Proceedings of the Sixth International Conference on Computer Vision (ICCV'98), January 4-7, 1998, Bo

Stereo Matching with Transparency and Matting

Richard Szeliski
Microsoft Research
One Microsoft Way
Redmond, WA 98052-6399
szeliski@microsoft.com

Polina Golland
Artificial Intelligence Laboratory
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 3-D (x, y, d) disparity space, and solve the problem using a combination of initial evidence aggregation followed by iterative energy minimization.

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 3-D scenes and environments for virtual reality and virtual studio applications. 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 [13] or iterative evidence aggregation [20] 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.

Unfortunately, for challenging applications such as *z-keying* (the insertion of graphics between different depth layers in video) [12], 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 [22]. 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* [17].

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 3-D (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.

2 Previous Work

Stereo matching (and the more general problem of stereo-based 3-D reconstruction) are fields with very rich histories [2, 5]. 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. Additional citations may be found in [23].

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 [12] or squared intensity differences [14], edges, or over an area, e.g., using correlation.

Aggregating support is necessary to disambiguate potential matches. Aggregation has been done using fixed square windows (traditional), windows with adaptive sizes [13], and iterative (non-linear) evidence aggregation [20].

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 [1], or try to recover occluded regions after the matching by cross-checking [6]. Several authors have addressed occlusions explicitly, using Bayesian models and dynamic programming [3, 7, 10]. However, such techniques require the strict enforcement of *ordering constraints*.

Finally, the topic of transparent surfaces has not received much study in the context of computational stereo [18, 24]. Relatively more work has been done in the context of transparent motion estimation [4, 11]. 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.

3 Disparity space representation

To formulate our (potentially multiframe) stereo problem, we use a *generalized disparity space* which can be any projective sampling (collineation) of 3-D space. 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 3-D space, we can make the camera orthographic, with the camera's (x, y, d) axes being orthogonal and evenly sampled (as in [21]).

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 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 [16]. 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. Note that in many imaging situations, integral steps in disparity will correspond to fractional shifts in displacement in order to achieve acceptable accuracy.

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 4-D (x, y, d, k) space. 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

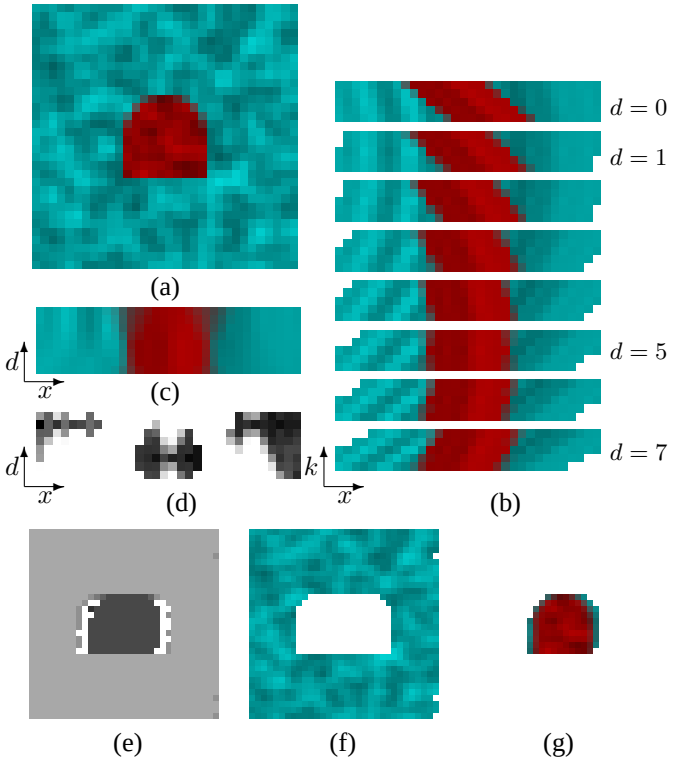

Figure 1: Sample slices through a 4-D disparity space: (a) sample input image—darker red object at $d = 5$ in front of lighter blue background at $d = 1$, (b) (x, d, k) slice for scanline 17, (c) means and (d) variances as a function of (x, d) (smaller variances are darker), (e) results of winner-takes-all for whole image (undecided columns in white), (f–g) colors and opacities at disparities 1 and 5. For easier interpretation, all images have been composited over an opaque white background (see [23] for color images).

