Stereo Matching with Reflections and Translucency

Yanghai Tsin[†] Sing Bing Kang[‡] Richard Szeliski[‡]

[†] The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
ytsin@cs.cmu.edu

[‡]Interactive Visual Media Group
Microsoft Research
Redmond, WA 98052
{sbkang, szeliski}@microsoft.com

Abstract

In this paper, we address the stereo matching problem in the presence of reflections and translucency, where image formation can be modeled as the additive superposition of layers at different depth. The presence of such effects violates the Lambertian assumption underlying traditional stereo vision algorithms, making it impossible to recover component depths using direct color matching based methods. We develop several techniques to estimate both depths and colors of the component layers. Depth hypotheses are enumerated in pairs, one from each layer, in a nested plane sweep. For each pair of depth hypotheses, we compute a component-color-independent matching error per pixel, using a spatial-temporal-differencing technique. We then use graph cut optimization to solve for the depths of both layers. This is followed by an iterative color update algorithm whose convergence is proven in our paper. We show convincing results of depth and color estimates for both synthetic and real image sequences.

1. Introduction

Stereo matching is one of the central problems in computer vision research. For the past several decades intensive research has been conducted in this area [9]. Stereo algorithms have shown improving accuracy and steady progress in handling difficult problems such as textureless regions and occluding boundaries. For example, recent progress in efficiently solving Markov Random Field problems has significantly improved stereo matching algorithm performance [3].

One of the remaining problems in this area is how to handle commonly occurring non-Lambertian effects such as partial filling [13, 14, 10], translucency [1], and reflection [11]. Proper modeling and analysis of these effects are essential for many vision and graphics applications, including video editing, image based rendering and video compression. We present in this paper several techniques to solve one class of such problems, namely, stereo matching in the presence of reflections and translucency.

Szeliski *et al.* [12] proposed an iterative method for recovering component layer depths and colors for the special case of global (parametric) motion. Their method is not designed to handle optical flow like motions such as those introduced by piece-wise planar or curved surfaces, nor can it handle occlusions. Ju *et al.* [5] used a layered piecewise parametric (affine) motion model to simultaneously estimate multiple motions in both occlusive and transparent motion sequences. Their method, however, does not produce the same accurate per-pixel disparity estimates as the method developed in this paper. Swaminathan *et al.* [11] handled the problem of highlight detection and removal by epipolar plane image (EPI) analysis, i.e., explicitly detecting the saturated highlights in a local EPI strip. In our case, image sequences are assumed to not be saturated, so there are no consistent cues such as saturated strips to be exploited.

2. Problem Definition

According to the law of reflection, a light ray's angle of reflection is equal to its angle of incidence. For a planar reflector, it is not difficult to show that the position of a scene point's virtual image is independent of the viewpoint, as if the reflection were at a fixed position behind the reflector (Figure 1(a)). We can thus ignore the dependency between a scene point and its reflection, and use the *layered model* to depict the reflector (*frontal layer* I_0) and reflection (*rear layer* I_1) (Figure 1(b)), and model the observed images as a composition of the two layers. For curved reflectors, as long as the image sequence is taken within a small temporal window, the layered model can still be applied using a model consisting of instantaneous depths and instantaneous colors [11]. The layered model also applies to translucency.

When light is reflected from a glossy surface or transmitted through a translucent colored surface, we get the linear superposition of two (real or virtual) images

$$C = I_0 + \beta_0 I_1. \quad (1)$$

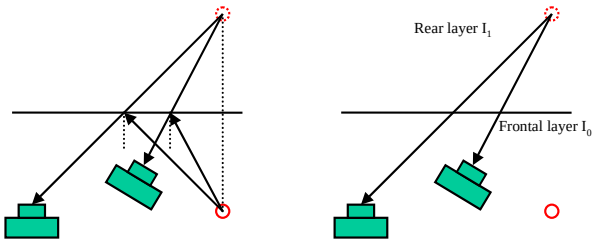

Figure 1: Modeling reflection using a layered model. *(a) The reflection model. The position of the virtual image is independent of the viewpoint. (b) The equivalent layered model.*

where C is the composite color, I_0 is the image associated with the *frontal* (reflecting/translucent) layer, I_1 is the image associated with the reflected or transmitted layer, and β_0 is an attenuation factor that is a property of the frontal layer. For a photo behind a reflecting glass surface (Figure 7), I_0 denotes the photo and I_1 denotes the reflection ($\beta_0 = 1$ in the glass area, and $\beta_0 = 0$ in the other matte regions). For a translucent material (Figure 3(b)), I_0 is the colored material, while I_1 denotes the transmitted image.

