Boston University Computer Science Technical Report No. 99-006, July 1999.

Non-Rigid Shape from Image Streams

Stan Sclaroff and Jonathan Alon
Computer Science Department
Boston University, Boston, MA 02215

Abstract

We present a framework for estimating 3D relative structure (shape) and motion given objects undergoing nonrigid deformation as observed from a fixed camera, under perspective projection. Deforming surfaces are approximated as piece-wise planar, and piece-wise rigid. Robust registration methods allow tracking of corresponding image patches from view to view and recovery of 3D shape despite occlusions, discontinuities, and varying illumination conditions. Many relatively small planar/rigid image patch trackers are scattered throughout the image; resulting estimates of structure and motion at each patch are combined over local neighborhoods via an oriented particle systems formulation. Preliminary experiments have been conducted on real image sequences of deforming objects and on synthetic sequences where ground truth is known.

1 Introduction

Estimation of 3D structure (shape) and motion from 2D image sequences has been a central problem in computer vision for many years. Many early studies focused on methods of relating pixel coordinates to 3D coordinates via camera calibration [51, 54], that is computing the projection matrix which relates image coordinates to a world coordinate frame. In recent years, the focus has shifted to non-metric reconstruction from uncalibrated cameras [25], by computing the fundamental matrix (two views) [28], and the trilinear tensor (three views) [42]. Also, different camera models were assumed; *i.e.*, orthographic [49, 53], perspective projection [27, 54], or a unified model [4, 41].

Determining the geometric relationship between various views of the environment and its 3D structure is a key component in a myriad of practical applications: reverse engineering, virtual reality, visualization, surgical planning, movie special effects, computer aided design, non-tactile inspection, manufacturing, image compression, *etc.* When 3D shape and motion estimates are computed in real time, they can be used to support applications where a computer (or robot) must interact with its environment: manipulation, navigation and control, tracking, *etc.* Furthermore, such estimates can be utilized to determine the locations, postures, and configurations of humans in order to enable a computer to assist (or avoid hampering) in a task.

Despite the many exciting applications and the energetic progress of research in structure and motion recovery algorithms, many problems remain unsolved. Some of these issues are related to numerical stability and/or ambiguity of the solution under general conditions [2, 33, 43, 45, 54, 55]. Other problems stem from the rich variety of shapes and motions that are possible in the world. In particular, many shapes can be non-planar and/or their motion can be nonrigid. Unfortunately, all of the above-mentioned approaches assume that object points in 3D space must remain at fixed distances from each other during motion.

Our goal is to extend these approaches to non-rigid objects. We propose a method for recovering 3D shape and motion estimates for objects undergoing nonrigid deformation as observed from a fixed camera, under perspective projection.[1] A natural first step to take towards solving this problem is to assume that the deforming object consists of small patches that are rigid and planar when considering small enough regions. In other words, we will employ a representation where deforming surfaces will be approximated as piece-wise planar, and piece-wise rigid.

A second assumption common to many of these approaches is that correspondence between features in different views is given. As will be outlined later, we utilize a tracker that automatically registers moving image patches from frame to frame [38]. Each corresponding warped image patch is then used directly in estimating the 3D orientation of the piece-wise planar surface patch, and its 3D position up to a scale factor. A robust image registration formulation provides stability to shadows, highlights, and partial occlusions. Furthermore, changes in illumination are modeled explicitly.

Two different approaches for acquiring piece-wise rigid/planar models are possible: top-down and bottom-up. In the top-down method, the initial hypothesis could be that an object's motion can be adequately modeled as a single moving rigid/planar patch [3]; the model would then be subdivided and augmented as needed to account for non-planar/non-rigid motion via an adaptive triangulation procedure. In the second, bottom-up approach, many relatively small planar/rigid image patch trackers could be scattered throughout the image; resulting estimates of structure and motion at each patch would then be combined over local neighborhoods via an extension of Szeliski's oriented particle systems formulation [16, 46].

In our preliminary system, we have developed the bottom-up approach, and will report these results. The botton-up framework is evaluated using synthetic data in which ground truth, deformation, and noise levels are known. The method's efficacy is also demonstrated on real image sequences of deforming objects. Implementation of the top-down approach, and experimental comparison of both strategies, is saved as future work

[1] It is assumed that self calibration of the camera will be given or obtained via a standard technique (*e.g.*, [29])

2 Background

The many years of work in structure from motion have led to significant advances in recovery of detailed, texture mapped models and motion estimates from video to support graphics, visualization, and compression. A number of researchers have demonstrated systems that can recover planar models and texture maps from image streams; *e.g.*, [4, 5, 12, 24] to name a few. Other researchers have demonstrated methods for recovering polygonal models of an object that is positioned on a rotating platform [1, 40, 44].

Other approaches focus on the problem of structure from tracked feature points (or lines) with known correspondence from two or more frames, under orthographic or perpsective projection [15, 54]. If desired, a polygonal model can be recovered from the resulting collection of unorganized 3D point position estimates via triangulation [8, 15, 20] or via surface approximation [14, 26].

In point based methods, feature tracking and correspondence is assumed. Such tracking can be attained via any number of techniques. Typically, image correlation or sum of squared differences methods are used [50]. A point feature is essentially a small image patch, which is tracked by optimizing some matching criterion with respect to translation or affine image deformation. Selection of good points to track can be based on a number of factors, including corners, texture, sufficient zero crossings in the Laplacian of image intensity, *etc.* [50]. Unfortunately, even a "good" feature can be difficult to track if it lies on a depth discontinuity, or across the boundary of a specular highlight, or if it is occluded during tracking. Such problems beg the use of smaller feature windows, since smaller windows tend to be less likely to straddle discontinuities. However, there is a tradeoff: estimates based on smaller windows tend to be more susceptible to noise and outliers, since there are fewer pixels per feature window tracked.