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. Figure 1 shows a set of sample images, together with an (x, d, k) slice through the 4-D 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 4-D (x, y, d, k) space with colors obtained by resampling the K input images,

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

where $\mathbf{c}_k(u,v)$ is the kth input image,[1] $\mathbf{H}_k + \mathbf{t}_k[0\ 0\ d]$ is the homography mapping this image to disparity plane d (see Section 5), $\mathcal{W}_f$ is the forward warping operator,[2] and $\mathbf{c}(x,y,d,k)$ is the pixel mapped into the 4-D generalized disparity space.

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. Examples of the mean and variance values for our sample image are shown in Figures 1c and 1d, 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.

To help disambiguate matches, we aggregate evidence using a variant of the algorithm described in [20],

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

where σ_i^t is the variance of pixel i 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. For the results in Figure 1, 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 [14, 16]. 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.[3] 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.

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

$$\mathbf{x}_k = \mathbf{M}_k \hat{\mathbf{M}}_0^{-1} \hat{\mathbf{x}}_0 \tag{3}$$

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. 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

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

where $\mathbf{x}_0 = (x,y,1)$, and the layers are then composited back-to-front.

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

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

where $\hat{\mathbf{c}} = [r\ g\ b\ \alpha]^T$ 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). 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 [17],

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

where $\mathbf{f}$ and $\mathbf{b}$ are the premultiplied foreground and background colors, and α_f is the opacity of the foreground [17]. Using the over operator, we can form a composite image

$$\begin{aligned} \tilde{\mathbf{c}}_k(u,v) &= \bigwedge_{d=d_{\max}}^{d_{\min}} \tilde{\mathbf{c}}_k(u,v,d) \\ &= \tilde{\mathbf{c}}_k(u,v,d_{\max}) \wedge \cdots \wedge \tilde{\mathbf{c}}_k(u,v,d_{\min}) \end{aligned} \tag{6}$$

[1] The color values $\mathbf{c}$ can be replaced with gray-level intensity values without affecting the validity of our analysis.

[2] In our current implementation, the warping (resampling) algorithm uses bi-linear interpolation of the pixel colors and opacities.

[3] 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 $\theta = \theta_{\min} + \theta_s \mathrm{Var}_{3\times3}$, with $\theta_{\min} = 10$ and $\theta_s = 0.02$.

(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). To construct the visibility map, we use a recursive front-to-back algorithm

$$\begin{aligned} V_k(u, v, d-1) &= V_k(u, v, d)\,(1 - \tilde{\alpha}_k(u, v, d)) \\ &= \prod_{d'=d}^{d_{\max}} (1 - \tilde{\alpha}_k(u, v, d')), \end{aligned} \tag{7}$$

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,

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

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 (Figure 2a). 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 (Figures 2c and 2d).

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 (Figure 2b–d). Figure 2e–g 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

