10^{th} British Machine Vision Conference (BMVC'99)

Stereo Algorithms and Representations for Image-Based Rendering

Richard Szeliski
Vision Technology Group
Microsoft Research
One Microsoft Way
Redmond, WA 98052-6399

Abstract

This paper reviews a number of recently developed stereo matching algorithms and representations. It focuses on techniques that are especially well suited for *image-based rendering* applications such as novel view generation and the mixing of live imagery with synthetic computer graphics. The paper begins by reviewing some recent approaches to the classic problem of recovering a depth map from two or more images. It then describes a number of newer representations (and their associated reconstruction algorithms), including volumetric representations, layered plane-plus-parallax representations, and multiple depth maps. Each of these techniques has its own strengths and weaknesses, which are discussed.

1 Introduction

Stereo matching, which is one of the oldest problems in computer vision, now appears to be a maturing research area. Real-time stereo matching, which a few years ago required special-purpose hardware [16, 20], is now implementable on regular personal computers (see [20] for some references). Depth maps computed with such systems can now be used as basic building blocks for higher level processes such as background subtraction and tracking.

But is stereo really a solved problem? Consider, for example, one of the more recent applications of real-time stereo matching: the ability to composite live video with synthetic computer graphics using the process of *z-keying* [16]. Or, consider the ability to film or photograph a scene or activity from multiple views, and to then look at the same scene from novel viewpoints, i.e., *virtualized reality* [18]. These applications are certainly very exciting, but is the quality of existing algorithms adequate for their use in real production environments?

Judging from the results in recent papers, it appears that we are not there yet. The reconstructions produced by today's algorithms still often leave a "halo" of background pixels clinging to the foreground object. Furthermore, even if a stereo algorithm were to assign a correct depth to each pixel in an image, it would still fail to correctly handle *mixed pixels*, i.e., pixels whose color is a combination of foreground and background colors (which occur at nearly all pixels along a depth discontinuity). What we really need are algorithms where a foreground layer can be pulled or *matted* from the background, based on the results of a stereo reconstruction algorithm.

Applications such as z-keying and virtualized reality are related to a recent trend in computer graphics, which is called *image-based rendering* [19]. While some of image-based rendering is concerned with re-using synthetically generated images to accelerate rendering speeds, a lot of recent work has focused on acquiring scene or object models from multiple images and re-synthesizing novel views from the original images [23, 14]. When stereo matching is used in such applications, there are several demanding requirements that are not present in more traditional robotics applications of stereo.

First of all, the stereo algorithm must be able to assign correct (or at least reasonable) depths at *all* pixels, especially those near depth discontinuities. In Section 2, I discuss how some recent stereo algorithms are able to avoid the systematic "fattening" of layers associated with traditional area-based methods. A second requirement is the ability to pull mattes, i.e., to separate foreground and background elements while correctly describing the true colors of individual pixels. A third requirement is to generate novel views with as few gaps (missing pixels) as possible, and to also account for partially occluded regions during the matching process. The single depth map representation used in Section 2 is inadequate on all of these counts.

Section 3 describes how a *volumetric* representation of space, combined with real-valued opacities, can be used to overcome most of these problems. Section 4 describes a different, more compact, representation based on arrangements of colored quasi-planar cutouts, which can also overcome these problems. Section 5 describes how to use multiple depth maps (and associated images) to solve the problem of partially occluded areas, and how this representation can also serve as a preliminary step towards a more complete reconstruction of the scene. I conclude this paper with a comparison of the approaches presented and some prospects for further progress in this field.

2 Depth maps

The classical problem of computing a dense depth map from two or more images has been extensively studied. Some good (although slightly dated) surveys of the field can be found in [3, 11, 8]. In this section, we first present a formulation for this problem, and then discuss several recently developed algorithm that attempt to accurately solve for depth near discontinuities.

2.1 Generalized disparity space

Assume we are given as input a collection of K images, $I_1(x,y), I_2(x,y), \ldots, I_K(x,y)$, captured by K cameras with known projection (camera) matrices, $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_K$. To formulate the multiframe stereo problem, we use a *generalized disparity space*, which can be any projective sampling (collineation) of 3-D space [9, 34]. This space is a generalization of the notion of *disparity space* [41, 15, 28], i.e., the enumeration of all possible disparities at every pixel. The goal of stereo matching is then to find the elements in disparity space which lie on the surfaces of the objects in the scene. The benefits of such an approach include the equal and efficient treatment of a large number of images [9] and the possibility of modeling occlusions [15].

To formulate the generalized disparity space, we first choose a *virtual camera* position and orientation. This virtual camera may be coincident with one of the input images, or it can be chosen based on the application demands and the desired accuracy of the results. For instance, if we wish to regularly sample a volume of 3-D space, we can make the camera orthographic, with the camera's (x, y, d) axes being orthogonal and evenly sampled (as in [29]).

Having chosen a virtual camera position, we then choose the orientation and spacing of the *disparity planes*, i.e., the constant d planes. The relationship between d and 3-D space can be projective. For example, we can choose d to be inversely proportional to depth, which is the usual meaning of disparity [24]. The information about the virtual camera's position and disparity plane orientation and spacing can be captured in a single 4×4 matrix $\hat{\mathbf{M}}_0$, which represents a collineation of 3-D space, $w(x\ y\ d\ 1)^T = \hat{\mathbf{M}}_0(X\ Y\ Z\ 1)^T$.

2.2 Area-based approaches

Having presented the representation used for describing the output of the matching algorithm, we can now state its goal: For each (x, y) location in disparity space, find the disparity d that aligns corresponding locations in the input images (ignoring, for now, the possibility that pixels may be occluded). In traditional area-based correlation, the quality of a match is measured by comparing windows centered at corresponding locations, for example, using the sum of squared intensity differences (SSD) [17].

A more general way of characterizing area-based algorithms is the following [28]:

1. For each disparity under consideration, compute a per-pixel matching cost, e.g., squared intensity difference or variance of colors across the k input images.

2. Aggregate support spatially (e.g., by summing over a window, or by diffusion).

3. At each pixel (x, y), find the best matching disparity d based on the aggregated support.

4. Compute a sub-pixel disparity estimate (optional).

Let us look at the components in this framework in more detail.

