaches

Although many agree that SfM is fundamentally a nonlinear problem, several attempts at representing it linearly have been made which provide mathematical elegance as well as direct solution methods. On the other hand, nonlinear techniques require iterative optimization and must contend with local minima. However, they promise good numerical accuracy and flexibility.

A. Perspective Camera Models

Most Structure from Motion (linear *and* non-linear) techniques begin by assuming a perspective projection model as shown in Figure 3 which can be traced back to Durer and Renaissance painters. Alternative projection models include paraperspective or orthographic cases. Here, three 3D feature points are projecting onto an image plane (Π) with perspective rays originating at the *center of projection* (COP), which would lie within the physical camera. The origin of the coordinate system is traditionally taken to be the COP and the *focal length*, f is the distance from the COP to the image plane along the *principal axis* (or *optical axis*). The optical axis is traditionally aligned with the $\vec{z}$ axis. The projection of the COP onto the image plane along the optical axis is called the *principal point.*

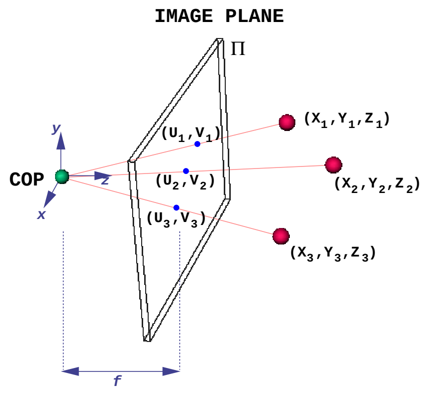


Fig. 3. Perspective Projection

Applying Thales theorem, we obtain the perspective projection formula as in Equation 1. Typically, the focal length f is set to 1 to simplify the expression since, in this model, f only varies the scaling of the image.

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} X \\ Y \end{pmatrix} \frac{f}{Z} \tag{1}$$

This perspective projection is often referred to as a pinhole camera. Although the focal length is the most emphasized internal camera geometry parameter, there exist more complex full parameterizations. In fact, real cameras have many other internal geometry variables. A more complete camera parameterization is shown in Equation 2 [41]. Here, the K matrix includes s_x and s_y, the scalings of the image plane along the $\vec{x}$ and $\vec{y}$ axes. Also note s_θ the skew between the $\vec{x}$ and $\vec{y}$ axes and (u_0, v_0) the coordinates of the principal point in the image plane. In addition to the linear effects summarized in the K matrix, there are other nonlinear and second order effects such as lens distortion. Typically, though, these second-order effects and even variables in K can be approximated and compensated for via standard corrective warping techniques [9].

$$\begin{pmatrix} u \\ v \end{pmatrix} = K \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \qquad K = \begin{pmatrix} s_x & s_\theta & u_0 \\ 0 & s_y & v_0 \end{pmatrix} \tag{2}$$

B. Algebraic Projective Geometry

The perspective camera model falls nicely into the mathematical realm of projective geometry which was invented by the French mathematician, Desargues (1591-1661). This formalism has grown to contain some extremely graceful mathematics ranging from invariants to injection of affine spaces to duality theories. While it is beyond the

[2] More specifically, three orthographic views are used and the solution is unique up to a reflection.

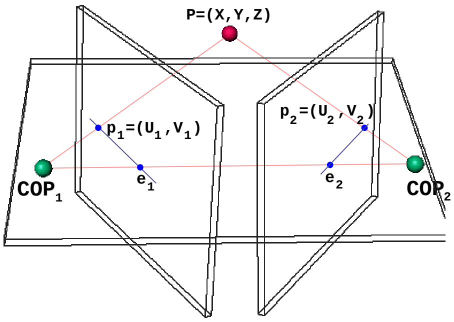


Fig. 4. Epipolar Geometry

scope of this paper to delve into this formalism, further reading can be found in [41] [45]. In the following, we shall discuss its practical implementation and implications in the SfM techniques that have adopted it.

C. Epipolar Geometry

The application of projective geometry techniques in computer vision is most notable in the *Stereo Vision* problem which is very closely related to Structure-from-Motion. Unlike general motion, stereo vision assumes that there are only two shots of the scene. In principle, then, one could apply stereo vision algorithms to a structure from motion task.

Applying projective geometry to stereo vision is not new and can be traced back from 19th century photogrammetry to work in the late sixties by Thompson [54]. However, interest in the subject was recently rekindled in the computer vision community thanks to important works in projective invariants and reconstruction by Faugeras [16] and Hartley [26].

Figure 4 depicts the imaging situation for stereo vision. The application of projective geometry to this situation results in the now popular epipolar geometry approach. The three points $[COP_1, COP_2, P]$ form what is called an *epipolar plane* and the intersections of this plane with the two image planes form the *epipolar lines.* The line connecting the two centers of projection $[COP_1, COP_2]$ intersects the image planes at the conjugate points e_1 and e_2 which are called *epipoles.* Assume that the 3D point P projects into the two image planes as the points p_1 and p_2 which are expressed in homogeneous coordinates $(u_1, v_1, 1)$ and $(u_2, v_2, 1)$ respectively. After some manipulations, the main result of the epipolar geometry is that the following *linear* relationship (Equation 3) can be written.

$$p_1^t F p_2 = 0 \tag{3}$$

Here, F is the so-called *fundamental matrix* which is a 3×3 entity with 9 parameters. However, it is constrained to have rank 2 (i.e. $||F|| = 0$) and can undergo an arbitrary scale factor. Thus, there are only 7 degrees of freedom in F. It defines the geometry of the correspondences between two views in a compact way, encoding intrinsic camera geometry as well as the extrinsic relative motion between the two cameras. Due to the linearity of the above equation, the epipolar geometry approach maintains a clean elegance in its manipulations. In addition, the structure of the scene is eliminated from the estimation of F and can be recovered in a separate step. Given the matrix F, identifying a point in one image identifies a corresponding epipolar line in the other image. [3]

Hartley proposes an elegant technique for recovering the parameters of the fundamental matrix when at least 8 points are observed [24]. Expanding the expression in Equation 3 gives one linear constraint on F per observed point as in Equation 4. Combining N of these equations from N corresponded features results in the linear system of the form $Af = 0$.

$$\begin{aligned} u_1u_2f_{11} + u_1v_2f_{12} + u_1f_{13} + v_1u_2f_{21} + v_1v_2f_{22} + \\ v_1f_{23} + u_2f_{31} + v_2f_{32} + f_{33} = 0 \end{aligned} \tag{4}$$

Typically, one solves such a linear system using more than 8 points in a least squares minimization $\min ||Af||^2$ subject to the constraint $||f|| = 1$. This constraint fixes the scale of the fundamental which otherwise is arbitrary. In addition, the rank 2 constraint must also be enforced. The algorithm employed utilizes an SVD computation but can be quite unstable. One way to alleviate this numerical ill-conditioning is to normalize pixel coordinates to span $[-1, 1]$. For robust fundamental matrix estimation techniques, refer to [63].

The fundamental matrix F is recovered independently of the structure and can be useful on its own, for example in a robotics application [16]. Hartley also uses it to derive Kruppa equations for recovering camera internal parameters [41] [25]. Ultimately, it becomes possible to recover Euclidean 3D coordinates for the structure which are often desirable for most typical application purposes.

At this point it is worthwhile to study the stability of such techniques. The reader should consider the case where the centers of projection of both images are close to each other (COP_1 and COP_2). Note the degeneracy when the centers overlap, which is the case when there is no translation and only rotation. A point in one image does not project to an epipolar line in the other for these cases. Degeneracy also occurs when all 3D points in the scene are coplanar. The result is that it is not possible to determine the epipolar geometry between close consecutive frames and it cannot be determined from image correspondences alone. The linearization in epipolar geometry creates these degeneracies and numerical ill-conditioning near them. Therefore, one requires a

[3] Thus, once F is solved, finding further corresponding points given a location in one image is reduced to searching along the epipolar line instead of the whole image.

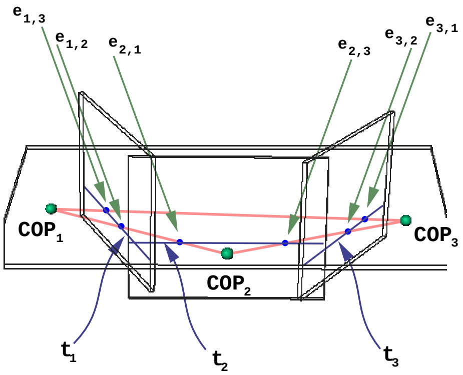


Fig. 5. Trifocal Geometry

large base-line or translation between the image planes for small errors. One way to overcome these degeneracies, is provided by Torr *et al.* [56]. Their technique involves switching from epipolar feature matching to a homography approach which can automatically detect and handle degenerate cases such as pure camera rotation.

The linear epipolar geometry formulation also exhibits sensitivity to noise (i.e. in the 2D image measurements) when compared to nonlinear modeling approaches. One reason is that each point can be corresponded to any point along the epipolar line in the other image. Thus, the noise properties in the image are not isotropic with noise along the epipolar line remaining completely unpenalized. Thus, solutions tend to produce high residual errors along the epipolar lines and poor reconstruction. Experimental verification of this can be found in [3].