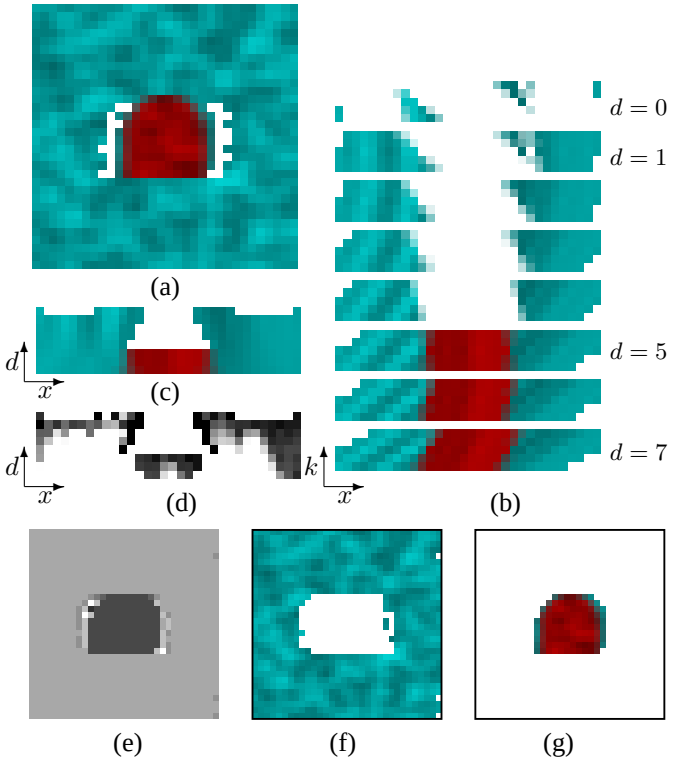

Figure 2: After modifying input images by visibility $V_k(u, v, d)$: (a) re-synthesized view of sample images, (b) (x, d, k) slice for scanline 17, (c) means and (d) variances as a function of (x, d), (e) results of winner-takes-all for whole image, and (f–g) colors and opacities at disparities 1 and 5 after one iteration of the reprojection algorithm.

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 [15], 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 reprojection), 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 reprojected images $\tilde{\mathbf{c}}_k(u, v)$ and the original input images $\mathbf{c}_k(u, v)$

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

2. a (weak) smoothness constraint on the colors and opaci-

ties,

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

3. a prior distribution on the opacities,

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

In the above equations, ρ_1 and ρ_2 are either quadratic functions or robust penalty functions [9], and ϕ is a function which encourages opacities to be 0 or 1, e.g., $\phi(x) = x(1-x)$.[4]

To minimize the total cost function

$$\mathcal{C} = \lambda_1 \mathcal{C}_1 + \lambda_2 \mathcal{C}_2 + \lambda_3 \mathcal{C}_3, \tag{12}$$

we use a preconditioned gradient descent algorithm. First, we compute the gradient of the cost function with respect to the color and opacity parameters $\hat{\mathbf{c}}(x,y,d)$. We also compute the diagonal of the approximate Hessian matrix[19, pp. 681-685]. Then, we divide the gradient by the diagonal of the Hessian, and adjust the color and opacity values accordingly. A complete description of this procedure is given in [23].

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.[5] We should emphasize that our algorithm does not require colored images as inputs (see Figure 5), nor does it require the use of standard epipolar geometries.

The first stimulus we generated was a traditional random-dot stereogram, where the choice of camera geometry and filled disparity planes results in integral pixel shifts. This example also contains no partially transparent pixels. Figure 3 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). 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 $d = 3$.

[4] All color and opacity values are, of course, constrained to lie in the range $[0,1]$, making this a constrained optimization problem.

[5] For a color version of these figures, please see [23].

As we can see in Figure 3, 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 3b 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 $\mathcal{C}_2$ (Figure 3c) ameliorates both of these problems. Adding the (weak) 0/1 opacity constraint $\mathcal{C}_3$ (Figure 3d–e) further removes stray pixels at wrong disparity levels.

For comparison, Figure 3f shows the results of a traditional winner-take-all algorithm (the same as Figure 3a 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 Figures 1 and 2. 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 4a 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 4b 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 $\mathcal{C}_2$ (Figure 4c) ameliorates both of these problems. Adding the (weak) 0/1 opacity constraint $\mathcal{C}_3$ (Figure 4d) 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 background, as in the last column). As before, Figure 4e shows the results of a traditional winner-take-all algorithm.