Another set of methods is based on image registration. Take for example, the plane plus parallax methods of [3, 11, 22, 37]. These methods exploit a dominant planar motion to compute the epipoles and perform a projective reconstruction. Such methods can use robust minimization methods [6] to overcome the influence of outliers.

All of the methods mentioned so far assume *rigid motion* in order to recover a model. This limits the utility of the above methods to recovery of rigid structure and motion estimates. In images, the deformational motion of objects is sometimes due to changes in viewing geometry. In many such cases, the above mentioned methods are sufficient. However, in general, these parameterizations are inadequate for representing motions that arise due to a general nonrigid deformation. For instance, most biological objects are flexible and articulated: fingers bend, cheeks bulge, fish swim, trees sway in the breeze, *etc.* Shapes are stretched, bent, tapered, dented, *etc.*, and so it seems logical to employ a model that can encode the ways in which real objects deform.

This rationale led to the development of 3D active shape models [48]. These models utilize a predefined structure that incorporates prior knowledge about a shape's smoothness and its resistance to deformation. A number of different 3D deformable model formulations have been proposed; *e.g.*, deformable tubes [32, 48], ellipsoidal models [9, 30], superquadrics [31, 36], *etc.* Perhaps the major limitation of such methods is the requirement that every object be described as the deformations of a single prototype object. This limits the kinds of shapes (and topologies) that can be recovered in general, since we can only recover shapes that are achievable via the specific geometric model and nonrigid motion formulation.

Some researchers attempt to overcome this limitation through the use of more general, 3D deformable part decompositions [35], local deformations [30, 31, 34], shape evolution models [13], or adaptive subdivision [21, 23, 52]. These methods offer greater generality, but are still somewhat limited in the shapes and deformations they can describe in general. Furthermore, these techniques sometimes require careful initial placement of the model, reliable feature detection for model-image correspondence, or the delicate choice of model parameters (*e.g.*, stiffness).

A second assumption common to many of the above approaches is that the correspondence between features in the different views is known. To get around this problem, we will use a tracker that automatically determines correspondence via registration of image patches from frame to frame, as described in the next section.

3 Tracking Deforming Image Patches

A key component of the proposed approach is tracking visible parts of objects from frame to frame. A promising family of approaches is based on tracking of deforming image regions [7, 17, 18, 38, 39]. These approaches integrate information over an image patch, and therefore tend to be more immune to noise and/or low-contrast, especially if a robust estimator formulation is employed [6]. Typically, use of a robust approach requires batch processing, though multiscale techniques offer some hope for realtime performance. Real-time approaches for tracking of parameterized patches have been developed [17, 18]; however, they do not address general nonrigid motion tracking.

3.1 Active Blobs Formulation

More general nonrigid motion tracking can be accomplished via the *active blobs* formulation of [38]. The formulation provides robustness to occlusions, wrinkles, shadows, and specular highlights. Furthermore, it is tailored to take advantage of texture mapping hardware available in many workstations, PC's, and game consoles. This enables nonrigid tracking at speeds approaching video rate.

In the active blobs formulation, shape of the image patch is modeled with a deformable triangular mesh. The construction of an example active blob model is shown in

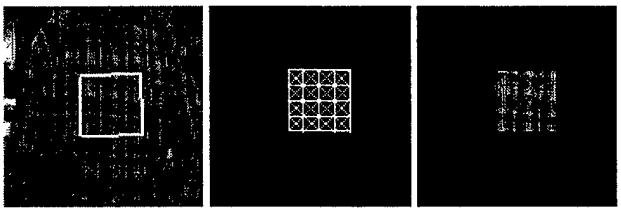

Figure 1: Construction of an example image patch model via *active blobs*. From left to right: a.) input image with region of interest overlaid, b.) triangle mesh model, c.) texture mapped model.

Fig. 1. Fig. 1(a) shows the first image in a sequence with regions of interest outlined. A 2D active triangular image patch model is then constructed for the region of interest as shown in Fig. 1(b). The blob's appearance is then captured as a color texture map and applied directly to the triangulated model as shown in Fig. 1(c).

For tracking, the active blob model is warped such that it is registered with the incoming image sequence. Warping is defined as a deformation of the triangular mesh and then a bilinear resampling of the texture mapped triangles. In essence, texture mapping is used to define a warping function for the input image, $\mathbf{I}$:

$$\mathbf{I}' = cW(\mathbf{I}, \mathbf{u}) + b = \mathcal{W}(\mathbf{I}, \mathbf{a}), \tag{1}$$

where $\mathbf{u}$ is a vector containing deformation parameters, and $\mathbf{b}$ and $\mathbf{c}$ model brightness and contrast variations. For notational convenience, we concatenate the parameters $\mathbf{u}, \mathbf{b}, \mathbf{c}$ together in a generic parameter vector $\mathbf{a}$, and define a generic warping function $\mathcal{W}$. In our current system, the photometric correction terms are defined as bilinear functions that scale the red, green, and blue channels equally.

Perhaps the simplest deformation functions to be used in Eq. 1 are those of an eight parameter projective model. Such functions are suitable for approximating the rigid motion of a planar patch. However, since the piece-wise planar/rigid assumption is likely to be violated, we utilize a parameterization that can accommodate greater variability.

A more general parameterization of nonrigid motion can be obtained via the modal representation [34], where deformation is represented in terms of eigenvectors of a finite element (FE) model. The underlying FE formulation offers the added advantage that it can be used in obtaining a regularized solution to the nonrigid tracking problem. For a given modal parameter vector obtained in tracking, we can compute the strain energy associated with deformation:

$$E_{strain} = \sum_{j=1}^{m} \tilde{u}_j^2 \omega_j^2, \tag{2}$$

where ω_j is the stiffness associated with the j^{th} modal deformation parameter. Note that these stiffnesses are determined directly from the FE shape model [34, 38].