D. The Trifocal Tensor

The next natural step from the stereo formalism and the fundamental matrix is a multi-camera situation (i.e. 3 or more projections). The trifocal tensor approach is such an extension and maintains a similar projective geometry spirit. This model has been proposed and developed by Sashua [46], Hartley [23] and Faugeras [19] among others. Figure 5 represents the imaging scenario.

Here, the trifocal plane is formed by the three optic centers COP_1, COP_2 and COP_3. Intersecting this plane with the three image planes produces three lines called the trifocal lines t_1, t_2 and t_3. There are now two epipoles (the $e_{i,j}$). One could use standard epipolar geometry and consider three fundamental matrices (one for each pair of COPs) F_{12}, F_{23} and F_{31}. However, the fundamental matrices are subject to some standard limitations which might be avoidable here. For instance, if a point P is in the trifocal plane, the fundamental matrices cannot determine if its 3 images belong to a single 3D point. In fact, there is additional information in the three plane case. Given a point in one image, it is possible to construct a line in another using the fundamental matrix. However, given a point in the first image and a point in the second image, one can directly compute the coordinates of the third point using a structure called the *trifocal tensor* which is the analog of the fundamental matrix for 3 view situations. Typically, one uses this tensor (denoted $\mathcal{T}$) to map a line in image 1 (l_1) and a line in image 2 (l_2) to a line in image 3 (l_3). This mapping is again a *linear* expression as in Equation 5.

$$l_3 = \mathcal{T}(l_1, l_2) \tag{5}$$

To map points, one merely considers intersections of mapped lines. The tensor $\mathcal{T}$ can be considered as a $3 \times 3 \times 3$ cube operator (i.e. defined by 27 scalars in total). It can also be represented as the concatenation of three 3×3 matrices: G_1, G_2 and G_3 which allow us to expand the above into the more straightforward Equation 6.

$$l_3 = \begin{pmatrix} l_{3_x} \\ l_{3_y} \\ l_{3_z} \end{pmatrix} = \begin{pmatrix} l_1^T G_1 l_2 \\ l_1^T G_2 l_2 \\ l_1^T G_3 l_2 \end{pmatrix} \tag{6}$$

If a set of corresponded points are known in each of the 3 images, the tensor can be estimated in a similar way as the fundamental matrix. For instance, one can perform a least-squares linear computation to recover the 27 parameters [23] [46]. However, the trifocal tensor's 27 scalar parameters are not all independent unknowns. Not every $3 \times 3 \times 3$ cube is a tensor. It too has constraints (like the fundamental matrix) and really has only 18 degrees of freedom. The above linear methods for recovering the tensor do not impose the constraints and can therefore produce invalid tensors.

By making an appeal to Grassmann-Cayley algebra, Faugeras gracefully derives the algebraic constraints on trifocal tensors which can be viewed as higher order (4th degree) polynomials on the parameters [19]. The 9 constraints are folded into a nonlinear optimization scheme which recovers the 18 remaining degrees of freedom of the tensor from image correspondences.

E. Application to Motion

Despite the elegance of the mathematics, these epipolar techniques do not address some pertinent practical issues in the Structure from Motion problem. In particular, they are reliable for perfect features and images with wide baselines but are sensitive to noise [55]. The formalism focuses on linear reformulations and only considers 2D measurement errors as an after thought. Thus it can exhibit numerical instabilities. These are especially evident when the baseline (i.e. relative camera translation between frames) is small. The case of no translation and pure rotation in the motions are actually degenerate cases for epipolar geometry. As the camera configurations approach degeneracies, the epipolar results vary wildly and noise causes numerical ill-conditioning. In addition, in the trifocal tensor case, the intermediate computation of higher order polynomials could also be prone to noise sensitivity.

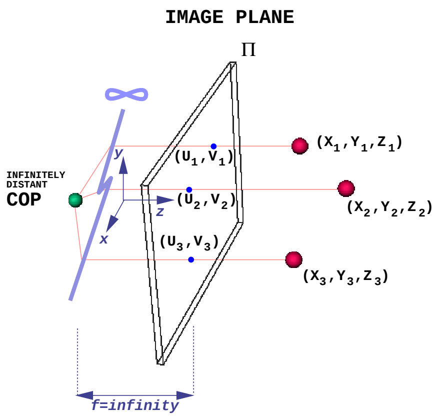


Fig. 6. Orthographic Projection

Essentially, these techniques focus on and perform best in 2-frame and 3-frame structure from motion with large baseline and small 2D measurement error. Their application to image sequences involves further processing and often requires some manual supervision. For instance, Faugeras discusses the special treatment that the trifocal technique requires on image sequences in post-production type applications [17]. Herein, a human user must preselect the triples of appropriate frames in a sequence to guarantee a wide range of camera motion (i.e. wide baseline) in the trifocal tensor calculations. In addition, the technique does not use all the images in a sequence, only appropriate triples. Therefore, the estimates of structure and motion are combined together using a somewhat unprincipled interpolation calculation. Since not all image frames are used and interpolation is relied on (be it linear or higher order), this technique discards data and hence compromises some accuracy.

F. Tomasi-Kanade Factorization

An alternative way to simplify the SfM problem is to consider a different projection model. The perspective projection case is characteristic of real cameras however, the corresponding equations are difficult to deal with. The orthographic case in Figure 6 greatly simplifies projection into the almost trivial form $u = X$ and $v = Y$.

One orthographic approach which has gained popularity is the factorization method proposed by Tomasi and Kanade [55]. Once again, the result is a *linear* formulation however the linearity is fundamentally different from the one induced in the previous epipolar geometry approaches. The technique begins with P tracked feature points over F frames and these are all combined into a matrix W of size $2F \times P$. For each frame (or row of W), the P feature points are registered by subtracting off their mean (recovering and factoring out the 2D translation). The resulting $2F \times P$ matrix $\hat{W}$ is then described as the product $\hat{W} = RS$ where R is a $2F \times 3$ matrix and S is $3 \times P$. These matrices are obtained from $\hat{W}$ via a singular value decomposition and some direct linear operations.

The algorithm is robust in many situations however it is tuned for orthographic projection, not for perspective effects. Degeneracies may occur when the camera translates forward and this forward motion parameter is not recovered by the system. Only two image-plane translations, camera yaw, roll and pitch are estimated. Therefore, it may not be applicable in some situations. The factorization method has subsequently been extended by Poelman and Kanade to the paraperspective case which is a closer approximation to perspective projection than orthographic projection [44].

V. Issues for Motion Sequences

Before moving on to nonlinear and dynamic approaches, we first discuss some of the issues that arise with continuous motion sequences (i.e. video) versus 2 or 3 frame static situations.

A. Probabilistic Feature Tracking

In video situations, be it real-time or offline processing, as in the above cases corresponded features are needed for the SfM problem. However, in video, features are computed in a temporally incremental way or are 'tracked' through the sequence. Tracked features may thus exhibit significant noise levels which should be estimated and treated rigorously. A feature is simply a measurement and all measurements have error (even at the pixel level due to image digitization). Therefore, one should model the error, for example via a typical Gaussian distribution (which identifies an ellipsoidal iso-probability curve around 2D feature points). In addition, over a video sequence, this error is likely to vary over time when the feature gets occluded (i.e. generates a wide distribution) or is very clearly observable (i.e. generates a tight distribution). This information is clearly useful and hence should be included in an SfM framework.

B. Causality and Recursion

In video and real-time applications, one critical constraint arises: causality and temporal continuity. As we begin considering more than three images, it becomes unlikely that the observations are generated by individual cameras but that a single camera was swept around the scene instead. A physical camera does not move instantaneously from one view point to another or equivalently objects being tracked can not teleport around the scene. Thus, one can assume that in image sequences the relative position between the camera and the scene changes incrementally. When applicable, this constraint or redundancy should be folded into the Structure from Motion estimation algorithm. For instance, one may consider the use of dynamic system theory.

In addition, if a sequence is available or if real-time video is streaming in, one may consider the use of online and recursive techniques. Instead of waiting for all future data to arrive, it makes sense to take advantage of the causal continuity and process each incoming frame when it arrives, summarizing all the past into a state vector. This has computational efficiency advantages as well as providing a real-time output. This allows the SfM estimates to be used in a closed loop control action such as navigating a robot. For a review of recursive structure from motion algorithms, consult [48] [49].

C. Small Baselines

One fundamental difference between causal methods and epipolar and trifocal techniques is that only small baselines are available as the camera or objects are displaced incrementally. Thus, techniques that are sensitive in this small displacement range will exhibit numerical problems. The reality is that there is little to no SfM information on a frame to frame basis, especially given the possible noise on the image features. This is a difficulty for almost any two-frame SfM algorithm. Thus, one must consider integrating (in batch or recursive form) the information over the sequence.

An important advantage arises in small baseline situations, though. The correspondence problem and feature tracking become easier. The proximity of features over adjacent frames can be used to guide correspondence and most feature detection techniques will exhibit more consistent behavior as images change slowly.

VI. Nonlinear Approaches

Many nonlinear frameworks can be related to the classic *Relative Orientation* problem proposed by Horn [29] [30]. The technique is a two-frame one with a perspective projection camera model. Structure and motion (but not camera internal geometry) are recovered by minimizing a nonlinear cost function.