At the base of any matching algorithm is a matching cost that measures the similarity of corresponding locations. Matching costs can be defined locally (at pixel level), or over a certain area of support. Examples of local costs are absolute intensity differences, squared intensity differences, binary pixel matches, edges, filtered images, and measures based on gradient direction or gradient vectors. Matching costs that are defined over a certain area of support include correlation and non-parametric measures. These can be viewed as a combination of the matching cost and aggregation stages. More than two images are used in multiframe stereo to increase stability of the algorithm [24].

Aggregating support is necessary for stable matching. A support region can either be two-dimensional at a fixed disparity (favoring fronto-parallel surfaces), or three-dimensional in x-y-d space (supporting slanted surfaces). Two-dimensional evidence aggregation has been done using square windows (traditional), Gaussian convolution, multiple windows anchored at different points [15], and windows with adaptive sizes [17]. Three-dimensional support functions that have been proposed include limited disparity difference, limited disparity gradient [25], and Prazdny's coherence principle [26].

Sub-pixel disparity estimates can be computed by fitting a curve to the matching costs at the discrete disparity levels [22, 17]. This provides an easy way to increase the resolution of a stereo algorithm with little additional computation. However, to work well, the intensities being matched must vary smoothly. A related refinement is to compensate for discrete sampling of disparity space by linearly interpolating errors computed at adjacent disparity levels, and analytically finding the minimum matching error in this interval [4].

2.3 Global optimization approaches

Optimization (regularization) approaches start with the same computation of matching costs as area-based techniques, but then add a controlled smoothness penalty (prior) on the disparity field $d(x, y)$. A variety of optimization algorithms can then be used to find a good solution to this problem [2, 28, 7].

An elegant mathematical approach to formulating these energy function and finding their minimum is to use a Bayesian (probabilistic) estimation framework. The Bayesian model of stereo image formation consists of two parts. The first part, a *prior model* for the disparity surface, uses a traditional Markov Random Field (MRF) to encode preferences for smooth surfaces [13]. This model is specified as a Gibbs distribution p_P, the exponential of a potential function E_P:

$$p_P(\mathbf{d}) = \frac{1}{Z_P} \exp\left(-E_P(\mathbf{d})\right), \tag{1}$$

where $\mathbf{d}$ is the vector of all disparities $d(x, y)$ and Z_P is a normalizing factor. The potential function itself is the sum of clique potentials that only involve neighboring sites in the field. The simplest such field is a first order field, where

$$E_P(\mathbf{d}) = \sum_{i,j} \rho_P(d(x+1, y) - d(x, y)) + \rho_P(d(x, y+1) - d(x, y)) \tag{2}$$

(see [38, 31] for generalizations to higher order fields).

When $\rho(x)$ is a quadratic, $\rho(x) = x^2$, the field is a Gauss-MRF, and corresponds in a probabilistic sense to a first order regularized (*membrane*) surface model [38, 31]. When $\rho(x)$ is a unit impulse, $\rho(x) = 1 - \delta(x)$, it corresponds to a MRF that favors fronto-parallel surfaces [13]. In between these two extremes are functions derived from *robust statistics*, which behave much like surface models with discontinuities [5].

The second part of a Bayesian model is the *data* or *measurement model* which accounts for differences in intensities between corresponding image locations. This model assumes independent, identically distributed measurement errors,

$$p_M(I_1, \ldots, I_K | \mathbf{d}) = \prod_{i,j} p_M(x, y, d)), \tag{3}$$

where $\log p_M(x, y, d) = \rho_M(x, y, d)$ is the initial, unaggregated, matching cost. Traditional stereo matching methods use either a squared intensity error metric (Gaussian noise), or an exact binary matching criterion (e.g., for random-dot stereograms or binary features such as edges or the sign of the Laplacian). A more general model is a contaminated Gaussian model, which models both Gaussian noise and allows possible outliers due to occlusions or non-modeled photometric effects such as specularities.

The posterior distribution, $p(\mathbf{d}|I_1, \ldots, I_K)$ can be derived from the prior and measurement models using Bayes' rule,

$$p(\mathbf{d}|I_1, \ldots, I_K) \propto p_P(\mathbf{d}) p_M(I_1, \ldots, I_K|\mathbf{d}). \tag{4}$$

As is often the case, it is more convenient to study the negative log probability distribution

$$\begin{aligned} E(\mathbf{d}) &= -\log p(\mathbf{d}|I_1, \ldots, I_K) && (5) \\ &= \sum_{i,j} \rho_P(d_{i+1,j} - d_{i,j}) + \rho_P(d_{i,j+1} - d_{i,j}) + \sum_{i,j} \rho_M(x_i, y_j, d_{i,j}). \end{aligned}$$

While $p(\mathbf{d}|I_1, \ldots, I_K)$ specifies a complete distribution, usually only a single optimal estimate of $d(x, y)$ is desired ([31] explains why modeling of uncertainties may be useful). The most commonly studied estimate is the peak of the distribution, or *Maximum A Posteriori* (MAP) estimate, which is equivalent to minimizing the energy given in (5).

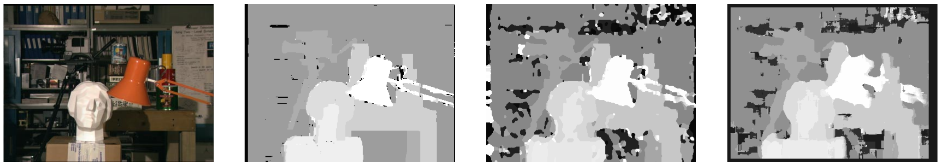
A variety of techniques have been developed for minimizing equations like (5). Two of the most popular are the Gibbs Sampler [13] and mean field theory [12]. The Gibbs Sampler randomly chooses values for each $d_{i,j}$ site according to the local distribution determined by the current guesses for a site's neighbors [13, 2]. This process will in theory converge to a statistically optimal sample, given enough time. Mean field theory updates an estimate of the *mean* value of $d_{i,j}$ at each site using a deterministic update rule derived from the original probability distribution [13].