Recall that in Eq. 1, we concatenate the deformation and lighting parameters $\mathbf{u}, \mathbf{b}, \mathbf{c}$ together in a generic parameter vector $\mathbf{a}$. Therefore, generalized stiffnesses are needed. We define a diagonal, generalized stiffness matrix Ψ that contains the modal stiffnesses ω_j and stiffnesses for the lighting parameters along the diagonal. The lighting stiffnesses are inversely proportional to the expected variance in lighting, and estimated via statistical methods [10, 38].

Tracking is then posed as a problem of regularized *active blob registration*. For each frame, the image template is warped to minimize a regularized registration function:

$$E = \frac{1}{n} \sum_{i=1}^{n} \rho(e_i, \sigma) + \gamma \mathbf{a}^t \Psi^2 \mathbf{a} \tag{3}$$

$$e_i = \|\mathbf{I}'(x_i, y_i) - \mathbf{I}(x_i, y_i)\| \tag{4}$$

where $\mathbf{I}'(x_i, y_i)$ is a pixel in the warped template (Eq. 1), $\mathbf{I}(x_i, y_i)$ is the pixel at the same location in the input, σ and γ are scale parameters, and ρ is an influence function [19].

The influence function ρ is also known as a robust error norm [6]. It is equivalent to the incorporation of an analog outlier process in our objective function. This results in better robustness to specular highlights and occlusions. In our experiments, we have used the function $\rho(e_i, \sigma) = \log(1 + e_i^2/(2\sigma^2))$ [6, 38]. For efficiency, the log function can be implemented via table look-up.

3.2 Robust Registration Algorithm

Registration requires minimization of residual error (Eq. 3) with respect to the deformation and lighting parameters. A common approach to multi-dimensional minimization problems is the Marquardt-Levenberg method. Marquardt-Levenberg requires the calculation of $O(N)$ gradient images and $O(N^2)$ image products per iteration of minimization, where N is the number of model parameters. To decrease the number of gradient calculations needed, we can use a *difference decomposition* [17, 18, 38]. The approach only requires the equivalent of $O(1)$ image gradient calculations and $O(N)$ image products per iteration.

In the difference decomposition, a set of difference images is generated by adding small changes to each of the blob parameters. Each difference image takes the form:

$$\mathbf{b}_k = \mathbf{I}_0 - \mathcal{W}(\mathbf{I}_0, \mathbf{n}_k), \tag{5}$$

where $\mathbf{I}_0$ is the template image, and $\mathbf{n}_k$ is the parameter displacement vector for the k^{th} difference image, $\mathbf{b}_k$. Each difference image becomes a column in the matrix $\mathbf{B}$. The difference matrix can be precomputed; this is the key to the difference decomposition's speed.

During tracking, an incoming image $\mathbf{I}$ is inverse warped into the blob's coordinate system using the most recent estimate of the warping parameters $\mathbf{a}$. The difference between the inverse-warped image and template is then computed:

$$\mathbf{D} = \mathbf{I}_0 - \mathcal{W}^{-1}(\mathbf{I}, \mathbf{a}). \tag{6}$$

This difference image $\mathbf{D}$ can be approximated in terms of a linear combination of the difference decomposition's vectors: $\mathbf{D} \approx \mathbf{Bq}$, where $\mathbf{q}$ is a vector of coefficients. Thus,

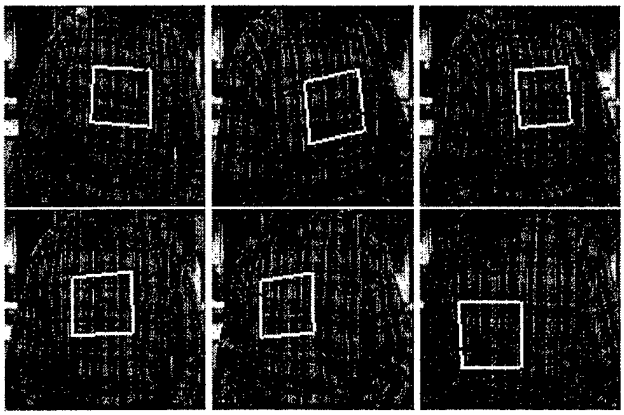

Figure 2: Tracking of a patch over a number of frames in a video sequence. The patch outlin is shown in white. The registration of the image patch from frame to frame implicitly establishes correspondence, allowing us to compute a least squares estimate of the local surface orientation and relative depth. The recovered surface normal is shown displayed over top the input sequence.

the maximum likelihood estimate of $\mathbf{q}$ can be obtained via least squares:

$$\mathbf{q} = (\mathbf{B}^t\mathbf{B})^{-1}\mathbf{B}^t\mathbf{D}. \tag{7}$$

The change in the image warping parameters is obtained via matrix multiplication

$$\Delta\mathbf{a} = \mathbf{N}\mathbf{q}, \tag{8}$$

where $\mathbf{N}$ has columns formed by the parameter displacement vectors $\mathbf{n}_k$ used in generating the difference basis.

A robust solution can be obtained through inclusion of a diagonal weighting matrix in Eq. 7:

$$\mathbf{q} = (\mathbf{B}^t\mathbf{S}^{-1}\mathbf{B})^{-1}\mathbf{B}^t\mathbf{S}^{-1}\mathbf{D}, \tag{9}$$

where entries in the diagonal matrix $\mathbf{S}$ take the form $s_{ii} = 2\sigma^2 + D_i^2$, as derived from the robust error norm ρ.

