International Journal of Computer Vision 32(1), 45–61 (1999)

Stereo Matching with Transparency and Matting

RICHARD SZELISKI
Microsoft Reasearch, One Microsoft Way, Redmond, WA 98052-6399
szeliski@microsoft.com

POLINA GOLLAND
Artificial Intelligence Laboratory, 545 Technology Square #810, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139
polina@ai.mit.edu

Abstract. This paper formulates and solves a new variant of the stereo correspondence problem: simultaneously recovering the disparities, true colors, and opacities of visible surface elements. This problem arises in newer applications of stereo reconstruction, such as view interpolation and the layering of real imagery with synthetic graphics for special effects and virtual studio applications. While this problem is intrinsically more difficult than traditional stereo correspondence, where only the disparities are being recovered, it provides a principled way of dealing with commonly occurring problems such as occlusions and the handling of mixed (foreground/background) pixels near depth discontinuities. It also provides a novel means for separating foreground and background objects (matting), without the use of a special blue screen. We formulate the problem as the recovery of colors and opacities in a generalized 3D (x, y, d) disparity space, and solve the problem using a combination of initial evidence aggregation followed by iterative energy minimization.

Keywords: stereo correspondence, 3D reconstruction, 3D representations, matting problem, occlusions, transparency

1. Introduction

Stereo matching has long been one of the central research problems in computer vision. Early work was motivated by the desire to recover depth maps and shape models for robotics and object recognition applications. More recently, depth maps obtained from stereo have been painted with *texture maps* extracted from input images in order to create realistic 3D scenes and environments for virtual reality and virtual studio applications (McMillan and Bishop, 1995; Szeliski and Kang, 1995; Kanade et al., 1996; Blonde et al., 1996). Unfortunately, the quality and resolution of most stereo algorithms falls quite short of that demanded by these new applications, where even isolated errors in the depth map become readily visible when composited with synthetic graphical elements.

One of the most common errors made by most stereo algorithms is a systematic "fattening" of depth layers near occlusion boundaries. Algorithms based on variable window sizes (Kanade and Okutomi, 1994) or iterative evidence aggregation (Scharstein and Szeliski, 1996) can sometimes mitigate such errors. Another common problem is that disparities are only estimated to the nearest pixel, which is typically not sufficiently accurate for tasks such as view interpolation. Different techniques have been developed for computing subpixel estimates, such as using a finer set of disparity hypotheses or finding the the analytic minimum of the local error surface (Tian and Huhns, 1986; Matthies et al., 1989).

Unfortunately, for challenging applications such as *z-keying* (the insertion of graphics between different depth layers in video) (Paker and Wilbur, 1994; Kanade

et al., 1996; Blonde et al., 1996), even this is not good enough. Pixels lying near or on occlusion boundaries will typically be *mixed*, i.e., they will contain blends of both foreground and background colors. When such pixels are composited with other images or graphical elements, objectionable "halos" or "color bleeding" may be visible.

The computer graphics and special effects industries faced a similar problem when extracting foreground objects using *blue screen* techniques (Smith and Blinn, 1996). A variety of techniques were developed for this *matting problem*, all of which model mixed pixels as combinations of foreground and background colors (the latter of which is usually assumed to be known). Practitioners in these fields quickly discovered that it is insufficient to merely label pixels as foreground and background: It is necessary to simultaneously recover both the true color of each pixel and its *transparency* or *opacity* (Porter and Duff, 1984; Blinn, 1994a). In the usual case of opaque objects, pixels are only partially opaque at the boundaries of objects—this is the case we focus on in this paper. True transparency (actually, translucency) has also been studied (Adelson and Anandan, 1990, 1993), but usually only for very simple stimuli.

In this paper, we develop a new, multiframe stereo algorithm which simultaneously recovers depth, color, and transparency estimates at each pixel. Unlike traditional blue-screen matting, we cannot use a known background color to perform the color and matte recovery. Instead, we explicitly model a 3D (x, y, d) *disparity space*, where each cell has an associated color and opacity value. Our task is to estimate the color and opacity values which best predict the appearance of each input image, using prior assumptions about the (piecewise-) continuity of depths, colors, and opacities to make the problem well posed. To our knowledge, this is the first time that the simultaneous recovery of depth, color, and opacity from stereo images has been attempted.

We begin this paper with a review of previous work in stereo matching. In Section 3, we discuss our novel representation for accumulating color samples in a generalized disparity space. We then describe how to compute an initial estimate of the disparities (Section 4), and how to refine this estimate by taking into account occlusions (Section 5). In Section 6, we develop a novel energy minimization algorithm for estimating disparities, colors and opacities. We present some experiments on both synthetic and real images in Section 7. We conclude the paper with a discussion of our results, and a list of topics for future research.

2. Previous Work

Stereo matching and stereo-based 3D reconstruction are fields with very rich histories (Barnard and Fischler, 1982; Dhond and Aggarwal, 1989). In this section, we focus only on previous work related to our central topics of interest: pixel-accurate matching with sub-pixel precision, the handling of occlusion boundaries, and the use of more than two images. We also mention techniques used in computer graphics to composite images with transparencies and to recover matte (transparency) values using traditional blue-screen techniques.

We find it useful to subdivide the stereo matching process into three tasks: the initial computation of matching costs, the aggregation of local evidence, and the selection or computation of a disparity value for each pixel (Scharstein and Szeliski, 1996).

The most fundamental element of any correspondence algorithm is a matching cost that measures the similarity of two or more corresponding pixels in different images. Matching costs can be defined locally (at the pixel level), e.g., as absolute (Kanade et al., 1996) or squared intensity differences (Matthies et al., 1989), using edges (Baker, 1980) or filtered images (Jenkin et al., 1991; Jones and Malik, 1992). Alternatively, matching costs may be defined over an area, e.g., using correlation (Ryan et al., 1980; Wood, 1983) (this can be viewed as a combination of the matching and aggregation stages). In this paper, we use squared intensity differences.

Support aggregation is necessary to disambiguate potential matches. A support region can either be two-dimensional at a fixed disparity (favoring fronto-parallel surfaces), or three-dimensional in (x, y, d) space (allowing slanted surfaces). Two-dimensional evidence aggregation has been done using both fixed square windows (traditional) and windows with adaptive sizes (Arnold, 1983; Kanade and Okutomi, 1994). Three-dimensional support functions include limited disparity gradient (Pollard et al., 1985), Prazdny's coherence principle (Prazdny, 1985) (which can be implemented using two diffusion processes (Szeliski and Hinton, 1985)), local winner-take-all (Yang et al., 1993), and iterative (nonlinear) evidence aggregation

(Scharstein and Szeliski, 1996). In this paper, our initial evidence aggregation uses an iterative technique, with estimates being refined later through a prediction/adjustment mechanism which explicitly models occlusions.

The easiest way of choosing the best disparity is to select at each pixel the minimum aggregated cost across all disparities under consideration ("winner-take-all"). A problem with this is that uniqueness of matches is only enforced for one image (the *reference image*), while points in the other image might get matched to multiple points. Cooperative algorithms employing symmetric uniqueness constraints are one attempt to solve this problem (Marr and Poggio, 1976). In this paper, we introduce the concept of a *virtual camera* which is used for the initial winner-take-all stage.

Occlusion is another very important issue in generating high-quality stereo maps. Many approaches ignore the effects of occlusion. Others try to minimize them by using a cyclopean disparity representation (Barnard, 1989), or try to recover occluded regions after the matching by cross-checking (Fua, 1993). Several authors have addressed occlusions explicitly, using Bayesian models and dynamic programming (Arnold, 1983; Ohta and Kanade, 1985; Belhumeur and Mumford, 1992; Cox, 1994; Geiger et al., 1992; Intille and Bobick, 1994). However, such techniques require the strict enforcement of *ordering constraints* (Yuille and Poggio, 1984). In this paper, we handle occlusion by re-projecting the disparity space into each input image using traditional back-to-front compositing operations (Porter and Duff, 1984), and eliminating from consideration pixels which are known to be occluded. (A related technique, developed concurrently with ours, traverses the disparity space from front to back (Seitz and Dyer, 1997).)

Sub-pixel (fractional) disparity estimates, which are essential for applications such as view interpolation, can be computed by fitting a curve to the matching costs at the discrete disparity levels (Lucas and Kanade, 1981; Tian and Huhns, 1986; Matthies et al., 1989; Kanade and Okutomi, 1994). 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.

Multiframe stereo algorithms use more than two images to increase the stability of the algorithm (Bolles et al., 1987; Matthies et al., 1989; Kang et al., 1995; Collins, 1996). In this paper, we present a new framework for formulating the multiframe stereo problem based on the concept of a *virtual camera* and a projective *generalized disparity space*, which includes as special cases the *multiple baseline stereo* models of (Okutomi and Kanade, 1993; Kang et al., 1995; Collins, 1996).

Finally, the topic of transparent surfaces has not received much study in the context of computational stereo (Prazdny, 1985; Szeliski and Hinton, 1985; Weinshall, 1989). Relatively more work has been done in the context of transparent motion estimation (Shizawa and Mase, 1991a, 1991b; Darrell and Pentland, 1991; Bergen et al., 1992; Ju et al., 1996). However, these techniques are limited to extracting a small number of dominant motions or planar surfaces. None of these techniques explicitly recover a per-pixel transparency value along with a corrected color value, as we do in this paper.

Our stereo algorithm has also been inspired by work in computer graphics, especially in image compositing (Porter and Duff, 1984; Blinn, 1994a) and blue screen techniques (Vlahos and Taylor, 1993; Smith and Blinn, 1996). While traditional blue-screen techniques assume that the background is of a known color, we solve for the more difficult case of each partially transparent surface pixel being the combination of two (or more) unknown colors.

