SURFACE RECONSTRUCTION USING 3–D MESHES AND PARTICLE SYSTEMS

P. Fua (fua@ai.sri.com) [1]
SRI International
333 Ravenswood Avenue, Menlo Park, CA 94025, USA

Abstract

We present a unified framework for 3–D shape reconstruction that allows us to combine multiple information sources—such as stereo and shape-from-shading—derived from images whose vantage points may be very different. A formal integration framework is critical because, in recovering complicated surfaces, the information from a single source is often insufficient to provide a unique answer.

We describe two complementary implementations of this paradigm that both rely on deforming a generic object-centered 3–D representation of the surface so as to minimize an objective function. The first implementation models the surface as a triangulated mesh. The second models it as a set of particles that interact with each other through forces that tend to align them.

Using several complex scenes, we demonstrate our approach and its ability to merge information and thus to go beyond what can be done with conventional techniques.

1 Introduction

The recovering of surface shape from image cues, the so-called "shape from X" problem, has received tremendous attention in the computer vision community. But no single source of information "X," be it stereo, shading, texture, geometric constraints or any other, has proved to be sufficient across a reasonable sampling of images. To get good reconstructions of a surface, it is necessary to use as many different kinds of cues with as many views of the surface as possible. In this paper, we present and demonstrate a working framework for surface reconstruction that combines image cues, such as stereo and shape-from-shading, obtained from multiple images whose vantage points may be very different.

To achieve our goal, we need a surface representation that can be used to generate images of the surface from arbitrary viewpoints, taking into account self-occlusion, self-shadowing, and other viewpoint-dependent effects. Clearly, a single image-centered representation is inadequate for this purpose. Instead, object-centered surface representations are required. Here, we advocate the use of two such representations that have proved effective under different circumstances:

- **Triangulated Meshes** A regular 3–D triangulation is an example of a surface representation that meets the criteria stated above. Our approach to recovering surface

[1] Support for this research was provided by contracts from the Advanced Research Projects Agency.

shape and reflectance properties from multiple images is to deform a 3–D representation of the surface so as to minimize an objective function. The free variables of this objective function are the coordinates of the vertices of the triangulation, and the process is started with an initial surface estimate.

We have successfully used these meshes to model surfaces whose topology is known *a priori*, such as human faces or relatively low-resolution terrain.

- **Sets of 3–D Particles** Real-world scenes often contain several objects whose topology may not be known in advance: some surfaces are best modeled as sheets, while others are topological spheres or contain holes. One cannot typically assume that there is only one object and one surface of interest.

  To deal with these complex issues, we replace the triangulations by a set of connected surface patches or "oriented particles" as defined by Szeliski and Tonnesen [1992]. These particles are instantiated by fitting local surfaces to traditional stereo data, and their positions are refined by minimizing an objective function analogous to the one used for triangulations.

We view the contribution of this paper as providing both the framework that allows us to combine diverse sources of information in a unified and computationally effective manner, and the specific details of how these diverse sources of information are derived from the images.

2 Using Meshes to Model Single Surfaces

We represent a surface $\mathcal{S}$ by a hexagonally connected set of vertices called a *mesh*. Such a mesh is shown in Figure 1(a). The position of a vertex v_j is specified by its Cartesian coordinates (x_j, y_j, z_j).

For each input image, we generate a "Facet-ID" image by encoding the index i of each facet f_i as a unique color, and projecting the surface into the image plane, using a standard hidden-surface algorithm. We use it to determine which surface points are occluded in a given view.

2.1 Objective Function and Optimization Procedure

The objective function $\mathcal{E}(\mathcal{S})$ that we use to recover the surface is a sum of terms that take into account the image-based constraints—stereo and shape-from-shading—that are brought to bear on the surface. To minimize $\mathcal{E}(\mathcal{S})$, we use an optimization method that is inspired by the heuristic technique known as a continuation method [Terzopoulos, 1986] in which we add a regularization term to the objective function and progressively reduce its influence. We define the total energy of the mesh, $\mathcal{E}_T(\mathcal{S})$, as

$$\mathcal{E}_T(\mathcal{S}) = \lambda_D \mathcal{E}_D(\mathcal{S}) + \mathcal{E}(\mathcal{S}) \text{ with } \mathcal{E}_(\mathcal{S}) = \sum_i \lambda_i \mathcal{E}_i(\mathcal{S}) \ . \tag{1}$$