Finally, the formulation can be extended to include a regularizing term that enforces the priors on the model parameters. This is accomplished using a constrained least squares formulation:

$$\mathbf{q} = \mathbf{P}\mathbf{D} - \mathbf{Q}\mathbf{a}, \tag{10}$$

where $\mathbf{P} = \left[\mathbf{B}^t\mathbf{S}^{-1}\mathbf{B} + \gamma\mathbf{N}^t\Psi^2\mathbf{N}\right]^{-1}\mathbf{B}^t\mathbf{S}^{-1}$ and $\mathbf{Q} = \gamma\left[\mathbf{B}^t\mathbf{S}^{-1}\mathbf{B} + \gamma\mathbf{N}^t\Psi^2\mathbf{N}\right]^{-1}\mathbf{N}^t\Psi^2$. If needed, this minimization procedure can be iterated at each frame until the percentage change in the error residual is below a threshold, or the number of iterations exceeds some maximum.

An example of tracking and image patch via difference decomposition is shown in Fig. 2. Image warping and registration implicitly establishes correspondences between views; every pixel within an image patch now has a corresponding location in the next frame. Given these corresponding pixel locations, we can recover estimates of local planar structure and surface normal via least squares [54] as described in the next section.

4 Piece-Wise Planar Structure Recovery

For a given collection of corresponding image points in two views, we estimate the planar patch's relative position and orientation via an algorithm proposed by Weng, *et al.* [54] and similarly presented by Faugeras in [15]. The approach employs a linear algorithm that yields a closed form solution. The formulation is briefly restated here. We consider this as a preliminary formulation, since it is standard in the literature; however, we plan to evaluate other methods for planar structure recovery in future work. In particular, multiple frame approaches [5], constrained approaches [47], and more stable approaches [43] seem promising.

Weng, *et al.* [54] use an ideal pin hole camera model with unit focal length. A conventional camera can be calibrated so that every point in the actual image plane can be transformed to a point in the image plane of this normalized model. Consider a point on the object that is visible at two time instants. The 3D spatial position of the point in the first instant is denoted $\mathbf{x} = (x, y, z)^t$, and in the second $\mathbf{x}' = (x', y', z')^t$. The image coordinates of the point, in the first and second images are denoted $\mathbf{X} = (u, v, 1)^t = (\frac{x}{z}, \frac{y}{z}, 1)^t$ and $\mathbf{X}' = (u', v', 1)^t = (\frac{x'}{z'}, \frac{y'}{z'}, 1)^t$, where (u, v) and (u', v') are the image coordinates of the point, in the first and second images respectively. Therefore, the spatial vector and image vector are related by $\mathbf{x} = z\mathbf{X}, \mathbf{x}' = z'\mathbf{X}'$.

The basic rigid motion equation that relates spatial points at the two time instances is:

$$\mathbf{x}' = R\mathbf{x} + \mathbf{T}. \tag{11}$$

where R and $\mathbf{T}$ are a rotation matrix and translation vector respectively. It is assumed that the camera undergoes rotation around an axis going through the origin followed by a translation. It is further assumed that the world coordinate system is centered at the optical center. Note that in monocular sequences, the translation vector $\mathbf{T}$ and the depths of the object points z and z' can only be determined up to a scale factor. Therefore translation is described in terms of a unit vector $\frac{\mathbf{T}}{\|\mathbf{T}\|}$, and depth estimates are similarly normalized $\frac{z}{\|\mathbf{T}\|}$.

The plane where the points are located in 3D space can be represented

$$\mathbf{N}^t\mathbf{x} = 1. \tag{12}$$

where $\mathbf{N}$ is the plane's normal vector. The distance d between the origin and the plane is $d = ||\mathbf{N}||^{-1}$. Note that $d \neq 0$ thus excluding cases in which the plane goes through the origin. Furthermore, since we can only determine depth up to a scale factor, we can only determine the normal up to a scale factor.

From Eqs. 11 and 12 we get

$$\mathbf{x}' = (R + \mathbf{T}\mathbf{N}^t)\mathbf{x}. \tag{13}$$

We define the *intermediate parameter matrix*:

$$F = R + \mathbf{T}\mathbf{N}^t, \tag{14}$$

which can be rewritten in terms of image vectors:

$$z'\mathbf{X}' = Fz\mathbf{X}. \tag{15}$$

Applying a cross product with X' on both sides of the equation yields:

$$\mathbf{X}' \times F\mathbf{X} = 0. \tag{16}$$

This can be rewritten in terms of the product of a matrix with a vector that contains the elements of intermediate parameter matrix $\mathbf{h} = (f_{11}, f_{12}, f_{13}, f_{23}, \cdots, f_{33})^t$:

$$\begin{bmatrix} \mathbf{X^t} & 0 & -u'\mathbf{X^t} \\ 0 & \mathbf{X^t} & -v'\mathbf{X^t} \\ v'\mathbf{X^t} & -u'\mathbf{X^t} & 0 \end{bmatrix} \mathbf{h} = 0. \tag{17}$$

The third row is a linear combination of the other two and thus can be omitted.

If we stack these 2 rows n times in a matrix where n is the number of points we get a $2n \times 9$ matrix such that

$$A = \begin{bmatrix} \mathbf{X_1^t} & 0 & -u_1'\mathbf{X_1^t} \\ 0 & \mathbf{X_1^t} & -v_1'\mathbf{X_1^t} \\ \vdots & \vdots & \vdots \\ \mathbf{X_n^t} & 0 & -u_n'\mathbf{X_n^t} \\ 0 & \mathbf{X_n^t} & -v_n'\mathbf{X_n^t} \end{bmatrix} \tag{18}$$

We then solve for unit vector $\mathbf{h} = \min_h \|A\mathbf{h}\|$, subject to: $\|\mathbf{h}\| = 1$ If $\text{rank}(A) = 8$, $\mathbf{h}$ can be solved up to a scale factor. Weng et al [54] show that $\text{rank}(A) = 8$ if and only if there exists a set of four object points such that no image projections of any three points in this set are collinear in any of the two images. Then assuming $\text{rank}(A) = 8$ the solution of $\mathbf{h}$ is a unit eigenvector of A^tA associated with the smallest eigenvalue.

