Direct Methods for Visual Scene Reconstruction

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

Abstract

There has been a lot of activity recently surrounding the reconstruction of photorealistic 3-D scenes and high-resolution images from video sequences. In this paper, we present some of our recent work in this area, which is based on the registration of multiple images (views) in a projective framework. Unlike most other techniques, we do not rely on special features to form a projective basis. Instead, we directly solve a least-squares estimation problem in the unknown structure and motion parameters, which leads to statistically optimal estimates. We discuss algorithms for both constructing planar and panoramic mosaics, and for projective depth recovery. We also speculate about the ultimate usefulness of projective approaches to visual scene reconstruction.

1 Introduction

The recovery of 3-D scene information from multiple views has long been one of the central problems in computer vision. Over the last decade, many researchers observed that such a full reconstruction may not be necessary for many vision-based tasks, e.g., face or object recognition. More recently, however, there has been a resurgence of interest in 3-D scene reconstruction, motivated both by improvements in algorithms and processing speeds, and the emergence of interesting applications such as virtual reality and model-based video compression.

Traditionally, 3-D scene reconstruction has been the focus of both stereo and structure from motion, two subfields with complementary sets of assumptions and techniques. In this paper, we present some of our recent techniques in this area, which blend aspects of both stereo and structure from motion [14, 15, 13]. We call our techniques *direct*, since they both directly minimize an image-based misregistration measure (without special algebraic or geometric transformations), and because they are (usually) based on the direct minimization of intensity errors.

Our techniques share a number of characteristics which distinguish them from traditional approaches to structure from motion and stereo. Whenever possible, we use many views instead of just two views, since this leads to more reliable estimates. We formulate our reconstruction algorithms using projective geometry, which allows them to work with uncalibrated cameras as well as cameras with time-varying parameters (e.g., zoom). We also formulate our problems as the direct (iterative) minimization of image-based measures of misregistration, instead of using algebraic manipulations which can result in marked sensitivity to noise. Under small Gaussian noise in either feature position or intensity samples, such techniques are statistically optimal. Finally, our projective depth recovery algorithm yields a dense estimate of scene depth, unlike most structure from motion algorithms.

The current focus for our work has been the creation of realistic high-resolution imagery and 3-D environments from low-resolution, uncalibrated video. Our applications range from automatically creating 360° panoramas from video or photographs (e.g., of an office or a whiteboard), to reconstructing the 3-D shape of individual objects. Our long term goal is to automatically construct 3-D indoor and outdoor environments for applications such as home sales, virtual supermarket shopping, and tele-travel (Section 6).

We begin the paper with the construction of high-resolution image mosaics from low-resolution video (Section 2). We then present our algorithm for projective depth recovery, and discuss its application to view interpolation and extrapolation (Section 3). For those cases when it is necessary to bootstrap the intensity-based shape from motion algorithm with a feature-based algorithm, we present our affine and projective structure from motion algorithms (Section 4). Finally, we discuss possible visual scene representations based on our techniques, and some potential applications.

2 Video Mosaics

The first technique we describe automatically aligns and composites multiple images into high-resolution mosaics [13]. Building aerial photomosaics has long been a staple of photogrammetry, but only recently have fully automated techniques for building mosaics been developed. Most techniques still only estimate pure translations or affine transformations [4], but some recent work has dealt with the full projective case [8]. Our approach is, to our knowledge, the first to combine full projective warping with near real-time performance.

Our techniques for automatically aligning images into photomosaics exploit the particularly simple form of the motion field resulting from two specific imaging situations. The first case is when the images cover a portion of a planar scene, e.g., a whiteboard, a desktop, or a wall. The second case is when the camera rotates around an axis through its focal point (or when all scene objects are very far from the camera). Under