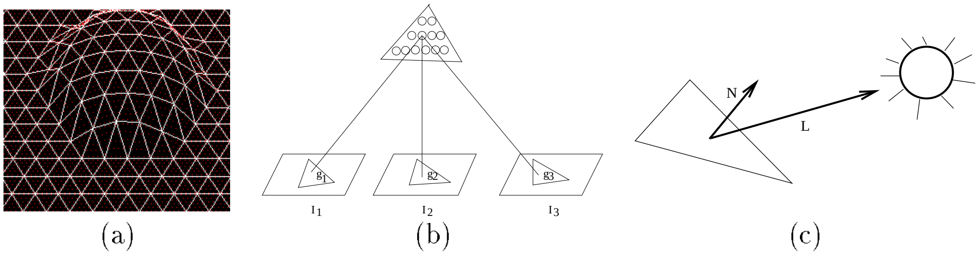

Figure 1: Mesh representation and computation of the objective function's image terms. (a) Wireframe representation of a mesh. (b) Facets are sampled at regular intervals; the circles represent the sample points. The stereo term is computed by summing the variance of the gray level of the projections of these sample points, the g_is. (c) Each facet's albedo is estimated using its normal N, the light source direction L, and the average gray level of the projection of the facet into the images. The shading term is the sum of the squared differences in estimated albedo across neighboring facets.

The $\mathcal{E}_i(\mathcal{S})$ represent the image-based constraints discussed below, and the λ_i their relative weights. $\mathcal{E}_D(\mathcal{S})$, the regularization term, serves a dual purpose. First, we define it as a quadratic function of the vertex coordinates, so that it "convexifies" the energy landscape when λ_D is large and improves the convergence properties of the optimization procedure. Second, in the presence of noise, some amount of smoothing is required to prevent the mesh from overfitting the data, and wrinkling the surface excessively [Fua and Leclerc, 1994].

In our implementation, we take $\mathcal{E}_D$ to be a measure of the curvature or local deviation from a plane at every vertex. Using finite differences, $\mathcal{E}_D$ can be expressed as a quadratic form [Fua and Leclerc, 1994]

$$\mathcal{E}_D(\mathcal{S}) = 1/2(X^T K X + Y^T K Y + Z^T K Z) \; , \tag{2}$$

where X, Y, and Z are the vectors of the x, y and z coordinates of the vertices, and K is a sparse and banded matrix.

Because $\mathcal{E}_D$ is quadratic and decouples the three spatial coordinates, our energy term is amenable to a "snake-like" optimization technique [Kass *et al.*, 1988]. We treat S as a physical surface embedded in a viscous medium and evolving under the influence of the potential $\mathcal{E}_T$. We solve the minimization problem by solving the dynamics equation of this system. We can optimize the three spatial components X, Y, and Z either simultaneously or separately.

To speed the computation and prevent the mesh from becoming stuck in undesirable local minima, we typically use several levels of mesh size to perform the computation. We start with a relatively coarse mesh that we optimize. We then refine it by splitting every facet into four smaller ones and reoptimizing. Finally, we repeat the split and optimization processes one more time.

In general we optimize all three spatial components simultaneously. However, when dealing with surfaces for which motion in one direction leads to more dramatic changes than motions in others, as is typically the case with the z direction in Digital Elevation Models (DEMs), we have found the following heuristic to be useful. We first fix the x and y coordinates of vertices and adjust z alone. Once the surface has been optimized, we then allow all three coordinates to vary.

2.2 Combining the Components

The total energy of Equation 1 is a sum of terms whose magnitudes are image- or geometry-dependent and, as a result, not necessarily commensurate. One therefore needs to scale them appropriately, that is to define the λ weights so as to make the magnitude of their contributions commensurate and independent of the specific radiometry or geometry of the scene under consideration. Since the dynamics of the optimization are controlled by the gradient of the objective function, an effective way to normalize the contributions is to introduce a set of weights λ'_i such that $\lambda'_D = 1 - \sum_{1 \leq i \leq n} \lambda'_i > 0$. The λs are taken to be

$$\lambda_i = \frac{\lambda'_i}{\| \overrightarrow{\nabla} \mathcal{E}_i(\mathcal{S}^0) \|} \ , \ \lambda_D = \frac{\lambda'_D}{\| \overrightarrow{\nabla} \mathcal{E}_D(\mathcal{S}^0) \|} \ , \tag{3}$$

where $\mathcal{S}^0$ is the surface estimate at the start of each optimization step. In practice we have found that, because the normalization makes the influence of the various terms comparable irrespective of actual radiometry or dimensions, the user-specified λ'_i weights are context-specific but not image-specific. In other words, we use one set of parameters for images of faces when combining stereo and shape-from-shading, and another when dealing with aerial images using stereo alone, but we do not have to change them for different faces or different landscapes. The continuation method of Section 2.1 is implemented by first taking λ'_D to be 0.5 and then reducing it while keeping the relative values of the λ'_is constant.