Since all the necessary information for F is contained in $\mathbf{h}$ we are now ready to solve for the rotation, translation, and plane normal from F. There are four cases to consider corresponding to the multiplicity of F^tF's eigenvalues. For brevity, these details are omitted. For the four cases and their geometric interpretation see [54].

5 Combining Surface Estimates

The strategy is to scatter many relatively small planar/rigid image patch trackers throughout the image. Using the procedure described above, a separate 3D position and orientation estimate is recovered for each image patch. It is possible that structure estimates will be noisy. A regularized solution can be obtained by combining the piece-wise shape estimates over local neighborhoods via an extension of Szeliski and Tonnesen's oriented particle systems formulation [16, 46]. Using this approach, complex surfaces are modeled as sets of local surface elements that interact with each other. Interaction potentials are devised that cause particles to move into locally smooth arrangements subject to external forces that are derived from the image-based piece-wise structure estimates.

Unlike the particle systems commonly used in computer graphics, our oriented particle system is massless. Instead, the formulation utilizes potentials that enforce priors on surface bending. This difference in formulation is due to the particular goal of our application: regularization of the piece-wise planar/rigid structure estimates. Following [16, 46], we define a co-normality potential ϕ_{ij}^N and co-planarity potential ϕ_{ij}^P between particles i and j:

$$\phi_{ij}^N = 1 - \mathbf{n}_i \cdot \mathbf{n}_j, \tag{19}$$

$$\phi_{ij}^P = (\mathbf{n}_i \cdot \mathbf{r}_{ij})^2 + (\mathbf{n}_i \cdot \mathbf{r}_{ij})^2, \tag{20}$$

where $\mathbf{n}_i$ and $\mathbf{n}_j$ are the unit normals for two piece-wise planar patches, and $\mathbf{r}_{ij}$ is the vector connecting the two patch centers. These two terms determine the surface's resistance to bending.

In the simulation, the potentials are combined in an internal energy term that sums the inter-particle energies:

$$E_{internal} = \sum_{i,j} (\phi_{ij}^N + \alpha\phi_{ij}^P)\beta(r_{ij}), \tag{21}$$

where α is a scale factor that controls the relative importance of the terms, and β is a monotonically decreasing function used to limit the range of the forces and torques derived from the potential energy function. For this application, the function we use is $\beta(r_{ij}) = \max(1 - ||r_{ij}||^m/d^m, 0)$, where d is the desired falloff distance, and m controls the rate of falloff.

Due to this falloff, a particle is affected by forces and torques exerted by the other particles only within its local neighborhood $\mathcal{N}_i$. Equations for the forces and torques can be found in [46]. For numerical conditioning, a damping term is added to both force and torque equations.

To gain a regularized estimate of the piece-wise surface, we run a particle simulation. We define two sets of particles in the simulation: *surface particles* and *data particles*. One surface particle and one data particle are defined for each piece-wise planar surface estimated in the image. The initial value of each surface and data particle is the position and orientation estimated via tracking as described in Sec. 4. Data particles remain fixed during the simulation, while surface particles are free to move. Each pair of data and surface particles can be joined by a linear spring.

The particle system's behavior is described by an ordinary differential equation [46], and integrated in time via Euler's method until the change in the potential energy between iterations goes beneath a threshold. The regularized piece-wise surface is taken as the position/orientation of the surface particles at the end of the simulation.

It is possible that there are depth discontinuities present in the scene, and therefore particles may lie on different sides of a depth discontinuity. The forces that bind particles should therefore be modeled as springs that break apart if particles are too far out of alignment [16].

The advantage of using the oriented particle system approach is that it requires no *a priori* knowledge of the

piece-wise surface's topology. One disadvantage is that the approach requires careful parameter setting. Furthermore, the computational complexity of simulation is prohibitive for large particle systems; each update of the system requires the calculation of $O(n^2)$ inter-particle forces. The complexity issue can be addressed through the use of spatial data structures [46].

6 "Good" Image Patches to Track

Piece-wise structure recovery depends on the registration of deforming image patches from frame to frame. In our proposed system, the strategy is to track many patches at a time. Some patches will be relatively "good" and will allow accurate tracking of deformation. Other patches may present problems in deformable region tracking, and should be detected.

For instance, some image patches may have relatively low contrast and therefore will be unfit for tracking. More generally, we need to anticipate and deal with the aperture problem in estimating patch motion. At each pixel, it is only possible to estimate that component of image velocity that is orthongal to an image isobrightness contour. One solution to this problem is to calculate motion over larger image patches. Since we are tracking relatively large image patches (on the order of 16×16 or 32×32 pixels), it is often possible to resolve the aperture problem, assuming sufficient image contrast.

However, in general, there will still be some image patches for which it is impossible to reliably estimate the motion parameters due to the aperture problem. In certain cases, parameter estimates may be ambiguous or underconstrained. This is a generalization of the aperture problem [18]. It effects not only estimates of translational motion, but estimates of deformational motion as well. It may be possible to reliably estimate only a subset of deformation parameters given an image patch of a particular texture. This ambiguity can be detected by computing the rank of the matrix $\mathbf{B}^t\mathbf{B}$ employed in image registration (Sec.3.2). If this matrix is rank deficient, then there will be an inherent ambiguity in tracking for that patch.