The Gibbs Sampler and its variants can produce good solutions, but at the cost of long computation times. Mean field techniques, on the other hand, are not very good at modeling ambiguous estimates, such as multiple potential matches at each pixel. Instead of using either of these two traditional approaches, we developed a novel estimation algorithm based on modeling the probability distribution of $d(x, y)$ at each site [28] . To do this, we associate a scalar value between 0 and 1 with each possible discrete value of d at each pixel (x, y), and require that the probabilities sum up to 1. This representation is therefore the same as that used by aggregation-based algorithms, i.e., it explicitly models all possible disparities at each pixel, rather than modeling a single estimated disparity as in traditional Gibbs Sampler or mean-field approaches [2].

The algorithm is initialized by calculating the probability distribution for each pixel (x, y) based on the intensity errors between matching pixels, i.e., using the measurement model (3). To derive the update formula, we approximate the true Markov Random Field distribution with a *factored* approximation, i.e., we assume that the neighboring disparity columns have independent distributions. Minimizing the Kullback-Leibler divergence between the true posterior Gibbs distribution and its factored (mean-field) approximation leads to a set of update formulas on the probability distributions that use non-linear *diffusion* (see [28] for details).

The results of running this algorithm on difficult stereo pairs are quite promising. The algorithm is particularly good at correctly matching pixels near depth discontinuities, since the robust smoothness constraint can be violated at the appropriate places, and also at stereograms that have a lot of potential matches, such as random-dot stereograms.

Another recent development in optimization-based stereo matching is the use of graph algorithms [27, 7]. Here, techniques from discrete optimization are used to find good minima (in some cases, even global minima) of the global energy function (5). These algorithms have both good discontinuity localization, since they are based on robust smoothness



Input image Graph cuts Bayesian diffusion SAD (13×13)

Figure 1: Results for several stereo algorithms on the University of Tsukuba imagery.

models, and also excel at filling in good disparities inside uniform color/intensity regions (since they take large steps in state space).

Figure 1 shows the results of applying three different matching algorithms on a multi-camera stereo data set provided by the University of Tsukuba. You can readily see that the two energy minimization-based approaches (Graph cuts and Bayesian) have much crisper depth discontinuities, compared with the sum of absolute differences (SAD) technique. The graph-cut approach also does an excellent job of filling in uniform intensity areas. These results are part of a larger evaluation of two-frame stereo matching algorithms that we are currently undertaking [37].

The original color image can be *texture-mapped* onto the surface defined by the depth map to produce novel views [22]. Unfortunately, the depth map behaves as a "rubber sheet", i.e., background regions that are not visible in the reference image are not correctly synthesized. As more and more images are used in stereo matching, this effect become even more pronounced. For this reason, we now turn our attention to algorithm that explicitly represent and reason about partially occluded regions.

3 Volumetric representations

The depth map representation presented in the previous section is unable to represent and hence render partially occluded background regions. This is due to our insistence (enforced during the winner-take-all stage) that only a single depth value be assigned to each pixel in the reference image.

What if we were to relax this assumption? What if in addition to being able to have several depth along each ray in the reference image, we also represented the colors of these pixels and their (potentially partial) opacities? In principle, we should be able to represent and reason about partially occluded pixels, and to correctly estimate the color values of mixed pixels. These are the intuitions that led to the development of the volumetric stereo reconstruction algorithm presented in [34]. (Simultaneously with our work, Seitz and Dyer [29] developed a volumetric stereo algorithm that uses binary (filled/empty) opacities and a front-to-back plane sweep (*voxel coloring*) algorithm. DeBonet and Viola also have a volumetric reconstruction technique that estimates partial opacities [10].)

The algorithm starts by performing the same matching cost computation, aggregation, and winning depth value selection as described in the previous section. However, instead of insisting that every pixel in the reference image pick a winning depth, we only select depth values that have a good match (good aggregated evidence), using a threshold to mark other pixels as currently "unassigned". These pixels will typically not have correspondences either because they are partially occluded, or because they are mixed pixels with different backgrounds in different images. A new (x, y, d) volume can now be created, where each cell now contains a color value, initially set to the mean color computed in the first stage, and the opacity is set to 1 for cells which are winners, and 0 otherwise.

Once we have an initial (x, y, d) volume containing estimated RGBA (color and 0/1 opacity) values, we can re-project this volume into each of the input cameras using the known transformation

$$\mathbf{x}_k = \mathbf{P}_k \hat{\mathbf{M}}_0^{-1} \hat{\mathbf{x}}_0 \tag{6}$$

where $\hat{\mathbf{x}}_0$ is a (homogeneous) coordinate in (x, y, d) space, $\hat{\mathbf{M}}_0$ is the complete camera matrix corresponding to the virtual camera, $\mathbf{P}_k$ is the kth camera matrix, and $\mathbf{x}_k$ are the image coordinates in the kth image. In our approach, we interpret the (x, y, d) volume as a set of (potentially) transparent acetates stacked at different d levels. Each acetate is first warped into a given input camera's frame using the known homography $\mathbf{H}_k$. Once the layers have been resampled, they are then composited using the standard *over* operator [6].

After the re-projection step, we refine the disparity estimates by preventing visible surface pixels from voting for potential disparities in the regions they occlude. More precisely, we build an (x, y, d, k) *visibility map*, which indicates whether a given camera k can see a voxel at location (x, y, d) [34].

Once we have computed the visibility volumes for each input camera, we can update the list of color samples we originally used to get our initial disparity estimates to obtain a distribution of colors in (x, y, d, k) where each color has an associated visibility value. Voxels that are occluded by surfaces lying in front in a given view k will now have fewer (or potentially no) votes in their local color distributions. We can therefore recompute the local mean and variance estimates using weighted statistics, where the visibilities $V(x, y, d, k)$ provide the weights.

With these new statistics, we are now in position to refine the disparity map. In particular, voxels in disparity space that previously had an inconsistent set of color votes (large variance) may now have a consistent set of votes, because voxels in (partially occluded) regions will now only receive votes from input pixels that are not already assigned to nearer surfaces.