When a sequence of images is taken from different viewpoints, we observe relative motions between the two layers due to their different apparent depths in the scene. We denote the frontal and rear layer depths as d_0 and d_1, and the corresponding warping function as $T_f(d_0)$ and $T_f(d_1)$, where f is an index for the camera position. A sequence of images $\{C_f\}$ can thus be modeled by blending the two warped layers,

$$C_f = T_f(d_0) \circ I_0 + (T_f(d_0) \circ \beta_0)(T_f(d_1) \circ I_1). \quad (2)$$

(Note that β_0 is a property of the frontal layer and must be warped according to the frontal layer depth.) For simplicity, we denote $T_f(d_0) = T_{f0}$ and $T_f(d_1) = T_{f1}$, which represent the warping functions from reference frame k to frame f using the frontal and rear depth maps, respectively.

Given the rendering equation (2), our problem proceeds in the other direction. Given a sequence of images $C_f(x)$, where $f = 0, ..., K-1$, we pick the middle image to be the reference frame $k = K/2$. The goal is to recover d_0, d_1, I_0, I_1 and β_0 from the sequence of observed images.

In the next section, we describe our new algorithm to estimate depth maps for two-layer models. We introduce an iterative color estimation method in section 4, given a known β-map. The β-map estimation is postponed to section 5. Experimental results are given in section 6.

3. Depth Estimation

Our novel depth estimation algorithm has three steps: enumerating the depth hypotheses, computing a matching error for each depth hypothesis, and depth estimation using a graph cut algorithm.

3.1. Nested Plane Sweep

Enumerating depth hypotheses in a traditional stereo matching problem is trivial. The search space per pixel is simply the full range of possible disparities, $d = d_{\min}, \ldots, d_{\max}$. However, the search space for the new problem is much larger. At each pixel, we need to determine a pair of depth hypotheses (d_0, d_1), under the constraint $d_0 \geq d_1$. If there are D discrete disparity levels, the number of depth hypotheses is $D(D+1)/2$. Without any prior knowledge, it is necessary to consider all depth pairs.

The depth hypotheses are enumerated in a plane sweep fashion [4]. Without loss of generality, we let the sweeping of the frontal plane be dominant. We warp all the input images to the reference view according to the frontal disparity d_0. With d_0 fixed, we conduct a second pass of sweeping for all valid $d_1 \leq d_0$. We can think of this method as sweeping the depth space using two planes simultaneously, with the rear plane d_1 nested within the frontal plane d_0. We call this extension of plane sweep the *nested plane sweep* method.

3.2. Computing Initial Matching Errors

A more challenging problem is to compute a meaningful color matching error without knowing the component colors in either layer. To make the problem tractable, we make a local piece-wise fronto-parallel scene assumption. This assumption introduces spatial constancy in a local neighborhood and makes the component motion predictable. We only need this assumption to hold in a small spatio-temporal window, so our method can deal with curved surfaces.

When have guessed the right depth values, we expect to see small matching errors across multiple views (Figure 2). Consider two different viewing rays that pass through the same rear layer scene point R from two different viewpoints f and $f+1$. Since these rays also pass through frontal points P and Q, respectively, we denote them as rays (f, P) and $(f+1, Q)$. Let us compute the color difference along the two viewing rays, $D_f = C_{f,P} - C_{f+1,Q}$. This difference is computed for each depth hypothesis (d_0, d_1). At the right depth, the difference is independent of the rear layer color, i.e.,

$$\begin{aligned} D_f &= C_{f,P} - C_{f+1,Q} \\ &= (I_0(P) + I_1(R)) - (I_0(Q) + I_1(R)) \\ &= I_0(P) - I_0(Q). \end{aligned} \quad (3)$$

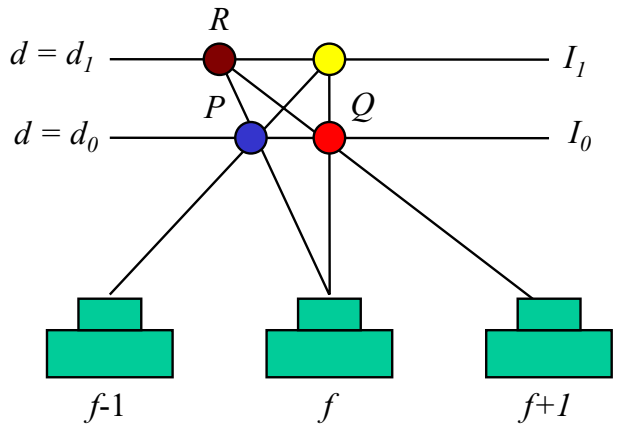

Figure 2: *Spatial-temporal differencing technique for estimating initial matching errors.*

Similarly, it can be shown

$$D_{f-1} = C_{f-1,P} - C_{f,Q} = I_0(P) - I_0(Q) = D_f. \quad (4)$$

Without noise, $\{D_f\}$ is a constant sequence. We can thus use the variance of the sequence $\{D_f\}$ as the initial matching error. The variance is expected to be small if the correct depths have been recovered.