3. Disparity Space Representation

To formulate our (potentially multiframe) stereo problem, we use a *generalized disparity space* which can be any projective sampling of 3D space (Fig. 1). More concretely, 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 3D space, we can make the camera orthographic, with the camera's (x, y, d) axes being orthogonal and evenly sampled (as in (Seitz and Dyer, 1997)). As another example, we may wish to use a *skewed camera model* for constructing a Lumigraph (Gortler et al., 1996).

Having chosen a virtual camera position, we can also choose the orientation and spacing of the *disparity planes*, i.e., the constant d planes. The relationship between d and 3D space can be projective. For example,

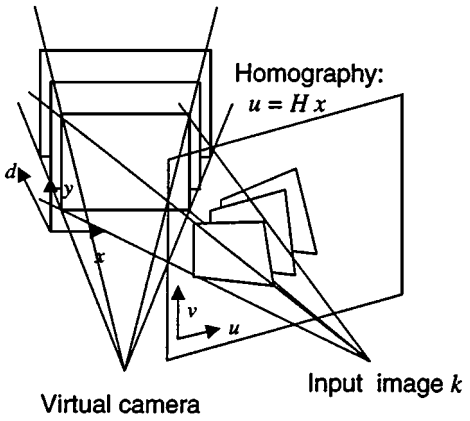

Figure 1. The virtual camera defines the (x, y, d) generalized disparity space.

we can choose d to be inversely proportional to depth, which is the usual meaning of disparity (Okutomi and Kanade, 1993). 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 3D space. The matrix $\hat{\mathbf{M}}_0$ can also capture the sampling information inherent in our disparity space, e.g., if we define disparity space (x, y, d) to be an integer valued sampling of the mapping $\hat{\mathbf{M}}_0\mathbf{x}$, where $\mathbf{x}$ represents point in 3D (Euclidean) space.

An example of a possible disparity space representation is the standard epipolar geometry for two or more cameras placed in a plane perpendicular to their optical axes, in which case a natural choice for disparity is inverse depth (since this corresponds to uniform steps in inter-camera displacements, i.e., the quantity which can be measured accurately) (Okutomi and Kanade, 1993). Other choices include the traditional *cyclopean camera* placed symmetrically between two verged cameras, or a uniform sampling of 3D which is useful in a true verged multi-camera environment (Seitz and Dyer, 1997) or for motion stereo. Note that in all of these situations, integral steps in disparity may correspond to fractional shifts in displacement, which may be desirable for optimal accuracy.

Regardless of the disparity space selected, it is always possible to project each of the input images onto the $d = 0$ plane through a simple homography (2D perspective transform), and to work with such re-projected (*rectified*) images as the inputs to the stereo algorithm. What are the possible advantages of such a rectification step? For two or more cameras whose optical centers are collinear, it is always possible to find a rectification in which corresponding epipolar lines are horizontal, greatly simplifying the stereo algorithm's implementation. For three or more cameras which are coplanar, after rectification, displacements away from the $d = 0$ plane (i.e., changes in disparity) will correspond to uniform steps along fixed directions for each camera (e.g., horizontal and vertical under a suitable camera geometry). Finally, for cameras in general position, steps in disparity will correspond to zooms (scalings) and sub-pixel shifts of the rectified images, which is quicker (and potentially more accurate) than general perspective resampling (Collins, 1996). A potential disadvantage of pre-rectification is a slight loss in input image quality due to multiple re-samplings, but this can be mitigated using higher-order (e.g., bicubic) sampling filters, and potentially re-sampling the rectified images at higher resolution. The Appendix derives the equations for mapping between input image (both rectified and not) and disparity space.

In this paper, we introduce a generalization of the (x, y, d) space. If we consider each of the $k = 1, \ldots, K$ images as being samples along a fictitious "camera" dimension, we end up with a 4D (x, y, d, k) space (Fig. 2). In this space, the values in a given (x, y, d) cell as k varies can be thought of as the color distributions at a given location in space, assuming that this location is actually on the surface of the object and is visible in all cameras. We will use these distributions as the inputs to our first stage of processing, i.e., by computing mean and variance statistics. A different slice through (x, y, d, k) space, this time by fixing k, gives the series of shifted images seen by one camera. In particular, compositing these images in a back-to-front

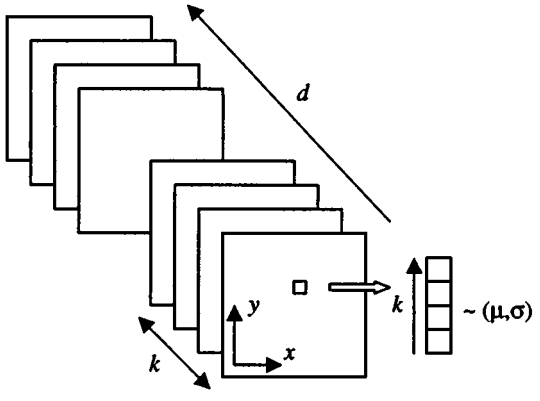

Figure 2. Resampled images can be stacked into a 4D (x, y, d, k) space, with mean values and variances being computed across k.

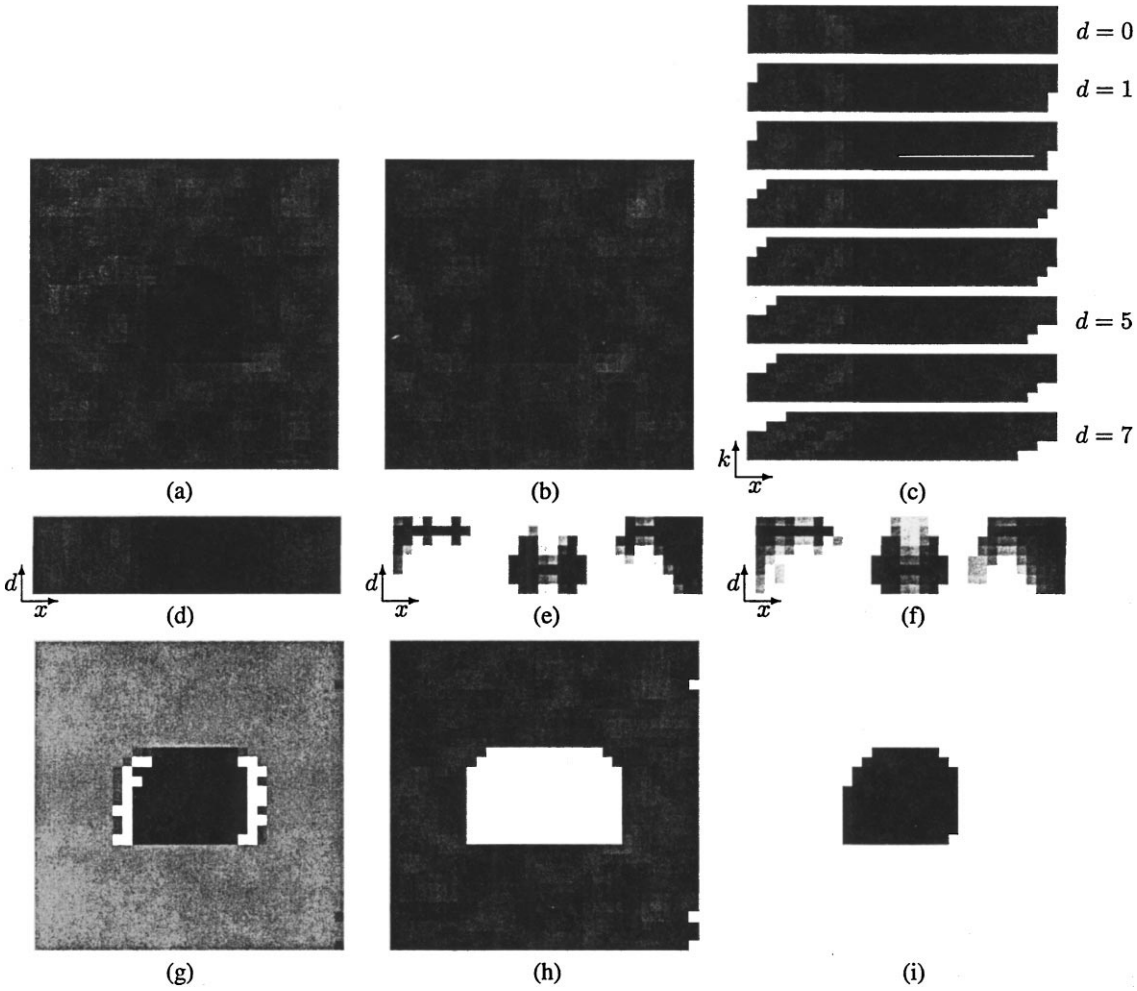

Figure 3. Sample slices through a 4D disparity space: (a, b) sample input images (arranged for free fusion)—darker red object at $d = 5$ in front of lighter blue background at $d = 1$, (c) (x, d, k) slice for scanline 17 ($k = 5$), (d) means and (e) variances as a function of (x, d) (smaller variances are darker), (f) variances after evidence accumulation, (g) results of winner-takes-all for whole image (undecided columns in white), (h, i) colors and opacities at disparities 1 and 5. For easier interpretation, all images have been composited over an opaque white background.

order, taking into account each voxel's opacity, should reconstruct what is seen by a given (rectified) input image (see Section 5).[1]

Figure 3 shows a set of sample images from a $k = 5$ image random-dot stereogram, together with an (x, d, k) slice through the 4D space (y is fixed at a given scanline), where color samples varying in k are grouped together.