While the above process of computing visibilities and refining disparity estimates will in general lead to a higher quality disparity map (and better quality mean colors, i.e., texture maps), it will not recover the true colors and transparencies in *mixed pixels*, e.g., near depth discontinuities, which is one of the main goals of this research.

In the second phase of our algorithm, we adjust the opacity and color values $\hat{\mathbf{c}}(x, y, d)$ to match the input images (after re-projection), while favoring continuity in the color and opacity values. This can be formulated as a non-linear minimization problem, where the cost function has three parts: a weighted error norm on the difference between the re-projected images $\tilde{\mathbf{c}}_k(u, v)$ and the original input images $\mathbf{c}_k(u, v)$; a (weak) smoothness constraint on the colors and opacities; and a prior distribution on the opacities [34]. To minimize the total cost function, we use a preconditioned gradient descent algorithm. A complete description of this procedure is given in [34].

Figure 2 shows the results of this algorithm when run on a cropped portion of the *SRI Trees* multibaseline stereo dataset. A small region (64×64 pixels) was selected in order to better visualize pixel-level errors. While the overall reconstruction is somewhat noisy, the final reconstruction with a synthetic blue layer inserted shows that the algorithm has

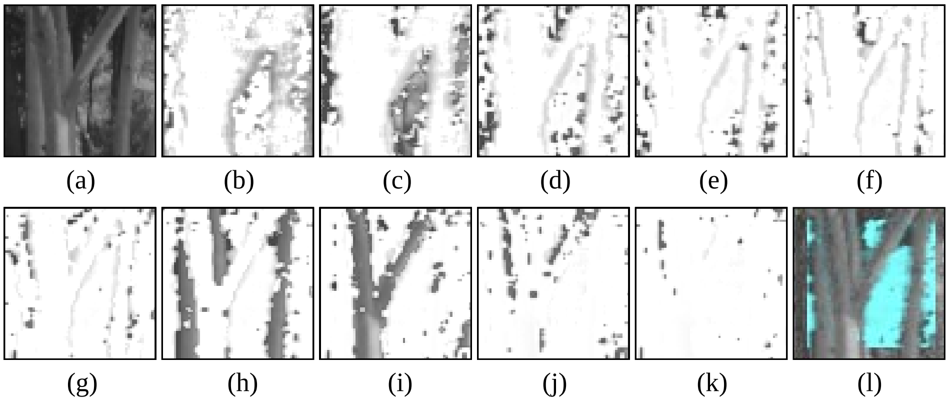


Figure 2: Volumetric reconstruction example: (a) cropped subimage from *SRI Trees* data set, (b–k) disparity layers $d = 0 \ldots 9$, (l) re-synthesized input image with inserted $d = 4$ blue layer.

done a reasonable job of assigning pixel depths and computing partial transparencies near the tree boundaries.

From this example, we see that the volumetric approach is a much more powerful representation for dealing with partially occluded regions and mixed pixels. Unfortunately, this power comes at the expense of two problems: the depth are quantized, which can lead to aliasing effects, and the representation has a very large number of degrees of freedom, which makes it difficult to find the optimal solution. The first problem could be fixed, in principle, by using fractional disparities (although these would have to be relative to one preferred camera). The second problem we address in the next section, where a much more parsimonious description is used.

4 Layered representations

To overcome the problem with the volumetric representation, we draw some inspiration from recent work in layered motion estimation [40]. Here, the goal is to decompose the images into sub-images, commonly referred to as *layers*, such that the pixels within each layer move in a manner consistent with a parametric transformation. The motion of each layer is determined by the values of the parameters. An important transformation is the 8–parameter homography (collineation), because it describes the motion of a rigid planar patch as either it or the camera moves.

While existing techniques have been successful in detecting multiple independent motions, layer extraction for scene modeling has not been fully developed. One fact that has not been exploited is that, when simultaneously imaged by several cameras, each of the layers implicitly lies on a fixed plane in the 3D world. Another omission is the proper treatment of transparency. With a few exceptions, the decomposition of an image into layers that are partially transparent has not been attempted. In contrast, scene modeling using multiple partially transparent layers is common in the graphics community [6].

In our own work [1], we have developed a framework for reconstructing a scene as a collection of approximately planar layers. Each of the layers has an explicit 3D plane equation and is recovered as a *sprite*, i.e. a colored image with per-pixel opacity (transparency) [6]. To model a wider range of scenes, a per-pixel depth offset relative to the plane is also added.

Our layered approach to stereo shares many of the advantages of the volumetric approach. In addition, it offers a number of other advantages:

- The combination of the global model (the plane) with the local correction to it (the per-pixel depth offset) results in very robust performance. In this respect, the framework is similar to the *plane + parallax* work of [21].
- The output (a collection of approximately planar regions) is more suitable than a discrete collection of voxels for many applications, including, rendering [30] and video parsing.

Our representation consists of a collection of L approximately planar layers, each of which is an alpha-matted color image (layer sprite) $L_l(x, y)$ with *pre-multiplied opacities* [6]. We also associate a homogeneous vector $\mathbf{n}_l$ with each layer (which defines the plane equation of the layer via $\mathbf{n}_l^T \mathbf{x} = 0$) and a per-pixel residual depth offset $Z_l(x, y)$.

The goal of our layer decomposition algorithm is to estimate these quantities. To do so, we wish to use techniques for parametric motion estimation. Unfortunately, most such techniques assume boolean-valued opacities α_l (i.e., unique layer assignments). We therefore split our framework into two parts. In the first part, we assume boolean opacities to get a first approximation to the structure of the scene. If the opacities are boolean, each point in each image I_k is only the image of a point on one of the layers L_l. We therefore introduce boolean masks B_{kl} which denote the pixels in image I_k that are images of points on layer L_l. So, in addition to L_l, $\mathbf{n}_l$, and Z_l, we also need to estimate the masks B_{kl}. Once we have estimates of the masks, we immediately compute masked input images $M_{kl} = B_{kl} \cdot I_k$. In the second part of our framework, we use the initial estimates of the layers made by the first part as input into a re-synthesis algorithm which refines the layer sprites L_l, *including* the opacities α_l. This second step requires a generative or forward model of the image formation process.