2.3 Image Constraints

In this work, we use two complementary image-based constraints: stereo and shape-from-shading.

The stereo component of the objective function is derived by comparing the gray levels of the points in all the images for which the projection of a given point on the surface is visible, as determined using the Facet-ID image. As shown in Figure 1(b), this comparison is done for a uniform sampling of the surface. More specifically, we take the stereo component of our objective function to be the variance in gray-level intensity of the projections in the various images of a given sample point on a particle, summed over all sample points in a facet, summed over all facets. This method allows us to deal with arbitrarily slanted regions and to discount occluded areas of the surface. This term can be viewed as a 3–D generalization to the purely image-based correlation techniques that use deformable windows [Nishihara, 1984, Baltsavias, 1991, Devernay and Faugeras, 1994].

The shading component of the objective function is computed using a method that does not invoke the traditional constant albedo assumption. Instead, it attempts to minimize the variation in albedo across the surface, and can therefore deal with surfaces whose albedo varies slowly. This term is depicted by Figure 1(c).

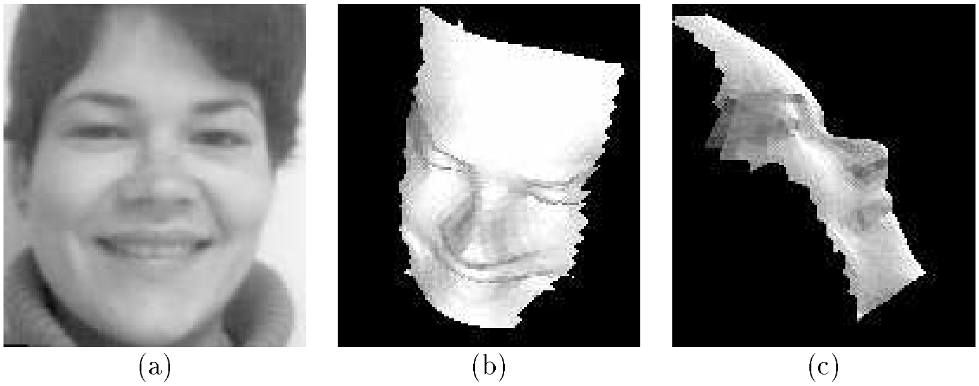

Figure 2: Combining stereo and shape-from-shading. (a) First image of a triplet (courtesy of INRIA). (b,c) Shaded views of the reconstructed surface.

Stereo information is very robust in textured regions but potentially unreliable elsewhere. We therefore use it mainly in textured areas by weighting the stereo component most strongly for facets of the triangulation that project into textured image areas. Conversely, the shading information is more reliable where there is little texture and is weighted accordingly.

The stereo and shape-from-shading terms are central to our approach. They allow the combination of geometric information with image information. However, since their behavior and implementation have been extensively discussed elsewhere, we do not describe them any further here and refer the interested reader to our previous publication [Fua and Leclerc, 1994]. In Figure 2, we show the reconstruction of a face using stereo and shape-from-shading.

3 Using Particles to Model Multiple Surfaces

Our approach to recovering multiple surfaces is to model them as sets of local surface elements that interact with one another. Following Szeliski and Tonnesen [1992], we refer to the surface patches as “oriented particles.” The forces that bind them can be understood as “breakable springs” [Terzopoulos and Vasilescu, 1991] that tend to align the particles with each other but may break for particles that are too far out of alignment.

Our surface elements are disks whose geometry is defined by the positions of their centers, the orientations of their normals and their radii. In theory, these disks have six degrees of freedom. However, in the modeling of a global surface in terms of such disks, translations along the tangent plane of the surface can be ignored as long as the disks remain roughly equidistant from one another and the radius can be chosen so that the disks approximately cover the surface. Therefore, in practice, we deal with only three degrees of freedom.

