Shape from Rotation

Richard Szeliski
Digital Equipment Corporation, Cambridge Research Lab
One Kendall Square, Bldg. 700, Cambridge, MA 02139

Abstract

This paper examines the construction of a 3-D surface model of an object rotating in front of a camera. Previous research in depth from motion has demonstrated the power of using an incremental approach to depth estimation. In this paper, we extend this approach to more general motion and use a full 3-D surface model instead of a $2\frac{1}{2}$-D depth map. The algorithm starts with a flow field computed using local correlation. It then projects individual measurements into 3-D points with associated uncertainties. Nearby points from successive frames are merged to improve the position estimates. These points are then used to construct a deformable surface model, which is itself refined over time. We demonstrate the application of our new techniques to several real image sequences.

1 Introduction

This paper examines the construction of a 3-D surface model from image sequences of an object rotating in front of a stationary camera. Because the motion of the object between frames is known, we can use traditional depth from motion techniques to directly recover the depth of points in the image. Our approach uses a large number of images with a small amount of motion between successive images. This makes it easier to compute flow, but makes individual flow measurements much less reliable. To compensate for this, we use an incremental estimation algorithm to integrate measurements from successive frames and reduce the uncertainty over time.

The incremental approach to depth estimation was previously developed by Matthies *et al.* [10]. In this paper, we extend their work to true 3-D surface models. A simpler method for creating such models from the same image sequence is to use the object silhouettes to "carve out" a bounding volume for the model [15]. However, to obtain a more detailed description, we need to use the optic flow of the texture marks to give us a dense estimate of surface shape. Our new *shape from rotation* algorithm builds such a model, and also provides us with a framework within which we can explore a number of important issues in computer vision. These include flow estimation, uncertainty modeling, incremental estimation, 3-D surface representation and reconstruction, and massively parallel algorithms.

Our algorithm for the automatic acquisition of 3-D object models can be used in a number of applications. These include robotics manipulation, where the object must first be described and/or recognized before it can be manipulated; Computer Aided Design (CAD), where automatic model building can be used as an input stage to the CAD system; and computer graphics animation or *virtual reality*, where it facilitates the task of an animator, allowing him easy access to a large catalog of real-world objects. All of these applications become much more interesting if the acquisition can be performed quickly and without the need for special equipment or environments. Our aim is to build such a system, using the motion of the turntable and object to provide most of the system calibration automatically.

1.1 Previous work

Some of the early work in object motion estimation identified Kalman filtering as a useful framework for incremental estimation since it incorporates representations of uncertainty and provides a mechanism for incrementally reducing uncertainty over time. Applied to depth from motion, this framework was at first restricted to estimating the positions of a sparse set of trackable features such as points or line segments [7]. Another line of work addressed the problem of extracting denser depth or displacement estimates from image sequences. However, these approaches either were restricted to two frame analysis [9, 2] or used batch processing of the image sequence, either through line fitting [5, 3] or spatio-temporal filtering [1]. The work of [10] overcame these limitations by combining a recursive estimation procedure with dense flow measurement to produce a $2\frac{1}{2}$-D depth map. In this paper, we extend this work to use a full 3-D shape model.

1.2 Framework

Our shape from rotation algorithm operates in the following stages. First, the 2-D optical flow between successive image pairs is extracted over the whole image (Section 2). The correlation surface corresponding to the Sum of Squared Differences (SSD) measure is used to compute both the best flow estimate at each point and its 2-D uncertainty. Next, using the known object motion, we project this flow into a 3-D position measurement with an associated 3×3 uncertainty at each point (Section 3). This "cloud" of intensity-tagged depth val-

CH2983-5/91/0000/0625/$01.00 © 1991 IEEE

ues is then refined by merging nearby points from successive frames whose uncertainties overlap sufficiently (Section 4). A locally parametrized surface is then fitted to this collection of points (Section 5) to reduce noise in nearby measurements and to fill in the data where it is unreliable. In Section 6 we present some experiments with real image sequences acquired in our lab. In Section 7 we compare our approach with alternative shape acquisition techniques, and we suggest a number of extensions to our work.

2 Optical flow

Given two or more images, we can compute a two-dimensional vector field called the *optic flow* which measures the interframe motion of each pixel in the image. A number of different algorithms have been developed previously for extracting the optic flow. In this paper, we use a variant of correlation called the *Sum of Squared Differences* (SSD) measure [2], since it provides us not only with flow estimates but also with uncertainty estimates for each measurement. Alternative approaches to computing optic flow include gradient-based techniques [9, 11] and spatio-temporal filtering [1] (see [11, 2] for a comparison of several of these techniques).