Note that this variance is similar to the sum of summed squared differences (SSSD) measure introduced in [7], but does not rely on subtracting frames from a specified keyframe. Using pixel variance (across time/views) as a measure of matching consistency has been suggested by a number of previous researchers (see, e.g., [13]). Since the difference D_f uses viewing rays from different view points (temporal axis) and different neighboring pixels (spatial axis), we call it the *spatial-temporal differencing* method.

To handle occlusions in both layers, we adopt the spatial-temporal selection method of [6]. In our case, spatial selection involves choosing the smaller variance of $\{C_{f,P} - C_{f+1,Q}\}$ and $\{C_{f,P} - C_{f-1,Q}\}$, and temporal selection involves selection from matching errors of sub-sequences $f = 0, 1, 2, \ldots, K/2 - 1$ and $f = K/2, \ldots, K - 1$.

3.3. Estimating Depths Using Graph Cuts

The initial matching errors are stored in a 3D volume called the *disparity space image* (DSI). The DSI is a function of the pixel location and the depth labels (d_0, d_1). The DSI is the evidence derived from the input images supporting each depth hypothesis.

The stereo matching problem is known to be under-constrained. Small matching error is only a necessary condition for a correct depth. To resolve the ambiguity, other constraints, such as the smoothness constraint, need to be added. Two neighboring pixels should have the same depth unless there is strong evidence supporting the separation. When the smoothness is enforced, the stereo problem is equivalent to finding the optimal labeling of a multiple label Markov Random Field. We adopt the graph cut based method of [3] to solve this combinatorial optimization problem. The graph cut method we adopted (which uses the $\alpha - \beta$ swapping algorithm) has been shown to produce high quality depth maps.

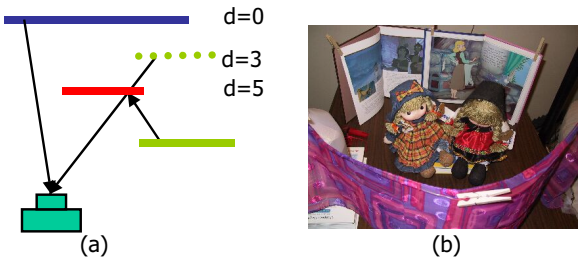

Figure 3: Experimental setup for *(a) the random dot sequence and (b) the doll sequence (Fig. 9(c)).*

We illustrate our algorithm in Figure 4 using a sequence of random dot images. The experimental setup is shown in Figure 3(a). Notice in this first example that each depth pair (d_0, d_1) is treated as a single label. The depth label output by the graph cut determines the depths of both layers simultaneously. Two frames from the five frame random dot sequence are shown in Figure 4(a)-(b). A frontal layer (disparity 5) is composed of a planar mirror with red random dots. Between the frontal layer and the background (blue pixels, disparity 0), there is a reflected random dot layer (green pixels, disparity 3). Due to the mixing of the frontal layer and the reflected layer, the original graph cut based algorithm fails to estimate depths in the mixing region (see Figure 4(c)). The proposed method generates a pair of clean depth maps shown in Figure 4(d)-(e). This result may appear to be incorrect because the estimated frontal layer depth is 6 corresponding to the background area, instead of being 0 as expected. However, this is due to the fundamental ambiguity of the image formation process, as shown in the following Lemma.

Lemma 1 Ambiguity of the Single Layer Model. *An image sequence of any single layer model is equivalent to image sequences of a class of two layer models, each with a textureless reflective/translucent frontal layer.*

The proof of the Lemma is straightforward. We can always insert a planar layer between the camera and the scene point P and consider the image as being the reflection of scene point P' in Figure 5(b), or through transparent material in 5(c). The seemingly wrong frontal depth explanation in Figure 4(d)-(e) is just one such two layer model.

The ambiguity problem for this simple noiseless case can be easily fixed. We can add a prior favoring single layer

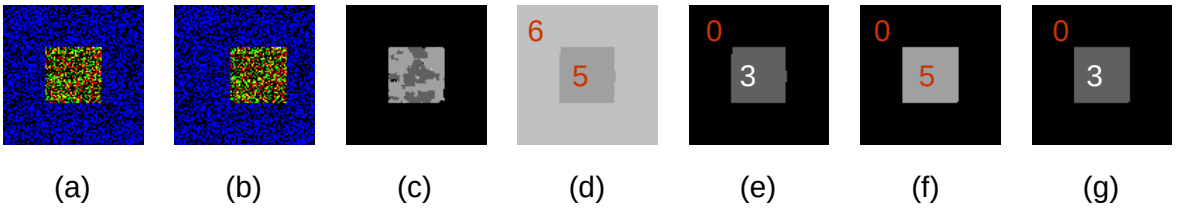

Figure 4: Random dot stereo matching with reflections. *Numbers in (d)-(g) are estimated disparities.(a)-(b) Frame 3 and 5 of a five frame image sequence. (c) Failed depth estimation using traditional stereo algorithm. (d)-(e) Estimated frontal and rear depths without adding single layer bias. (f)-(g) Estimated frontal and rear depths with added single layer bias.*