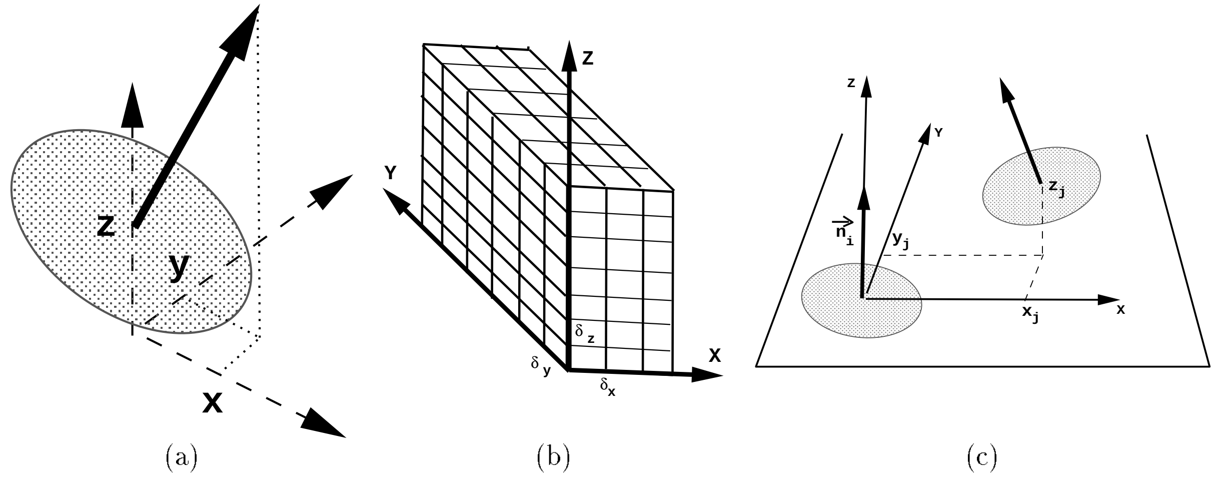

Figure 3: Data structures and metric. (a) A particle is a disk to which we associate a local referential. We allow the center of gravity to shift along the z axis and parametrize the orientation using the projections of the normal vector on the x and y axes. (b) The input 3–D points are stored in a cube-shaped set of voxels and we instantiate a particle in each voxel containing enough such points. (c) The "distance" between two particles is primarily a function of the distance of the center of gravity of one particle from the tangent plane of another.

As shown in Figure 3, to achieve an orientation-independent implementation, we assign to each particle a local referential. We define a particle's position by the translation of the center along the local vertical and its orientation by the x and y projections of the normal on the local x and y axes. This particular parametrization is most favorable when the local vertical is relatively close to the normal of the surface under consideration for two reasons. First, the x and y projections of the particle's normal vector will then be relatively small and the interaction forces between particles almost quadratic in terms of those parameters. Second, displacements along the local z axis will be close to being normal to the underlying surface and thus precisely the ones that are most significant in terms of recovering its shape.

Our procedure involves three steps: (1) instantiating a set of particles from raw stereo data, (2) refining their position and orientation by minimizing an image-based objective function, and (3) eliminating spurious particles and clustering those that appear to belong to the same global surfaces.

3.1 Initialization

We typically start with a set of stereo pairs or triplets, and corresponding disparity maps, of a given scene as our input data. We then turn each valid disparity into a 3–D point. These points typically form an extremely noisy and irregular sampling of the underlying global 3–D surfaces.

To generate a set of regularly spaced particles from such data, we pick spatial step sizes

δ_x,δ_y, and δ_z along the X, Y and Z axes of an absolute referential. We use them to define a cube-shaped set of 3–D buckets, such as the one of Figure 3(b). We then store the 3–D points computed from our initial correlation data into the appropriate buckets. By fitting a local surface to every bucket containing enough points, we generate particles whose center is the projection of the bucket's center onto the surface and whose orientation is given by the surface's normal at that point. In the presence of very noisy data, the projection may fall outside the bucket. In this case, we reject the particle, thereby ensuring that there is only one particle per bucket and that the particles are regularly distributed.

In general, most of the 3–D buckets will be empty. Therefore we do not store the set of 3–D buckets as a cube but as a hash-table allowing for both compact storage and easy computation of neighborhood relationships.