The Sum of Squared Differences method integrates the squared intensity difference between two shifted images over a small area to obtain an error measure $e(u, v; x, y)$ (see [2, 16] for details). The SSD flow estimator selects at each pixel (x, y) the flow $(\tilde{u}, \tilde{v})$ which minimizes this SSD measure. It also uses the shape (steepness) of the error surface to determine the confidence in this estimate [2, 14, 16]. The shape of the error surface can also be used to estimate regions of the image when the flow estimates are suspect (e.g., because of occlusion [2]) or where no motion is present [16].

3 Constrained flow and depth recovery

The general 2-D flow estimator described in the previous section is a useful first step in determining shape from motion when the object motion (*egomotion*) is unknown. In our work, however, we know the angular position of the turntable in each frame, and therefore the relative 3-D motion of the object (or equivalently, of the camera). This makes the problem of depth recovery much easier. Using the known motion, we can compute for each pixel a constraint line for the flow at that point, with the actual (ideal) flow observed depending only on the depth of the surface at that pixel. Alternatively, since we know the motion between the two frames, we could use the standard epipolar geometry to find the set of corresponding epipolar lines in the two images [5].

The flow extraction algorithm we use first extracts corresponding rows from the two images, and then interpolates each row by a factor of 4. For each candidate (fractional) horizontal displacement, a discrete approximation to the SSD measure is computed at each pixel [16]. A parabola is then fit to the minimum SSD value and its two neighbors and is used to compute the sub-pixel flow estimate $\tilde{u}$ and its variance $\sigma_{\tilde{u}}^2 = 2\sigma_n^2/a$, where σ_n^2 is the variance of the image noise and a is the second derivative of the parabola [16].

For each flow estimate $\tilde{u}$ we compute the corresponding 3-D object space location $\mathbf{p}$ using the inverse perspective projection [16]. For each 3-D point, we also compute a 3×3 covariance matrix $\mathbf{C_p}$ which characterizes the shape and magnitude of the point's positional uncertainty. The component of this covariance along the viewing ray can be approximately computed using the flow variance and the Hessian of the inverse projection operator. The other two axes of the covariance ellipsoid can be chosen arbitrarily and their length (standard deviation) set to a suitably chosen constant value σ_0.

4 Incremental estimation (points)

The result of our two-frame optic flow analysis and backprojection into object space gives us a "cloud" of uncertainty-tagged points lying on the surface of the object. As the object continues to rotate and more points are acquired, point collections from successive frames must be merged in order to reduce the noise in point location estimates. To represent the 3-D position of the points, we use an *object-centered* coordinate reference frame whose origin is fixed to the top of the turntable and rotates with it.

To merge neighboring 3-D points from different frames, we start by computing an uncertainty-weighted distance measure

$$d_{ij} = (\mathbf{p}_i - \mathbf{p}_j)^T (\mathbf{C}_i^{-1} + \mathbf{C}_j^{-1})(\mathbf{p}_i - \mathbf{p}_j). \quad (1)$$

If this distance is sufficiently small, we can merge the two points and replace them with a single measurement

$$\mathbf{p}_k = \mathbf{C}_k(\mathbf{C}_i^{-1}\mathbf{p}_i + \mathbf{C}_j^{-1}\mathbf{p}_j) \quad (2)$$

with a reduced uncertainty

$$\mathbf{C}_k = (\mathbf{C}_i^{-1} + \mathbf{C}_j^{-1})^{-1}. \quad (3)$$

The problem with this simple approach is that there may be many candidate matches for a given point, especially if one elongated uncertainty ellipsoid overlaps several other points. To reduce this problem, we limit merges to points whose uncertainty ellipsoid major axes are nearly parallel and which also meet the previous distance criteria. In practice, we make the merging step even simpler by re-projecting the 3-D locations and their uncertainties into the camera image plane. Two points are merged if their image plane centers lie within a small distance of each other (say, $1/2$ pixel) and their depths overlap sufficiently (using a 1-D version of the uncertainty-weighted distance). The thresholds for merging points are set high enough so that neighboring measurements from the same frame are not merged (we want our final model to be at least as accurate as the input image) but low enough so that oversampling (the density of 3-D points per image pixel) is not too great.

5 Local surface fitting

Once the 3-D point estimates acquired from multiple frames have been integrated sufficiently to make them reliable, we can start building a 3-D surface model. This model serves both to reduce the noise in the position estimates (through smoothing) and to fill-in areas on the object surface where no reliable flow information is available.