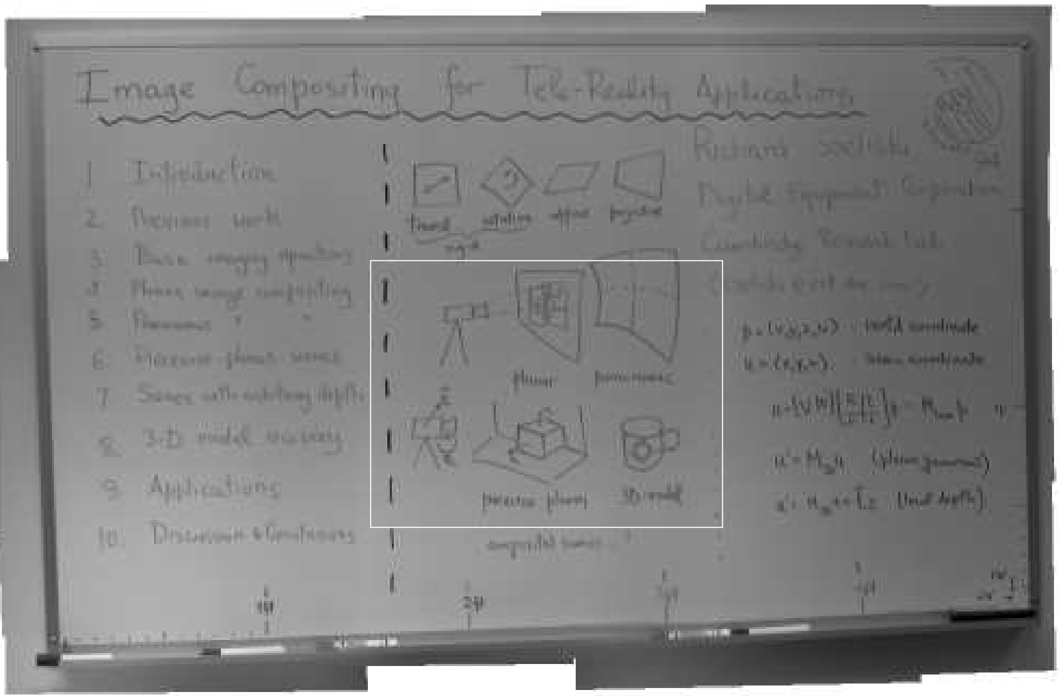

Figure 1: Whiteboard image mosaic example. The central square shows the size of one input image (*tile*).

either of these two conditions, the inter-frame motion can be represented by a *homography*, i.e., a linear function of projective image coordinates $\mathbf{u}' = \mathbf{M}\mathbf{u}$ (see [13] for a simple proof).

In the subsections below, we describe our algorithm in more detail, and give examples of its application to the specific cases of planar scenes, panoramas, and multiresolution mosaics.

2.1 Planar Scenes

The compositing of multiple images into larger mosaics requires two basic steps: an image-to-image alignment (preferably to sub-pixel precision), and a method for seamlessly blending images. Many different solutions are possible for the first problem, including matching four or more feature points and then solving for the homography, or manually adjusting image positions using a blink comparator.

The approach we have taken is to directly minimize the discrepancy in intensities between pairs of images after applying the transformation we are recovering. Our technique does not require the location and correspondence of feature points, and is statistically optimal in the vicinity of the true solution [14]. Let us write the 2-D homography as

$$x_i' = \frac{m_0 x_i + m_1 y_i + m_2}{m_6 x_i + m_7 y_i + 1}, \; y_i' = \frac{m_3 x_i + m_4 y_i + m_5}{m_6 x_i + m_7 y_i + 1}. \quad (1)$$

Our technique minimizes the sum of the squared intensity errors

$$E = \sum_i [I'(x_i', y_i') - I(x_i, y_i)]^2 = \sum_i e_i^2 \quad (2)$$

over all corresponding pairs of pixels i which are inside both images. Once we have found the best transformation $\mathbf{M}$, we can *warp* image I' into the reference frame of I using $\mathbf{M}$ and then blend the two images together. To reduce visible artifacts, we weight images being blended together more heavily towards the center, using a bilinear weighting function.

To perform the minimization, we use the Levenberg-Marquardt iterative non-linear minimization algorithm (see [14, 13] for details). The advantage of using Levenberg-Marquardt over straightforward gradient descent is that it converges in fewer iterations.

Unfortunately, both gradient descent and Levenberg-Marquardt only find locally optimal solutions. If the motion between successive frames is large, we use *hierarchical matching*, which first registers smaller, subsampled versions of the images where the apparent motion is smaller. For even larger displacements, we use *phase correlation*, which is a technique based on 2-D Fourier transforms [6].

To demonstrate the performance of our algorithm, we digitized an image sequence with a camera panning over a whiteboard. Figure 1 shows the final mosaic of the whiteboard, with the location of a constituent image shown as a white outline. This mosaic is 1300×2046 pixels, based on compositing 39 NTSC (640×480) resolution images.

2.2 Panoramic Mosaics

In order to build a panoramic image mosaic, we rotate a camera around its optical center. Images taken in this manner are related by 2-D projective transformations, just as in the planar case [13]. Intuitively, we cannot tell the relative depth of points in the scene as we rotate (there is no *motion parallax*), so they could be located on a plane.