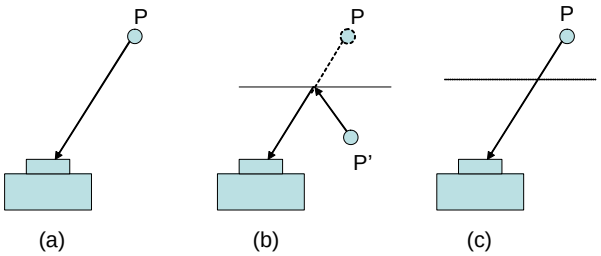

Figure 5: Ambiguity of the single layer models: *(a) A single layer model for a scene point. (b) The equivalent reflective model. (c) The equivalent translucent model.*

models. Whenever a single layer model and a two layer model explain the observed data equally well, we choose the single layer model. This approach is similar in philosophy to the model selection methods of [8]. Simpler models tend to better explain unseen data.

In our formulation, we denote the depth hypotheses where $d_0 = d_1$ as single layer models. Once we add a constant bias to the matching errors of two layer models ($d_0 \neq d_1$), we get the desired clean depth map shown in Figure 4(f)-(g).

3.4. More efficient depth estimation

For more complex scenes, we can choose not to store the matching errors corresponding to all (d_0, d_1) pairs. There are two reasons for this decision. First, storing all (d_0, d_1) pairs is not efficient in terms of storage and computational cost. Assuming fixed image size, the memory required for a DSI with D disparity levels is $O(D^2)$, while the computational cost for graph cut is $O(D^4)$.

Second, it is difficult to model partial smoothness in an energy minimization framework if we treat (d_0, d_1) as a single label. For example, it is straightforward to define the smoothness cost for neighboring pixels x_1 and x_2 in Figure 6, since $(d_0(x_1), d_1(x_1)) = (d_0(x_2), d_1(x_2))$. But it is not clear how to define the error for neighboring pixels x_2 and x_3 in the $\alpha - \beta$ swap algorithm, where $d_0(x_2) = d_0(x_3)$, but $d_1(x_2) \neq d_1(x_3)$.

Our solution is to keep two separate DSIs for d_0 and d_1.

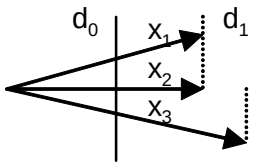

Figure 6: Partial (frontal) connectivity of real world scenes. *Smooth frontal layer, broken rear layer.*

We apply nested plane sweep and spatial-temporal differencing as usual, but keep only the minimum matching errors, i.e.,

$$\begin{aligned} DSI_0(d_0, x) &= \min_{d_1} e(d_0, d_1, x) \\ DSI_1(d_1, x) &= \min_{d_0} e(d_0, d_1, x) \end{aligned} \tag{5}$$

where $e(d_0, d_1, x)$ is the matching error corresponding to depth hypothesis (d_0, d_1). We expect $DSI_{0,1}(d, x)$ to be consistently low over a range at the correct disparity, thus enabling the graph cut algorithm to find the right answers.

As a result of this min finding step, we only need to keep two DSIs, each of which requires a storage of $O(D)$, and apply the graph cut on each of the DSIs to get the two depth maps, each at a computational cost of $O(D^2)$. At the same time, the partial smoothness problem is solved because we consider the two DSIs separately.

Although the min finding step is non-optimal, we find it works well in practice, as shown in section 6.

4. Estimating the Component Colors

We formulate the component color estimation problem as an optimization problem. The cost function is defined to be the prediction error between the synthesized image

$$\tilde{C}_f = T_{f0} \circ I_0 + (T_{f0} \circ \beta_0)(T_{f1} \circ I_1) \tag{6}$$

and the observed image C_f, i.e.,

$$\mathcal{C}(I_0, I_1) = \sum_f \left\| C_f - \tilde{C}_f \right\|^2 . \tag{7}$$

We have to ensure that each pixel x in view f used in (7) is valid, i.e., it has to be observed at the reference view. To prevent occluded pixels from contributing, we perform a visibility check by forward warping from the reference view to view f and then warping back from view f to the reference view. Missing pixels (forming gaps or holes) in the reference view are deemed invalid. This check is done separately for both frontal and rear layers.