Generating a parametric surface from a sparse and scattered collection of points is in general quite difficult. To solve this problem, we have developed a new 3-D surface interpolation model based on interacting *oriented particles* [17]. These particles, which represent local surface patches, have energy functions which favor the alignment of normals of neighboring particles, thus endowing the surface with an elastic resistance to bending. The particles also have a preferred inter-particle spacing distance, which encourages a uniform sampling density over the surface.

Once a reasonably accurate surface model has been constructed, we can dispense with the optic flow computation altogether. As each new image arrives, it directly modifies the deformable surface model and its associated intensities by making small local changes which better register the model and the image.

6 Experimental results

We have performed a number of experiments with our shape from rotation algorithms on both live and off-line image sequences. The experimental setup consists of a spring-wound microwave turntable with a position encoding grid taped to its side (Figures 1–3a) and a stationary camera mounted on a tripod. A rough calibration of the intrinsic and extrinsic camera parameters can be obtained by locating the ellipse that defines the turntable top and measuring the camera to turntable distance. A more exact calibration can be obtained using multiple images of a calibration cube [15].

The live experiments involve building an octree bounding volume of the object, processing a 512×480 monochrome image every 3.4 seconds on a RISC-based workstation [15]. The algorithm is first adapted to the empty turntable while it is spinning, both to memorize the background, and to locate the position encoding ring. After the object is placed on the table, each new image is then thresholded and the turntable angle computed from the binary codes averaged over 32 columns (accurate to about $0.1°$). The bounding volume is then computed from the object silhouettes [15].

For the off-line experiments, we first recorded onto videotape a number of image sequences of different objects spinning on the turntable (Figures 1–3a). We then digitized each sequence using the single-frame playback capabilities of our video recorder to obtain a high resolution image sequence of about 500 frames (about $0.72°$ rotation between frames). For the experiments presented in this paper, each image was subsampled from 512×480 to 256×240 with only every second frame being used. The resulting interframe rotation is about $1.44°$, with a maximum horizontal flow (on the turntable edge) of about 2.9 pixels.

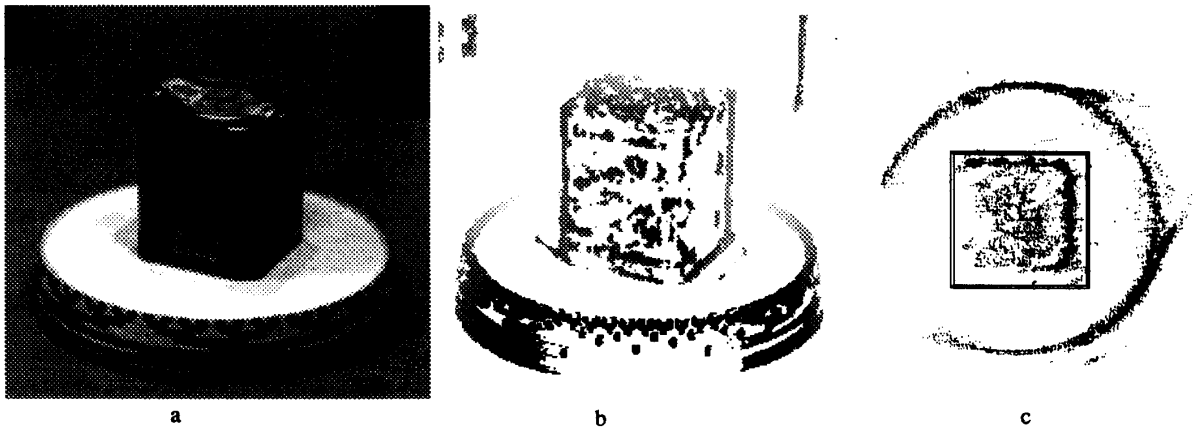
These image sequences were input into our optic flow extraction algorithm, whose output was then backprojected into 3-D world coordinates. Figures 1, 2, and 3 show three of the image sequences we are using and the results of these initial depth extraction stages. The first image (a) in each figure shows the first frame of the input intensity image sequence. The second image (b) shows an intensity-coded depth map extracted from the first pair of images (depth values with high uncertainty are not shown). The 3-D position estimates computed by backprojecting these depth values are shown in the third part (c) of each figure, using a top view of the object to better see its structure (the wireframe cube and axes are for reference only). Both the circular structure of the turntable edge, and the rectangular structure of the tea box (Figure 1) and the domino cube (Figure 3) are roughly recovered.