Figure 5 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

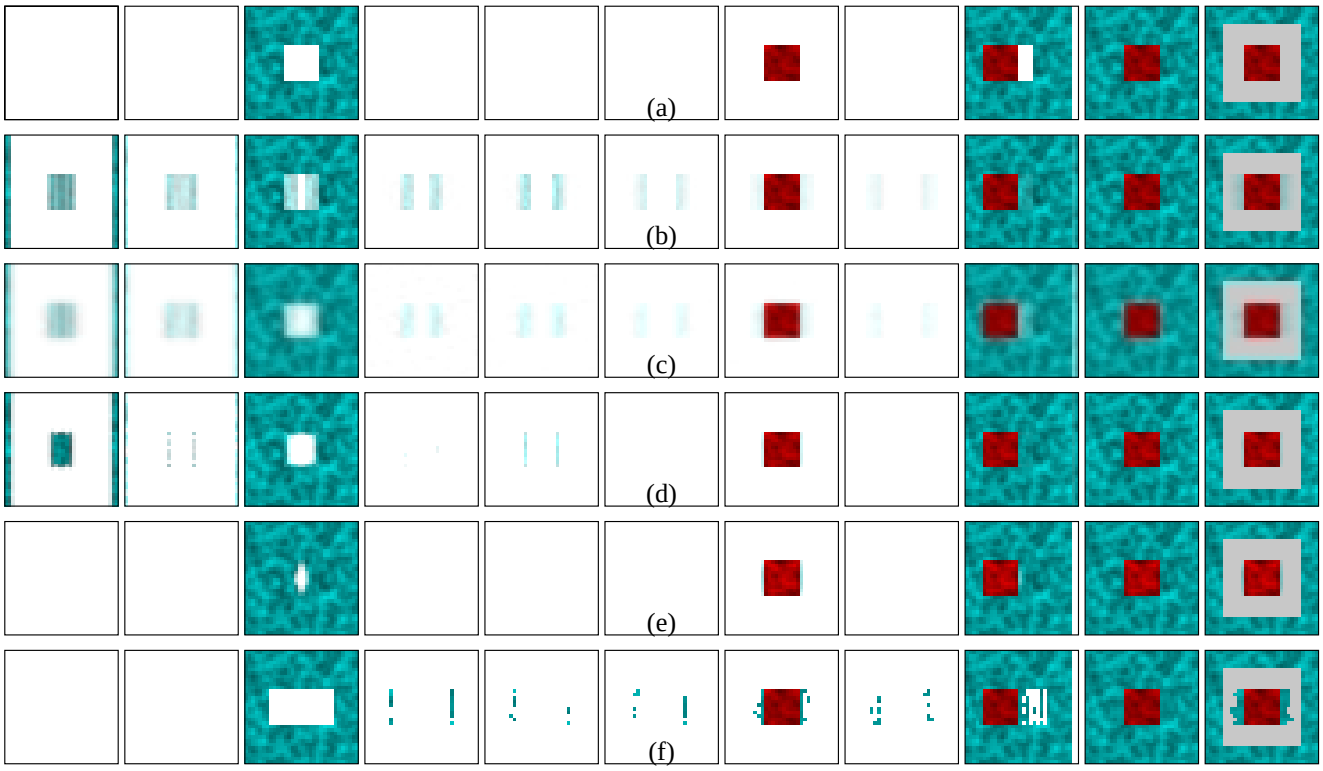

Figure 3: Traditional synthetic RDS results: (a) after iterative aggregation but before gradient descent, (b) without smoothness or opacity constraint, $\lambda_1 = 1, \lambda_2 = \lambda_3 = 0$, (c) without opacity constraint, $\lambda_1 = \lambda_2 = 1, \lambda_3 = 0$, (d) with all three constraints, $\lambda_1 = 50, \lambda_2 = 1, \lambda_3 = 50$, (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 $d = 3$.