More generally, $\mathbf{B}^t\mathbf{B}$ serves as the estimated covariance matrix of the standard errors in the recovered registration parameters for each patch. These covariances could be incorporated directly into the structure recovery and in the oriented particle simulation. This would allow resolution of possible ambiguities by pooling over neighborhoods, and is saved as future work.

Unfortunately, even a "good" image patch can be difficult to track if it lies on a depth discontinuity, across the boundary of a specular highlight, or if it is occluded during tracking. The use of the influence function formulation in registration provides improved robustness to these effects. The particular robust error norm employed reaches its theoretical break down point when the number of outliers exceeds 50%. As suggested by [50], patches that straddle depth discontinuities can be detected by inspecting the residual error in registration at each step.

7 Preliminary Experiments

To test the capabilities of our proposed framework, we built an experimental implementation of the piece-wise planar tracking system. Our system was implemented on an SGI O2 with a 180Mhz R5K processor, 128MB RAM. At this time, only the tracking and piece-wise structure modules have been fully-tested. The particle system module has undergone preliminary testing with planar motion sequences. Full integration/evaluation of the particle system module is expected for the final version of this paper.

The basic piece-wise structure approach was tested on synthetic sequences in which ground truth was known. The experimental setup for generating synthetic sequences was as follows. A polygonal, texture mapped model was rendered under perspective projection using OpenGL at 128×128 resolution. The resulting image sequence was then used as a test sequence. For visualization purposes, the recovered normal and patch location were then displayed overlaid on the input frames. Additional orthographic views were displayed for ease of viewing.

The system was tested on approximately twenty synthetic sequences under varying amounts of rotation, scaling, translation, and deformation. Two different 3D deformation functions were used: quadratic bending, and helical twisting. Illumination was kept fixed, since previous experiments with active blobs [38] already demonstrated showed robustness of the tracker to illumination. Each image region tracked was 32×32 pixels in size.

Results for two different synthetic sequences are shown in Figs. 3 and 4. In both figures, the first frame in the input sequence is shown in (a), with the initial position of image patches shown overlaid in white. Subsequent frames in the sequence are shown in (b). Ground truth normals are shown in green. Estimated normals are shown in red. To better visualize the result, orthographic views of the surface an normals are shown below each image in the sequence (c,d).

Since the polygonal model and the deformation were known, ground truth structure and normal information was readily available. This allowed us to compute error in orientation estimates. Throughout the synthetic sequences tested, the dot product between the estimated and ground truth normals had an average value of 0.97 (15^o).

The system has also been tested on real image sequences of deformable objects in motion. Frames taken from a tracking sequence of piece of a foam rubber block deforming are shown in Fig. 5. As before, tracked regions are shown outlined in white and estimated normals are shown in red (displayed under perspective projection). As can be seen, the results look reasonable despite the large deformation and nonrigidity. We expect that the results will improve further with inclusion of the particle system module.

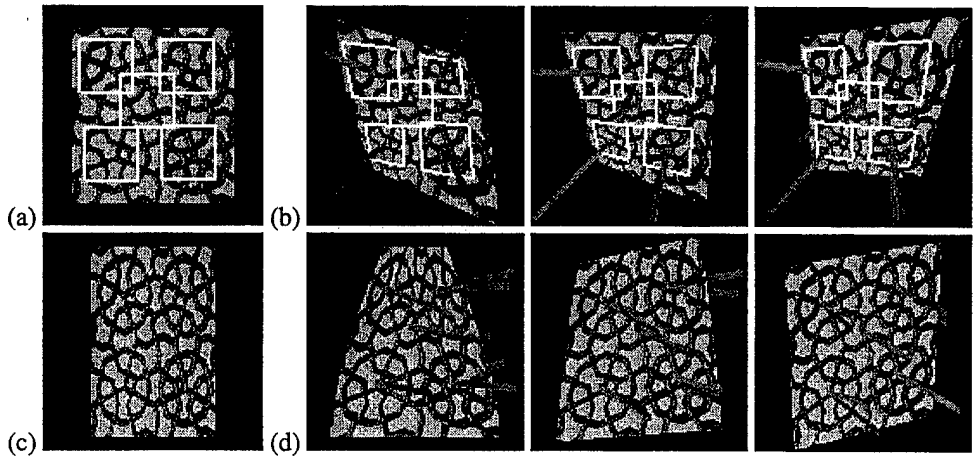

Figure 3: Example tracking with synthetic sequence: twisting. A perspective image sequence was generated for a deforming plane. The first frame in the input sequence is shown in (a), with the initial position of image patches shown overlaid in white. Frames taken from later in the input sequence are shown in (b). Ground truth normals are shown in green. Estimated normals are shown in red. To better visualize the result, corresponding orthographic side views are shown below each image in the sequence (c,d).

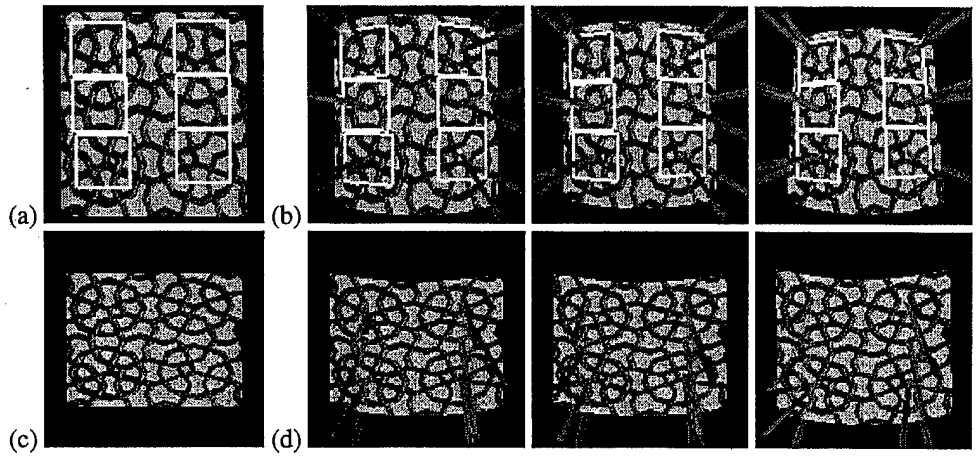

Figure 4: Second example of tracking with synthetic sequence: quadratic bending of planar sheet. A perspective image sequence was generated and piece-wise model estimates were obtained as in previous example. The first frame in the input sequence is shown in (a), with the initial position of image patches shown overlaid in white. Subsequent frames are shown in (b). As before, ground truth normals are shown in green and estimated normals are shown in red. Corresponding orthographic top views are shown below each image(c,d).