The next step in the shape from rotation algorithm consists of merging neighboring 3-D points acquired from different viewpoints. Figures 4, 5, and 6 show the results of this merging step, operating incrementally on the complete 250 image sequences. We present this data as isolated points shown in 4 different projections: top, front, side, and oblique. From these figures, we can see that the overall shape of the objects is recovered well, although the exact surface data is not very smooth. Adding a small amount of image-plane smoothing should help to reduce this effect [10]. Of course, once a complete surface model is fit to this sparse data, the resulting solution will also be smooth. Figures 5b–d show that in some cases, shadows will be incorporated into the object model. To remove these shadow points, we could either cut off the bottom of the model, or use a more sophisticated color-based image preprocessing stage.

7 Discussion

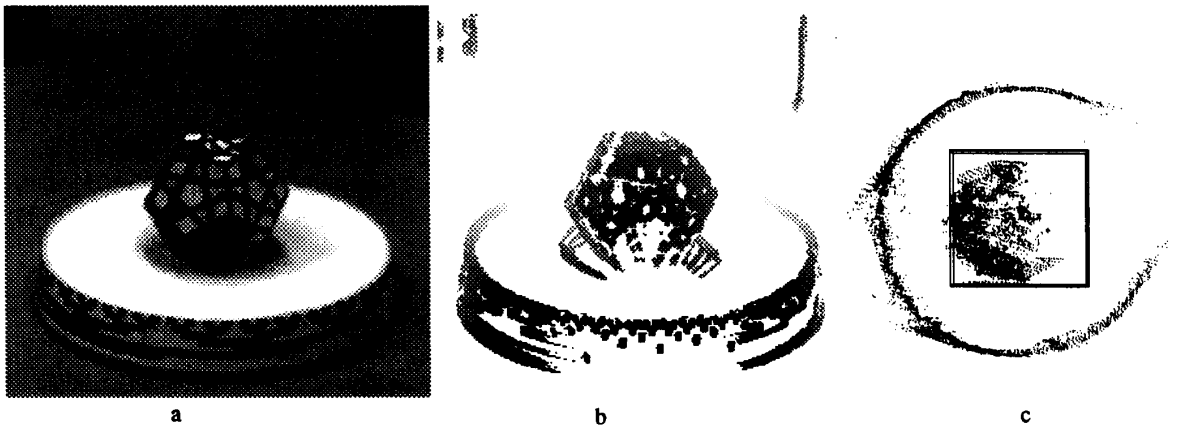
The techniques we have described in this paper perform a shape construction task similar to that usually associated with active range sensors [4]. An example of such a sensor is structured light, where an encoded light pattern falling on the object is used to give direct (and usually sparse) measurements of depth. Compared to active range sensors, our approach requires a far less structured environment, since no special lighting sources are required, and the calibration of the system is simple and fairly automatic. Our technique also has the potential for better accuracy since our measurements are dense (at least in textured areas), and because we see more views of the object. On the other hand, our flow-based approach will fail in areas where the surface has a uniform albedo. An experimental comparison of these two techniques needs to be performed to better quantify these effects.

An alternative to the approach presented in this paper is to use the silhouette of the object in each frame to construct



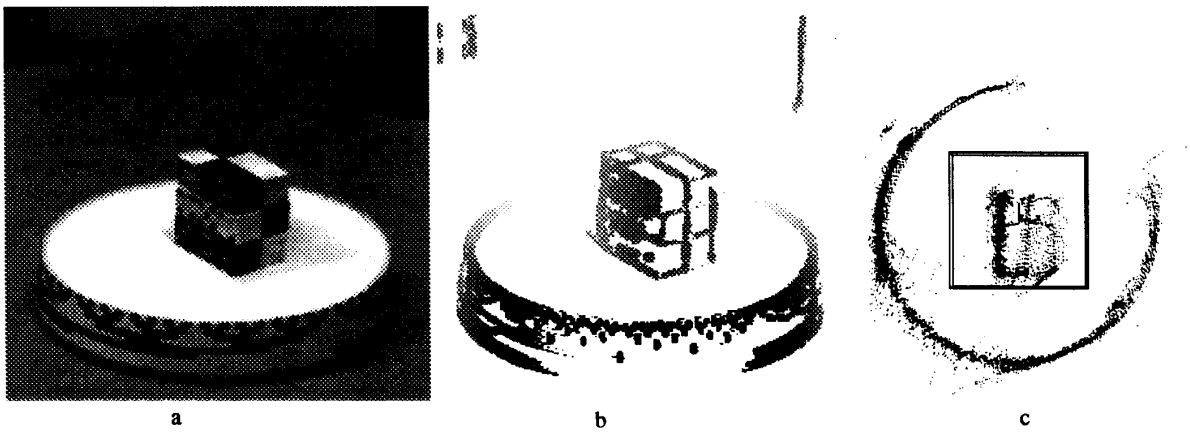
a b c

Figure 1: `assam` image sequence: (a) first image (b) depth map from flow (darker is nearer) (c) top view of 3-D point cloud