For the initialization phase to be successful, it is important both to choose the right kind of surface model and to use a robust method to perform the fitting. We have used both planar and quadric models. The quadrics, even though they involve more computation, have proven very effective because they allow the use of larger sets of points than planes without introducing any appreciable bias. In our implementation, we take advantage of this by fitting, to each bucket containing enough points, a plane of form

$$ax + by + cz = h \quad ,$$

when x, y and z are coordinates in the absolute referential. We then fit a quadric of form

$$z' = ax'x' + bx'y' + cy'y' + dx' + ez' + f \quad ,$$

where x', y', and z' are coordinates defined by the plane. We use not only the points in the bucket under consideration but also in the buckets that are its immediate neighbors. This method allows us to fit local surfaces of arbitrary orientation using a relatively large set of 3–D points, and tends to enforce consistency of orientation among neighboring particles.

Because of the noisiness of the input data, a robust surface-fitting method is essential. In this implementation, we use a variant of the Iterative Reweighted Least Squares [Beaton and Turkey, 1974] technique.

3.2 Clustering

To cluster the isolated particles into more global entities, we define a simple "same surface" relationship $\mathcal{R}$ between particles P_i and P_j as follows:

$$P_i \mathcal{R} P_j \iff d_{\mathrm{part}}(P_i, P_j) < \max_d \quad , \tag{4}$$

where d_{part} is a distance function and $\max_d$ a distance threshold. We could take d_{part} to be the Euclidean distance between particle centers. However, such a distance would not be discriminating enough for our purposes because it is circularly symmetric and does not take the particles'orientation into account. It has proved much more effective to define a distance function that penalizes more heavily the distance of one particle's center from the tangent plane of the other than the distance along the tangent plane. The simplest way to achieve this result is to define d_{part} as follows:

$$\begin{aligned} d_j &= kz_j^2 + (1-k)(x_j^2 + y_j^2) \ , \\ d_i &= kz_i^2 + (1-k)(x_i^2 + y_i^2) \ , \\ d_{\text{part}}(P_i, P_j) &= max(d_i, d_j) \end{aligned} \tag{5}$$

where x_j,y_j and z_j are the coordinates of the center of P_j in a referential whose Z direction is the normal of P_i and whose origin is the center of P_i, as shown in Figure 3(c), and k is a constant larger than 0.5. In this paper, we take $k = 0.9$; x_i, y_i, and z_i are defined symmetrically.

In essence, the threshold $\max_d$ on d_{part} limits the curvature of the common underlying surface to which particles may belong. As such it is domain-dependent; here we take $\max_d$ to be a multiple, typically 1.5, of the median value of d_{part} for all pairs of neighboring particles in the cube-shaped structure of Figure 3(b).

The data set equipped with the relationship $\mathcal{R}$ can now be viewed as a graph whose connected components are the surfaces we are looking for. In practice, there will usually be spurious particles that are weakly linked to legitimate clusters. In such cases, we have found that removing all points that do not have a minimum number of neighbors allows us to throw away the gross errors and generate meaningful clusters.

To demonstrate the effectiveness of our initialization and clustering methods given relatively clean stereo data, we use the sequence of forty 512x512 images depicted by Figure 4. The images were acquired with a video camera over a period of a few seconds by turning around the subject who was trying to stand still. Camera models were later computed using standard photogrammetric techniques at the Institute for Geodesy and Photogrammetry, ETH-Zürich.

We ran a correlation-based algorithm—once for each consecutive pair of images in the sequence—stored the resulting 3–D points in a 80x80x80 set of voxels, and instantiated particles in all voxels containing at least 200 points. The results are shown in the second row of Figure 4. Because we use a large number of images, the main features of the head—including the nose, mouth, chin, ears, and even the boundary of the skullcap—are clearly captured by our representation. However, because the correlation-based algorithm produced erroneous, but not random, disparities around occlusion boundaries, we also find a number of spurious particles around the nose, chin, and back of the head. To get rid of them we computed the interparticle distance described above and eliminated all particles not having at least four neighbors within 1.2 times the median distance between neighbors, as shown in the third row of Figure 4.