For a given pixel x, we define two sets of views $\mathcal{V}_0(x)$ and $\mathcal{V}_1(x)$. $\mathcal{V}_0(x)$ is related to the estimation of the frontal color at frame f, i.e., $C_f - T_{f1} \circ I_1$:

$$\mathcal{V}_0(x) = \{f | x \text{ valid in } T_{f0}^{-1}(V(C_f) \cap V(T_{f1} \circ I_1))\}, \quad (8)$$

where $V(I)$ is a binary image with valid pixels of I set to 1 and invalid pixels of I set to 0.

Let us define $\tilde{I}_0$ as the region of the frontal image where $\beta_0 = 1$. $\mathcal{V}_1(x)$ is related to the estimation of the rear color at frame f, i.e., $C_f - T_{f0} \circ \tilde{I}_0$, and is similarly defined as

$$\mathcal{V}_1(x) = \{f | x \text{ valid in } T_{f1}^{-1}(V(C_f) \cap V(T_{f0} \circ \tilde{I}_0))\}. \quad (9)$$

We define the *valid view set of* x as $\mathcal{V}(x) = \mathcal{V}_0(x) \cap \mathcal{V}_1(x)$, and the *upgradeable pixel set* as $\mathcal{P} = \{x | \beta_0(x) = 1 \text{ and } \mathcal{V}(x) \neq \emptyset\}$. The definition of $\mathcal{P}$ and $\mathcal{V}$ ensures that steps 2 and 3 in the following algorithm are optimizing the same cost function.

When the depths d_0 and d_1 are fixed, the two view sets $\{\mathcal{V}_0(x), \mathcal{V}_1(x)\}$ and the upgradeable pixel set $\mathcal{P}$ can be precomputed.

Having defined our cost function, we now present our color updating algorithm as follows:

Algorithm 1 *Component Color Estimation Algorithm*

1. Estimate an initial frontal color $I_0^{(0)}$ using the min-composite algorithm [12], and let $I_1^{(0)} = C_k - I_0^{(0)}$.

2. Fix $I_0^{(n)}$ and update $I_1^{(n)}$ as follows:

$$\hat{I}_1^{(n)}(x) = \frac{\sum_{f \in \mathcal{V}(x)} T_{f1}^{-1} \circ (C_f - T_{f0} \circ I_0)(x)}{\|\mathcal{V}(x)\|} \quad (10)$$

$$I_1^{(n+1)}(x) = \max\{\min\{\hat{I}_1^{(n)}(x), 255\}, 0\} \quad (11)$$

   The updating equation (10) will become clear once Lemma 2 is proven.

3. Fix $I_1^{(n+1)}$ and update $I_0^{(n)}$ symmetrically (use the same equations as step 2, reversing the roles of frontal layer 0 and rear layer 1).

4. Repeat steps 2 and 3 until I_0 and I_1 converge.

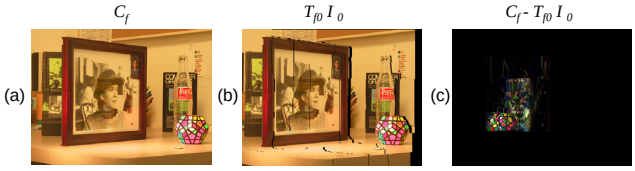

Figure 7: Color updating illustration. *(a) Observed view. (b) Forward warped frontal color. (c) Difference image, which provides color estimates for the rear layer when warped back to the reference view.*

An example of our component color estimation algorithm in action is shown in Figure 7. The difference between image C_f and the warped frontal color $T_{f0} \circ I_0$ gives us an estimate of the warped rear colors $T_{f1} \circ I_1$. We can then warp these estimates back by $T_{f1}^{-1}(C_f - T_{f0} \circ I_0)$ to provide noisy estimates of I_1. The updated estimates of I_1 will just be the average of all these noisy estimates.

Lemma 2 Convergence of the Component Color Estimation Algorithm. *Cost function (7) converges to a local minimum using the above component color estimation algorithm.*

Proof. We need to prove that each iteration of steps 2 and step 3 decreases the cost function $\mathcal{C}$. Since $\mathcal{C}$ has a lower bound ($\mathcal{C} \geq 0$), it must converge to a fixed point (local minimum). Due to the symmetry of steps 2 and 3, we only show that step 2 decreases the cost function.