The technique begins with a setup similar to the one in Figure 4 without any of the epipolar details. In addition, the focal length f is assumed to be given. Assume that the point P is actually represented as unknown 3D coordinates P_1 and P_2 in two different coordinate systems (one for each COP). These 3D coordinates are directly related to their 2D projections (u_1, v_1) and (u_2, v_2) by perspective projection as in Equation 7.

$$X_1 = u_1 \frac{Z_1}{f} \quad Y_1 = v_1 \frac{Z_1}{f} \tag{7}$$

The projection of P in one image plane can be defined as a translation and rotation of the 3D point in the coordinate system of the other. Thus, the 3D point P_1 in COP_1 can be computed from the 3D point P_2 in COP_2 as in Equation 8. Here, rotation R and translation $\mathbf{t}$ define the relative 3D motion between the two cameras.

$$\begin{aligned} \begin{pmatrix} X_2 \\ Y_2 \\ Z_2 \end{pmatrix} &= R \begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix} + \mathbf{t} \\ \begin{pmatrix} u_2 Z_2/Z_1 \\ v_2 Z_2/Z_1 \\ f Z_2/Z_1 \end{pmatrix} &= R \begin{pmatrix} u_1 \\ v_1 \\ f \end{pmatrix} + \mathbf{t}\frac{f}{Z_1} \end{aligned} \tag{8}$$

Each corresponded pair of points increases the system by three more equations with two more unknowns (the Z_1 and Z_2). The system can undergo an arbitrary scaling so a normality constraint is applied to translation to force a unique solution. In addition, there are orthonormality constraints on the rotation matrix (which only has 3 true degrees of freedom). Thus, the 5 unknowns for the relative 3D motion can be solved using 5 point correspondences. However, a least-squares version with more points is preferred for accuracy. The solution minimizes error using iterative nonlinear optimization. In the process, the values of Z_1 and Z_2 are solved for each corresponded point and 3D structure is recovered.

A. Batch Techniques

The above methodology can be extended to multiple images (i.e. more than 2) which becomes a batch optimization over the whole set of measurements. The relative motion must be estimated for each additional image but the structure is rigid and fixed so the system becomes more constrained. This approach was developed through work by Kumar *et al.* [34], Szeliski and Kang [52] and Weng *et al.* [61] where the SfM solution is found via Levenberg-Marquardt nonlinear minimization.

B. Filtered and Recursive Multi-Frame Techniques

Another type of extension to multi-frame is a recursive one where the multiple images are not available in batch form but rather are streaming in serially. One way of formulating this extension is by repeating the classic 2-frame estimate such as the Longuet-Higgens technique [36] or the algorithm described above [30]. Oliensis and Thomas [42] and Soatto *et al.* [50] sequentially compute such 2-frame estimates and post-process the output with a smoothing Kalman filter (KF). Here, the measurement vectors and the state vectors are the same so the KF is linear, completely observable and hence does not have any linearization problems. In fact, the KF is only acting as a smoothing filter. It is not really being used in its full capacity as a state estimator where the measurements are nonlinearly inverted to obtain state information while keeping track of the state's internal complex dynamics.

Extended Kalman Filters (EKFs) deal with nonlinearity explicitly and can be applied to nonlinearly uncover motion and structure instead of smooth the output of 2-frame techniques. EKF frameworks were utilized on image sequences by Ayache and Faugeras [1], Broida and Chellappa [11], Dickmanns and Graefe [15], Faugeras *et al.* [20], Heel [28], Matthies *et al.* [38] and Young and Chellappa [62]. A seminal paper by Broida, Chandrashekhar and Chellappa features a nonlinear EKF for recovering state information [10] . It does not rely on 2-frame techniques but rather folds the estimation into the Kalman filtering equations. The filter is used to nonlinearly invert the measurements to gather state information. One important deficiency of these techniques is that camera internal geometry is not always estimated. This is acceptable for some camera parameters such as skew, etc. which can be less significant and constant in modern cameras. However, the focal length (which is related to the zoom) readily changes in many different video situations. An additional problem is the perceived unreliability of these techniques due to the linearization at each time step in EKF calculations. We now outline our method which estimates focal length and has stronger stability properties due to parameterization changes.

VII. The Proposed Nonlinear Recursive Framework

In the following we review and discuss the formulation proposed by Azarbayejani and Pentland [2] for a *recursive* recovery of 3D structure, 3D motion and camera geometry from feature correspondences over an image sequence. The emphasis here is that the Structure from Motion is cast into a dynamical system framework with important representational improvements that provide an accurate and stable implementation. In addition, the system integrates information over a complete sequence of images in a probabilistic framework. Therefore, it is robust to errors in 2D feature tracking. The pertinent issue is the parameterization of the geometric concepts and the representation which bring forth numerical advantages, real-world reliability and new functionality. For instance, we demonstrate the recursive recovery of focal length which is free to span the full range from both perspective projection to orthographic projection. The recursive framework allows online and real-time recovery of structure, motion and focal length for true flexibility and applicability to real-time problems. Finally, the approach provides a fully metric 3D recovery of structure (versus a pseudo-structure recovery). Thus, it has important applications in graphics and post-production which traditionally operate in such representations.

A. The Dynamical System

We briefly discuss the dynamics of the Structure from Motion problem. As shown earlier, it is often the case (i.e. in cinematographic post-production, robotics, etc.) that cameras do not teleport around the scene and objects do not move about too suddenly. These bodies are governed by physical dynamics and it thus makes sense to constrain the possible configurations of the camera to have some smooth temporal changes over a causal time sequence. For instance, we consider the typical dynamic system: [4]

$$\mathbf{x}_{t+1} = \Phi \mathbf{x}_t + \mathcal{N}(0,Q) \tag{9}$$

$$\mathbf{y}_t = H(\mathbf{x}_t)\mathbf{x}_t + \mathcal{N}(0,R_t) \tag{10}$$

Here, the observations are the 2D features (in u,v coordinates) which are concatenated into an observation vector $\mathbf{y}_t$ for each moment in time. The observations are caused by the internal state of the system, $\mathbf{x}$ which contains the scene's 3D structure, the relative 3D motion between the camera and the scene and the camera's internal geometry. The mapping from $\mathbf{x}$ to $\mathbf{y}$ is tricky in SfM since it is nonlinear ($H(\mathbf{x})$ varies with $\mathbf{x}$) and is also corrupted by some noise. Here, the noise is represented as an additive Gaussian (normal $\mathcal{N}$) process with zero-mean and time-varying covariance R_t. The matrix R_t probabilistically encodes the accuracy of the measured 2D feature coordinates and can represent features that are missing in certain frames when large variances are imputed into R_t appropriately.

In addition, the dynamics of the internal state are constrained. The 3D structure, 3D motion and camera geometry do not vary wildly but are linearly dependent (via Φ) on their previous values at the past time interval plus Gaussian noise. The noise process is additive with zero-mean and covariance Q. For generality, we assume that the motion of the camera through the scene is not known a priori and thus, Φ is set to identity. Therefore, the internal state varies only through some Gaussian random noise process. This can be seen as a 'random walk' type of internal state space. In other words, the vector $\mathbf{x}$ varies randomly but smoothly with small deltas from its past values.

This dynamic system encodes the causal and dynamic nature of the SfM problem and allows an elegant integration of multiple frames from image sequences. It is also a probabilistic framework for representing uncertainty. These dynamical systems have been extensively studied are routinely solved via reliable Kalman Filtering (KF) techniques. In our nonlinear case, an Extended Kalman Filter (EKF) is utilized which linearizes $H(\mathbf{x}_t)$ at each time step.

The representation of the measurement vector $\mathbf{y}_t$ is simply the concatenation of the 2D feature point measurements. We now turn our attention to the representation of the internal state $\mathbf{x}_t$ of the unknowns of the system: the 3D structure, 3D motion and internal camera geometry. This step is critical since the effectiveness of the Kalman filtering framework depends strongly on the representation.

B. Representation

In order to develop a dynamic system with robustness and numerical stability for SfM geometry calculations,

[4] Note, more generally, all matrices can vary as functions of the state $\mathbf{x}$ and with time t.

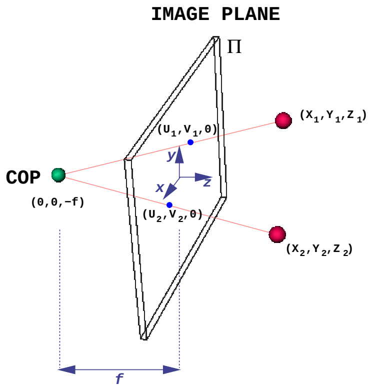


Fig. 7. Central Projection

one must interpret the SfM problem in an appropriate representation. For instance, the representation should not contain degeneracies, numerical ill-conditioning or intractable coupling between variables. Herein, we shall develop a practical derivation of the representation. A more elegant Riemannian manifold theory argument for the representations is found in [3].

A. Internal Geometry and Projection Model

In standard perspective projection, the mapping from a 3D coordinate onto the image plane is accomplished via the projection Equation 11.

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} X_C \\ Y_C \end{pmatrix} \frac{f}{Z_C} \tag{11}$$

However, we instead use the central projection representation as depicted in Figure 7. Here, the coordinate system's origin is fixed at the image plane instead of at the center of projection (COP). In addition, the focal length is parameterized by its inverse, $\beta = 1/f$. This camera model has long been used in the photogrammetry community and has also been adopted by Szeliski and Kang [52] in their nonlinear least squares formulation. The projection equation thus becomes Equation 12.