Before we can compute the layer sprites L_l, we need to choose 2D coordinate systems for the sprite images. Such coordinate systems can be specified by a collection of arbitrary (rank 3) camera matrices $\mathbf{Q}_l$. In [1] we show that the image coordinates $\mathbf{x}_k$ of the pixel in image M_{kl} that is projected onto the pixel $\mathbf{x}_l$ on the plane $\mathbf{n}_l^T \mathbf{x} = 0$ is given by

$$\mathbf{x}_k \;=\; \mathbf{P}_k \left((\mathbf{n}_l^T \mathbf{q}_l)\mathbf{I} - \mathbf{q}_l \mathbf{n}_l^T \right) \mathbf{Q}_l^* \mathbf{x}_l \;\equiv\; \mathbf{H}_k^l \mathbf{x}_l, \tag{7}$$

where $\mathbf{Q}_l^*$ is the pseudo-inverse of $\mathbf{Q}_l$, and $\mathbf{q}_l$ is a vector in the null space of $\mathbf{Q}_l$. The homography $\mathbf{H}_k^l$ can be used to warp the image M_{kl} forward onto the coordinate frame of the plane $\mathbf{n}_l^T \mathbf{x} = 0$, the result of which is denoted $\mathbf{H}_k^l \circ M_{kl}$. Then, we can estimate the layer sprite (with boolean opacities) by *blending* the warped images [35].

To compute the homographies $\mathbf{H}_k^l$ that align all masked image pieces M_{kl} into a consistent coordinate frame, we use a previously developed parametric motion (*mosaicing*) technique [35]. Once we have an initial estimate for the $\mathbf{H}_k^l$, we use a structure-from-motion algorithm to compute the plane equations (and, in the case where the original camera matrices $\mathbf{P}_k$ are unknown, to estimate them as well) [36]. A better approach might be to directly optimize over the plane normal $\mathbf{n}_l$ used in (7).

Since in general, the scene will not be piecewise planar, we allow the point $\mathbf{x}_l$ on the plane $\mathbf{n}_l^T \mathbf{x} = 0$ to be displaced slightly. We assume it is displaced in the direction of the ray through $\mathbf{x}_l$ defined by the camera matrix $\mathbf{Q}_l$. The distance it is displaced is denoted by $Z_l(\mathbf{x}_l)$, as measured in the direction *normal* to the plane. In this case, the homographic warps used in the previous section are not applicable, but using a similar argument, it is possible to show that

$$\mathbf{x}_k \;=\; \mathbf{H}_k^l \mathbf{x}_l + w(\mathbf{x}_l) Z_l(\mathbf{x}_l) \mathbf{t}_{kl}, \tag{8}$$

where $\mathbf{t}_{kl} = \mathbf{P}_k \mathbf{q}_l$ is the epipole and it is assumed that the vector $\mathbf{n}_l = (n_x, n_y, n_z, n_d)^T$ has been normalized such that $n_x^2 + n_y^2 + n_z^2 = 1$. The term $w(\mathbf{x}_l)$ is a projective scaling factor which equals the reciprocal of $\mathbf{Q}_l^3 \mathbf{x}$, where $\mathbf{Q}_l^3$ is the third row of $\mathbf{Q}_l$ and $\mathbf{x}$ is the world coordinate of the point. Equation (8) can be used to map plane coordinates $\mathbf{x}_l$ backwards to image coordinates $\mathbf{x}_k$, or to map the image M_{kl} forwards onto the plane. We denote the result of this warp by $(\mathbf{H}_k^l, \mathbf{t}_{kl}, Z_l) \circ M_{kl}$, or more concisely $\mathbf{W}_k^l \circ M_{kl}$.

Almost any stereo algorithm can be used to compute $Z_l(\mathbf{x}_l)$, although it is preferable to use one favoring small disparities. Currently, we use a simple plane-sweep algorithm, a simplified version of the algorithm described in [28]. Once the residual depth offsets have been estimated, the layer sprite images can be re-estimated by blending the warped images $\mathbf{W}_k^l \circ M_{kl}$.

To re-compute the pixel assignments, we compare the warped images $\mathbf{W}_k^l \circ M_{kl}$ with the layer sprites L_l. If the pixel assignment was correct (and neglecting resampling issues) these images should be identical where they overlap. Details of the heuristics used in re-computing the layer assignments are given in [1].

In [1], we also describe how the estimates of the layer sprites can be refined, now assuming that their opacities α_l are real-valued. We begin by formulating a generative model of the image formation process. Afterwards, we propose a measure of how well the layers re-synthesize the input images, and show how the re-synthesis error can be minimized to refine the estimates of the layer sprites. This approach is similar to the one we developed for the volumetric model with transparent voxels [34]. We are currently implementing this portion of our algorithm. Once it is complete, we are hoping to have layer descriptions that will correctly account for mixed pixels, and may even be able to reconstruct scenes with translucent surfaces such as dirty windows or scenes with additive phenomena such as reflections.

Figure 3 shows some results of applying our algorithm to five images from a 40-image stereo data set taken at a graphics symposium. Figure 3(a) shows the middle input image, Figure 3(b) shows the initial pixel assignment to layers, Figure 3(c) shows the recovered planar depth map, and Figure 3(f) shows the residual depth map for one of the layers. Figures 3(d) and (e) show the recovered sprites. Figure 3(g) shows the middle image re-synthesized from these sprites. Finally, Figures 3(h–i) show the same sprite collection seen from a novel viewpoint (well outside the range of the original views), first with and then without residual depth correction. The gaps in Figure 3 correspond to parts of the scene that were not visible in any of the five input images.

To summarize, the layered approach to 3D reconstruction represents the scene as a collection of approximately planar layers. Each layer consists of a plane equation, a layer sprite image, and a residual depth map. The framework exploits the fact that each layer implicitly lies on a fixed plane in the 3D world. This is both the algorithm's strength (using a compact description) and its weakness (it is limited to scenes where objects are "cutouts with relief"). The layered approach also requires solving a combinatorial optimization problem, since the number of layers needs to be determined, as well as figuring out the assignment of pixels to layers [39].