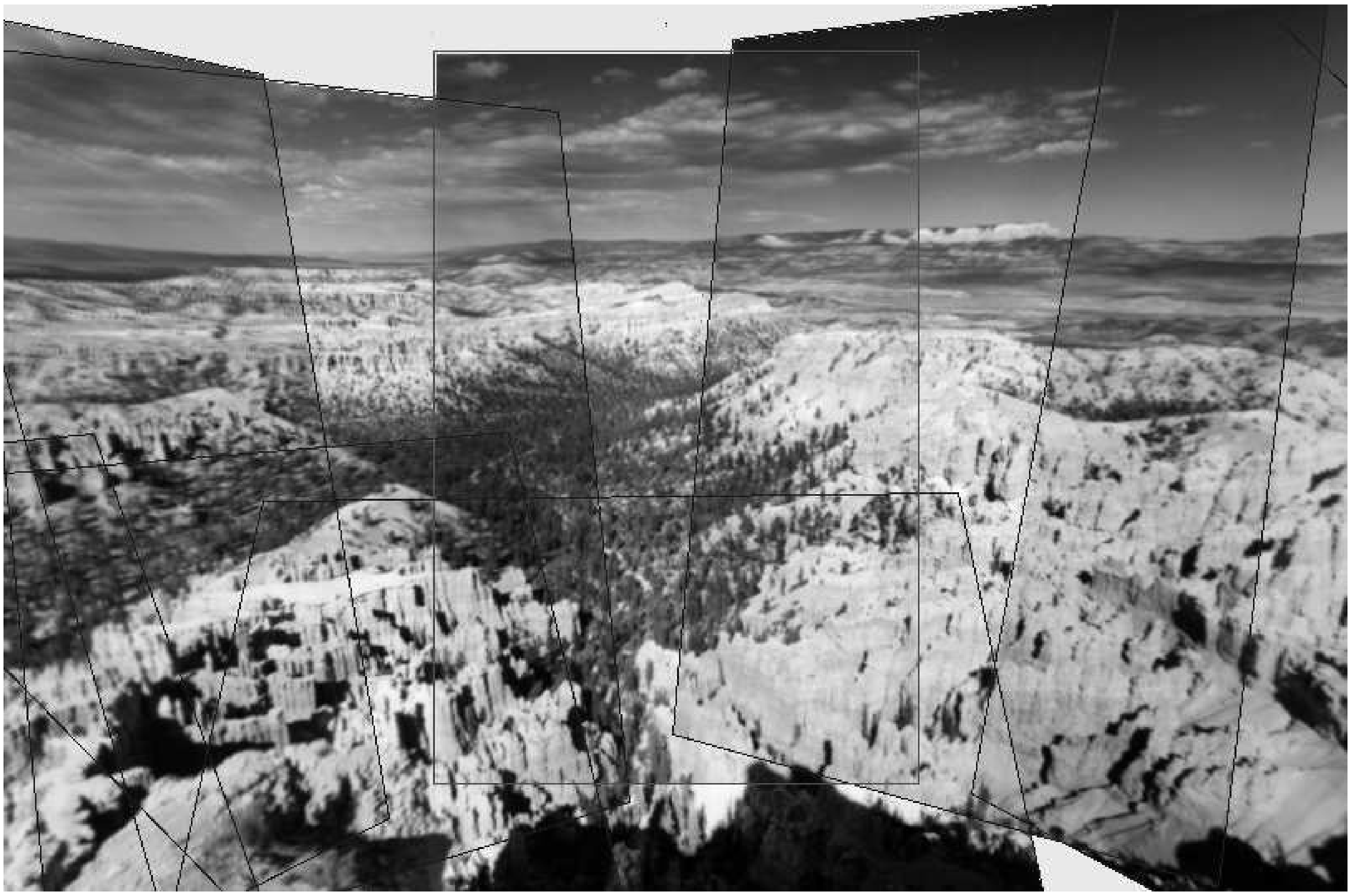

Figure 2: A portion of the Bryce Canyon mosaic. Because of the large motions involved, we cannot use a single plane for the whole mosaic. Instead, we can select different tiles as base images.

Figure 3: Circular panoramic image mosaic example (office interior). A total of 36 images were pasted onto a cylindrical viewing surface.

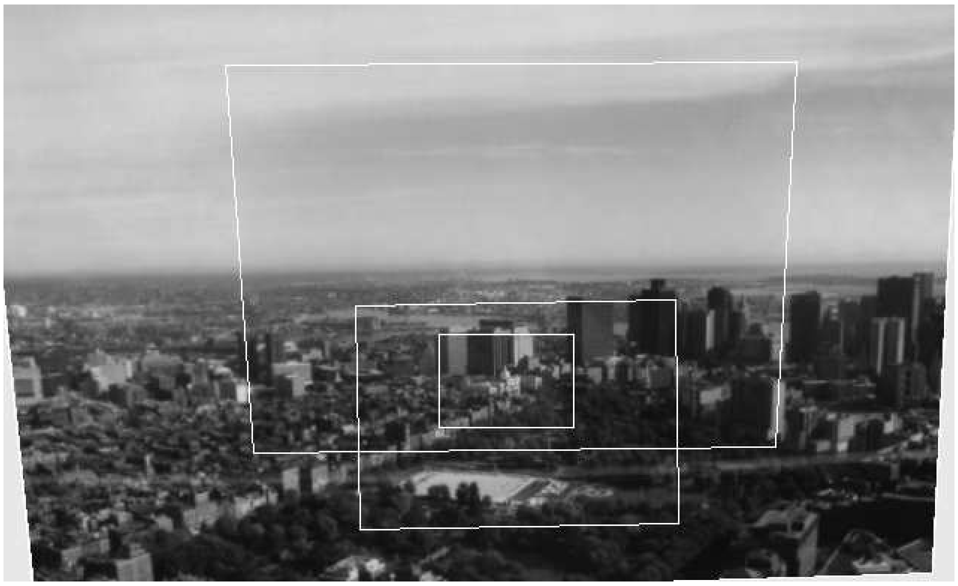

Figure 4: Zoom sequence. The outlines show the extents of the four constituent images.