According to the definition of $\mathcal{P}$ and $\mathcal{V}$, the warped difference image

$$T_{f1}^{-1} \circ \left(C_f - \tilde{C}_f\right) = T_{f1}^{-1} \circ (C_f - T_{f0} \circ I_0 - T_{f1} \circ I_1)$$

is defined for all $x \in \mathcal{P}$. As a result,

$$\mathcal{C}(I_0, I_1) = \sum_f \left\| T_{f1}^{-1} \circ \left(C_f - \tilde{C}_f\right) \right\|^2$$

$$= \sum_{x \in \mathcal{P}} \sum_{f \in \mathcal{V}(x)} \left\| T_{f1}^{-1} \circ (C_f - T_{f0} \circ I_0)(x) - I_1(x) \right\|^2$$

When we fix I_0, the color estimates at pixels $x \in \mathcal{P}$ are independent of each other. It's now easy to show that the above cost function is a quadratic function, whose minimum $\hat{I}_1$ is computed by (10). Since the min-composite provides us with initial color estimates within [0,255], the clamping operation (11) will always find an updated color in the negative gradient direction, but between $I_1^{(n)}$ and $\hat{I}_1^{(n)}$. □

To alleviate the streaking effect caused by pixels interacting mostly in the direction of motion [12], we add a regularization term that encourages intensity similarity between

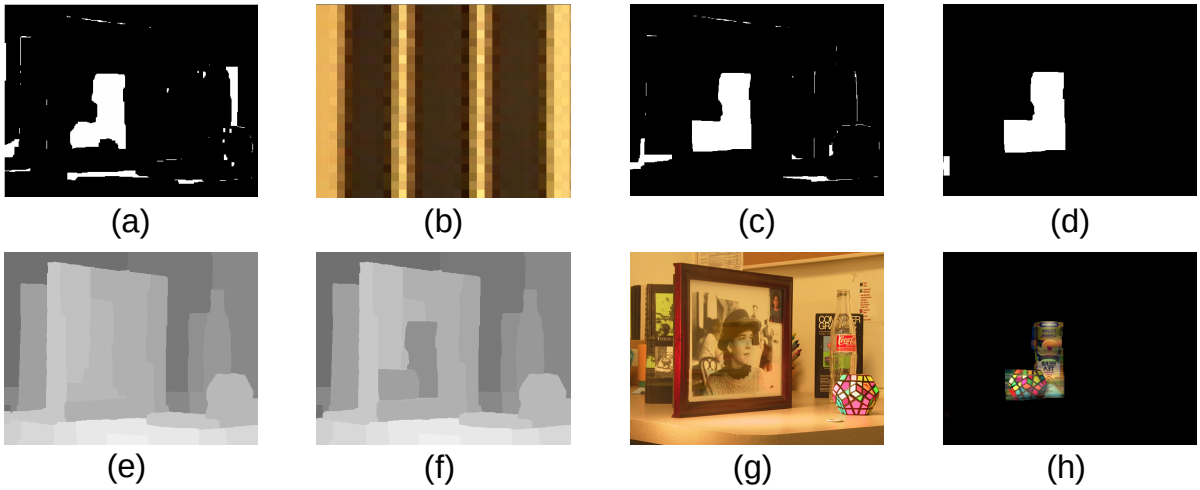

Figure 8: Automatic segmentation of single/two layer regions. Black: single-layer regions. White: two-layer regions. *(a)Segmentation using the original DSI. (b) Aliasing in a stabilized EPI strip. (c) Segmentation using scaled DSI. (d) Cleaned up β-map. (e)-(h) estimated depths and colors based on the automatically estimated β-map: (e) frontal depth, (f) rear depth, (g) frontal color, and (h) rear color.*

neighboring rows. If the pixels neighboring x are denoted by $\mathcal{N}(x)$, the regularization term can be written as

$$R(I_0, I_1) = \sum_{x \in \mathcal{P}} \sum_{y \in \mathcal{N}(x)} \|I_0(x) - I_0(y)\|^2 + \|I_1(x) - I_1(y)\|^2, \tag{12}$$

and the new cost function is now defined as

$$\mathcal{C}(I_0, I_1) = \sum_f \left\| C_f - \tilde{C}_f \right\|^2 + \lambda R(I_0, I_1). \tag{13}$$

In our experiments, we choose $\lambda = 0$ (no regularization) for the random dot sequence, and $\lambda = 1$ for all other sequences.

To solve the regularized problem we can use the same color updating strategy, namely, alternately updating I_0 and I_1. With the introduction of the regularization terms, color estimates of neighboring pixels are coupled. Fortunately, they can be efficiently estimated by solving a sparse linear system of equations. Convergence of the estimates can be similarly proven.

5. Determining the Number of Layers

Many real world scenes contain both single layer regions (non-reflective and non-translucent) and two layer regions (reflective or translucent). Figure 9(d) (the *Lee* sequence) shows one such example. To avoid false frontal layers (Lemma 1), it is desirable to explicitly segment these two types of regions.

In section 3.3, we went through a simple case where no aliasing is introduced and no noise is involved. The segmentation can easily be done by introducing a bias term that favors single layer models. After we get two depth maps, we label pixels with $d_0 = d_1$ as single layer model pixels.

In real images, however, aliasing effects and noise complicate the situation. We cannot rely on a constant bias term to segment the two types of regions. Figure 8(a) shows segmentation results for the *Lee* sequence using a constant bias. The bias term causes part of the tea box reflection (a two-layer region) to become a single-layer region, while allowing part of the computer graphics book (a single-layer region) to have a false frontal layer.