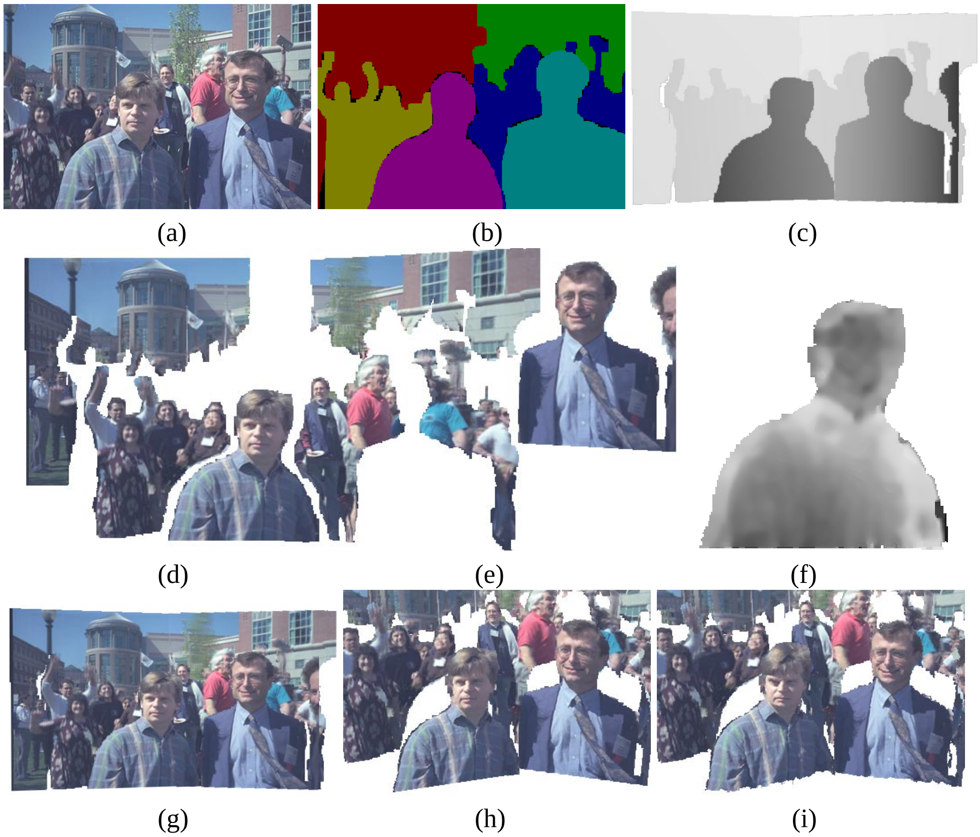


Figure 3: Layered stereo results: (a) third of five images; (b) initial segmentation into six layers; (c) recovered depth map (darker denotes closer); (d) and (e) the five layer sprites; (f) residual depth image for fifth layer. (g) re-synthesized third image (note extended field of view). (h) novel view without residual depth; (i) novel view with residual depth (note the “rounding” of the people).

5 Multiple depth maps

In our most recent work, we have been investigating an alternative to volumetric and layered representations that can also represent and reason about semi-occluded regions. Rather than estimating a single depth map, we associate a depth map with *each* input image (or some subset of them) [32]. Furthermore, we try to ensure consistency between these different estimates using a *depth compatibility* constraint, and reason about occlusion relationships by computing pixel *visibilities*. Our representation can be used as is for image-based rendering (view interpolation) applications, or it can be used as a low-level representation from which segmentation and layer extraction (or 3D model construction) can take place.

To formulate the multi-view stereo problem, we take the matching costs for all reference images and sum them together. This *brightness compatibility* term, which measures the degree of agreement in brightness or color between corresponding pixels, can be written

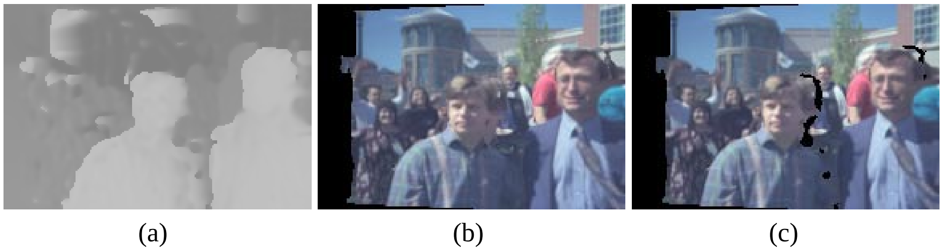


Figure 4: Results of multi-view stereo algorithm: (a) depth estimate for first frame; (b) warped (resampled) images without visibilities; (c) with visibility computation.

as

$$\mathcal{C}(\{\mathbf{x}_s\}) = \sum_{s \in S} \sum_{t \in \mathcal{N}(s)} w_{st} \sum_{\mathbf{x}_s} v_{st}(\mathbf{x}_s) \rho \left(I_s(\mathbf{x}_s) - I_t(\mathbf{x}_t) \right). \tag{9}$$

The images I_s form the set S of *keyframes* (or *key-views*) for which we will estimate a depth estimate $d_s(\mathbf{x}_s)$. The decision as to which images are keyframes is problem-dependent, much like the selection of I and P frames in video compression. For 3D view interpolation, one possible choice of keyframes would be a collection of *characteristic views*.

Images $I_t, t \in \mathcal{N}(s)$ are *neighboring frames* (or views), for which we require that corresponding pixel brightnesses (or colors) agree. The pixel coordinate $\mathbf{x}_t$ corresponding to a given keyframe pixel $\mathbf{x}_s$ with depth d_s can be computed according to the rigid motion model (6). The constants w_{st} are the *inter-frame weights* which dictate how much neighboring frame t will contribute to the estimate of d_s. Corresponding pixel brightness or color differences are passed through a robust penalty function ρ. The visibility factor $v_{st}(\mathbf{x}_s)$, which encodes whether pixel $\mathbf{x}_s$ is *visible* in image I_t, can be computed by comparing corresponding depth values, i.e., checking whether $d_t(\mathbf{x}_t) \leq d_s(\mathbf{x}_s)$.