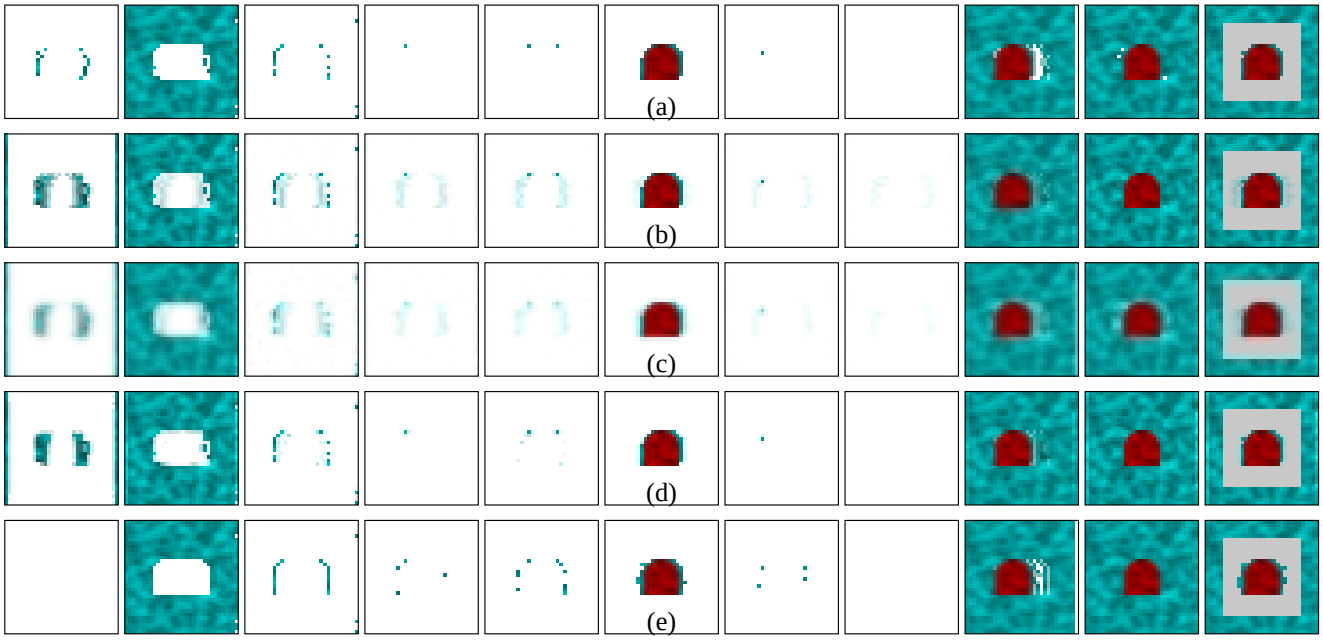

Figure 4: More challenging synthetic RDS results: see above caption for description of (a–d); (e) simple winner-take-all (shown for comparison).