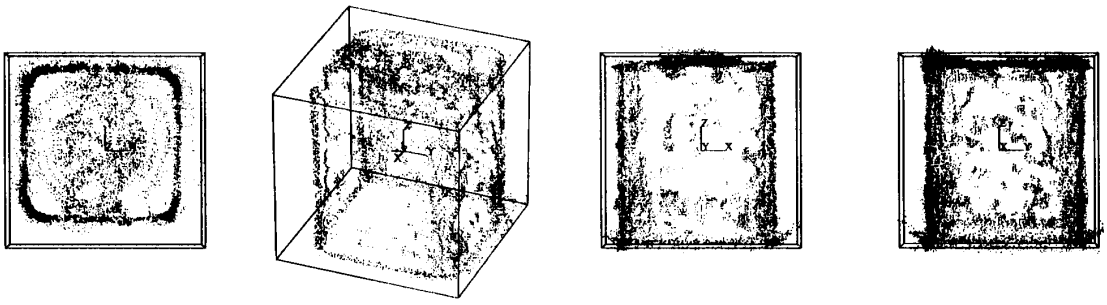
a b c

Figure 2: `dodecahedron` image sequence: (a) first image (b) depth map from flow (darker is nearer) (c) top view of 3-D point cloud



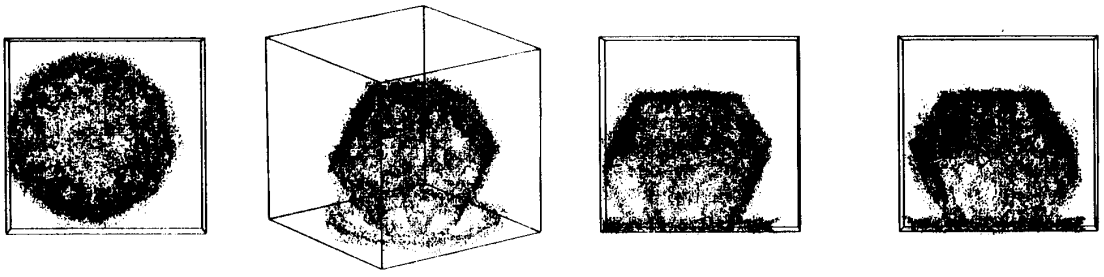
a b c

Figure 3: `domino` image sequence: (a) first image (b) depth map from flow (darker is nearer) (c) top view of 3-D point cloud



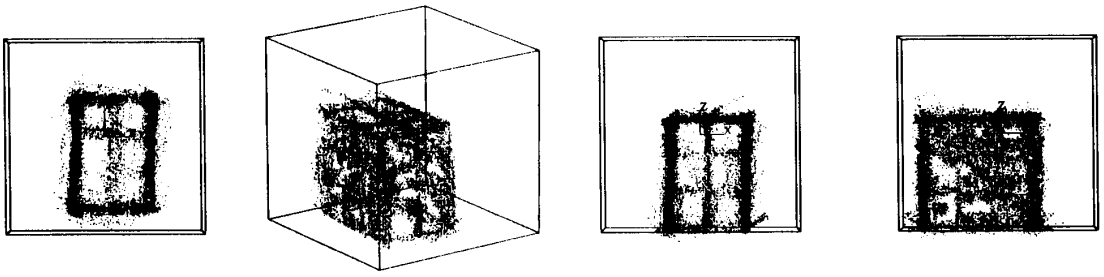
a b c d

Figure 4: Final merged data from `assam` image sequence: (a) top view (b) oblique view (c) front view (d) side view



a b c d

Figure 5: Final merged data from `dodecahedron` image sequence: (a) top view (b) oblique view (c) front view (d) side view



a b c d

Figure 6: Final merged data from `domino` image sequence: (a) top view (b) oblique view (c) front view (d) side view

a bounding volume for the object [15]. This bounding volume can provide a non-linear (inequality) constraint on the position of surface points. Tracking the silhouettes through three or more images can also be used to estimate the location and curvature of points on the surface of the object [8, 18, 6]. Combining silhouette-based and flow-based approaches should yield an algorithm that works for a much wider variety of object shapes and textures.