4. Estimating an Initial Disparity Surface

The first step in stereo matching is to compute some initial evidence for a surface existing at (or near) a location (x, y, d) in disparity space. We do this by conceptually populating the entire 4D (x, y, d, k) space with colors obtained by resampling the K input images,

$$\mathbf{c}(x, y, d, k) = \mathcal{W}_f(\mathbf{c}_k(u, v); \mathbf{H}_k + \mathbf{t}_k[0\ 0\ d]), \quad (1)$$

where $\mathbf{c}_k(u, v)$ is the kth input image,[2] $\mathbf{H}_k + \mathbf{t}_k[0\ 0\ d]$ is the homography mapping this image to disparity plane d (see the Appendix), $\mathcal{W}_f$ is the forward warping operator,[3] and $\mathbf{c}(x, y, d, k)$ is the pixel mapped into the 4D generalized disparity space.

Algorithmically, this can be achieved either by first rectifying each image onto the $d = 0$ plane, or by

directly using a homography (planar perspective transform) to compute each (d, k) slice.[4] Note that at this stage, not all (x, y, d, k) cells will be populated, as some of these may map to pixels which are outside some of the input images.

Once we have a collection of color (or luminance) values at a given (x, y, d) cell, we can compute some initial statistics over the K (or fewer) colors, e.g., the sample mean μ and variance σ^2.[5] Robust estimates of sample mean and variance are also possible (e.g., Scharstein and Szeliski (1996). Examples of the mean and variance values for our sample image are shown in Figs. 3(d) and (e), where darker values indicate smaller variances.

After accumulating the local evidence, we usually do not have enough information to determine the correct disparities in the scene (unless each pixel has a unique color). While pixels at the correct disparity should in theory have zero variance, this is not true in the presence of image noise, fractional disparity shifts, and photometric variations (e.g., specularities). The variance may also be arbitrarily high in occluded regions, where pixels which actually belong to a different disparity level will nevertheless vote, often leading to gross errors. For example, in Fig. 3(c), the middle (red) group of pixels at $d = 5$ should all have the same color in any given column, but they do not because of resampling errors. This effect is especially pronounced near the edge of the red square, where the red color has been severely contaminated by the background blue. This contamination is one of the reasons why most stereo algorithm make systematic errors in the vicinity of depth discontinuities.

To help disambiguate matches, we can use local evidence aggregation. The most common form is averaging using square windows, which results in the traditional sum of squared difference (SSD and SSSD) algorithms (Okutomi and Kanade, 1993). To obtain results with better quality near discontinuities, it is preferable to use adaptive windows (Kanade and Okutomi, 1994) or iterative evidence accumulation (Scharstein and Szeliski, 1996). In the latter case, we may wish to accumulate an evidence measure which is not simply summed error (e.g., the probability of a correct match (Scharstein and Szeliski, 1996)). Continuing our simple example, Fig. 3(f) shows the results of an evidence accumulation stage, where more certain depths are darker. To generate these results, we aggregate evidence using a variant of the algorithm described in (Scharstein and Szeliski, 1996),

$$\sigma_i^{t+1} \leftarrow a\hat{\sigma}_i^t + b \sum_{j \in \mathcal{N}_4(i)} \hat{\sigma}_j^t + c\sigma_i^0. \qquad (2)$$

Here, σ_i^0 is the original variance at pixel i computed by comparing all sampled colors from the k images, σ_i^t is the variance at iteration t, $\hat{\sigma}_i^t = \min(\sigma_i^t, \sigma_{\max})$ is a robustified (limited) version of the variance, and $\mathcal{N}_4$ are the usual four nearest neighbors. The effect of this updating rule is to *diffuse* variance values to their neighbors, while preventing the diffusion from totally averaging out the variances. For the results in Fig. 3, we use $(a, b, c) = (0.1, 0.15, 0.3)$ and $\sigma_{\max} = 16$.

At this stage, most stereo matching algorithms pick a winning disparity in each (x, y) column, and call this the final correspondence map. Optionally, they may also compute a fractional disparity value by fitting an analytic curve to the error surface around the winning disparity and then finding its minimum (Matthies et al., 1989; Okutomi and Kanade, 1993). Unfortunately, this does nothing to resolve several problems: occluded pixels may not be handled correctly (since they have "inconsistent" color values at the correct disparity), and it is difficult to recover the true (unmixed) color values of surface elements (or their opacities, in the case of pixels near discontinuities).

Our solution to this problem is to use the initial disparity map as the input to a refinement stage which simultaneously estimates the disparities, colors, and opacities which best match the input images while conforming to some prior expectations on smoothness. To start this procedure, we initially pick only winners in each column where the answer is fairly certain, i.e., where the variance ("scatter" in color values) is below a threshold and is a clear winner with respect to the other candidate disparities.[6] A new (x, y, d) volume is 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.[7]

5. Computing Visibilities Through Re-Projection

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

$$\mathbf{x}_k = \mathbf{M}_k \hat{\mathbf{M}}_0^{-1} \mathbf{x}_0 \qquad (3)$$

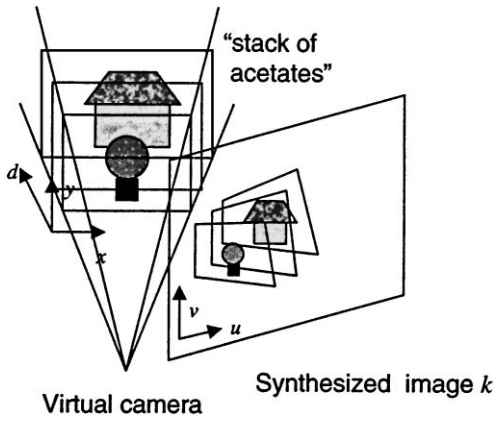

Figure 4. "Stack of acetates" model for image formation from (x, y, d) RGBA color/opacity volume.

(see (A1) in the Appendix), 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, $\mathbf{M}_k$ is the kth camera matrix, and $\mathbf{x}_k$ are the image coordinates in the kth image. There are several techniques possible for performing this projection, including classical *volume rendering* techniques (Levoy, 1990; Lacroute and Levoy, 1994). In our approach, we interpret the (x, y, d) volume as a set of (potentially) transparent acetates stacked at different d levels (Fig. 4). Each acetate is first warped into a given input camera's frame using the known homography

$$\mathbf{x}_k = \mathbf{H}_k \mathbf{x}_0 + \mathbf{t}_k d = (\mathbf{H}_k + \mathbf{t}_k [0\ 0\ d])\, \mathbf{x}_0 \qquad (4)$$

where $\mathbf{x}_0 = (x, y, 1)$, and the layers are then composited back-to-front (this is called a shear-warp algorithm (Lacroute and Levoy, 1994)).[8]

The resampling procedure for a given layer d into the coordinate system of camera k can be written as

$$\tilde{\mathbf{c}}_k(u, v, d) = \mathcal{W}_b(\hat{\mathbf{c}}(x, y, d); \mathbf{H}_k + \mathbf{t}_k [0\ 0\ d]), \qquad (5)$$

where $\hat{\mathbf{c}} = [r\ g\ b\ \alpha]^{\mathrm{T}}$ is the current color and opacity estimate at a given location (x, y, d), $\tilde{\mathbf{c}}_k$ is the resampled layer d in camera k's coordinate system, and $\mathcal{W}_b$ is the resampling operation induced by the homography (4).[9] The opacity value α is 0 for transparent pixels, 1 for opaque pixels, and in between for border pixels. Note that the warping function is *linear* in the colors and opacities being resampled, i.e., the $\tilde{\mathbf{c}}_k(u, v, d)$ can be expressed as a linear function of the $\hat{\mathbf{c}}(x, y, d)$, e.g., through a sparse matrix multiplication.

Once the layers have been resampled, they are then composited using the standard *over* operator (Porter and Duff, 1984),

$$\mathbf{f} \odot \mathbf{b} \equiv \mathbf{f} + (1 - \alpha_f)\mathbf{b},$$

where $\mathbf{f}$ and $\mathbf{b}$ are the premultiplied foreground and background colors, and α_f is the opacity of the foreground (Porter and Duff, 1984; Blinn, 1994a). Note that for $\alpha_f = 0$ (transparent foreground), the background is selected, whereas for $\alpha_f = 1$ (opaque foreground), the foreground is returned. Using the over operator, we can form a composite image

$$\tilde{\mathbf{c}}_k(u, v) = \bigodot_{d=d_{\min}}^{d_{\max}} \tilde{\mathbf{c}}_k(u, v, d)$$
$$= \tilde{\mathbf{c}}_k(u, v, d_{\max}) \odot \cdots \odot \tilde{\mathbf{c}}_k(u, v, d_{\min}) \qquad (6)$$

(note that the over operator is associative but not commutative, and that $d_{\max}$ is the layer closest to the camera).

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). A simple way to construct such a visibility map is to record the disparity value d_{top} for each (u, v) pixel which corresponds to the topmost opaque pixel seen during the compositing step.[10] The visibility value can then be defined as

$$V_k(u, v, d) = \text{if } d \geq d_{\text{top}}(u, v) \text{ then 1 else 0.}$$

The visibility and opacity (alpha) values taken together can be interpreted as follows:

$V_k = 1, \tilde{a}_k = 0$:	free space
$V_k = 1, \tilde{a}_k = 1$:	surface voxel visible in image k
$V_k = 0, \tilde{a}_k = ?$:	voxel not visible in image k

where $\tilde{a}_k$ is the opacity of $\tilde{\mathbf{c}}_k$ in (5).