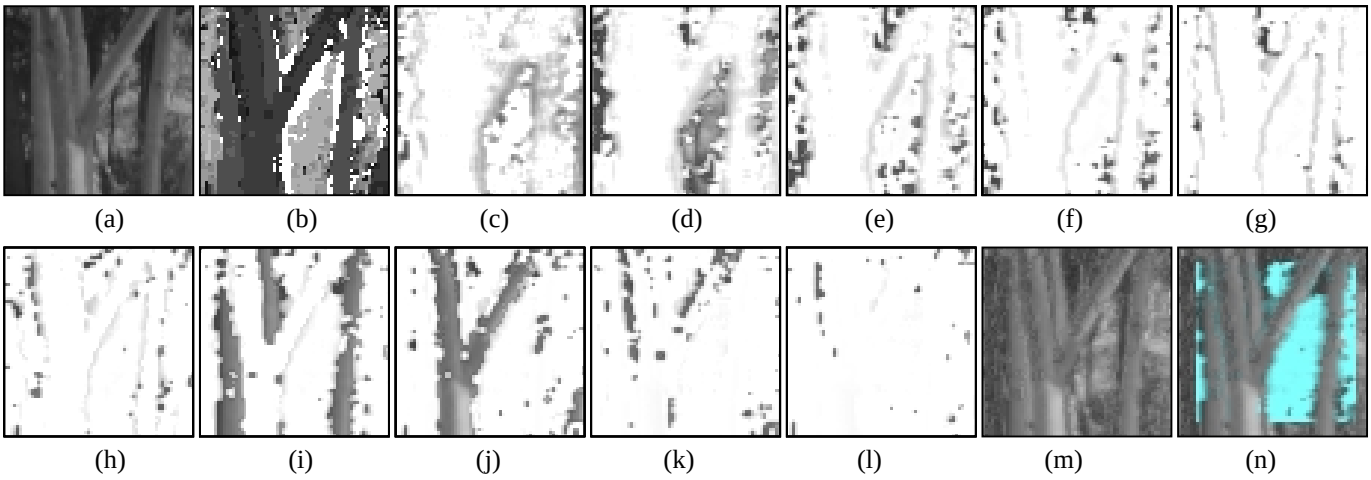

Figure 5: Real image example: (a) cropped subimage from *SRI Trees* data set, (b) depth map after initial aggregation stage, (c–l) disparity layers $d = 0 \ldots 9$, (m) re-synthesized input image, (n) with inserted $d = 4$ blue layer.

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.

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 re-sampling 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 plan to investigate. For example, we plan to try our algorithm on data sets with true transparency (not just mixed pixels), such as traditional transparent random dot stereograms [18] and reflections in windows [4].

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 [14]. 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.

Instead of representing our color volume $\hat{\mathbf{c}}(x, y, d)$ using colors pre-multiplied by their opacities, 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 [8], and they would permit us to reason more directly about underlying physical processes (shadows, shading,*etc.*).

We plan to investigate the relationship of our new disparity space model to more traditional layered motion models. We also plan to make more principled use of robust statistics, and investigate alternative search algorithms such as multiresolution (pyramidal) continuation methods and stochastic (Monte Carlo) gradient descent techniques.

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 3-D. 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.

References

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

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

[3] P. N. Belhumeur and D. Mumford. A Bayesian treatment of the stereo correspondence problem using half-occluded regions. In *Computer Vision and Pattern Recognition*, pages 506–512, 1992.

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

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

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

[7] D. Geiger, B. Ladendorf, and A. Yuille. Occlusions and binocular stereo. In *Second European Conference on Computer Vision (ECCV'92)*, pages 425–433, May 1992.

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

[9] P. J. Huber. *Robust Statistics*. John Wiley & Sons, New York, New York, 1981.

[10] S. S. Intille and A. F. Bobick. Disparity-space images and large occlusion stereo. In *Proc. Third European Conference on Computer Vision (ECCV'94)*, volume 1, May 1994.

[11] S. X. Ju, M. J. Black, and A. D. Jepson. 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)*, pages 307–314, June 1996.

[12] T. Kanade et al. 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)*, pages 196–202, June 1996.

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

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

[15] T. Mitsunaga, T. Yokoyama, and T. Totsuka. Autokey: Human assisted key extraction. In *Computer Graphics Proceedings, Annual Conference Series*, pages 265–272, August 1995.

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

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

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

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

[20] D. Scharstein and R. Szeliski. Stereo matching with non-linear diffusion. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'96)*, pages 343–350, June 1996.

[21] S. M. Seitz and C. M. Dyer. Photorealistic scene reconstrcution by space coloring. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97)*, pages 1067–1073, June 1997.

[22] A. R. Smith and J. F. Blinn. Blue screen matting. In *Computer Graphics Proceedings, Annual Conference Series*, pages 259–268, August 1996.

[23] R. Szeliski and P. Golland. Stereo matching with transparency and matting. Technical Report MSR-TR-97-13, Microsoft Research, May 1997. http://www.research.microsoft.com/pubs/msr-bib.htm

[24] R. Szeliski and G. Hinton. Solving random-dot stereograms using the heat equation. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'85)*, pages 284–288, June 1985.