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} X_C \\ Y_C \end{pmatrix} \frac{1}{1 + Z_C\beta} \tag{12}$$

Note how this projection decouples the camera focal length (f) from the depth of the point (Z_C). In the traditional projection Equation 11, if Z_C is fixed and the f is altered, the imaging geometry remains the same while the scale of the image changes. In other words, the cone of perspective rays remains fixed while the focal plane (Π) translates along the optical (Z) axis. We note that in the standard projection model, the imaging geometry (i.e. the perspective rays) are only altered by varying depth Z_C which is the only way to alter the imaging geometry. Thus, f only acts as a scaling factor and the imaging geometry *and* the depth are encoded in Z_C.

In our representation, however, the inverse focal length β alters the imaging geometry independently of the depth value Z_C. State variable decoupling is known to be critical in Kalman filtering frameworks and is applicable here since we plan on putting *both* camera internal geometry β and structure Z_C into the internal hidden state $\mathbf{x}$.

Another critical property of β as opposed to f is that it does not exhibit numerical ill-conditioning. It can span the wide range of perspective projection but also the special case of orthographic projection which occurs when we set the focal length $f = \infty$ and all rays project orthogonally onto the image plane. However, under orthographic projection, $\beta = 0$ which does not 'blow up' and maintains numerical stability in KF frameworks. We can thus combine both perspective and orthographic projection into the same so-called central projection framework without any numerical instabilities (this is demonstrated experimentally in the next section). This flexibility is not typical in many traditional computer vision approaches where perspective and orthographic projection must be treated quite differently. We now begin building our internal state vector with this well-behaved parameter, β as in Equation 13.

$$\mathbf{x}_1 = (\text{camera internal geometry}) = \beta \tag{13}$$

B. 3D Structure Model

We assume that N feature points are to be tracked over F frames in an image sequence. In the first frame, each "feature point" is initially in terms of an image location (u,v). Subsequent frames are then observed and the image location of the feature is measured with some noisy zero-mean Gaussian error. One may think of the 3D structure of the model as the true (X_C, Y_C, Z_C) coordinates of each of the 3D points which then project into (u, v) 2D coordinates. However, this obvious parameterization is not compact and stable. For one thing, it contains 3 unknowns (or 3 degrees of freedom) per point being tracked resulting in the concatenation of $3 \times N$ dimensions to our internal state vector $\mathbf{x}$. Another problem with this representation is that it in no way encodes the rigidity of the object being tracked.

A more compact representation of the 3D location is shown in Equation 14 below. Here, there is only one degree of freedom per point, α. The first term represents the initial image location and the second term represents the perspective ray scaled by the unknown depth α:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} u \\ v \\ 0 \end{pmatrix} + \alpha \begin{pmatrix} u\beta \\ v\beta \\ 1 \end{pmatrix} \tag{14}$$

Point-wise structure, therefore, can be represented with one parameter per point (instead of 3).

This representation is consistent with early analyses such as [29], but inconsistent with representations used in much of the image sequence work, including [10;42;60], which use three parameters per point. It is critical for estimation stability, however, either to use this basic parameterization or to understand and properly handle the additional parameters. Here we describe the computational implications of our parameterization and in Section VII.D we show how it relates to alternate parameterizations.

First consider the total number of unknowns that are to be recovered in a batch solution of these F frames and N points. There are $6(F-1)$ motion parameters[5] and $3N$ structure parameters. Point measurements contribute $2NF$ constraints and one arbitrary scale constraint must be applied. Hence, the problem can be solved uniquely when $2NF+1 > 6(F-1)+3N$. Thus all motion and structure parameters can in principle be recovered from any batch of images for which $F \geq 2$ *and* $N \geq 6$.

However, in a recursive solution, not all constraints are applied concurrently. At each step of computation, one frame of measurements constrains all of the structure parameters and one set of motion parameters, i.e. $2N$ measurements constrain $6+3N$ degrees of freedom at each frame. This is always an under-determined computation having the undesirable property that the more features that are added, the more under-determined it is. Unless one already has low prior uncertainty on the structure (effectively reducing the dimensionality of unknowns), one should expect unstable and unpredictable estimation behavior from such a formulation. Indeed, in [10], it was proposed that such filters only be used for "tracking" after a batch procedure is applied for initialization.

On the other hand, in our formulation, constraints ($1+2N$) outnumber degrees of freedom ($6+1+N$) for motion, camera, and structure at every frame when $N > 7$. The more measurements available the larger the gap. Our experiments verify that the overdeterminancy results in better stability, allowing for good convergence and tracking in most cases without the requirement of good prior information. In both types of formulation, once structure (and camera, in our case) has converged, each step is effectively overconstrained; the only issue is stability when structure (and camera) is grossly uncertain.

It is clear, then, that excess parameters are undesirable for stability, but how can both $3N$- and N-parameter representations describe the same structure? Section VII.D relates the two and demonstrates that when measurement biases are exactly zero (or known) the $3N$ space really only has N degrees of freedom. Even in the presence of bias, most uncertainty remains along these N DOFs, justifying the structure parameterization

$$\mathbf{x}_2, \cdots, \mathbf{x}_{1+N} \; = \; (\text{structure}) = (\alpha_1, \cdots, \alpha_N)$$

We show experimentally that even relatively large biases do not have a strong adverse effect on accuracy using this more concise model.

C. 3D Motion Model - Translation

The translational motion is represented as the 3-D location of the object reference frame relative to the current camera reference frame using the vector

$$\mathbf{t} = (t_X, t_Y, t_Z)$$

The t_X and t_Y components correspond to directions parallel to the image plane, while the t_Z component corresponds to the depth of the object along the optical axis. As such, the sensitivity of image plane motion to t_X and t_Y motion will be similar to each other, while the sensitivity to t_Z motion will differ, to a level dependent upon the focal length of the imaging geometry.

For typical video camera focal lengths, even with "wide angle" lenses, there is already much less sensitivity to t_Z motion than there is to (t_X, t_Y) motion. For longer focal lengths the sensitivity decreases until in the limiting orthographic case there is zero image plane sensitivity to t_Z motion.

For this reason, t_Z cannot be represented explicitly in our estimation process. Instead, the product $t_Z\beta$ is estimated. The coordinate frame transformation equation

$$\begin{pmatrix} X_C \\ Y_C \\ Z_C\beta \end{pmatrix} = \begin{pmatrix} t_X \\ t_Y \\ t_Z\beta \end{pmatrix} + \begin{pmatrix} 1 & & \\ & 1 & \\ & & \beta \end{pmatrix} \mathbf{R} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{15}$$

combined with Equation 12 demonstrates that only $t_Z\beta$ is actually required to generate an equation for the image plane measurements (u, v) as a function of the motion, structure, and camera parameters (rotation $\mathbf{R}$ is discussed below).

Furthermore, the sensitivity of $t_Z\beta$ does not degenerate at long focal lengths as does t_Z. For example, the sensitivities of the u image coordinate to both t_Z and $t_Z\beta$ are

$$\frac{\partial u}{\partial t_Z} = \frac{-X_C\beta}{(1+Z_C\beta)^2} \quad \text{and} \quad \frac{\partial u}{\partial (t_Z\beta)} = \frac{-X_C}{(1+Z_C\beta)^2}$$

demonstrating that $t_Z\beta$ remains observable from the measurements and is therefore estimable for long focal lengths, while t_Z is not (β approaches zero for long focal lengths).

Thus we parameterize translation with the vector

$$(translation) = (t_X, t_Y, t_Z\beta)$$

True translation $\mathbf{t}$ can be recovered post-estimation simply by dividing out the focal parameter from $t_Z\beta$. This is valid only if β is non-zero (non-orthographic), which is desirable, because t_Z is not geometrically recoverable in the orthographic case. To see this mathematically, the error variance on t_Z will be the error variance on $t_Z\beta$ scaled by $1/\beta^2$, which gets large for narrow fields of view.

[5] These encode the relative 3D rotation and the 3D translation between the camera and the scene.

D. 3D Motion Model - Rotation

The 3-D rotation is defined as the relative rotation between the object reference frame and the current camera reference frame. This is represented using a unit quaternion, from which the rotation matrix ($\mathbf{R}$) can be generated:

$$\begin{pmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{pmatrix} \tag{16}$$

The four elements of the unit quaternion only have three degrees of freedom due to the normality constraint. Thus, all four cannot be estimated independently; only a nonlinear constrained minimization will work to recover the quaternion directly. Since the EKF utilizes a linearization at each step, the nonlinear normality constraint cannot be rigidly enforced within the EKF computational structure.

However, a 3-parameter incremental rotation representation, similar to that used in [11], can be used in the EKF to estimate interframe rotation at each frame. Incremental Euler angles centered about zero (or discrete-time "rotational velocity") do not overparameterize rotation and are approximately independent and therefore can be used reliably in a system linearization.

The incremental rotation quaternion is a function of these three parameters:

$$\delta\mathbf{q} = \left(\sqrt{1-\epsilon}, \omega_X/2, \omega_Y/2, \omega_Z/2\right) \tag{17}$$

$$\epsilon = (\omega_X^2 + \omega_Y^2 + \omega_Z^2)/4 \tag{18}$$

This incremental rotation can be computed at each frame and then composed with an external rotation quaternion to maintain an estimate of global rotation. The global quaternion is then used in the linearization process at the next frame.