3.3 Refining

Because it is extremely difficult to design a stereo algorithm that never produces correlated artifacts, we cannot expect any robust fitting technique to exclude all erroneous 3–D points. Furthermore, fitting local surfaces to the initial data amounts to smoothing and may result

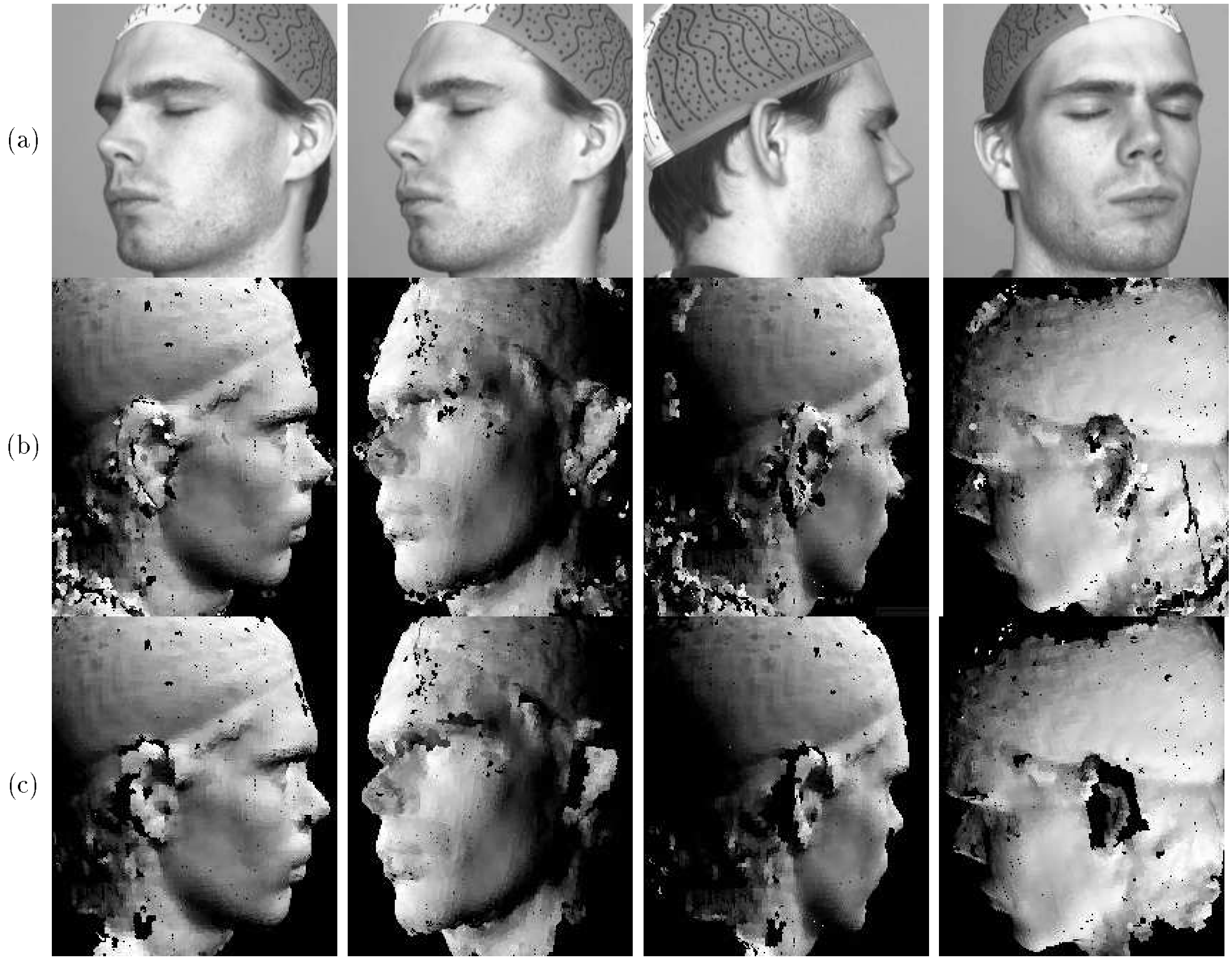

Figure 4: Modeling a complete head. Row (a): A stereo pair of a person's head and left images of two additional stereo pairs taken from different viewpoints (courtesy of ETH-Zürich). Row (b): Four shaded views of the particles instantiated by fitting local surfaces to the 3–D points derived by correlating the images of Figure 4. Row (c): Similar views of the subset of particles that belong to the same global surface. Erroneous ones have been eliminated.

in spurious particles that appear to line up with legitimate ones and become very hard to eliminate. To resolve such problems, it is necessary to return to the original images and assess the quality of each particle. For each disk-shaped particle, we define a stereo term by projecting the 3–D disks into 2–D elliptical patches in each image and measuring how well these patches correlate. This term is equivalent to the one defined in Section 2.3, except that it is now a disk as opposed to a triangular facet that is sampled at regular intervals [Fua, 1994]. It is a function of the three degrees of freedom of each particle and can therefore be used to perform optimization. We allow the particles to interact with one another and to rearrange themselves to minimize an energy term that is the sum of the multi-image intensity correlation term and of a deformation energy term that tends to enforce consistency between neighboring particles [Szeliski and Tonnesen, 1992]. While optimizing the energy term, the particles that actually correspond to the same underlying global surfaces will "stick together" and the ones that do not will tend to move in separate directions, stop lining up with each other, and be easily eliminated by the clustering technique of Section 3.2.