Our shape from rotation algorithm would be even more useful if we could change the position of the camera and/or the object. The former case is easier to handle: we simply re-calibrate the system and continue processing with the new camera parameters. Determining the change in object pose from the surface data itself is more difficult [13].

The algorithm described in this paper builds a detailed locally parameterized surface model of the object. The next step in processing would be to build a higher-level description of the object, either for more efficient CAD/graphics manipulation, or for object recognition. An example of such a model would be a superquadrics parts model, which could be fitted directly to our sparse collection of 3-D points [12].

8 Conclusions

Shape from rotation is a practical approach to building 3-D models from a sequence of images. The goal of this work is to produce a locally accurate model of shape and intensity of an unknown object As such, this technique should be useful in a variety of robotics and CAD tasks, as well as providing a novel source of objects for computer animation systems.

The design of our algorithm was motivated by the recent success of incremental algorithms in building high-quality depth maps from motion sequences. This work can be viewed as an extension of this work to full 3-D shape models.

The design of a complete shape from rotation system requires the solution of a number of fundamental computer vision problems. These include flow estimation, uncertainty modeling, incremental estimation, and 3-D surface representation and reconstruction. We have implemented and tested the main stages of processing (flow constraints, flow estimation, backprojection into 3-D, and 3-D point merging), but much interesting work remains to be done (surface reconstruction and refinement, evaluation, and enhancements). We expect that shape from rotation will prove to be an interesting and challenging problem to solve, as well as a good framework for studying various important computer vision algorithms.

References

[1] E. H. Adelson and J. R. Bergen. Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America*, A 2(2):284–299, February 1985.

[2] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2(3):283–310, January 1989.

[3] H. H. Baker and R. C. Bolles. Generalizing epipolar-plane image analysis on the spatiotemporal surface. *International Journal of Computer Vision*, 3(1):33–49, 1989.

[4] P. J. Besl and R. C.. Jain. Three-dimensional object recognition. *Computing Surveys*, 17(1):75–145, March 1985.

[5] R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1:7–55, 1987.

[6] R. Cipolla and A. Blake. The dynamic analysis of apparent contours. In *Third International Conference on Computer Vision (ICCV'90)*, pages 616–623, Osaka, Japan, December 1990. IEEE Computer Society Press.

[7] O. D. Faugeras, N. Ayache, and B. Faverjon. Building visual maps by combining noisy stereo measurements. In *IEEE International Conference on Robotics and Automation*, pages 1433–1438, San Francisco, California, April 1986. IEEE Computer Society Press.

[8] P. Giblin and R. Weiss. Reconstruction of surfaces from profiles. In *First International Conference on Computer Vision (ICCV'87)*, pages 136–144, London, England, June 1987. IEEE Computer Society Press.

[9] B. K. P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[10] L. H. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3:209–236, 1989.

[11] H.-H. Nagel. On the estimation of optical flow: Relations between different approaches and some new results. *Artificial Intelligence*, 33:299–324, 1987.

[12] A. P. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28(3):293–331, May 1986.

[13] R. Szeliski. Estimating motion from sparse range data without correspondence. In *Second International Conference on Computer Vision (ICCV'88)*, pages 207–216, Tampa, Florida, December 1988. IEEE Computer Society Press.

[14] R. Szeliski. *Bayesian Modeling of Uncertainty in Low-Level Vision*. Kluwer Academic Publishers, Boston, Massachusetts, 1989.

[15] R. Szeliski. Real-time octree generation from rotating objects. Technical Report 90/12, Digital Equipment Corporation, Cambridge Research Lab, December 1990. For ordering information, please send a message to techreports@crl.dec.com with the word `help` in the Subject line.

[16] R. Szeliski. Shape from rotation. Technical Report 90/13, Digital Equipment Corporation, Cambridge Research Lab, December 1990. For ordering information, please send a message to techreports@crl.dec.com with the word `help` in the Subject line.

[17] R. Szeliski and D. Tonnesen. Particle systems for surface interpolation. *Computer Graphics (SIGGRAPH'91)*, (submitted) 1991.

[18] R. Vaillant. Using occluding contours for 3D object modeling. In *First European Conference on Computer Vision (ECCV'90)*, pages 454–464, Antibes, France, April 23–27 1990. Springer-Verlag.