Thus, we have,

$$\text{(interframe rotation)} = (\omega_X, \omega_Y, \omega_Z)$$

$$\text{(global rotation)} = (q_0, q_1, q_2, q_3)$$

where interframe rotation is part of the EKF state vector and global rotation is maintained and used in the linearization at each step.

E. The Issue of Scale

It is well known that the shape and motion geometry in SfM problems such as this are subject to arbitrary scaling and that this scale factor cannot be recovered. (The imaging geometry β and the rotation *are* recoverable and not subject to this scaling.) In two-frame problems with no information about true lengths in the scene, scale factor is usually set by fixing the length of the "baseline" between the two cameras. This corresponds to the magnitude of the translational motion.

It is equally acceptable to fix any other single length associated with the motion or the structure. In many previous formulations, including [10;42] some component of the translational motion is fixed at a finite value. This is not a good practice for two reasons. First, if the fixed component, e.g. the magnitude of translation is actually zero (or small), the estimation becomes numerically ill-conditioned. Second, every component of motion is generally dynamic, which means the scale changes at every frame! This is disastrous for stability and also requires some post-process to rectify the scale.

A better approach to setting the scale is to fix a static parameter. Since we are dealing with rigid objects, all of the shape parameters $\{\alpha_i\}$ are static. Thus, fixing any one of these establishes a uniform scale for all motion and structure parameters over the entire sequence. The result is a well-conditioned, stable representation. Setting scale is simple and elegant in the EKF; the initial variance on, say, α_0 is set to zero, which will fix that parameter at its initial value. All other parameters then automatically scale themselves to accommodate this constraint. This behavior can be observed in the experimental results.

C. The EKF Implementation

Using the representations discussed thus far, our composite state vector consists of $7+N$ parameters—6 for motion, 1 for camera geometry, and N for structure—where N is the number of features tracked:

$$\mathbf{x} = (t_X, t_Y, t_Z\beta, \omega_X, \omega_Y, \omega_Z, \beta, \alpha_1, \cdots, \alpha_N) \tag{19}$$

The vector $\mathbf{x}$ is the state vector used in a standard EKF implementation, where the measurement vector contains the image locations of all the tracked features in a new frame. As described earlier, an additional quaternion is required for maintaining a description of the global rotation external to the EKF.

The dynamics model in the EKF can be chosen trivially as an identity transform plus noise, unless additional prior information on dynamics is available. The measurement equation is simply obtained by combining Equations 12, 15, and 14. The RHS (u,v) in Equation 14 is the defining image location of the feature in its initial frame, and the LHS (u,v) in Equation 12 is the measurement.

The final implementation of the EKF is straightforward (standard references include [10;12;21]), with the only additional computation being the quaternion maintenance. Computationally, the filter requires inverting a $2N$x$2N$ matrix (i.e. the size of the measurement vector) [12;21], which is not a large task for the typical number of features on a single object. Since all parameters are overdetermined with 7 or more points, N rarely needs to be more than 15 or 20 for good results, yielding filter steps which can be computed in real-time on modern workstations.

D. Biased Measurements

We turn attention here to the issue of biased measurement noise in the EKF and how it relates to representation of object structure.

We have assumed that features are identified in the first frame and that measurements are obtained by comparing new images to the previous images and that our measurements are zero-mean or very close to zero-mean. This thinking leads to the α description of structure given earlier in which the single unknown depth for each feature fully describes structure. These parameters can be computed very effectively using the EKF, which assumes zero-mean measurements.

It is common to use Kalman filters even when measurements are not truly zero-mean. Good results can be obtained if the biases are small. However, if the measurements are biased a great deal, results may be inaccurate. In the case of large biases, the biases are observable in the measurements and can therefore be estimated by augmenting the state vector with additional parameters representing the biases of the measurements. In this way, the Kalman filter can in principle be used to estimate biases in all the measurements.

However, there is a tradeoff between the accuracy that might be gained by estimating bias and the stability of the filter, which is reduced when the state vector is enlarged. When the biases are large, i.e. compared to the standard deviation of the noise, they can be estimated and can contribute to increased accuracy. But if the biases are small, they cannot be accurately estimated and they do not affect accuracy much. Thus, it is only worth augmenting the state vector to account for biases when the biases are known to be significant relative to the noise variance.

In the SfM problem, augmenting the state vector to account for bias adds two additional parameters per feature. This results in a geometry representation having a total of $7 + 3N$ parameters. Although we do not recommend this level of state augmentation, it is interesting because it can be related to the large state vector used in [10;42] and others, where each structure point is represented using three free parameters (X,Y,Z).

If we add noise bias parameters (b_u,b_v), Equation 14 can be written

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \begin{pmatrix} (1+\alpha\beta)(u+b_u) \\ (1+\alpha\beta)(v+b_v) \\ \alpha \end{pmatrix} \tag{20}$$

This relation is invertible so the representations are analytically equivalent. However, geometrically the (α, b_u, b_v) parameterization is more elucidating than (X, Y, Z) because it parameterizes structure along axes physically relevant to the measurement process. Thus, it allows us to more effectively tune the filter, ultimately reducing the dimensionality of the state space quite significantly.

It is clear that, in general, uncertainty in α trivializes uncertainty in the direction of the biases. By using initial error variance on α that is high in comparison to the error variances on (b_u, b_v), the state space is essentially reduced because the system responds more stiffly in the direction of the biases, favoring instead to correct the depths. In the limit (zero-mean-error tracking) the biases can be removed completely, resulting in the strictly lower dimensional formulation that we typically use in this paper. Our experimental results demonstrate that bias is indeed a second-order effect and is justifiably ignored in most cases.

VIII. Experiments

To test our recursive structure from motion system, a variety of simulations were carried out with synthetic data. These tests are detailed in [2] and demonstrate the accuracy and reliability of the method. Structure, 3D motion and focal length were stably recovered in situations that included increased noise levels, orthographic projection, increased noise bias and degenerate rotational motion. The estimator performed accurately despite extreme levels of noise and was robust to bias as well. In addition, it properly handled the orthographic and degenerate rotation cases which typically cause problems for other techniques and camera models. In addition, an extensive Monte Carlo analysis was performed with thousands of trials to confirm the stability of the technique and show smooth degradation in increasing noise conditions.

In this section, we present results from applying our estimation formulation to two sequences of real imagery. Additional real imagery results can also be found in [2].

A. Experiment 1: Egomotion, Models from Video

In this example, a texture-mapped model of a building is extracted from a 20-second video clip of a walk-around outside a building (the Media Laboratory, MIT). Figure 8(a) shows two frames of the original digitized video with feature points overlaid.

Twenty-one features on the building were tracked and used as measurement input to the EKF described earlier. The resulting estimates of camera geometry, camera motion, and pointwise structure are shown in Figure 10. The EKF is iterated once to remove the initial transient.

Recovered 3-D points were used to estimate the planar surfaces of the walls. The vertices were selected in an image by hand and back projected onto the planes to form 3-D polygons, depicted in wireframe in Figure 8(b). These polygons, along with the recovered motion and focal length were used to warp and combine video from 25 separate frames to synthesize texture maps for each wall using a procedure developed by Galyean [6]. In Figure 8(c,d), the texture-mapped model is rendered along the original trajectory and at some novel viewing positions.

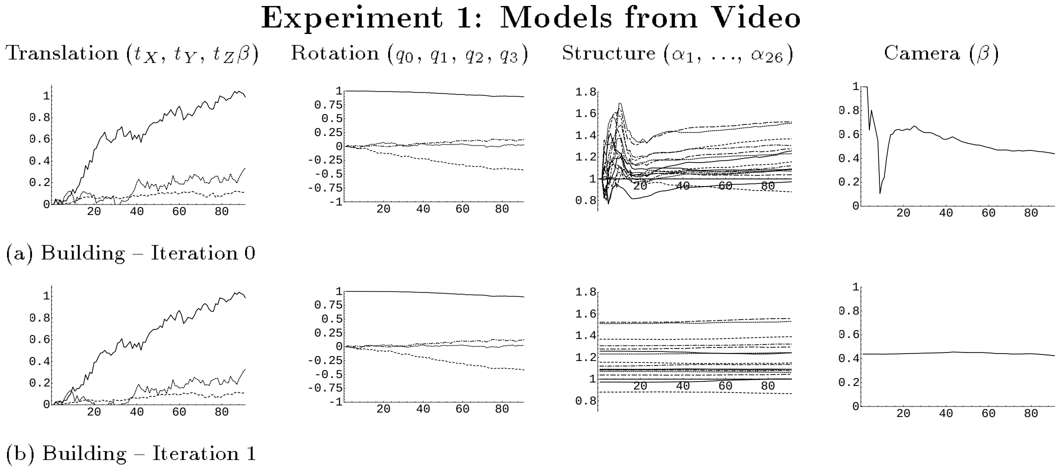


Fig. 10. Experiment 1: Models from Video. Structure, motion, and focal length recovered from 2D features.