More formally, the 2-D transformation denoted by $\mathbf{M}$ is related to the 3×3 viewing matrices $\mathbf{V}$ and $\mathbf{V}'$ and the inter-view rotation matrix $\mathbf{R}$ by [13]

$$\mathbf{M} = \mathbf{V}'\mathbf{R}\mathbf{V}^{-1} \tag{3}$$

(see Section 3 for definitions of $\mathbf{V}$ and $\mathbf{R}$). In the case of a completely calibrated camera, $\mathbf{M}$ is a pure rotation matrix (only three unknowns). If the focal lengths in the two images are unknown, then these two parameters must also be estimated. In either case, we can register any two overlapping images using the same technique as for the planar mosaic case.

How do we represent a panoramic scene composited using our techniques? One approach is to divide the viewing sphere into several large, potentially overlapping regions, and to represent each region with a plane onto which we paste the images. Examples of such mosaics are given in [13]. Another approach is to compute the relative position of each frame relative to some base frame, and to periodically choose a new base frame for doing the alignment. We can then re-compute an arbitrary view on the fly from all visible pieces, given a particular rotation matrix $\mathbf{R}$ and zoom factor f. This is the approach used to composite a large wide-angle mosaic of Bryce Canyon, as shown in Figure 2.

A third approach is to use a cylindrical viewing surface to represent the image mosaic [8]. In this approach, we map world coordinates $\mathbf{p} = (x, y, z)$ onto 2-D cylindrical screen locations $\mathbf{u} = (\theta, v)$, with $\theta = \tan^{-1}(x/z)$ and $v = y/\sqrt{x^2 + z^2}$. Figure 3 shows a complete circular panorama of an office unrolled onto a cylindrical surface.

2.3 Multiresolution Mosaics

The techniques described so far have used a single-resolution compositing surface to blend all of the images together. In many applications, we may wish to have spatially-varying amounts of resolution, e.g., for zooming in on areas of interest.

The modifications to the basic planar mosaic building algorithm are relatively straightforward, and affect only the image blending portion of the algorithm. To create the new composite mosaic, we weight each image by an amount proportional to the difference in scale from the desired view.

Figure 4 shows the result of compositing four images of a city scene taken from an office tower. These images were taken with a hand-held 35mm camera equipped with a 28-200mm zoom lens, and the resulting 4”×6” photographs were scanned in at 300dpi. The multiresolution mosaic has a 7:1 variation in original image scales. The video sequence seen by a user zooming in on the central feature of interest (the State House) shows an even wider range of scales. To zoom from an NTSC resolution wide-angle shot encompassing all four images down to a slightly magnified (4:1) version of the most detailed image involves a scaling of over 100:1.

3 Projective Depth Recovery

While mosaics of flat or panoramic scenes can be useful for some applications, other applications require the recovery of dense depth maps. When the camera motion is known, the problem of depth map recovery is called *stereo reconstruction* (or multi-frame stereo if more than two views are used). When the camera motion is unknown, we have the more difficult *structure from motion* problem [2, 15]. In this section, we present our solution to this latter problem based on recovering *projective depth*, which is particularly simple and robust and fits in well with the methods already developed in this paper.

To formulate the projective structure from motion recovery problem, we first write the perspective projection from world coordinates $\mathbf{p} = (X, Y, Z, W)$ to screen coordinates

$\mathbf{u} = (x, y, w)$ as

$$\mathbf{u} = \mathbf{V}\,[\mathbf{R}\,|\,\mathbf{t}]\,\mathbf{p}, \qquad (4)$$

where $\mathbf{V}$ is the upper triangular *viewing matrix*, and $\mathbf{R}$ and $\mathbf{t}$ are the usual rotational and translational components of the camera motion [2]. Without loss of generality, we can set $\mathbf{R} = \mathbf{I}$ and $\mathbf{t} = \mathbf{0}$ in the first frame. The world coordinates corresponding to an optical ray (in the first image) passing through $\mathbf{u}$ are therefore

$$\mathbf{p} = \begin{bmatrix} \mathbf{V}^{-1}\mathbf{u} \\ d \end{bmatrix}$$

where d is the *projective depth* of the world point [15]. The coordinates corresponding to a pixel $\mathbf{u}$ with projective depth d in some other frame can therefore be written as

$$\mathbf{u}' = \mathbf{V}'\mathbf{R}\mathbf{V}^{-1}\mathbf{u} + d\mathbf{V}'\mathbf{t} = \mathbf{M}\mathbf{u} + d\tilde{\mathbf{t}}, \qquad (5)$$

i.e., as the summation of a planar projective transformation (homography) and a depth-dependent parallax motion (in the direction of the epipole). This formulation has formed the basis of both our projective structure from motion algorithms [15] and our projective dense depth estimation algorithm [14]. More recently, it been used by other researchers under the names of *affine depth* [12] and *planar parallax* [10, 7] (see Section 4.2 for a more detailed discussion of projective depth).