Formally, we write the total energy of a set of particles $\mathcal{E}_T$ as

$$\mathcal{E}_T = \mathcal{E}_{St} + \lambda_D \mathcal{E}_D \quad , \tag{6}$$

where $\mathcal{E}_{St}$ is the multi-image intensity correlation term, $\mathcal{E}_D$ the deformation energy term, and λ_D a weighting coefficient that is dynamically adjusted using the method of Section 2.2.

To define $\mathcal{E}_D$, we follow Szeliski and Tonnesen [1992], and define a conormality potential $\mathcal{E}^{ij}_{cn}$ and a coplanarity potential $\mathcal{E}^{ij}_{cp}$ between particles i and j by writing

$$\begin{aligned} \mathcal{E}^{ij}_{cn} &= 1/2||\vec{n_i} - \vec{n_j}||^2 = 1 - \vec{n_i}\vec{n_j} \ , \\ \mathcal{E}^{ij}_{cp} &= 1/2((\vec{n_i}\vec{r_{ij}})^2 + (\vec{n_j}\vec{r_{ij}})^2) \ , \end{aligned} \tag{7}$$

where $\vec{n_i}$ and $\vec{n_j}$ are the normal vectors and $\vec{r_{ij}}$ the vector joining the centers of the two particles. These terms control the surface's resistance to bending and we take our overall regularization terms $\mathcal{E}_D$ to be

$$\mathcal{E}_D = \sum_{i,j} f(\mathcal{E}^{ij}_{cn} + \mathcal{E}^{ij}_{cp}) \quad , \tag{8}$$

where the summation over i and j denotes a summation over all pairs of particles that are neighbors in the cube-shaped structure of Section 3.1, and f is a monotonically increasing function. In practice we implement the concept of breakable springs by taking f to be

$$f(x) = log(1 + x/s) \quad ,$$

with s being a fixed constant so that, as in Section 3.1, the interaction forces have a Lorentzian behavior [Black and Rangarajan, 1994]. As the particles move out of alignment, the strength of the interaction increases up to a point, after which the interaction strength decreases and eventually vanishes.

In our implementation, we have not found it necessary to weight the interactions: By construction, our particles tend to be equidistant and cannot slide along the surfaces because they have only three degrees of freedom.

3.4 Global Optimization

Recall that the total energy of a set of n particles is written as

$$\mathcal{E}_T = \mathcal{E}_{St} + \lambda_D \mathcal{E}_D \quad .$$

Since each particle has three degrees of freedom, $\mathcal{E}_T$ is a function of $3n$ state variables. This is different from the case of meshes; $\mathcal{E}_D$ is not quadratic and we cannot use the "snake-like" optimization technique of Section 2.1. Instead, we use more conventional techniques, such as conjugate gradient, that tend to be much slower, especially for large sets of particles. An effective way to alleviate this problem and to achieve a desirable minimum of the objective function is to cluster the particles into smaller subsets, to optimize each subset independently, and to reiterate the process until a stable solution is found. In practice, this can be achieved either by using the metric of Section 3.2 to break the set of particles into smaller subsets or by histogramming the median gray levels of the particles—computed using the gray-levels of the projections of sample points on the disks—and grouping those that fall into the same histogram peaks. The latter makes sense in the absence of surface markings because particles that have similar gray levels are more likely to belong to the same underlying surfaces than particles that do not.