The primary reason for this failure is aliasing, which is illustrated in Figure 8(b) using an EPI strip. Vertical lines correspond to stabilized pixels. Due to aliasing, the colors of these vertical lines change considerably from view to view, making the matching errors for single-layer hypotheses very large. On the other hand, two-layer hypotheses are more expressive since they can model some of these color changes using an additional layer of component colors. In the *Lee* sequence, the adoption of a two-layer model causes a greater reduction in matching error for aliased pixels than for reflected pixels. As a result, a single bias term cannot segment the two regions.

Our solution is to scale the original DSI, or equivalently, to adapt the bias term based on local gradients. We change the DSI volume as follows:

$$e(d_0, d_1, x) \Leftarrow (1 - \gamma(x)) e(d_0, d_1, x), \tag{14}$$

where $\gamma(x) \in [0, 1)$ is proportional to the intensity gradient of pixel x along the epipolar line direction.

The scaling operation (14) does not change the ordering of $e(d_0, d_1, x)$. It only changes the dynamic range of the

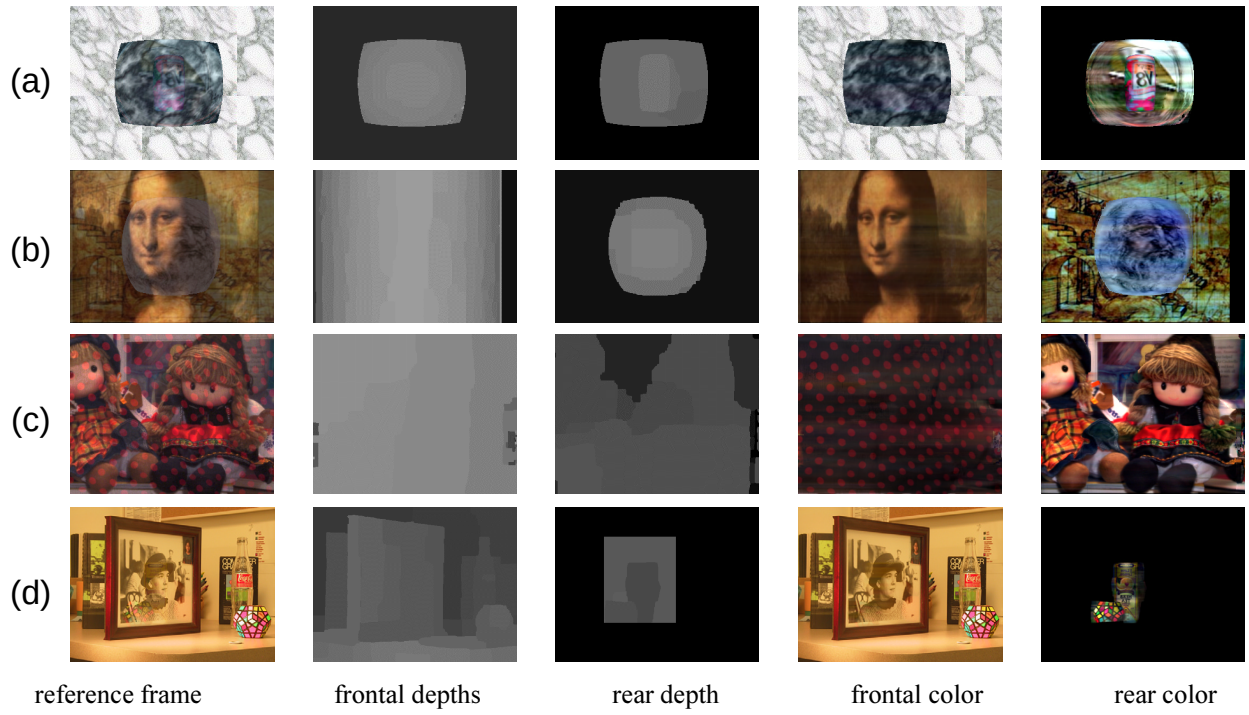

Figure 9: Experimental results on four test sequences. *(a)-(b) are synthetic sequences, (c)-(d) are real sequences. For display purpose, the images in the last column are scaled up by a factor of 4 in (a)-(b), and 2 in (c)-(d).*

matching errors and adjusts their weights in graph cut operations. It therefore reduces the influence of the high contrast edges, which are most vulnerable to aliasing.

After scaling of the DSI volume, we are able to segment the two types of regions using a single bias term (Figure 8(c)). By applying the morphological operations of erosion followed by dilation, we were able to estimate the β-map shown in Figure 8(d), and the depths and colors shown in 8(e)-(h).

6. Experiments

In this section, we present four sets of experimental results, two synthetic and two real. The results are shown in Figure 9.