The above formulation extends naturally to multiframe depth recovery by simply associating a separate $\mathbf{M}_j$ and $\tilde{\mathbf{t}}_j$ with each frame and minimizing the summed intensity error

$$E = \sum_{j\neq 0}\sum_i [I_j(x'_{ij}, y'_{ij}) - I_0(x_i, y_i)]^2 = \sum_{j\neq 0}\sum_i e_{ij}^2. \quad (6)$$

To recover the parameters in $\mathbf{M}_j$ and $\tilde{\mathbf{t}}_j$ for each frame along with the depth values d_i (which are the same for all frames), we use the same Levenberg-Marquardt algorithm as before. Once the projective depth values are recovered, they can be used directly in viewpoint interpolation (using a new $\mathbf{M}$ and $\tilde{\mathbf{t}}$), or they can be converted to true Euclidean depth using at least 4 known depth measurements [2].

In more detail, we can write the projection equation (5) as

$$\begin{aligned} x'_{ij} &= \frac{m_0^{(j)} x_i + m_1^{(j)} y_i + t_0^{(j)} d_i + m_2^{(j)}}{m_6^{(j)} x_i + m_7^{(j)} y_i + t_2^{(j)} d_i + 1}, \\ y'_{ij} &= \frac{m_3^{(j)} x_i + m_4^{(j)} y_i + t_1^{(j)} d_i + m_5^{(j)}}{m_6^{(j)} x_i + m_7^{(j)} y_i + t_2^{(j)} d_i + 1}. \end{aligned} \qquad (7)$$

To estimate the unknown parameters, we alternate iterations of the Levenberg-Marquardt algorithm over the motion parameters $\{m_0^{(j)}, \ldots, t_2^{(j)}\}$ and the depth parameters $\{d_i\}$. In our current implementation, in order to reduce the total number of parameters being estimated, we represent the depth map using a tensor-product spline, and only recover the depth estimates at the spline control vertices (the complete depth map is available by interpolation) [14].

Figure 5 shows an example of using our projective depth recovery algorithm. The image sequence was taken by moving the camera up and over the scene of a table with stacks of papers (Figure 5a). The resulting depth-map is shown in Figure 5b as intensity-coded range values.

3.1 View Interpolation

Once a dense depth map has been recovered for the scene, we can use this information to synthesize (interpolate or extrapolate) novel views [1, 11, 13]. When a Euclidean depth map is available, regular 3-D graphics can be used for the view synthesis [1]. In other situations, corresponding points must be found between the original views and the novel view in order to compute the required transformations [11], or the projective depth description must be converted to a Euclidean one [2].

A simpler approach, which often produces results of acceptable quality, is to simply re-scale the projective depths by an amount which yields a sensible 3-D scene when viewed from moderate viewing angles. This is the approach we used to generate the pictures in Figure 5. Figure 5c shows the original intensity image texture mapped onto the surface seen from a side viewpoint which is not part of the original sequence (an example of view extrapolation). Figure 5d shows a set of grid lines overlayed on the recovered surface to better judge its shape.

4 Affine and Projective Structure from Motion

In the preceding section, we ignored the problem of local minima in the search space. Our experience has been that our direct intensity-based projective depth recovery algorithm converges to a good solution with only a small hint as to the camera translation direction (e.g., vertical for Figure 5). In some situations, however, it may be necessary to bootstrap the dense depth recovery algorithm by first estimating the camera motion using a feature-based structure from motion algorithm.

Traditional structure from motion algorithms attempt to recover a Euclidean reconstruction of the world [2]. More recent algorithms, motivated by the difficulty of obtaining metrically accurate 3-D reconstructions, have attacked the problem of recovering an affine [5, 16] or projective [3, 9, 11] description. The advantage of this approach is that it does not require camera calibration and can lead to more reliable estimates [3]. It may also be sufficient for many vision-based tasks such as re-projection and object recognition [11].

Our structure from motion algorithm [15] directly minimizes (using Levenberg-Marquardt) the squared difference between predicted and measured screen coordinates

$$E = \sum_j \sum_i \sigma_{ij}^{-2}[(u_{ij} - x'_{ij})^2 + (v_{ij} - y'_{ij})^2], \qquad (8)$$

where (u_{ij}, v_{ij}) is the screen location of the ith feature in the jth frame, and (x'_{ij}, y'_{ij}) are given by (7). Each measurement can be weighted by its inverse variance σ_{ij}^{-2}, which can be set to zero for missing measurements. Such as weighting leads to