The cost function used in [32] consists of two additional terms. The controlled *depth compatibility* constraint, enforces *mutual consistency* between depth estimates at different neighboring keyframes. The controlled *depth smoothness* constraint, encourages the depth maps to be piecewise smooth. The shape of this robust penalty function is affected by the brightness/color difference between neighboring pixels (see [32] for details).

Our algorithm operates in two phases. During an initialization phase, we estimate the depths independently for each keyframe. Since we do not yet have any good motion estimates for other frames, the depth compatibility term $\mathcal{C}_T$ is ignored, and no visibilities are computed (i.e., $v_{st} = 1$). In the second phase, we enforce depth compatibility and compute visibilities based on the current collection of depth estimates $\{d_s\}$. Details on the optimization algorithm can be found in [32].

Figure 4 shows some representative results from running our algorithm. The depth map estimated by the algorithm is shown in Figure 4a. Figure 4b shows the results of warping the last image to the first image, based on the depth computed in the first image. Displaying these warped images as the algorithm progresses is a very useful way to debug the algorithm and to assess the quality of the motion estimates. Figure 4c shows the same warped image with invisible pixels flagged as black. Notice how the algorithm correctly labels most of the occluded pixels to the right of the two people's heads.

The experimental results we have obtained so far are encouraging, but still leave room for improvement. In particular, the smoothness of the final estimates and the *sharpness* of the motion discontinuities is not as high as that obtainable with layered models [1]. This is particularly true in occluded regions: layered models will apply the layer's motion to the occluded regions, while we use a weak smoothness constraint.

The multi-view stereo matching framework described in this section produces estimates for a subset of the input images, thereby representing depth in partially occluded regions and explicitly modeling the variation in appearance between different views. Compared with the volumetric and layered representations, the multiple depth map representation is potentially not as compact (although it can be more compact than the search space of the volumetric technique), nor does it correctly model mixed pixels (because the concept of opacity is not built in). It also does not ensure that corresponding surface elements in different views have the same 3D location, although it attempts to ensure this with the weak compatibility constraint. The representation does, however, capture the variation in appearance between different view, for example, when there are strong illumination effects. The representation is useful for performing image-based rendering tasks such as novel view generation, and can also be used to *bootstrap* a more parsimonious representation such as 3D layers.

6 Discussion and Conclusions

In this paper, I have presented a number of representation and algorithms for reconstructing 3D scenes and objects using stereo matching techniques. My emphasis has been on techniques that are well suited to image-based rendering, i.e., approaches that can re-synthesize observed and novel views with a high degree of realism.

The desire to predict the performance of these approaches in image-based rendering applications has led me to propose a new quality metric for stereo matching. Instead of measuring deviations from ground truth depth maps (which are generally hard to come by), I suggest measuring how well the representation predicts *novel* views, i.e., images in a calibrated multi-image stereo data set that have intentionally been held back from the matcher [33] (this is similar to the statistical method of cross-validation). Such data sets are relatively easy to acquire, e.g., by taking a video of a rigid scene and applying a tracking and structure from motion algorithm to recover the camera positions.

Ramin Zabih and I are also currently performing a comparative evaluation of two-frame stereo matching algorithms [37]. While this study excludes some of the novel representations presented in this paper, we hope that it will shed light on underlying principles that make stereo matching work better.

To summarize, I have presented four different representations for stereo matching. A single depth map, the traditional representation used for matching, is a very compact and useful representation that can yield good results when the amount of occlusion is not large, i.e., when the surface is smoothly varying (e.g., a human face) and the range of viewpoints is limited. The volumetric representation (with partial opacities) can be used to represent and reason about partially occluded regions and mixed pixels. Unfortunately, it also has many degrees of freedom, which makes it tricky to find the best reconstruction. Layered representations have the same advantages as volumetric ones, and are potentially more compact, and hence easier to recover. However, determining the best number of planes and the correct pixel assignment is a tricky problem, which we are currently trying to solve. These representations are also inherently limited to scenes that are well approximated by a collection of embossed cutouts. Finally, multiple depth maps can be used to obtain some of the same advantages with respect to partially occluded regions, and also to model the variation in appearance between viewpoints. Unfortunately, they are not guaranteed to have consistent representations of 3D shape, and also do not correctly predict the appearance of mixed pixels.

Thus, we see that all of the representations suggested so far have their limitations. Still, a tremendous amount of progress has been made in recent years in obtaining better and better stereo reconstructions, especially for image-based rendering applications where recovering the true shape of a scene is not paramount. I expect that by re-visiting issues in representation, e.g., by more closely studying the role of discontinuities in shape and depth representations, we will be able to make even further progress, and thereby expand the utility and applicability of stereo-based reconstruction techniques.

References

[1] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *IEEE Comp. Soc. Conf. on Comp. Vision and Patt. Recog. (CVPR'98)*, pp. 434–441, June 1998.

[2] S. T. Barnard. Stochastic stereo matching over scale. *Intl. J. Comp. Vision*, 3(1):17–32, 1989.

[3] S. T. Barnard and M. A. Fischler. Computational stereo. *Computing Surveys*, 14(4):553–572, December 1982.

[4] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Trans. on Patt. Analysis and Machine Intelligence*, 20(4):401–406, April 1998.

[5] M. J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Intl. J. Comp. Vision*, 19(1):57–91, 1996.

[6] J. F. Blinn. Jim Blinn's corner: Compositing, part 1: Theory. *IEEE Comp. Graphics and Applications*, 14(5):83–87, September 1994.

[7] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *Seventh Intl. Conf. on Comp. Vision (ICCV'99)*, September 1999.

[8] L. G. Brown. A survey of image registration techniques. *Computing Surveys*, 24(4):325–376, December 1992.

[9] R. T. Collins. A space-sweep approach to true multi-image matching. In *IEEE Comp. Soc. Conf. on Comp. Vision and Patt. Recog. (CVPR'96)*, pp. 358–363, June 1996.