A more principled way of defining visibility, which takes into account partially opaque voxels, uses a recursive front-to-back algorithm

$$V_k(u, v, d-1) = V_k(u, v, d)\,(1 - \tilde{a}_k(u, v, d))$$
$$= \prod_{d'=d}^{d_{\max}} (1 - \tilde{a}_k(u, v, d')), \qquad (7)$$

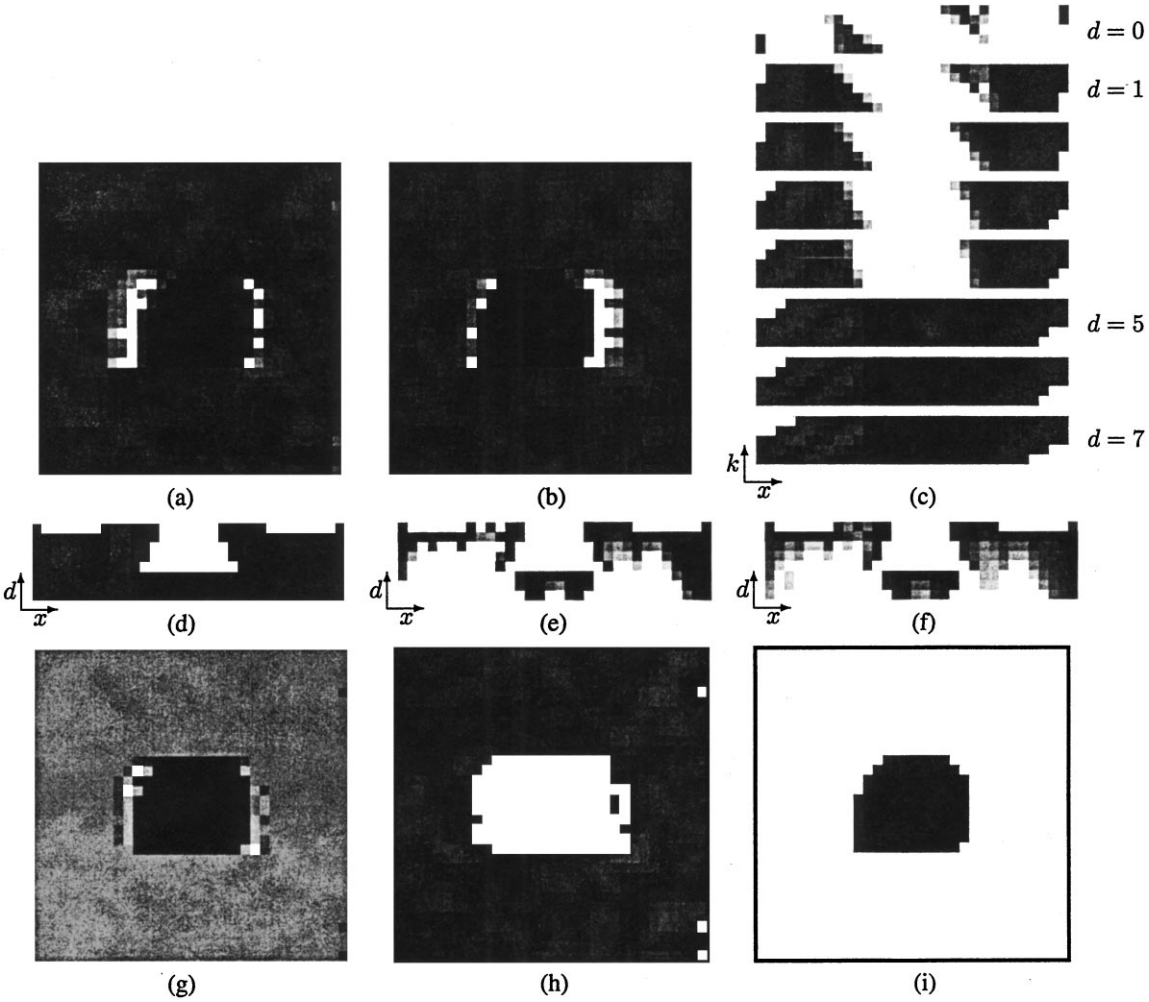

Figure 5. After modifying input images by visibility $V_k(u, v, d)$: (a, b) re-synthesized views of sample images, (c) (x, d, k) slice for scanline 17, (d) means and (e) variances as a function of (x, d), (f) variances after evidence accumulation, (g) results of winner-takes-all for whole image, and (h, i) colors and opacities at disparities 1 and 5 after one iteration of the reprojection algorithm.

with the initial visibilities all being set to 1, $V_k(u, v, d_{\max}) = 1$. We now have a very simple (linear) expression for the compositing operation,

$$\tilde{\mathbf{c}}_k(u, v) = \sum_{d=d_{\min}}^{d_{\max}} \tilde{\mathbf{c}}_k(u, v, d) V_k(u, v, d). \qquad (8)$$

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. Let

$$\mathbf{c}_k(u, v, d) = \mathbf{c}_k(u, v) V_k(u, v, d)$$

be the input color image multiplied by its visibility at disparity d. If we substitute $\mathbf{c}_k(u, v, d)$ for $\mathbf{c}_k(u, v)$ in (1), we obtain a distribution of colors in (x, y, d, k) where each color has an associated visibility value (Fig. 5(c)). Voxels which 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 (Figs. 5(d) and (e)).

With these new statistics, we are now in position to refine the disparity map. In particular, voxels in disparity space which 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 which are not already assigned to nearer surfaces (Figs. 5(c)–(f)). Figure 5(g)–(i) show the results after one iteration of this algorithm.

6. Refining Color and Transparency Estimates

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.

A simple way to approach this problem is to take the binary opacity maps produced by our stereo matching algorithm, and to make them real-valued using a low-pass filter. Another possibility might be to recover the transparency information by looking at the magnitude of the intensity gradient (Mitsunaga et al., 1995), assuming that we can isolate regions which belong to different disparity levels.

In our work, we have chosen instead to 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:

1. a weighted error norm on the difference between the re-projected images $\tilde{\mathbf{c}}_k(u, v)$and the original (or rectified) input images $\mathbf{c}_k(u, v)$

$$\mathcal{C}_1 = \sum_{(u,v)} w_k(u, v)\rho_1(\tilde{\mathbf{c}}_k(u, v) - \mathbf{c}_k(u, v)), \quad (9)$$

   where the weights $w_k(u, v)$ may depend on the position of camera k relative to the virtual camera;[11]
2. a (weak) smoothness constraint on the colors and opacities,

$$\mathcal{C}_2 = \sum_{(x,y,d)} \sum_{\substack{(x',y',d') \\ \in \mathcal{N}(x,y,d)}} \rho_2(\hat{\mathbf{c}}(x', y', d') - \hat{\mathbf{c}}(x, y, d)); \quad (10)$$

3. a prior distribution on the opacities,

$$\mathcal{C}_3 = \sum_{(x,y,d)} \phi(\alpha(x, y, d)). \quad (11)$$

In the above equations, ρ_1 and ρ_2 are either quadratic functions or robust penalty functions (Huber, 1981), and ϕ is a function which encourages opacities to be 0 or 1, e.g., $\phi(x) = x(1 - x)$.[12]

The smoothness constraint on colors makes more sense with non-premultiplied colors. For example, a voxel lying on a depth discontinuity will be partially transparent, and yet should have the same non-premultiplied color as its neighbors. An alternative, which allows us to work with premultiplied colors, is to use a smoothness constraint of the form

$$\mathcal{C}_2' = \sum_{(x,y,d)} \sum_{\substack{(x',y',d') \\ \in \mathcal{N}(x,y,d)}} \rho_2(D) \quad (12)$$

where

$$D = \alpha(x, y, d)\mathbf{c}(x', y', d') - \alpha(x', y', d')\mathbf{c}(x, y, d).$$

To minimize the total cost function

$$\mathcal{C} = \lambda_1\mathcal{C}_1 + \lambda_2\mathcal{C}_2 + \lambda_3\mathcal{C}_3, \quad (13)$$

we use a preconditioned gradient descent algorithm. The Appendix contains details on how to compute the required gradients and Hessians.

7. Experiments

To study the properties of our new stereo correspondence algorithm, we ran a small set of experiments on some synthetic stereo datasets, both to evaluate the basic behavior of the algorithm (aggregation, visibility-based refinement, and energy minimization), and to study its performance on mixed (boundary) pixels. Being able to visualize opacities/transparencies is very important for understanding and validating our algorithm. For this reason, we chose color stimuli (the background is blue-green, and the foreground is red). Pixels which are partially transparent will show up as "pale" colors, while fully transparent pixels will be white. We should emphasize that our algorithm does not require colored images as inputs (see Fig. 8), nor does it require the use of standard epipolar geometries.