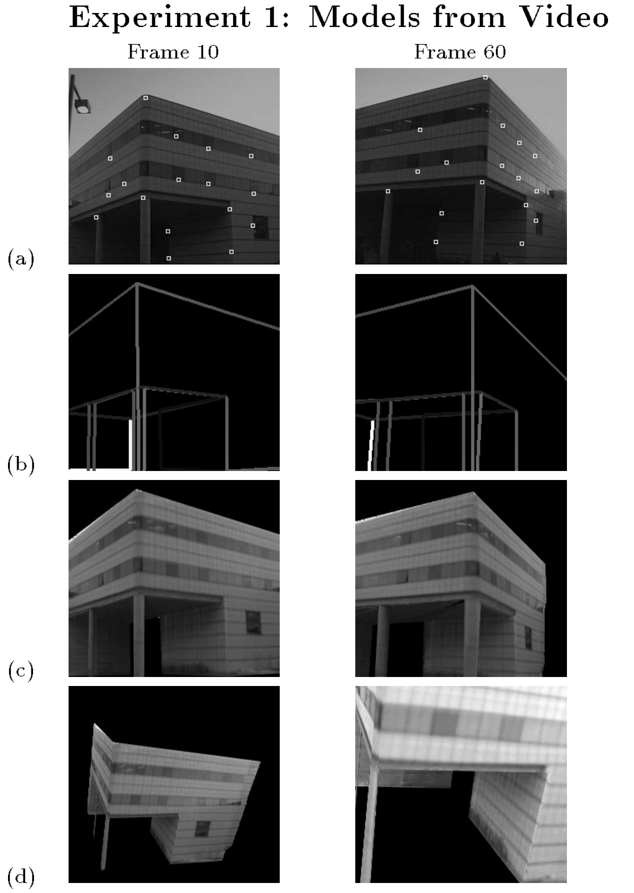


Fig. 8. Experiment 1: Recovering Models from Video. (a) Features are tracked from video using normalized correlation. (b) 3-D polygons are obtained by segmenting a 2-D image and back-projecting the vertices onto a 3-D plane. This 3-D plane is computed from the recovered 3-D points corresponding to image features in the 2-D polygon. (c) Texture maps are obtained by projecting the video onto the 3-D polygons. The estimated motion and camera parameters are used to warp and combine the video from 25 separate frames to create the texture map for each polygon. (d) Alternate views of the recovered model.

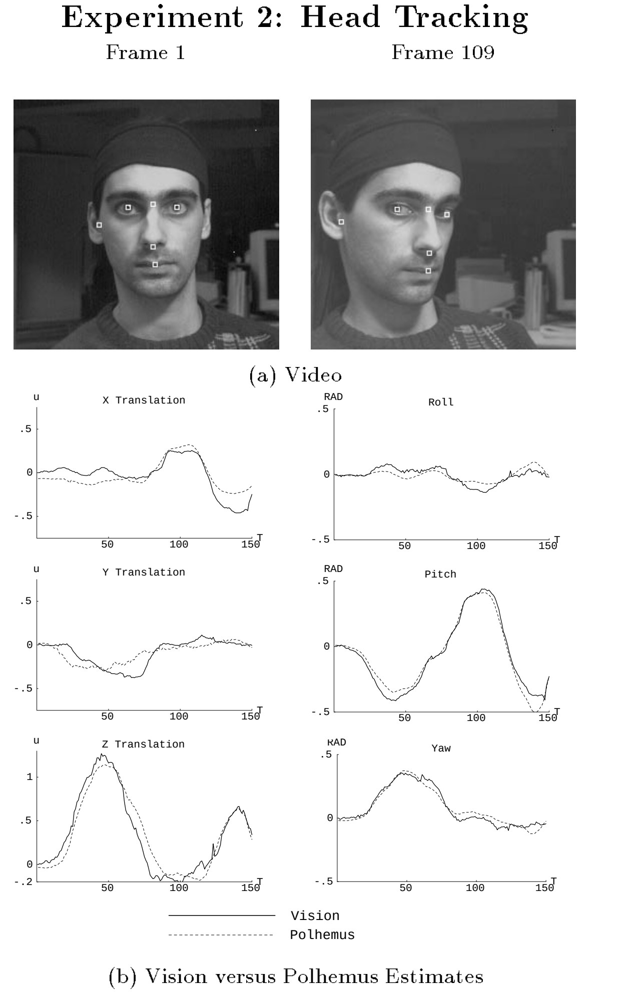


Fig. 9. Experiment 2: Head Tracking. (a) 2D Feature tracking, (b) Vision and Polhemus estimates of head position. Much of the observed error is known to be due to Polhemus error. RMS differences are 0.11 units and 2.35 degrees.

B. Experiment 2: Object Tracking – Head

In this experiment, a person's head was tracked using both the vision algorithm and the Polhemus magnetic sensor [6] simultaneously. Figure 9 shows the vision estimate and Polhemus measurements (after an *absolute orientation* [29] was performed to align the estimates properly). The RMS difference in translation is 0.11 units and the RMS difference in rotation is 2.35°. (The scale of translation is, of course, unknown, but is approximately 10–12cm per unit, yielding a RMS tracking error of approximately 1 cm.) This yields accuracy on the order of the observed accuracy of the Polhemus sensor, indicating that the vision estimate is at least as accurate as the Polhemus sensor.

This example is identical to the example presented in our earlier work on vision-based head tracking [5], except here we recover focal length and structure simultaneously with motion. The previous work relied on a rough, *a priori* structural model and calibration of focal length. The RMS errors between vision and Polhemus estimates for this example were slightly better than those in the previous study, ($\sim$1cm versus 1.67cm and 2.35 degrees versus 2.4 degrees).

C. Independent Evaluation

In an independent evaluation performed by Soatto and Perona [49], a technique similar to the proposed one demonstrated good results when compared to other variants. The similar algorithm (referred to as the "Structure Integral Filter" in their paper) was cast into a generic evaluation framework [48] and then compared with subspace filters [27], essential filters and fixation constraint methods. The system performed well in terms of accuracy, robustness and noise tolerance. The authors reported some sensitivity to initialization errors in the technique. However, the aforementioned bias estimation component did not seem to be included in the estimation process. Erroneous initialization is precisely the reason for including a bias estimation stage and it can easily be added to the framework when these situations are expected.

IX. Extending the Framework: Feedback and Feedforward Tracking

Since the development of the above recursive SfM technique, we have explored its interaction as a module in other vision frameworks. The module is fed 2D data and computes 3D estimates. However, it is not necessarily a processing dead-end. One can consider SfM as a subcomponent in larger vision systems with multiple loops. Thus, SfM can feed its results back to lower level vision processes or forward to higher level modules.

One real-time application developed by Jebara and Pentland [32] is the automatic real-time 3D face tracking system shown in Figure 11. An automatic initialization module finds the face, locating eyes, nose and mouth coordinates in under a second. These are then used to initialize 8 normalized correlation tracking squares (i.e. sum-squared distance minimization [22]) on the face.

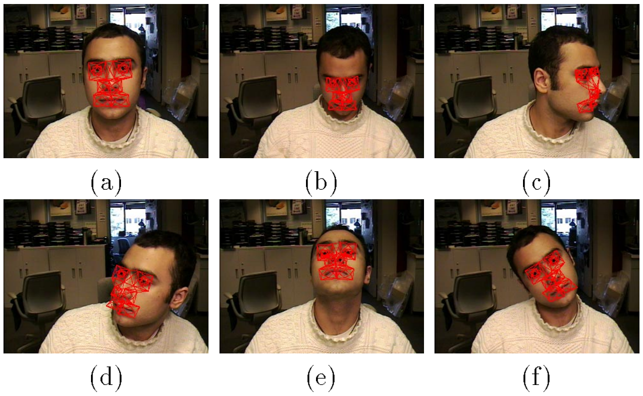


Fig. 11. Real-Time 3D Face Tracker

Each square can translate, rotate and scale and so is equivalent to two 2D point features (Figure 12(a)(b)(c)). The resulting 16 features are fed into the SfM algorithm resulting in the recovery of 16 *rigid* 3D points. This estimated rigid 3D model is then reprojected onto the image plane to generate a set of 16 *rigidly constrained* 2D points. These points are used to relocate the individual trackers for tracking the motion in the next frame. The trackers estimate an instantaneous trajectory yet are not permitted to follow through with it (i.e. in a nearest-neighbor tracking framework). Instead, this estimate is used in the SfM which computes the corresponding rigid trajectory and repositions the trackers along this rigid 'path' for the next frame in the sequence. Thus, instead of letting each square individually track, the SfM couples them all, forcing them to behave as if they were glued onto a rigid 3D body (i.e. a 3D face). Furthermore, the 8 trackers output an error level which can be used in the R matrix in the SfM Kalman filtering to adaptively weight good features more than bad features in the 3D estimates. Feature errors are mapped into a Gaussian uncertainty in localization by an initial perturbation analysis which computes each tracker's error sensitivity under small displacements.

The end result is a much more stable tracking framework (operating at 30Hz). If some trackers are occluded or fail, the others pull them along via the imposed rigidity constraint. The feedback from the adaptive Kalman filter maintains a sense of 3D structure and enforces a global collaboration between the separate 2D trackers. Thus, tracking remains stable for minutes instead of seconds (if no feedback SfM is used). Figure 12(d) depicts the stability under occlusion where a mouth and eye tracker are distracted by the presence of the user's finger. Similarly in Figure 12(e), the mouth tracker is distracted by deformation (smiling) where the mouth is no longer similar to the closed mouth the template was initialized with. These conditions remain stable due to the feedback loop.