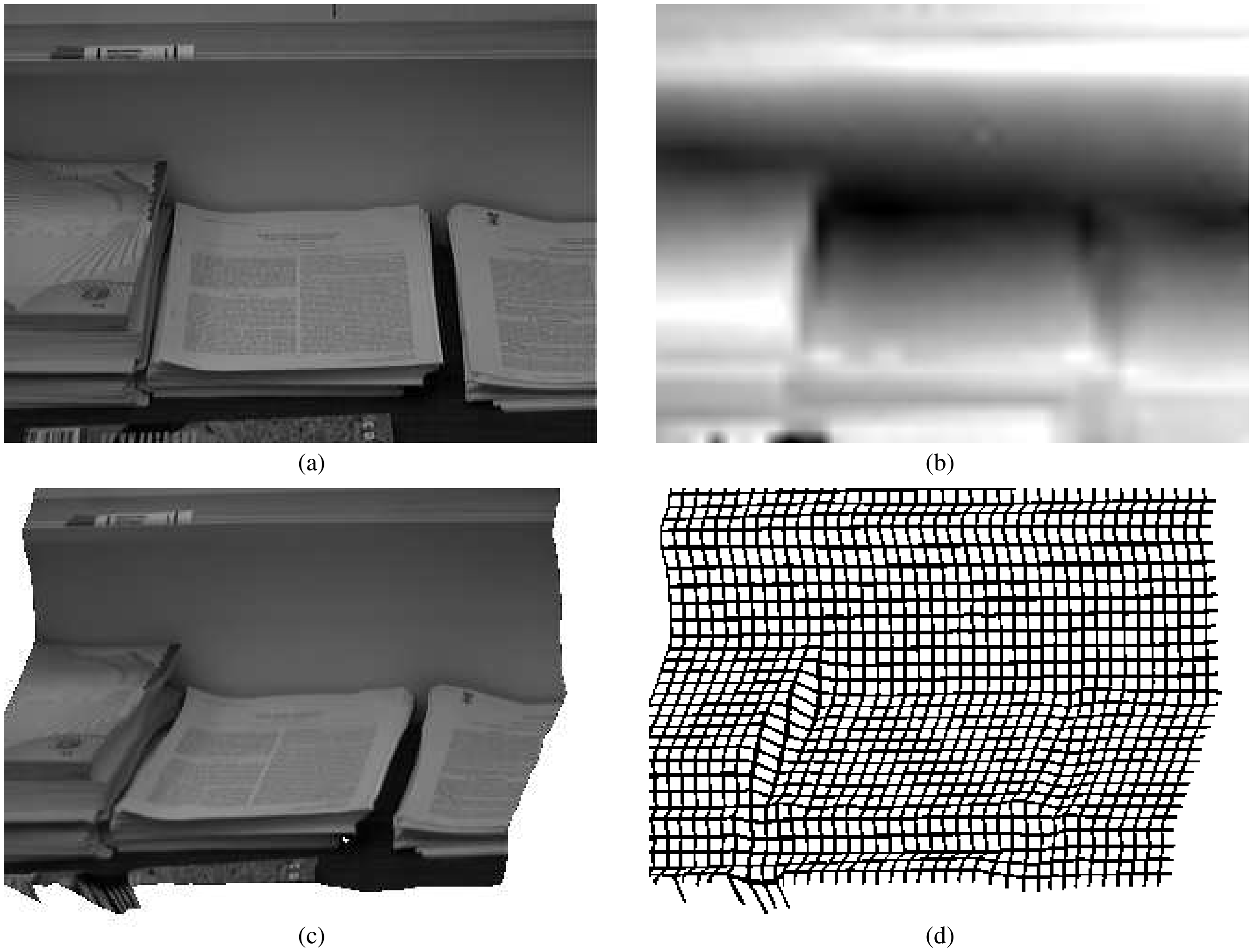

Figure 5: Depth recovery example—table with stacks of papers: (a) input image, (b) intensity-coded depth map (dark is farther back) (c) texture-mapped surface seen from novel viewpoint, (d) gridded surface.

a statistically optimal (maximum likelihood) estimate of the unknown parameters.

For our feature-based algorithm, we optimize over all frames (including frame zero), and the x_i and y_i coordinates of the 3-D points in (7) are also treated as unknowns. However, since we set $\mathbf{M}_0 = 0$ and $\tilde{\mathbf{t}}_0 = 0$ as before, the (x_i, y_i) values remain close to the screen positions measured in frame zero. Unlike most other projective reconstruction algorithms, we do not choose a set of feature points as a projective basis. This allows the algorithm to work with tracks where features may disappear at any time, and avoids the sensitivity of the results to the choice of basis points.

4.1 Algorithm initialization

To initialize our non-linear least-squares algorithm, we have tried two approaches (a third approach to bootstrapping the algorithm, which we have not investigated, is to use fundamental matrices). The first is to simply set $(x_i, y_i, d_i) = (u_{i0}, v_{i0}, 0)$, $\mathbf{M}_j = \mathbf{I}$, and $\tilde{\mathbf{t}}_j = 0$, i.e., to set the 3-D points to lie on a null plane, and to assume no motion. In our experiments, the algorithm usually converges in under a dozen iterations.

Our second approach, which yields much quicker results, is to first solve for the $\mathbf{M}_j$ by computing a planar projective transformation, i.e., to fix $(x_i, y_i, d_i) = (u_{i0}, v_{i0}, 0)$ and to optimize (8). Then, a guess for the focus of expansion for each frame, which corresponds to $\tilde{\mathbf{t}}_j$, can be computed by finding the dominant eigenvalue of the moment matrix of the residual vectors $(u_{ij} - x_{ij'}, v_{ij} - y'_{ij})$. It turns out that in the orthographic case, i.e., for affine structure from motion (where the denominators in (7) are unity), this two step approach results in an exact solution (in the noise-free case), and is equivalent to singular value decomposition [16] but at a lower computational cost. For perspective projection, the planar motion computed by the first step may not correspond to the motion of an actual plane, but this will be corrected during the iterative minimization, which often converges in just a single step.