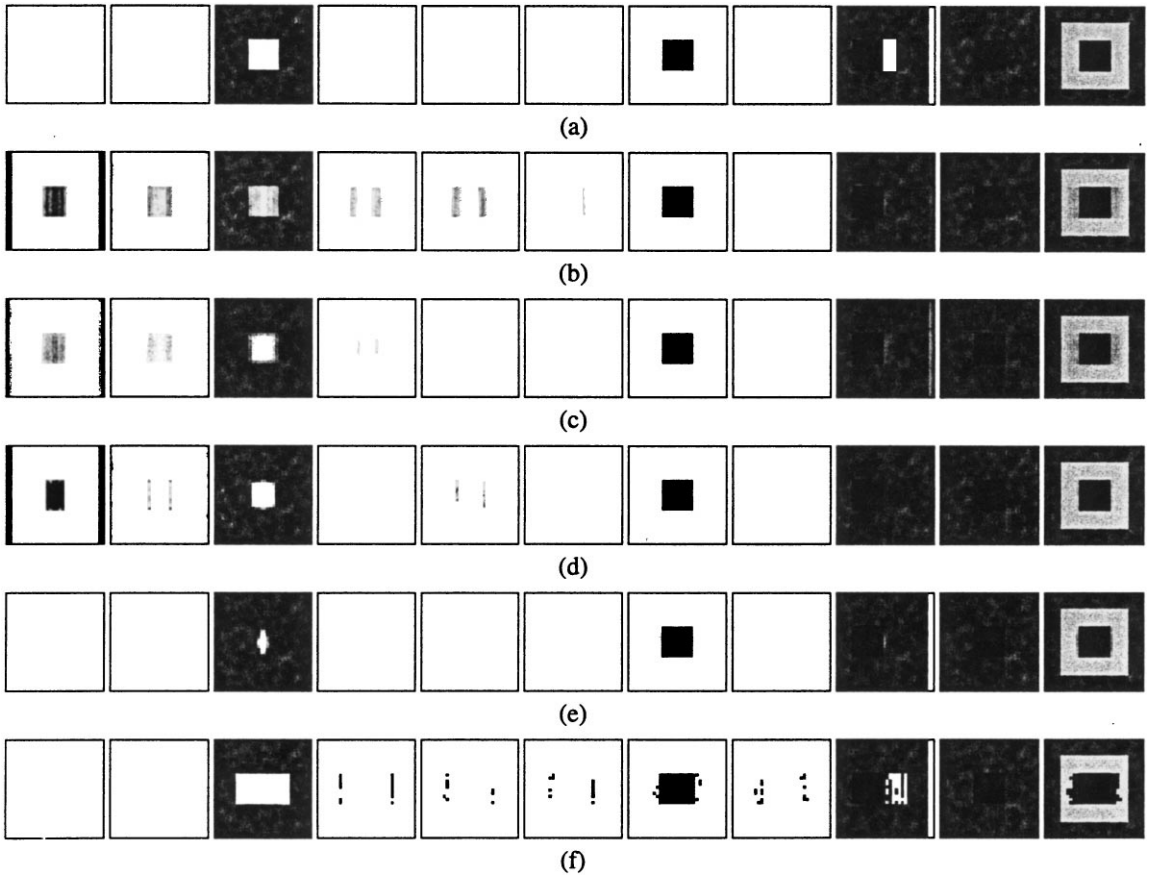

Figure 6. Traditional synthetic RDS results: (a) after iterative aggregation but before gradient descent, (b) without smoothness or opacity constraint, $\lambda_1 = 1, \lambda_2 = \lambda_3 = 0$, (c) without opacity constraint, $\lambda_1 = \lambda_2 = 1, \lambda_3 = 0$, (d) with all three constraints, $\lambda_1 = 50, \lambda_2 = 1, \lambda_3 = 50$, (e) with all three constraints, $\lambda_1 = 50, \lambda_2 = 1, \lambda_3 = 100$, (f) simple winner-take-all (shown for comparison). The first eight columns are the disparity layers, $d = 0, \ldots, 7$. The ninth and tenth columns are re-synthesized sample views. The last column is a re-synthesized view with a synthetic gray square inserted at disparity $d = 3$.

The first stimulus we generated was a traditional random-dot stereogram with $k = 5$ images, where the choice of camera geometry and filled disparity planes results in integral pixel shifts. This example also contains no partially transparent pixels. Figure 6 shows the results on this stimulus. The first eight columns are the eight disparity planes in (x, y, d) space, showing the estimated colors and opacities (smaller opacities are shown as lighter colors, since the RGBA colors are composited over a white background). The ninth and tenth column are two re-synthesized views (leftmost and middle). The last column is the re-synthesized middle view with a synthetic light-gray square inserted at disparity $d = 3$.

As we can see in Fig. 6, the basic iterative aggregation algorithm results in a "perfect" reconstruction, although only one pixel is chosen in each column. For this reason, the re-synthesized leftmost view (ninth column) contains a large "gap".

Figure 6(b) shows the results of using only the first $\mathcal{C}_1$ term in our cost function, i.e., only matching re-synthesized views with input images. The re-synthesized view in column nine is now much better, although we see that a bit of the background has bled into the foreground layers, and that the pixels near the depth discontinuity are spread over several disparities.

Adding the smoothness constraint $\mathcal{C}_2$ (Fig. 6(c)) ameliorates both of these problems. Adding the (weak) 0/1 opacity constraint $\mathcal{C}_3$ (Fig. 6(d) and (e)) further removes stray pixels at wrong disparity levels. Figure 6(d) shows a "softer" variant of the opacity constraint ($\lambda_3 = 50 = \lambda_1$), where more levels end up being

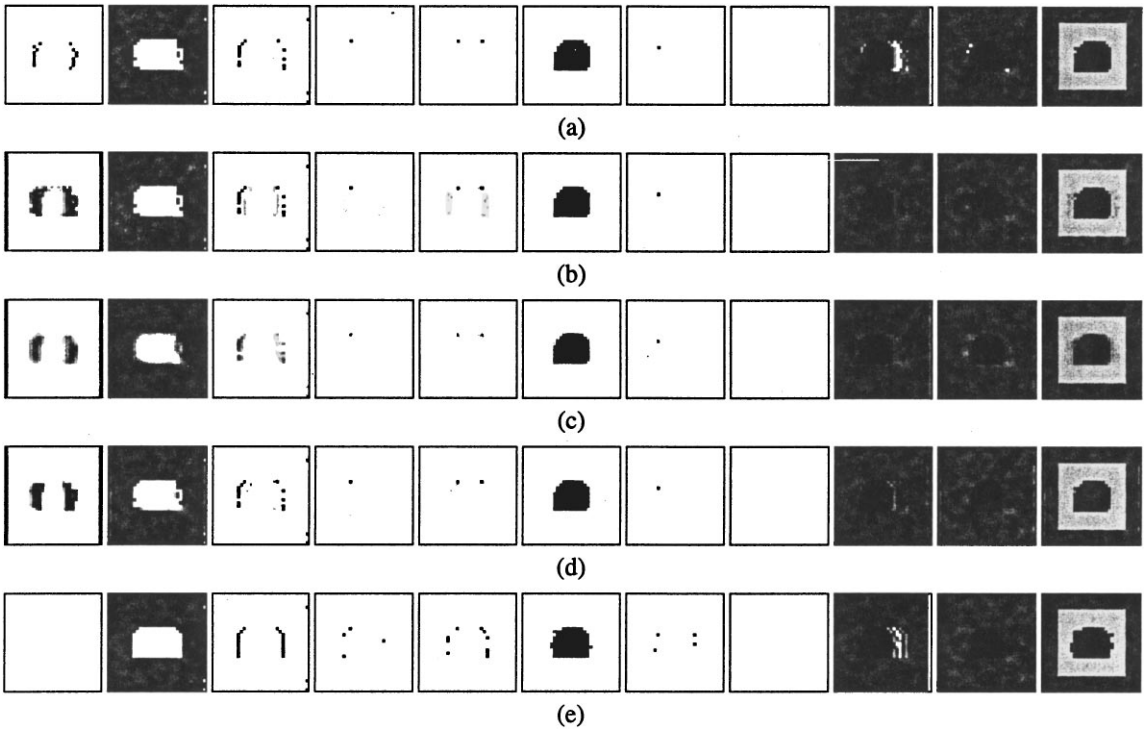

Figure 7. More challenging synthetic RDS results: (a) after iterative aggregation but before gradient descent, (b) without smoothness or opacity constraint, $\lambda_1 = 1$, $\lambda_2 = \lambda_3 = 0$, (c) without opacity constraint, $\lambda_1 = \lambda_2 = 1$, $\lambda_3 = 0$, (d) with all three constraints, $\lambda_1 = 50$, $\lambda_2 = 1$, $\lambda_3 = 50$, (e) simple winner-take-all (shown for comparison). The first eight columns are the disparity layers, $d = 0, \ldots, 7$. The ninth and tenth columns are re-synthesized sample views. The last column is the re-synthesized view with a synthetic gray square inserted at disparity $d = 3$.

filled in, but the re-synthesized views are very good. Figure 6(e) shows a "harder" constraint ($\lambda_3 = 100 = 2\lambda_1$), where only pixels adjacent to initial estimates are filled in, at the cost of a gap in some re-synthesized views.

For comparison, Fig. 6(f) shows the results of a traditional winner-take-all algorithm (the same as Fig. 6(a) with a very large $\theta_{\min}$ and no occluded pixel removal). We can clearly see the effects of background colors being pulled into the foreground layer, as well as increased errors in the occluded regions.

Our second set of experiments uses the same synthetic stereo dataset as shown in Figs. 3 and 5, again with $k = 5$ input images. Here, because the background layer is at an odd disparity, we get significant re-sampling errors (because we currently use bilinear interpolation) and mixed pixels. The stimulus also has partially transparent pixels along the edge of the top half-circle in the foreground shape. This stereo dataset is significantly more difficult to match than previous random-dot stereograms.

Figure 7(a) shows the results of applying only our iterative aggregation algorithm, without any energy minimization. The set of estimated disparities are insufficient to completely reconstruct the input images (this could be changed by adjusting the thresholds $\theta_{\min}$ and θ_s), and several pixels are incorrectly assigned to the $d = 0$ layer (due to difficulties in disambiguating depths in partially occluded regions).

Figure 7(b) shows the results of using only the first $\mathcal{C}_1$ term in our cost function, i.e., only matching re-synthesized views with input images. The re-synthesized view in column nine is now much better, although we see that a bit of the background has bled into the foreground layers, and that the pixels near the depth discontinuity are spread over several disparities.