The algorithm also re-initializes when it detects that

[6] The Polhemus sensor is physically attached to the head

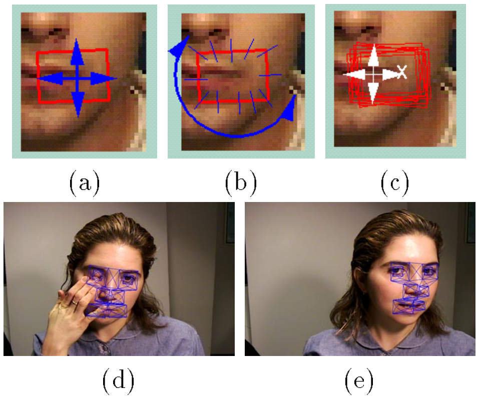


Fig. 12. Correlation Tracking and Feedback

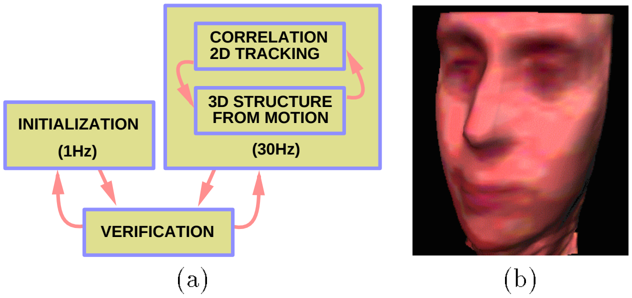


Fig. 13. Feedback for 3D Stable Tracking and Feeding Forward for Detailed 3D Model Recovery

it has lost the face as in Figure 13(a). This detection is performed via the so-called "Distance-from-Face-Space" calculation which essentially computes the probability of a face pixel image with respect to a constrained Gaussian distribution [40]. While multiple real and synthetic tests show very strong convergence we have also used the system extensively in the above real-time application settings where it behaved consistently and reliably.

We can also feed forward the SfM results. Recall that SfM recovers 3D pose (or motion) as well as 3D structure. In the above, the 16 3D points recovered are not detailed enough to generate a 3D graphical model of the face (adding points is too slow, with complexity $\approx O(N^3)$). Instead, 3D pose is fed forward into a module that unwarps the face into a standard mug-shot pose for a second stage where a fast, linear 3D shape estimator can be used (see Jebara, Russel and Pentland [33]), to compute a full 3D model. The result is shown in Figure 13(b) which is a full 3D model computed in real-time from facial images of the user seen in Figure 11.

X. Practical Experience in Commercial Post-Production Applications

Years of experience and comparative testing in the film and video post-production industry by the authors and software developers has demonstrated the effectiveness and importance of our computational foundation of nonlinear and probabilistic modeling for 3D computer vision over competing approaches. In the film and video production industry, software based on the principles of the above SfM technique has been used in a commercial setting by Alchemy 3D Technology's MatchMaker(TM) and Alias—Wavefront's MayaLive(TM) software products [43] and has recently been chosen to contribute to the feature-based vision subsystem of SynaPix's SynaFlex(TM) [51] system.

In this industry, there has been a proliferation of demand for computer-graphics based special effects. An emerging staple in producing such effects is the process of "3D compositing", in which 2D-source imagery (i.e. film, video, or digital image sequences) is combined with 3D-source imagery (i.e. 3D computer graphics) in a realistic and metrically accurate fashion by first recovering an accurate 3D representation of the 2D-source imagery using computer vision techniques. The computer vision component is known as "3D matchmoving" and results in a 3D representation of camera motion, scene geometry, and camera imaging geometry.

Product developers have sought for years to develop reliable vision front-ends to facilitate this growing need and have considered all available published work in the field as candidate technology, including linear algebraic techniques, photogrammetric techniques, and optical-flow-based techniques. The selection of software based on our technology as the basis of several major matchmoving systems is testimony to the practical importance of the theoretical foundation as borne out in results of objective testing in the field against competing approaches. In particular, software based on our technology has exhibited substantially greater efficiency, reliability, accuracy, flexibility, and extensibility. There is sound theoretical grounding for these observations.

Efficiency arises from the ability to combine probabilistic representations of information recursively. In typical cinematic sequences of 200-300 frames, software based on our techniques usually obtain complete solutions in 00:00:30 (30 seconds) to 00:08:00 (8 minutes), whereas comparable solutions with photogrammetric or other nonlinear approaches on the same sequences typically require many hours, often overnight processing.

Reliability arises from the stability associated with probabilistic rather than rigid linear algebraic modeling of spatial and dynamic processes. In the presence of noisy input, our techniques have proven to be resilient where competing methods produce nonsensical output. Long "dolly" shots (primarily translation along z-axis) in particular have proven difficult for most solution techniques, but our techniques routinely acquire accurate solutions, including in extreme conditions, e.g. a 1550-frame helicopter shot with over 70 features and large turnover of features [4].

Accuracy arises from nonlinear 3D scene-based modeling. Linear algebraic techniques are fragile in the presence of real-world data and modeling imperfections and often do not even produce a useful 3D Euclidean output.

For post-production applications which depend upon useful 3D (Euclidean) output to match 3D CG representations, there is no substitute for Euclidean modeling. Optical flow methods produce dense depth maps, but since they are view-based and based on pairs of closely spaced images, there is no easy or sufficiently general way of producing a consistent and accurate scene-based 3D description and there is no general way of controlling scaling and drift.

Flexibility arises from probabilistic modeling. Since probabilistic modeling facilitates accumulation and propagation of information, such modeling allows efficient solution of otherwise difficult sequences. Among these are sequences in which features disappear and reappear and those in which there is insufficient visual information throughout all frames. Competing systems have found it particularly difficult to solve cinematic shots in which features appear and disappear due to foreground occlusions caused by, e.g., actors and vehicles, those in which only a camera pan (pure rotation) is present, those in which almost the entire feature set changes from the start to the end, and those in which large segments of the sequence are completely unusable (e.g. due to practical effects such as steam, explosions, or blinding light). Software based on our techniques routinely solve these types of cinematic shots because probabilistic modeling can be used to account for missing information.

Finally, extensibility arises from probabilistic modeling. Many shots encountered in cinematic post-production do not have ideal camera motions for 3D recovery purely from 2D visual motion. In these cases, additional information about scene structure is necessary to obtain complete solutions. The information can come in many forms and must be integrated in some consistent fashion. Probabilistic modeling has long been used as the foundation for integrating information from qualitatively different sources, and this application is no exception. Since the visual process is already modeled probabilistically, the integration of, e.g., scene-based measurements with visual feature measurements has been able to take place quite naturally and allow shots to be solved that techniques based purely on visual relationships could not possibly have solved completely.

In short, the theoretical foundation of nonlinear and probabilistic modeling for 3D computer vision has borne itself out in at least one industry with an important application using these techniques. The objective nature of the arena in which the technology has competed and is now enjoying growing preference lends credence to the fundamental practical advantages of the formulation. From an evolutionary standpoint, the observed flexibility and extensibility in particular offer the greatest indication that the technology can find an important place both in further software applications and in a larger framework for perceptual information processing.

XI. Concluding Remarks

From the original inspirations of AI and vision to the aforementioned practical uses for Structure from Motion, multiple approaches have been proposed. These range from linear to non-linear, perspective to orthographic, 2-frame to multi-frame, and recursive to batch tradeoffs. The methods have their own idiosyncrasies with different input features, different accuracies, degeneracies, and flexibilities. Ultimately, the choice of which framework to use is application dependent.

We have also discussed the use of these techniques on video and continuous motion sequences and the implications that has on the algorithms. We emphasized our nonlinear recursive probabilistic approach for its stability and ease of use in these environments. These capabilities were confirmed in synthetic experiments, in real imagery experiments, under independent evaluation by others, within large vision implementations for real-time tracking, and with post-production industry evaluation. Fundamentally, this technique proved accurate, practical and resilient to a wide variety of such tests.

XII. Acknowledgments

The authors thank Alias—Wavefront and Alchemy 3D Technology for data and results of testing and evaluation.

References

[1] Nicholas Ayache and Olivier Faugeras. Maintaining representations of the environment of a mobile robot. *IEEE Trans. Robotics Automation*, 5(6):804–819, 1989.

[2] A. Azarbayejani and A Pentland. Recursive estimation of motion, structure, and focal length. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(6), 1995.

[3] A.J. Azarbayejani. *Nonlinear Probabilistic Estimation of 3-D Geometry from Images.* PhD thesis, Massachusetts Institute of Technology, February 1997.

[4] Ali Azarbayejani, Chris Perry, and Alex Pentland. Vision-based modeling for production-quality integration of photographic imagery and 3D graphics. In *Visual Proceedings, Siggraph '96*, page 155, New York, NY, August 1996. ACM Siggraph, Association for Computing Machinery.

[5] Ali Azarbayejani, Thad Starner, Bradley Horowitz, and Alex Pentland. Visually controlled graphics. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(6):602–605, June 1993.