4.2 Ambiguities in solution

The projective depths d_i and motion descriptors $(\mathbf{M}_j, \tilde{\mathbf{t}}_j)$ have a four-parameter ambiguity associated with them, even after we have set $\mathbf{M}_0 = \mathbf{I}$ and $\tilde{\mathbf{t}}_0 = 0$. One of these ambiguities corresponds to the scale ambiguity present in Euclidean structure from motion, i.e., we can scale d_i and $\tilde{\mathbf{t}}_j$ by reciprocal factors and still obtain the same solution (predicted feature positions).

In a similar manner, we can add a multiple of $\tilde{\mathbf{t}}_j$ to any column of $\mathbf{M}_j$ and modify d_i appropriately, which corresponds to adding a plane equation to the d_i. This three-parameter ambiguity corresponds to choosing the plane relative to which the projectives depths d_i are defined. Planar parallax techniques [10, 7] assume that this plane is the one with the dominant motion. Our structure from motion technique finds a plane which is close to a least-squares plane fit to the depths.[1]

5 Representations for complex scenes

The reconstruction of 3-D scenes using a projective framework raises some interesting questions about the representation of the scene. At the most primitive level, the output of a structure from motion algorithm may just be a collection of points and camera matrices. While this may be adequate for certain tasks such as navigation, it is not that useful for tasks such as view-based recognition or virtual reality.

The dense range maps available from multiframe stereo techniques are more interesting. They can be used to synthesize novel views using view interpolation [1], even in the absence of full metric information [11]. For true virtual environments, however, multiple depth maps must be combined into a richer structure, which may require segmentation.

Several alternatives exist for the representation of such environments. One possibility would be to reduce the world to a collection of (hopefully continuous) planar surfaces [17], which could then be texture mapped. Another possibility is to have a collection of contiguous depth maps and images, which could then be rendered using either conventional graphics or multi-frame view interpolation [1]. The question of how to merge such multiple depth maps is an active research area. Such systems would also have to include multiresolution representations, at least if a large range of viewing positions or virtual camera settings were permitted. For true 3-D objects, however, volumetric or parametric 3-D models may be the best choice.

6 Applications

The reconstruction of visual scenes has many potential applications, including object recognition, model-based video compression, and the construction of highly detailed virtual environments. Our research has concentrated on this last class of applications. In the simplest case, planar mosaics can be used for scanning whiteboards as an aid to videoconferencing or as an easy way to capture ideas. Scanning can produce images of much greater resolution than single wide-angle lens shots; the techniques developed in this paper enable any video camera attached to a computer to be used. Piecewise planar mosaics could also be used to model certain virtual environments, e.g., the aisles at your local supermarket.

Panoramic mosaics can have many applications, including tele-tourism (e.g., looking at the views from the Eiffel Tower or the rim of the Grand Canyon), education (tours of museums), and home sales (views of room interiors). True walkthroughs of existing building or outdoor environments require the solution of a much more difficult problem, i.e., full 3-D reconstruction. They also require the rapid display of very complex scenes, for which view interpolation may be useful.

The ultimate in virtual reality systems is true *telepresence*, which composites video from multiple source in real-time to create the illusion of being in a dynamic (and perhaps reactive) 3-D environment. An example of such an application might be to view a 3-D version of a concert or sporting event with control over the camera shots, even being able to see the event from the players' point of view. Other examples might be to participate or consult in a surgery from a remote location (*telemedicine*), or to remotely participate in a *virtual classroom*.

7 Discussion and Open Questions

The recent interest in projective approaches to visual scene reconstruction and representation appears to be motivated by two main concerns. The first is a desire to avoid camera calibration. The second is a disappointment with the (metric) quality of the results available with Euclidean techniques.

The need for accurate camera calibration depends very much on the task at hand. For example, robot systems that handle or inspect parts benefit greatly from accurate calibration. On the other hand, visual servoing, which does not require precise calibration, is sufficient for robot systems that are capable of hybrid force and position control. Accurate calibration is necessary for veridical scene reconstruction, e.g., for virtual reality environments and games. Of course, computing a projective description first and then converting it to a Euclidean representation later through control points may be a reasonable approach.

We believe that the quality of Euclidean reconstructions must be examined in more detail, since its underlying problems can also plague projective reconstruction techniques. We see four main reasons why reconstruction techniques may not produce reliable results: a poor choice of technique, using an inappropriate representation, using too little data, and fundamental limitations on the achievable accuracies.

Traditionally, structure from motion algorithms have been developed using geometric arguments about point, lines, and planes, followed by a reduction to an algebraic formulation or series of estimation steps. The problem with this approach is that while geometric or algebraic constructs are correct in the noisefree case, there is no guarantee that they will produce reasonable estimates for noisy data. Our approach has been to estimate the unknown structure and motion parameters using a non-linear least-squares minimization of the image plane measurement errors, which is statistically optimal for small

[1] We can enforce this constraint, if desired, by modifying the d_i and $\mathbf{M}_j$ after each iteration to maintain a zero bias in the d_i.