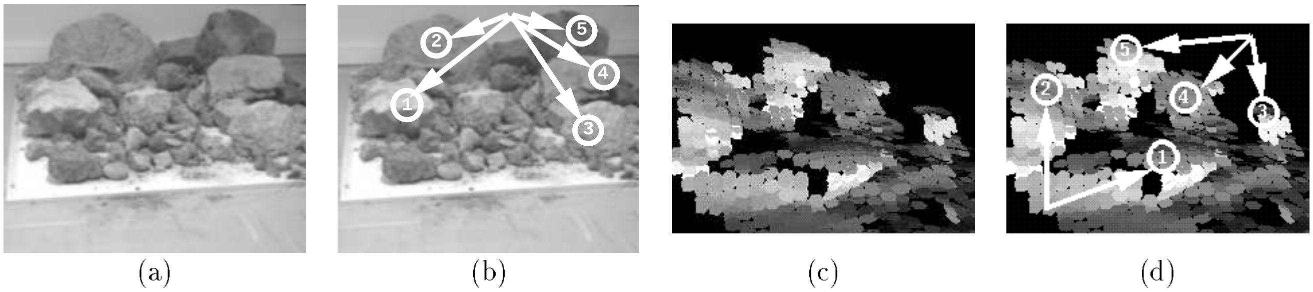

Figure 5: Modeling a pile of rocks. (a) The first image of a triplet (courtesy of INRIA). (b) The same image with the five largest rocks labeled from 1 to 5. (c) Shaded view of the particle set after refinement and clustering seen from a viewpoint located on the left side of the rock pile. (d) The same view with the five largest rocks labeled as in (b). Note that the overhang of rock number 1 is well recovered, a result that would be difficult to achieve using a 2-1/2–D representation.

In our final example, shown in Figure 5, we used all three steps of our surface reconstruction procedure to reconstruct a ground-level scene using three triplets of images acquired by the INRIA mobile robot and the precomputed robot's ego-motion. Note that the five main rocks in the scene, including one that overhangs, appear in the reconstruction.

4 Conclusion

We have presented a surface reconstruction framework that uses object-centered representations to recover geometry and reflectance properties from multiple images.

When the surface's topology is known *a priori*, we represent it as a triangulated mesh and are able to handle self-occlusions while merging information from several viewpoints, thereby allowing us to eliminate blindspots and make the reconstruction more robust where more than one view is available. The reconstruction process relies on both monocular shading cues and stereoscopic cues. We use these cues to drive an optimization procedure that takes advantage of their respective strengths while eliminating some of their weaknesses.

When the surfaces's topology is unknown or there may be more than one surface in the scene, we replace the triangulation by a set of oriented particles that allow us to achieve similar results, but with a higher computational cost.

References

[Baltsavias, 1991] E. P. Baltsavias. *Multiphoto Geometrically Constrained Matching.* PhD thesis, Institute for Geodesy and Photogrammetry, ETH Zurich, December 1991.

[Beaton and Turkey, 1974] A. E. Beaton and J.W. Turkey. The Fitting of Power Series, Meaning Polynomials, Illustrated on Band-Spectroscopic Data. *Technometrics*, 16:147–185, 1974.

[Black and Rangarajan, 1994] M. J. Black and A. Rangarajan. The Outlier Process: Unifying Line Processes and Robust Statistics. In *Conference on Computer Vision and Pattern Recognition*, pages 121–128, Seattle, WA, June 1994.

[Devernay and Faugeras, 1994] F. Devernay and O. D. Faugeras. Computing Differential Properties of 3–D Shapes from Stereoscopic Images without 3–D Models. In *Conference on Computer Vision and Pattern Recognition*, pages 208–213, Seattle, WA, June 1994.

[Fua and Leclerc, 1994] P. Fua and Y. G. Leclerc. Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading. *International Journal of Computer Vision*, 1994. In press, available as Tech Note 535, Artificial Intelligence Center, SRI International.

[Fua, 1994] P. Fua. Reconstructing Complex Surfaces from Multiple Stereo Views. Tech Note 550, Artificial Intelligence Center, SRI International, December 1994.

[Kass *et al.*, 1988] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active Contour Models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[Nishihara, 1984] H.K. Nishihara. Practical Real-Time Imaging Stereo Matcher. *Optical Engineering*, 23(5), 1984.

[Szeliski and Tonnesen, 1992] R. Szeliski and D. Tonnesen. Surface Modeling with Oriented Particle Systems. In *Computer Graphics (SIGGRAPH)*, volume 26, pages 185–194, July 1992.

[Terzopoulos and Vasilescu, 1991] D. Terzopoulos and M. Vasilescu. Sampling and Reconstruction with Adaptive Meshes. In *Conference on Computer Vision and Pattern Recognition*, pages 70–75, 1991.

[Terzopoulos, 1986] D. Terzopoulos. Regularization of Inverse Visual Problems Involving Discontinuities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:413–424, 1986.