[6] Ali J. Azarbayejani, Tinsley Galyean, Bradley Horowitz, and Alex Pentland. Recursive estimation for cad model recovery. In *2nd CAD-based Vision Workshop*, Los Alamitos, CA, February 1994. IEEE Computer Society, IEEE Computer Society Press. (Champion, PA).

[7] P.A. Beardsley, A. Zisserman, and D.W. Murray. Navigation using affine structure from motion. In *ECCV94*, pages B:85–96, 1994.

[8] A. Blake and A. Yuille. *Active vision.* MIT Press, 1992.

[9] P. Brand, R. Mohr, and Ph. Bobet. Distorsion optique: Correction dans un modele projectif. In *Actes du 9eme Congres AFCET de Reconnaissance des Formes et Intelligence Artificielle*, pages 87–98, Paris, France, January 1994.

[10] Ted J. Broida, S. Chandrashekhar, and Rama Chellappa. Recursive estimation of 3-d motion from a monocular image sequence. *IEEE Trans. Aerosp. Electron. Syst.*, 26(4):639–656, July 1990.

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

[12] Robert Grover Brown. *Introduction to Random Signal Analysis and Kalman Filtering.* John Wiley & Sons, New York, 1983.

[13] P.E. Debevec, C.J. Taylor, and J. Malik. Modeling and rendering architecture from photographs. In *SIGGRAPH '96*, August 1996.

[14] U. Dhond and J. Aggarwal. Structure from stereo - a review. *IEEE Transactions on Systems, Man and Cybernetics*, 19(6), 1989.

[15] Ernst Dieter Dickmanns and Volker Graefe. Dynamic monocular machine vision. *Machine Vision and Applications*, 1:223–240, 1988.

[16] O. Faugeras. What can be seen in three dimensions from an uncalibrated stereo rig? In *Proceedings of the 2nd European Conference on Computer Vision*, pages 563–578, Santa Margherita Ligure, Italy, 1992. Springer-Verlag.

[17] O. Faugeras. From geometry to variational calculus: Theory and applications of three-dimensional vision. In *Computer Vision for Virtual Reality Based Human Communications, CVVRHC'98*, Bombay, India, January 1993. IEEE Computer Society.

[18] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint.* MIT Press, 1993.

[19] O. Faugeras and T. Papadopoulo. A nonlinear method for estimating the projective geometry of 3 views. In *Sixth International Conference on Computer Vision*, pages 477–484, January 1998.

[20] Olivier D. Faugeras, Nicholas Ayache, and B. Faverjon. Building visual maps by combining noisy stereo measurements. In *Proc. IEEE Conf. on Robotics and Automation*, April 1986. (San Francisco, CA.).

[21] Arthur Gelb, editor. *Applied Optimal Estimation.* MIT Press, Cambridge, MA, 1974.

[22] G.D. Hager and P.N. Belhumeur. Real time tracking of image regions with changes in geometry and illumination. In *CVPR96*, pages 403–410, 1996.

[23] R. Hartley. Lines and points in three views - an integrated approach. In *Proceedings of the ARPA IU Workshop.* DARPA, Morgan Kaufmann, 1994.

[24] R. Hartley. In defence of the 8-point algorithm. In *Proceedings of the 5th International Conference on Computer Vision*, pages 1064–1070, Cambridge, Massachusetts, USA, 1995.

[25] R. Hartley. Kruppa's equations derived from the fundamental matrix. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(2):133–135, February 1997.

[26] R. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 761–764, Urbana-Champaign, Illinois, 1992.

[27] D. Heeger and A. Jepson. Subspace methods for recovering rigid motion i: Algorithm and implementation. *International Journal of Computer Vision*, 7(2):95–117, 1992.

[28] Joachim Heel. Temporally integrated surface reconstruction. In *ICCV '90.* IEEE, 1990.

[29] Berthold Klaus Paul Horn. *Robot Vision.* MIT Press, 1986.

[30] Berthold Klaus Paul Horn. Relative orientation. *International Journal of Computer Vision*, 4(1):59–78, January 1990.

[31] T. Huang and A. Netravali. Motion and structure from feature correspondences: A review. *Proceedings of the IEEE*, 82(2), 1994.

[32] T. Jebara and A. Pentland. Parametrized structure from motion for 3d adaptive feedback tracking of faces. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[33] T. Jebara, K. Russel, and A. Pentland. Mixtures of eigenfeatures for real-time structure from texture. In *Proceedings of the International Conference on Computer Vision*, 1998.

[34] Ratnam V. Raja Kumar, Arun Tirumalai, and Ramesh C. Jain. A non-linear optimization algorithm for the estimation of structure and motion parameters. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 136–143, June 1989. (San Diego, CA.).

[35] K. Kutulakos. Altering reality through interactive image and video manipulation. In *Computer Vision for Virtual Reality Based Human Communications, CVVRHC'98*, Bombay, India, January 1993. IEEE Computer Society.

[36] H. C. Longuet-Higgens. A computer algorithm for reconstructinga scene from two projections. *Nature*, 293:133–135, 1981.

[37] D. Marr and T. Poggio. Cooperative computation of stereo disparity. *Science*, 194:282–287, October 1976.

[38] Larry Matthies, Takeo Kanade, and Richard Szeliski. Kalman filter based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3(3):209–236, 1989.

[39] S. Maybank. *Theory of Reconstruction from Image Motion.* Springer-Verlag, 1993.

[40] B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *ICCV95*, pages 786–793, 1995.

[41] R. Mohr and B. Triggs. Projective geometry for image analysis. Technical report, International Society for Photogrammetry and Remote Sensing, Vienna Congress, July 1996. WG III/2 Tutorial.

[42] J. Oliensis and J. Inigo Thomas. Incorporating motion error in multi-frame structure from motion. In *IEEE Workshop on Visual Motion*, pages 8–13, Los Alamitos, CA, October 1991. IEEE Computer Society, IEEE Computer Society Press. (Nassau Inn, Princeton, NJ.).

[43] Harri Paakkonen. Alias—wavefront's mayalive: Plug-in tracking helps meld animation and live action. *Millimeter*, 26(6), October 1998.

[44] Conrad J. Poelman and Takeo Kanade. A paraperspective factorization method for shape and motion recovery. CMU-CS 92-208, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213-3890, October 1992.

[45] J.G. Semple and G.T. Kneebone. *Algebraic Projective Geometry.* Oxford Science Publication, 1952.

[46] A. Shashua and M. Werman. On the trilinear tensor of three perspective views and its underlying geomtry. In *International Conference on Computer Vision*, 1995.

[47] C.C. Slama, editor. *Manual of Photogrammetry.* American Society of Photogrammetry and Remote Sensing, Falls Church, Virginia, 4 edition, 1980.

[48] S. Soatto and P. Perona. Reducing "structure from motion": A general framework for dynamic vision part 1: Modeling. *Pattern Analysis and Machine Intelligence*, 20(9), September 1998.

[49] S. Soatto and P. Perona. Reducing "structure from motion": A general framework for dynamic vision part 1: Implementation and experimental assessment. *Pattern Analysis and Machine Intelligence*, 20(9), September 1998.

[50] S. Soatto, P. Perona, R. Fraezza, and G. Picci. Recursive motion and structure estimation with complete error characterization. In *1993 IEEE Conference on Computer Vision and Pattern Recognition*, pages 428–433, Los Alamitos, CA, June 1993. IEEE Computer Society, IEEE Computer Society Press. (New York).

[51] Bruce Stockler. The tommy awards: Millimeter's second annual "best of nab" awards. *Millimeter*, 26(6), June 1998.

[52] Richard Szeliski and Sing Bing Kang. Recovering 3d shape and motion from image streams using non-linear least squares. In *1993 IEEE Conference on Computer Vision and Pattern Recognition*, pages 752–753, Los Alamitos, CA, June 1993. IEEE Computer Society, IEEE Computer Society Press. (New York).

[53] Richard Szeliski and Heung-Yeung Shum. Creating full view panoramic image mosaics and environment maps. In *Computer Graphics Proceedings, Annual Conference Series (Proc. SIGGRAPH '97)*, pages 251–258, 1997.

[54] E. Thompson. The projective theory of relative orientation. *Photogrammetria*, 23(1):67–75, 1968.

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

[56] P. Torr, W. Fitzgibbon, and A. Zisserman. Maintaining multiple motion model hypotheses over many views to recover matching and structure. In *Sixth International Conference on Computer Vision*, pages 485–491, January 1998.

[57] M. Turk, editor. *Workshop on Perceptual User Interfaces*, San Francisco, CA, 1998.

[58] S. Ullman. *The Interpretation of Visual Motion.* MIT Press, Cambridge, MA, 1979.

[59] W.M. Wells III. Visual estimation of 3-d line segments from motion: A mobile robot vision system. *RA*, 5:820–825, 1989.

[60] Juyang Weng, Narendra Ahuja, and Thomas S. Huang. Optimal motion and structure estimation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 144–152, June 1989. (San Diego, CA.).

[61] Juyang Weng, Narendra Ahuja, and Thomas S. Huang. Optimal motion and structure estimation. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(9):864–884, September 1993.

[62] G-S. Young and Rama Chellappa. 3-d motion estimation using a sequence of noisy stereo images: Models, estimation and uniqueness. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(8):735–759, January 1990.

[63] Z. Zhang, R. Deriche, O. Faugeras, and Q.T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence Journal*, 78:87–119, October 1995.