Adding the smoothness constraint $\mathcal{C}_2$ (Fig. 7(c)) ameliorates both of these problems. Adding the (weak) 0/1 opacity constraint $\mathcal{C}_3$ (Fig. 7(d)) further removes stray pixels at wrong disparity levels, but at the cost of an incompletely reconstructed image (this is less of a problem if the foreground is being layered on a synthetic

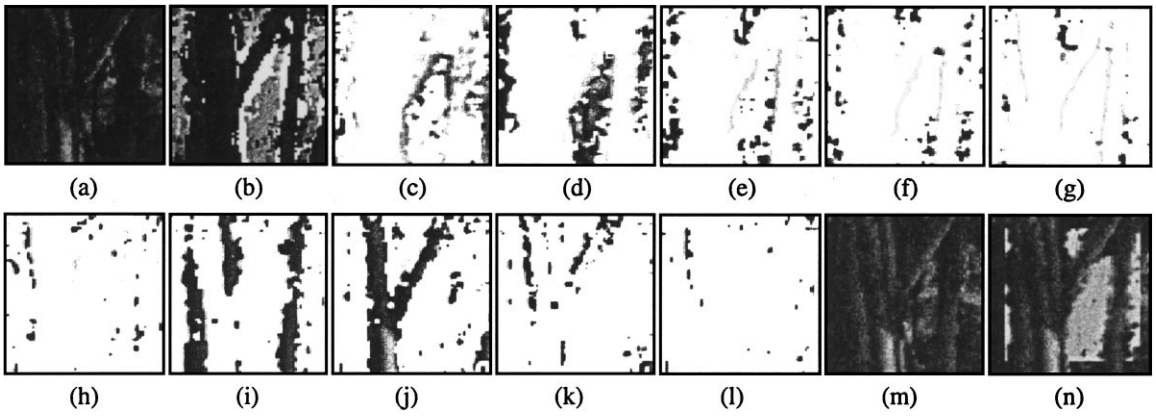

Figure 8. Real image example: (a) cropped subimage from *SRI Trees* data set, (b) depth map after initial aggregation stage, (c–l) disparity layers $d = 0, \ldots, 9$, (m) re-synthesized input image, (n) with inserted $d = 4$ blue layer.

background, as in the last column). As before, Fig. 7(e) shows the results of a traditional winner-take-all algorithm.

Figure 8 shows the results 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 done a reasonable job of assigning pixel depths and computing partial transparencies near the tree boundaries.

From these examples, it is apparent that the algorithm is currently sensitive to the choice of parameters used to control both the initial aggregation stage and the energy minimization phase. Setting these parameters automatically will be an important area for further research.

8. Discussion

While our preliminary experimental results are encouraging, the simultaneous recovery of accurate depth, color, and opacity estimates remains a challenging problem. Traditional stereo algorithms search for a unique disparity value at each pixel in a given reference image. Our approach, on the other hand, is to recover a sparsely populated volume of colors and opacities. This has the advantage of correctly modeling mixed pixels and occlusion effects, and allows us to merge images from very disparate points of view. Unfortunately, it also makes the estimation problem much more difficult, since the number of free parameters often exceeds the number of measurements, hence necessitating smoothness constraints and other prior models.

Partially occluded areas are problematic because very few samples may be available to disambiguate depth. A more careful analysis of the interaction between the measurement, smoothness, and opacity constraints will be required to solve this problem. Other problems occur near depth discontinuities, and in general near rapid intensity (albedo) changes, where the scatter in color samples may be large because of resampling errors. Better imaging and sensor models, or perhaps working on a higher resolution image grid, might be required to solve these problems.

8.1. Future Work

There are many additional topics related to transparent stereo and matting which we would like to investigate. For example, we would like to try our algorithm on data sets with true transparency (not just mixed pixels), such as traditional transparent random dot stereograms (Prazdny, 1985; Weinshall, 1989) and reflections in windows (Bergen et al., 1992).

Estimating disparities to sub-integer precision should improve the quality of our reconstructions. Such fractional disparity estimates can be obtained by interpolating a variance vs. disparity curve $\sigma(d)$, e.g., by fitting a parabola to the lowest variance and its two neighbors (Tian and Huhns, 1986; Matthies et al., 1989). Alternatively, we can linearly interpolate individual color errors $\mathbf{c}(x, y, d, k) - \mu(x, y, d)$ between disparity levels, and find the minimum of the summed

squared error (which will be a quadratic function of the fractional disparity).

Instead of representing our color volume $\hat{\mathbf{c}}(x, y, d)$ using colors pre-multiplied by their opacities (Blinn, 1994a), we could keep these quantities separate. Thus, colors could "bleed" into areas which are transparent, which may be a more natural representation for color smoothness (e.g., for surfaces with small holes). Different color representations such as hue, saturation, intensity (HSV) may also be more suitable for performing correspondence (Golland and Bruckstein, 1995), and they would permit us to reason more directly about underlying physical processes (shadows, shading, etc.).

In recent work, we have extended our stack of acetates model to use a smaller number of tilted acetates with arbitrary plane equations (Baker et al., 1998). This work is closely related to more traditional layered motion models (Wang and Adelson, 1993; Ju et al., 1996; Sawhney and Ayer, 1996; Weiss and Adelson, 1996), but focuses on recovering 3D descriptions instead of 2D motion estimates. Each layer can also have an arbitrary out-of plane parallax component (Baker et al., 1998). The layers are thus used to represent the gross shape and occlusion relationships, while the parallax encodes the fine shape variation. We are also investigating efficient rendering algorithm for doing view synthesis from such *sprites with depth* (Shade et al., 1998).

9. Conclusions

In this paper, we have developed a new framework for simultaneously recovering disparities, colors, and opacities from multiple images. This framework enables us to deal with many commonly occurring problems in stereo matching, such as partially occluded regions and pixels which contain mixtures of foreground and background colors. Furthermore, it promises to deliver better quality (sub-pixel accurate) color and opacity estimates, which can be used for foreground object extraction and mixing live and synthetic imagery.

To set the problem in as general a framework as possible, we have introduced the notion of a virtual camera which defines a generalized disparity space, which can be any regular projective sampling of 3D. We represent the output of our algorithm as a collection of color and opacity values lying on this sampled grid. Any input image can (in principle) be re-synthesized by warping each disparity layer using a simple homography and compositing the images. This representation can support a much wider range of synthetic viewpoints in view interpolation applications than a single texture-mapped depth image.

To solve the correspondence problem, we first compute mean and variance estimates at each cell in our (x, y, d) grid. We then pick a subset of the cells which are likely to lie on the reconstructed surface using a thresholded winner-take-all scheme. The mean and variance estimates are then refined by removing from consideration cells which are in the occluded (shadow) region of each current surface element, and this process is repeated.

Starting from this rough estimate, we formulate an energy minimization problem consisting of an input matching criterion, a smoothness criterion, and a prior on likely opacities. This criterion is then minimized using an iterative preconditioned gradient descent algorithm.

While our preliminary experimental results look encouraging, there remains much work to be done in developing truly accurate and robust correspondence algorithms. We believe that the development of such algorithms will be crucial in promoting a wider use of stereo-based imaging in novel applications such as special effects, virtual reality modeling, and virtual studio productions.

Appendix

Camera Models, Disparity Space, and Induced Homographies

The homographies mapping input images (rectified or not) to planes in disparity space can be derived directly from the camera matrices involved. Throughout this appendix, we use projective coordinates, i.e., equality is defined only up to a scale factor.

Let $\mathbf{M}_k$ be the 3×4 *camera matrix* which maps real-world coordinates $\mathbf{x} = [X\ Y\ Z\ 1]^{\mathrm{T}}$ into a camera's screen coordinates $\mathbf{x}_k = [u\ v\ 1]^{\mathrm{T}}$, $\mathbf{x}_k = \mathbf{M}_k\mathbf{x}$. Similarly, let $\hat{\mathbf{M}}_0$ be the 4×4 collineation which maps world coordinates $\mathbf{x}$ into disparity space coordinates $\hat{\mathbf{x}}_0 = [x\ y\ 1\ d]^{\mathrm{T}}$, $\hat{\mathbf{x}}_0 = \hat{\mathbf{M}}_0\mathbf{x}$.

We can write the mapping between a pixel in the dth disparity plane, $\mathbf{x}_0 = [x\ y\ 1]^{\mathrm{T}}$, and its corresponding coordinate $\mathbf{x}_k$ in the kth input image as

$$\begin{aligned}\mathbf{x}_k &= \mathbf{M}_k\hat{\mathbf{M}}_0^{-1}\hat{\mathbf{x}}_0 = \mathbf{H}_k\mathbf{x}_0 + \mathbf{t}_k d \\ &= (\mathbf{H}_k + \mathbf{t}_k[0\ 0\ d])\,\mathbf{x}_0, \end{aligned} \tag{A1}$$

where $\mathbf{H}_k$ is the homography relating the rectified and non-rectified version of input image k (i.e., the homography for $d = 0$), and $\mathbf{t}_k$ is the image of the virtual camera's center of projection in image k, i.e., the epipole (this can be seen by setting $d \to \infty$).

If we first rectify an input image, we can re-project it into a new disparity plane d using

$$\mathbf{x}_k = \mathbf{H}_k \mathbf{x}_0' = \mathbf{H}_k \mathbf{x}_0 + \mathbf{t}_k d$$

where $\mathbf{x}_0'$ is the new coordinate corresponding to $\mathbf{x}_0$ at $d = 0$. From this,

$$\mathbf{x}_0' = \mathbf{x}_0 + \hat{\mathbf{t}}_k d = (\mathbf{I} + \hat{\mathbf{t}}_k [0\ 0\ d])\mathbf{x}_0 = \hat{\mathbf{H}}_k \mathbf{x}_0,$$

where $\hat{\mathbf{t}}_k = \mathbf{H}_k^{-1} \mathbf{t}_k$ is the focus of expansion, and the new homography $\hat{\mathbf{H}}_k = \mathbf{I} + \hat{\mathbf{t}}_k [0\ 0\ d]$ represents a simple shift and scale. It can be shown (Collins, 1996) that the first two elements of $\hat{\mathbf{t}}_k$ depend on the horizontal and vertical displacements between the virtual camera and the kth camera, whereas the third element is proportional to the displacement in depth (perpendicular to the d plane). Thus, if all of the cameras are coplanar (regardless of their vergence), and if the d planes are parallel to this common plane, then the re-mappings of rectified images to a new disparity correspond to pure shifts.