Gaussian noise, and can be made robust against gross errors using robust statistics. Furthermore, this approach provides explicit measures of uncertainty in the estimates, which can be used to great advantage when processing sequences of data.

Carefully choosing the coordinate frame for the structure reconstruction, i.e., using an *object-centered representation*, can dramatically improve the quality of Euclidean reconstruction [16, 15]. This advantage is shared by many projective reconstruction techniques, which often choose the reconstruction plane to be located near the interesting structure. Many structure from motion algorithms are also restricted to using only a few points or frames. Our estimation-theoretic approach encourages the use of as much redundant data as possible, and can easily accommodate missing or noisy estimates.

Finally, it is important to understand that structure from motion, and scene reconstruction in general, are fundamentally limited in accuracy by the quality of the feature tracks, regardless of the choice of algorithm and representation. Therefore, the importance of feature tracker accuracy cannot be overemphasized. Discarding unreliable feature tracks using robust statistics, as is the case in our structure from motion algorithm, will greatly improve the quality of the reconstructions.

A large number of open questions remain in this domain. In terms of efficiency, there is the question of the relative accuracy of recursive vs. batch estimation algorithms. Within this context, a better understanding of the structure of the uncertainty (covariance) in the estimates should improve the quality of recursive algorithms.

Another interesting question is whether the recovery of a projective scene description is a useful intermediate step in the process of recovering Euclidean structure. How reliable is such an approach compared with direct Euclidean estimation? Does it offer significant improvements in terms of speed? What are the limitations on the accuracy of Euclidean reconstructions, and what kind of metric information is most useful when constructing such estimates?

To summarize, we have described our philosophy and our algorithms in the area of scene reconstruction from multiple views. In particular, we believe that the approach to scene reconstruction should be dictated by the task requirements, which is consistent with the notion of task-oriented vision. For example, for scene interpretation tasks where relative depths are used to qualitatively describe the spatial ordering of objects in the scene, recovery of projective depth is adequate. For applications such as virtual reality environment construction, Euclidean (true or scaled) is required. In either case, it is important to understand the nature of structure recovery errors both to optimize the algorithms we use and to understand the fundamental limitations of these techniques.

References

[1] S. Chen and L. Williams. View interpolation for image synthesis. *Computer Graphics (SIGGRAPH'93)*, pages 279–288, August 1993.

[2] O. Faugeras. *Three-dimensional computer vision: A geometric viewpoint*. MIT Press, Cambridge, Massachusetts, 1993.

[3] O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In *Second European Conference on Computer Vision (ECCV'92)*, pages 563–578, Santa Margherita Liguere, Italy, May 1992. Springer-Verlag.

[4] M. Hansen, P. Anandan, K. Dana, G. van der Wal, and P. Burt. Real-time scene stabilization and mosaic construction. In *IEEE Workshop on Applications of Computer Vision (WACV'94)*, pages 54–62, Sarasota, Florida, December 1994.

[5] J. J. Koenderink and A. J. van Doorn. Affine structure from motion. *Journal of the Optical Society of America A*, 8:377–385538, 1991.

[6] C. D. Kuglin and D. C. Hines. The phase correlation image alignment method. In *IEEE 1975 Conference on Cybernetics and Society*, pages 163–165, New York, September 1975.

[7] R. Kumar, P. Anandan, and K. Hanna. Direct recovery of shape from multiple views: A parallax based approach. In *Twelfth International Conference on Pattern Recognition (ICPR'94)*, volume A, pages 685–688, Jerusalem, Israel, October 1994. IEEE Computer Society Press.

[8] S. Mann and R. W. Picard. Virtual bellows: Constructing high-quality images from video. In *First IEEE International Conference on Image Processing (ICIP-94)*, volume I, pages 363–367, Austin, Texas, November 1994.

[9] R. Mohr, L. Veillon, and L. Quan. Relative 3D reconstruction using multiple uncalibrated images. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'93)*, pages 543–548, New York, New York, June 1993.

[10] H. S. Sawhney. 3D geometry from planar parallax. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 929–934, Seattle, Washington, June 1994. IEEE Computer Society.

[11] A. Shashua. Projective depth: A geometric invariant for 3D reconstruction from two perspective/orthographic views and for visual recognition. In *Fourth International Conference on Computer Vision (ICCV'93)*, pages 583–590, Berlin, Germany, May 1993. IEEE Computer Society Press.

[12] A. Shashua and N. Navab. Relative affine structure: Theory and applications to 3D reconstruction from perspective views. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 483–489, Seattle, Washington, June 1994. IEEE Computer Society.

[13] R. Szeliski. Image mosaicing for tele-reality applications. In *IEEE Workshop on Applications of Computer Vision (WACV'94)*, pages 44–53, Sarasota, Florida, December 1994. IEEE Computer Society.

[14] R. Szeliski and J. Coughlan. Hierarchical spline-based image registration. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 194–201, Seattle, Washington, June 1994. IEEE Computer Society.

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

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

[17] J. Y. A. Wang and E. H. Adelson. Layered representation for motion analysis. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'93)*, pages 361–366, New York, New York, June 1993.