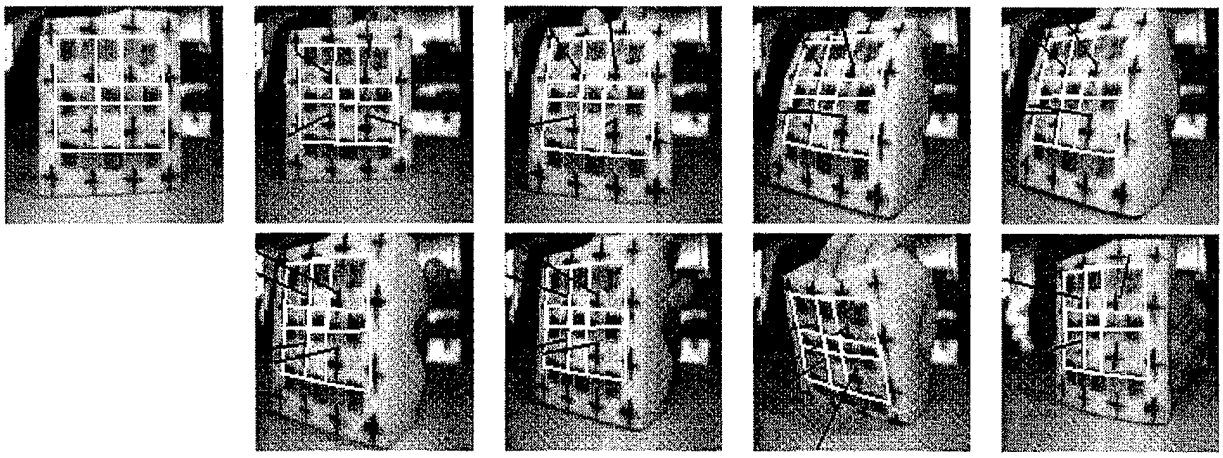

Figure 5: Example tracking with a real image sequence: a foam rubber block deforming. As before, tracked regions are shown outlined in white and estimated normals are shown in red (displayed under perspective projection).

References

[1] G. Cross A. Fitzgibbon and A. Zisserman. Automatic 3D model construction for turn-table sequences. In *SMILE Workshop*, 1998.

[2] G. Adiv. Inherent ambiguities in recovering 3-D motion and structure from a noisy flow field. *PAMI*, 11(5), 1989.

[3] P. Anandan, K. Hanna, and R. Kumar. Shape recovery from multiple views: A parallax based approach. In *ICPR*, 1994.

[4] A. Azarbayejani and A.P. Pentland. Recursive estimation of motion, structure, and focal length. *PAMI*, 17(6), 1995.

[5] P.A. Beardsley, P.H.S. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In *ECCV*, 1996.

[6] M.J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *IJCV*, 19(1), 1996.

[7] M.J. Black, Y. Yacoob, A.D. Jepson, and D.J. Fleet. Learning parameterized models of image motion. In *CVPR*, 1997.

[8] J.D. Boissonnat. Representing 2D and 3D shapes with the delaunay triangulation. In *ICPR*, 1984.

[9] L.D. Cohen and I. Cohen. Finite-element methods for active contour models and balloons for 2D and 3D images. *PAMI*, 15(11), 1993.

[10] T. Cootes. Combining point distribution models with shape models based on finite element analysis. In *BMVC*, 1994.

[11] A. Criminisi, I. Reid, and A. Zisserman. Duality, rigidity and planar parallax. In *ECCV*, 1998.

[12] P.E. Debevec, C.J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *SIGGRAPH*, 1996.

[13] D. DeCarlo and D. Metaxas. Shape evolution with structural and topological changes using blending. *PAMI*, 20(11), 1998.

[14] M. Eck and H. Hoppe. Automatic reconstruction of b-spline surfaces of arbitrary topological type. In *SIGGRAPH*, 1996.

[15] O. Faugeras. *Three-Dimensional Computer Vision*. MIT Press, 1993.

[16] P. Fua. From multiple stereo views to multiple 3D surfaces. *IJCV*, 24(1), 1997.

[17] M. Gleicher. Projective registration with difference decomposition. In *CVPR*, 1997.

[18] G.D. Hager and P.N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *PAMI*, 20(10), 1998.

[19] F. Hampel, E. Ronchetti, P. Rousseeuw, and W. Stehel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley and Sons, New York, 1986.

[20] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. In *SIGGRAPH*, 1992.

[21] W.C. Huang and D.B. Goldgof. Adaptive-size meshes for rigid and nonrigid shape analysis and synthesis. *PAMI*, 15(6), 1993.

[22] M. Irani, B. Rousso, and S. Peleg. Recovery of ego-motion using region alignment. *PAMI*, 19(3), 1997.

[23] I. Kakadiaris, D. Metaxas, and R. Bajcsy. Active part-decomposition, shape and motion estimation of articulated objects: A physics-based approach. In *CVPR*, 1994.