[10] J. S. De Bonet and P. Viola. Poxels: Probabilistic voxelized volume reconstruction. In *Seventh Intl. Conf. on Comp. Vision (ICCV'99)*, September 1999.

[11] U. R. Dhond and J. K. Aggarwal. Structure from stereo—a review. *IEEE Trans. on Systems, Man, and Cybernetics*, 19(6):1489–1510, November/December 1989.

[12] D. Geiger and F. Girosi. Parallel and deterministic algorithms for MRF's: Surface reconstruction. *IEEE Trans. on Patt. Analysis and Machine Intelligence*, 13(5):401–412, May 1991.

[13] S. Geman and D. Geman. Stochastic relaxation, Gibbs distribution, and the Bayesian restoration of images. *IEEE Trans. on Patt. Analysis and Machine Intelligence*, PAMI-6(6):721–741, November 1984.

[14] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Comp. Graphics Proceedings, Annual Conf. Series*, pp. 43–54, Proc. SIGGRAPH'96, August 1996.

[15] S. S. Intille and A. F. Bobick. Disparity-space images and large occlusion stereo. In *Proc. Third European Conf. on Comp. Vision (ECCV'94)*, vol. 1, May 1994. Springer-Verlag.

[16] T. Kanade et al. A stereo machine for video-rate dense depth mapping and its new applications. In *IEEE Comp. Soc. Conf. on Comp. Vision and Patt. Recog. (CVPR'96)*, pp. 196–202, 1996.

[17] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory

and experiment. *IEEE Trans. on Patt. Analysis and Machine Intelligence*, 16(9):920–932, September 1994.

[18] T. Kanade, P. W. Rander, and P. J. Narayanan. Virtualized reality: constructing virtual worlds from real scenes. *IEEE MultiMedia Magazine*, 1(1):34–47, Jan-March 1997.

[19] S. B Kang. A survey of image-based rendering techniques. In *Videometrics VI*, vol. 3641, pp. 2–16, 1999. SPIE.

[20] R. Kimura et al. A convolver-based real-time stereo machine (SAZAN). In *IEEE Comp. Soc. Conf. on Comp. Vision and Patt. Recog. (CVPR'99)*, vol. 1, pp. 457–463, June 1999.

[21] R. Kumar, P. Anandan, and K. Hanna. Direct recovery of shape from multiple views: A parallax based approach. In *Twelfth Intl. Conf. on Patt. Recognition (ICPR'94)*, volume A, pp. 685–688, Jerusalem, Israel, October 1994. IEEE Comp. Soc. Press.

[22] L. H. Matthies, R. Szeliski, and T. Kanade. Kalman filter-based algorithms for estimating depth from image sequences. *Intl. J. Comp. Vision*, 3:209–236, 1989.

[23] L. McMillan and G. Bishop. Plenoptic modeling: An image-based rendering system. *Comp. Graphics (SIGGRAPH'95)*, pp. 39–46, August 1995.

[24] M. Okutomi and T. Kanade. A multiple baseline stereo. *IEEE Trans. on Patt. Analysis and Machine Intelligence*, 15(4):353–363, April 1993.

[25] S. B. Pollard, J. E. W. Mayhew, and J. P. Frisby. PMF: A stereo correspondence algorithm using a disparity gradient limit. *Perception*, 14:449–470, 1985.

[26] K. Prazdny. Detection of binocular disparities. *Biological Cybernetics*, 52:93–99, 1985.

[27] S. Roy, , and I. J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *Sixth Intl. Conf. on Comp. Vision (ICCV'98)*, pp. 492–499, January 1998.

[28] D. Scharstein and R. Szeliski. Stereo matching with nonlinear diffusion. *Intl. J. Comp. Vision*, 28(2):155–174, July 1998.

[29] S. M. Seitz and C. M. Dyer. Photorealistic scene reconstrcution by voxel coloring. In *IEEE Comp. Soc. Conf. on Comp. Vision and Patt. Recog. (CVPR'97)*, pp. 1067–1073, June 1997.

[30] J. Shade, S. Gortler, L.-W. He, and R. Szeliski. Layered depth images. In *Comp. Graphics (SIGGRAPH'98) Proceedings*, pp. 231–242, July 1998.

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

[32] R. Szeliski. A multi-view approach to motion and stereo. In *IEEE Comp. Soc. Conf. on Comp. Vision and Patt. Recog. (CVPR'99)*, volume 1, pp. 157–163, Fort Collins, June 1999.

[33] R. Szeliski. Prediction error as a quality metric for motion and stereo. In *Seventh Intl. Conf. on Comp. Vision (ICCV'99)*, September 1999.

[34] R. Szeliski and P. Golland. Stereo matching with transparency and matting. *Intl. J. Comp. Vision*, 32(1):45–61, August 1999. Special Issue for Marr Prize papers.

[35] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and texture-mapped models. *Comp. Graphics (SIGGRAPH'97)*, pp. 251–258, August 1997.

[36] R. Szeliski and P. Torr. Geometrically constrained structure from motion: Points on planes. In *European Workshop on 3D Structure from Multiple Images of Large-Scale Environments (SMILE)*, pp. 171–186, Freiburg, Germany, June 1998.

[37] R. Szeliski and R. Zabih. An experimental comparison of stereo algorithms. In *Vision Algorithms 99 Workshop*, submitted 1999.

[38] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Trans. on Patt. Analysis and Machine Intelligence*, PAMI-8(4):413–424, July 1986.

[39] P. H. S. Torr, R. Szeliski, and P. Anandan. An integrated Bayesian approach to layer extraction from image sequences. In *Seventh Intl. Conf. on Comp. Vision (ICCV'98)*, September 1999.

[40] J.Y.A. Wang and E.H. Adelson. Layered representation for motion analysis. In *IEEE Comp. Soc. Conf. on Comp. Vision and Patt. Recognition*, pp. 361–366, 1993.

[41] Y. Yang, A. Yuille, and J. Lu. Local, global, and multilevel stereo matching. In *IEEE Comp. Soc. Conf. on Comp. Vision and Patt. Recog. (CVPR'93)*, pp. 274–279, June 1993.