The first column of Figure 9 shows the reference frame image. The second to fifth columns are the estimated frontal depth, rear depth, frontal color and rear color. In these experiments, the β-maps are given as input. The β-maps for the first three examples are trivial, i.e., we set the whole image to $\beta_0 = 1$. We add a constant bias to segment the single/two layer model regions. In the fourth example, we set $\beta_0 = 1$ in the glass area of the picture frame and $\beta_0 = 0$ elsewhere.

Figure 9 (a) shows the result on a synthetic curved reflector sequence. In the foreground is a slab of curved marble, which reflects a V8 can and a picture of a flower garden. The can is closer to the marble so it occludes part of the picture. Figure 9 (b) shows results on a synthetic translucent sequence. The foreground is a translucent layer texture mapped with the painting *Mona Lisa*. Behind the translucent layer is a "sculpture" of da Vinci (texture-mapped sphere) and one of his drawings at a distance. Figures 9 (c)-(d) are real image sequences. The experimental setup for the doll sequence (c) is shown in Figure 3(b).

Notice in all the results that occlusions in both layers are correctly handled. Figure 10 shows synthesized frames alongside the original views. Except at the large depth discontinuity boundaries, where invisible pixels are drawn in black, few artifacts are apparent. The occlusion effects of the tea box and the dodecahedron were faithfully reproduced.

Figure 11 shows some results of the color update algorithm. Min-composite initialization (Figure 11(a)) provides us with a noisy rear color image. Part of the tea box is missing and the color does not look right. After the color update, we get cleaner versions of the reflection, shown in Figure 11(b) and (c). The updated color estimate with regularization (and less streaking effects) is shown in Figure 11(b).

7. Conclusion

In this paper, we have developed a set of techniques to solve the stereo matching problem in the presence of reflection and translucency. The main contributions of our work include:

Figure 10: Synthesized views. *The three rows correspond to views 4, 8 and 16. The first column is the original images and the second column is the synthesized results.*

1. A nested plane sweep framework to estimate component depths in a systematic way, deriving evidences locally, while integrating them globally.
2. A component-color-independent color matching algorithm, the spatial-temporal differencing technique.
3. An iterative color updating algorithm that is guaranteed to converge.
4. A segmentation algorithm to automatically determine the number of layers.

When judiciously combined together, these techniques allow us to estimate accurate stereo correspondence for complex scenes that include multiple visual phenomena occurring at different apparent depths.

In our current work, we have assumed that the β-map is binary, i.e., that in regions where reflection or translucency occur, the strength of the gloss or attenuation is spatially constant. In future work, we plan to relax this assumption by allowing fractional values in the β-map. In addition, we would like investigate the feasibility of recovering more than two layers in order to model inter-reflections and more complex visual phenomena.

Acknowledgments

We would like to thank Antonio Criminisi and P. Anandan for their help and constructive suggestions.

Figure 11: Color update results. *(a) Min-composite initialization, quite noisy. (b) Output of the color update algorithm. (c) Without the regularization terms, the streaking effects are more obvious.*

References

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

[2] J. Blinn. Image compositing–theory. *IEEE Computer Graphics and Applications*, 14(5), 1994.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. PAMI*, 23(11):1222–1239, November 2001.

[4] R. T. Collins. A space-sweep approach to true multi-image matching. In *CVPR'96*, pp. 358–363, June 1996.

[5] S. X. Ju, M. J. Black, and A. D. Jepson. Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency. In *CVPR'96*, pp. 307–314, June 1996.

[6] S. B. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *CVPR'01*, vol. I, pp. 103–110, December 2001.

[7] M. Okutomi and T. Kanade. A multiple baseline stereo. *IEEE Trans. PAMI*, 15(4):353–363, April 1993.

[8] J. Rissanen. Modeling by shortest data desription. *Automatica*, 14(14):465–471, 1978.

[9] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1):7–42, May 2002.

[10] A. Smith and Blinn J. Blue screen matting. In *ACM SIGGRAPH*, pp. 259–268, August 1996.

[11] R. Swaminathan, S.B. Kang, R. Szeliski, A. Criminisi, and S.K. Nayar. On the motion and appearance of specularities in image sequences. In *ECCV*, vol. I, pp. 508–523, May 2002.

[12] R. Szeliski, S. Avidan, and P. Anandan. Layer extraction from multiple images containing reflections and transparency. In *CVPR'00*, vol. 1, pp. 246–253, June 2000.

[13] R. Szeliski and P. Golland. Stereo matching with transparency and matting. *IJCV*, 32(1):45–61, August 1999.

[14] Y. Wexler, A. Fitzgibbon, and A. Zisserman. Bayesian estimation of layers from multiple images. In *ECCV'02*, vol. III, pp. 487–501, May 2002.