Note that in the body of the paper, when we speak of the homography (A1) parameterized by $\mathbf{H}_k$ and $\mathbf{t}_k$, we can replace $\mathbf{H}_k$ and $\hat{\mathbf{t}}_k$ by $\mathbf{I}$ and $\hat{\mathbf{t}}_k$ if the input images have been pre-rectified.

Gradient Descent Algorithm

To implement our gradient descent algorithm, we need to compute the partial derivatives of the cost functions $\mathcal{C}_1, \ldots, \mathcal{C}_3$ with respect to all of the unknowns, i.e., the colors and opacities $\hat{\mathbf{c}}(x, y, d)$. In this section, we will use $\hat{\mathbf{c}} = [r\ g\ b\ \alpha]^{\mathrm{T}}$ to indicate the four-element vector of colors and opacities, and α to indicate just the opacity channel. In addition to computing the partial derivatives, we will compute the diagonal of the approximate Hessian matrix (Press et al., 1992, pp. 681–685), i.e., the square of the derivative of the term inside the ρ function.

The derivative of $\mathcal{C}_1$ given in (9) can be computed by first expressing $\tilde{\mathbf{c}}_k(u, v)$ in terms of $\tilde{\mathbf{c}}_k(u, v, d)$,

$$\begin{aligned}\tilde{\mathbf{c}}_k(u, v) &= \sum_{d=d_{\min}}^{d_{\max}} \tilde{\mathbf{c}}_k(u, v, d) V_k(u, v, d) \\ &= \sum_{d'=d}^{d_{\max}} \tilde{\mathbf{c}}_k(u, v, d') V_k(u, v, d') \\ &\quad + (1 - \tilde{\alpha}_k(u, v, d))\tilde{\mathbf{a}}_k(u, v, d-1),\end{aligned}$$

where

$$\mathbf{a}_k(u, v, d) = \sum_{d'=d_{\min}}^{d} \tilde{\mathbf{c}}_k(u, v, d') V_k(u, v, d')$$

is the *accumulated color/opacity* in layer d, with $\tilde{\mathbf{c}}_k(u, v) = \tilde{\mathbf{a}}_k(u, v, d_{\max})$. We obtain

$$\begin{aligned}\frac{\partial r_k(u, v)}{\partial r_k(u, v, d)} &= \frac{\partial g_k(u, v)}{\partial g_k(u, v, d)} = \frac{\partial b_k(u, v)}{\partial b_k(u, v, d)} \\ &= V_k(u, v, d)\end{aligned}$$

and

$$\frac{\partial \tilde{\mathbf{c}}_k(u, v)}{\partial \tilde{\alpha}_k(u, v, d)} = [0\ 0\ 0\ V_k(u, v, d)]^{\mathrm{T}} - \tilde{\mathbf{a}}_k(u, v, d-1).$$

Let $\mathbf{e}_k(u, v) = \tilde{\mathbf{c}}_k(u, v) - \mathbf{c}_k(u, v)$ be the color error in image k, and assume for now that $w_k = 1$ and $\rho_1(\mathbf{e}_k) = \|\mathbf{e}_k\|^2$ in (9). The gradient of $\mathcal{C}_1$ w.r.t. $\tilde{\mathbf{c}}_k(u, v, d)$ is thus

$$\begin{aligned}\tilde{\mathbf{g}}_k(u, v, d) = V_k(u, v, d)(&\mathbf{e}_k(u, v) \\ &- [0\ 0\ 0\ \mathbf{e}_k(u, v) \cdot \tilde{\mathbf{a}}_k(u, v, d-1)]^{\mathrm{T}})\end{aligned}$$

while the diagonal of the Hessian is

$$\begin{aligned}\tilde{\mathbf{h}}_k(u, v, d) = V_k(u, v, d)& \\ &\times [1\ 1\ 1\ 1 - \|\tilde{\mathbf{a}}_k(u, v, d-1)\|^2]^{\mathrm{T}}.\end{aligned}$$

Once we have computed the derivatives w.r.t. the *warped* predicted color values $\tilde{\mathbf{c}}_k(u, v, d)$, we need to convert this to the gradient w.r.t. the disparity space colors $\hat{\mathbf{c}}(x, y, d)$. This can be done using the transpose of the linear mapping induced by the backward warp $\mathcal{W}_b$ used in (5). For certain cases (pure shifts), this is the same as warping the gradient $\tilde{\mathbf{g}}_k(u, v, d)$ and Hessian $\tilde{\mathbf{h}}_k(u, v, d)$ through the forward warp $\mathcal{W}_f$,

$$\begin{aligned}\hat{\mathbf{g}}_1(x, y, d, k) &= \mathcal{W}_f(\tilde{\mathbf{g}}_k(u, v, d); \mathbf{H}_k + \mathbf{t}_k[0\ 0\ d]), \\ \hat{\mathbf{h}}_1(x, y, d, k) &= \mathcal{W}_f(\tilde{\mathbf{h}}_k(u, v, d); \mathbf{H}_k + \mathbf{t}_k[0\ 0\ d]).\end{aligned}$$

For many other cases (moderate scaling and shear), this is still a good approximation, so it is the approach we currently use.

Computing the gradient of $\mathcal{C}_2$ w.r.t. $\hat{\mathbf{c}}(x, y, d)$ is much more straightforward,

$$\hat{\mathbf{g}}_2(x, y, d) = \sum_{\substack{(x', y', d') \\ \in \mathcal{N}_4(x, y, d)}} \dot{\rho}_2(\mathbf{c}(x', y', d') - \mathbf{c}(x, y, d)),$$

where ρ_2 is applied to each color component separately. The Hessian will be a constant for a quadratic penalty function; for a non-quadratic function, the secant approximation $\dot{\rho}(r)/r$ can be used (Sawhney and Ayer, 1996).

Finally, the derivative of the opacity penalty function $\mathcal{C}_3$ can easily be computed for $\phi = x(1 - x)$,

$$\hat{\mathbf{g}}_3(x, y, d) = [0\ 0\ 0\ (1 - 2\alpha(x, y, d))]^{\mathrm{T}}.$$

To ensure that the Hessian is positive, we set $\hat{\mathbf{h}}_3(x, y, d) = [0\ 0\ 0\ 1]^{\mathrm{T}}$.

The gradients for the three cost functions can now be combined

$$\hat{\mathbf{g}}(x, y, d) = \lambda_1 \sum_{k=1}^{K} \hat{\mathbf{g}}_1(x, y, d, k) + \lambda_2 \hat{\mathbf{g}}_2(x, y, d) + \lambda_3 \hat{\mathbf{g}}_3(x, y, d),$$

$$\hat{\mathbf{h}}(x, y, d) = \lambda_1 \sum_{k=1}^{K} \hat{\mathbf{h}}_1(x, y, d, k) + \lambda_2 \hat{\mathbf{h}}_2(x, y, d) + \lambda_3 \hat{\mathbf{h}}_3(x, y, d),$$

and a gradient descent step can be performed,

$$\hat{\mathbf{c}}(x, y, d) \leftarrow \hat{\mathbf{c}}(x, y, d) + \epsilon_1 \hat{\mathbf{g}}(x, y, d) / (\hat{\mathbf{h}}(x, y, d) + \epsilon_2). \quad \text{(A2)}$$

In our current experiments, we use $\epsilon_1 = \epsilon_2 = 0.5$.[13]

Notes

1. Note that this 4D space is *not* the same as that used in the Lumigraph (Gortler et al., 1996), where the description is one of rays in 3D, as opposed to color distributions across multiple cameras in 3D. It is also not the same as an epipolar-plane image (EPI) volume (Bolles et al., 1987), which is a simple concatenation of warped input images.
2. The color values **c** can be replaced with gray-level intensity values without affecting the validity of our analysis.
3. In our current implementation, the warping (resampling) algorithm uses bi-linear interpolation of the pixel colors and opacities.
4. For certain epipolar geometries, even more efficient algorithms are possible, e.g., by simply shifting along epipolar lines (Kanade et al., 1996).
5. In many traditional stereo algorithms, it is common to effectively set the mean to be just the value in one image, which makes these algorithms not truly multiframe (Collins, 1996). The sample variance then corresponds to the squared difference or sum of squared differences (Okutomi and Kanade, 1993).
6. To account for resampling errors which occur near rapid color or luminance changes, we set the threshold proportional to the local image variation within a 3×3 window, $\mathrm{Var}_{3\times3}$. In our experiments, the threshold is set to $\theta = \theta_{\min} + \theta_{\mathrm{s}} \mathrm{Var}_{3\times3}$, with $\theta_{\min} = 10$ and $\theta_{\mathrm{s}} = 0.02$.
7. We may, for computational reasons, choose to represent this volume using colors premultiplied by their opacities (*associated colors* (Porter and Duff, 1984; Blinn, 1994a)), in which case voxels for which alpha (opacity) is 0 should have their color or intensity values set to 0. See Blinn (1994a, 1994b) for a discussion of the advantages of using premultiplied colors.
8. If the input images have been rectified, or under certain imaging geometries, this homography will be a simple scale and/or shift (see the Appendix).
9. This is the inverse of the warp specified in (1).
10. Note that it is not possible to compute visibility in (x, y, d) disparity space, as several opaque pixels in disparity space may potentially project to the same input camera pixel.
11. More precisely, we may wish to measure the angle between the viewing ray corresponding to (u, v) in the two cameras. However, the ray corresponding to (u, v) in the virtual camera depends on the disparity d.
12. All color and opacity values are, of course, constrained to lie in the range [0, 1], making this a constrained optimization problem.
13. A more sophisticated Levenberg-Marquardt minimization technique could also be implemented by adding an extra stabilization parameter (Press et al., 1992). However, implementing a full Levenberg-Marquardt with off-diagonal Hessian elements would greatly complicate the implementation.