[24] T. Kanade, P.W. Rander, and P.J. Narayanan. Virtualized reality - constructing virtual worlds from real scenes. *IEEE MultiMedia*, 4(1), 1997.

[25] J.J. Koenderink and A.J. vanDoorn. Affine structure from motion. *JOSA-A*, 8(2), 1991.

[26] C.W. Liao and G. Medioni. Surface approximation of a cloud of 3D points. *GMIP*, 57(1), 1995.

[27] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293, 1981.

[28] Q.T. Luong, R. Deriche, O.D. Faugeras, and T. Papadopoulo. On determining the fundamental matrix: Analysis of different methods and experimental results. Technical report, INRIA, 1993.

[29] Q.T. Luong and O.D. Faugeras. Self-calibration of a moving camera from point correspondences and fundamental matrices. *IJCV*, 22(3), 1997.

[30] T. McInerney and D. Terzopoulos. A finite element model for 3D shape reconstruction and nonrigid motion tracking. In *ICCV*, 1993.

[31] D. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *PAMI*, 15(6), 1993.

[32] C. Nastar and N. Ayache. Frequency-based nonrigid motion analysis: Application to 4D medical images. *PAMI*, 18(11), 1996.

[33] J. Oliensis. A critique of structure from motion algorithms. Technical report, NEC Research Institute, 1997.

[34] A. Pentland and S. Sclaroff. Closed-form solutions for physically-based shape modeling and recognition. *PAMI*, 13(7), 1991.

[35] A.P. Pentland. Automatic extraction of deformable part models. *IJCV*, 4(2), 1990.

[36] A.P. Pentland and B. Horowitz. Recovery of nonridid motion and structure. *PAMI*, 13(7), 1991.

[37] H.S. Sawhney. 3D geometry from planar parallax. In *CVPR*, 1994.

[38] S. Sclaroff and J. Isidoro. Active blobs. In *ICCV*, 1998.

[39] S. Sclaroff and A. Pentland. Physically-based combinations of views: Representing rigid and nonrigid motion. In *IEEE Workshop on Nonrigid and Articulate Motion*, 1994.

[40] W.B. Seales and O.D. Faugeras. Building 3D object models from image sequences. *CVIU*, 61(3), 1995.

[41] A. Shashua and N. Navab. Relative affine structure: Canonical model for 3D from 2D geometry and applications. *PAMI*, 18(9), 1996.

[42] A. Shashua and M. Werman. Trilinearity of three perspective views and its associated tensor. In *ICCV*, 1995.

[43] S. Soatto and R. Brockett. Optimal structure from motion: Local ambiguities and global estimates. In *CVPR*, 1998.

[44] S. Sullivan and J. Ponce. Automatic model construction and pose estimation from photographs using triangular splines. *PAMI*, 20(10), 1998.

[45] R. Szeliski and S. B. Kang. Shape ambiguities in structure from motion. *PAMI*, 19(5), 1997.

[46] R. Szeliski and D. Tonnesen. Surface modeling with oriented particle systems. In *SIGGRAPH*, 1992.

[47] R. Szeliski and P. H. S. Torr. Geometrically constrained structure from motion: Points on planes, MSR-TR-98-64. Technical report, Microsoft Research, 1998.

[48] D. Terzopoulos, A.P. Witkin, and M. Kass. Constraints on deformable models: Recovering 3D shape and nonrigid motion. *AI*, 36(1), 1988.

[49] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *IJCV*, 9(2), 1992.

[50] C. Tomasi and J. Shi. Good features to track. In *CVPR*, 1994.

[51] R.Y. Tsai and T.S. Huang. Uniqueness and estimation of 3D motion parameters of rigid objects with curved surfaces. *PAMI*, 6(1), 1984.

[52] L.V. Tsap, D.B. Goldgof, S. Sarkar, and W.C. Huang. Efficient nonlinear finite element modeling of nonrigid objects via optimization of mesh models. *CVIU*, 69(3), 1998.

[53] S. Ullman. *The Interpretation of Visual Motion*. MIT Press, 1979.

[54] J. Weng, T.S. Huang, and N. Ahuja. *Motion and Structure from Image Sequences*. Springer-Verlag, Berlin Heidelberg, 1993.

[55] G. Young and R. Chellapa. Statistical analysis of inherent ambiguities in recovering 3-D motion from a noisy low field. *PAMI*, 14(10), 1992.

REPORT DOCUMENTATION PAGE

OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 1999	

4. TITLE AND SUBTITLE
Non-Rigid Shape from Image Streams

5. FUNDING NUMBERS
G N00014-96-1-0661

6. AUTHOR(S)
Stan Sclaroff and Jonathan Alon

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Computer Science Department
Boston University
111 Cummington Street
Boston, MA 02215

8. PERFORMING ORGANIZATION REPORT NUMBER
sclaroff-ONR-TR99-006

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Department of the Navy
Office of Naval Research
Ballston Centre Tower One
800 North Quincy Street
Arlington, VA 22217-5660

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

We present a framework for estimating 3D relative structure (shape) and motion given objects undergoing nonrigid deformation as observed from a fixed camera, under perspective projection. Deforming surfaces are approximated as piece-wise planar, and piece-wise rigid. Robust registration methods allow tracking of corresponding image patches from view to view and recovery of 3D shape despite occlusions, discontinuities, and varying illumination conditions. Many relatively small planar/rigid image patch trackers are scattered throughout the image; resulting estimates of structure and motion at each patch are combined over local neighborhoods via an oriented particle systems formulation. Preliminary experiments have been conducted on real image sequences of deforming objects and on synthetic sequences where ground truth is known.

14. SUBJECT TERMS
Non-rigid shape description and recognition; model recovery from video.

15. NUMBER OF PAGES
8

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
unclassified	unclassified	unclassified	UL