References

Adelson, E.H. and Anandan, P. 1990. Ordinal characteristics of transparency. In *AAAI-90 Workshop on Qualitative Vision*, AAAI: Boston, MA, pp. 77–81.

Adelson, E.H. and Anandan, P. 1993. Perceptual organization and the judgement of brightness. *Science*, 262:2042–2044.

Arnold, R.D. 1983. Automated stereo perception. Technical Report AIM-351, Artificial Intelligence Laboratory, Stanford University.

Baker, H.H. 1980. Edge based stereo correlation. In *Image Understanding Workshop*, L.S. Baumann (Ed.), Science Applications International Corporation, pp. 168–175.

Baker, S., Szeliski, R., and Anandan, P. 1998. A layered approach to stereo reconstruction. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'98)*, Santa Barbara.

Barnard, S.T. 1989. Stochastic stereo matching over scale. *International Journal of Computer Vision*, 3(1):17–32.

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

Belhumeur, P.N. and Mumford, D. 1992. A Bayesian treatment of the stereo correspondence problem using half-occluded regions. In *Computer Vision and Pattern Recognition*, Champaign-Urbana, Illinois, pp. 506–512.

Bergen, J.R., Burt, P.J., Hingorani, R., and Peleg, S. 1992. A three-frame algorithm for estimating two-component image motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):886–896.

Blinn, J.F. 1994a. Jim Blinn's corner: Compositing, part 1: Theory. *IEEE Computer Graphics and Applications*, 14(5):83–87.

Blinn, J.F. 1994b. Jim Blinn's corner: Compositing, part 2: Practice. *IEEE Computer Graphics and Applications*, 14(6):78–82.

Blonde, L. et al. 1996. A virtual studio for live broadcasting: The Mona Lisa project. *IEEE Multimedia*, 3(2):18–29.

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

Collins, R.T. 1996. A space-sweep approach to true multi-image matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, San Francisco, California, pp. 358–363.

Cox, I.J. 1994. A maximum likelihood n-camera stereo algorithm. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'94)*, IEEE Computer Society: Seattle, Washington, pp. 733–739.

Darrell, T. and Pentland, A. 1991. Robust estimation of a multi-layered motion representation. In *IEEE Workshop on Visual Motion*, IEEE Computer Society Press: Princeton, New Jersey, pp. 173–178.

Dhond, U.R. and Aggarwal, J.K. 1989. Structure from stereo—A review. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(6):1489–1510.

Fua, P. 1993. A parallel stereo algorithm that produces dense depth maps and preserves image features. *Machine Vision and Applications*, 6:35–49.

Geiger, D., Ladendorf, B., and Yuille, A. 1992. Occlusions and binocular stereo. In *Second European Conference on Computer Vision (ECCV'92)*, Springer-Verlag: Santa Margherita Liguere, Italy, pp. 425–433.

Golland, P. and Bruckstein, A. 1995. Motion from color. Technical Report 9513, IS Lab, CS Department, Technion, Haifa, Israel.

Gortler, S.J., Grzeszczuk, R., Szeliski, R., and Cohen, M.F. 1996. The lumigraph. In *Computer Graphics Proceedings, Annual Conference Series*, ACM SIGGRAPH, Proc. SIGGRAPH'96 (New Orleans), pp. 43–54.

Huber, P.J. 1981. *Robust Statistics*. John Wiley & Sons: New York.

Intille, S.S. and Bobick, A.F. 1994. Disparity-space images and large occlusion stereo. In *Proc. Third European Conference on Computer Vision (ECCV'94)*, Springer-Verlag: Stockholm, Sweden.

Jenkin, M.R.M., Jepson, A.D., and Tsotsos, J.K. 1991. Techniques for disparity measurement. *CVGIP: Image Understanding*, 53(1):14–30.

Jones, D.G. and Malik, J. 1992. A computational framework for determining stereo correspondence from a set of linear spatial filters. In *Second European Conference on Computer Vision (ECCV'92)*, Springer-Verlag: Santa Margherita Liguere, Italy, pp. 397–410.

Ju, S.X., Black, M.J., and Jepson, A.D. 1996. Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, San Francisco, California, pp. 307–314.

Kanade, T. and Okutomi, M. 1994. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932.

Kanade, T., Yoshida, A., Oda, K., Kano, H., and Tanaka, M. 1996. A stereo machine for video-rate dense depth mapping and its new applications. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, San Francisco, California, pp. 196–202.

Kang, S.B., Webb, J., Zitnick, L., and Kanade, T. 1995. A multi-baseline stereo system with active illumination and real-time image acquisition. In *Fifth International Conference on Computer Vision (ICCV'95)*, Cambridge, Massachusetts, pp. 88–93.

Lacroute, P. and Levoy, M. 1994. Fast volume rendering using a shear-warp factorization of the viewing transformation. *Computer Graphics (SIGGRAPH'94)*, 451–457.

Levoy, M. 1990. Efficient ray tracing of volume data. *ACM Transactions on Graphics*, 9(3):245–261.

Lucas, B.D. and Kanade, T. 1981. An iterative image registration technique with an application in stereo vision. In *Seventh International Joint Conference on Artificial Intelligence (IJCAI-81)*, Vancouver, pp. 674–679.

Marr, D. and Poggio, T. 1976. Cooperative computation of stereo disparity. *Science*, 194:283–287.

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

McMillan, L. and Bishop, G. 1995. Plenoptic modeling: An image-based rendering system. *Computer Graphics (SIGGRAPH'95)*, 39–46.

Mitsunaga, T., Yokoyama, T., and Totsuka, T. 1995. Autokey: Human assisted key extraction. In *Computer Graphics Proceedings, Annual Conference Series*, ACM SIGGRAPH, Proc. SIGGRAPH'95 (Los Angeles), pp. 265–272.

Ohta, Y. and Kanade, T. 1985. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-7(2):139–154.

Okutomi, M. and Kanade, T. 1993. A multiple baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(4):353–363.

Paker, Y. and Wilbur, S. (Eds.) 1994. Image Processing for Broadcast and Video Production, Hamburg, 1994, Springer, Hamburg. Proceedings of the European Workshop on Combined Real and Synthetic Image Processing for Broadcast and Video Production, Hamburg, 23–24 November, 1994.

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

Porter, T. and Duff, T. 1984. Compositing digital images. *Computer Graphics (SIGGRAPH'84)*, 18(3):253–259.

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

Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T. 1992. *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press: Cambridge, England.

Ryan, T.W., Gray, R.T., and Hunt, B.R. 1980. Prediction of correlation errors in stereo-pair images. *Optical Engineering*, 19(3):312–322.

Sawhney, H.S. and Ayer, S. 1996. Compact representation of videos through dominant multiple motion estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(8):814–830.

Scharstein, D. and Szeliski, R. 1996. Stereo matching with non-linear diffusion. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, San Francisco, California, pp. 343–350.

Seitz, S.M. and Dyer, C.M. 1997. Photorealistic scene reconstrcution by space coloring. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, San Juan, Puerto Rico, pp. 1067–1073.

Shade, J., Gortler, S., He, L.-W., and Szeliski, R. 1998. Layered depth images. In *Computer Graphics (SIGGRAPH'98) Proceedings*, ACM SIGGRAPH, Orlando.

Shizawa, M. and Mase, K. 1991a. Principle of superposition: A common computational framework for analysis of multiple motion. In *IEEE Workshop on Visual Motion*, IEEE Computer Society Press: Princeton, New Jersey, pp. 164–172.

Shizawa, M. and Mase, K. 1991b. A unified computational theory of motion transparency and motion boundaries based on eigenenergy analysis. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'91)*, IEEE Computer Society Press: Maui, Hawaii, pp. 289–295.

Smith, A.R. and Blinn, J.F. 1996. Blue screen matting. In *Computer Graphics Proceedings, Annual Conference Series*, ACM SIGGRAPH, Proc. SIGGRAPH'96 (New Orleans), pp. 259–268.

Szeliski, R. and Hinton, G. 1985. Solving random-dot stereograms using the heat equation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'85)*, IEEE Computer Society Press: San Francisco, California, pp. 284–288.

Szeliski, R. and Kang, S.B. 1995. Direct methods for visual scene reconstruction. In *IEEE Workshop on Representations of Visual Scenes*, Cambridge, Massachusetts, pp. 26–33.

Tian, Q. and Huhns, M.N. 1986. Algorithms for subpixel registration. *Computer Vision, Graphics, and Image Processing*, 35:220–233.

Vlahos, P. and Taylor, B. 1993. Traveling matte composite photography. In *American Cinematographer Manual*, American Society of Cinematographers, Hollywood, pp. 430–445.

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

Weinshall, D. 1989. Perception of multiple transparent planes in stereo vision. *Nature*, 341:737–739.

Weiss, Y. and Adelson, E.H. 1996. A unified mixture framework for motion segmentation: Incorporating spatial coherence and estimating the number of models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, San Francisco, California, pp. 321–326.

Wood, G.A. 1983. Realities of automatic correlation problems. *Photogrammetric Engineering and Remote Sensing*, 49(4):537–538.

Yang, Y., Yuille, A., and Lu, J. 1993. Local, global, and multilevel stereo matching. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'93)*, IEEE Computer Society: New York, pp. 274–279.

Yuille, A.L. and Poggio, T. 1984. A generalized ordering constraint for stereo correspondence. A.I. Memo 777, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.