Boundary Matting for View Synthesis

Samuel W. Hasinoff[1] Sing Bing Kang[2] Richard Szeliski[2]

[1] Dept. of Computer Science
University of Toronto
Toronto, Canada M5S 3G4
hasinoff@cs.toronto.edu

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

Abstract

In the last few years, new view synthesis has emerged as an important application of 3D stereo reconstruction. While the quality of stereo has improved, it is still imperfect, and a unique depth is typically assigned to every pixel. This is problematic at object boundaries, where the pixel colors are mixtures of foreground and background colors. Interpolating views without explicitly accounting for this effect results in objects with a "cut-out" appearance.

To produce seamless view interpolation, we propose a method called boundary matting*, which represents each occlusion boundary as a 3D curve. We show how this method exploits multiple views to perform fully automatic alpha matting and to simultaneously refine stereo depths at the boundaries. The key to our approach is the unifying 3D representation of occlusion boundaries estimated to subpixel accuracy. Starting from an initial estimate derived from stereo, we optimize the curve parameters and the foreground colors near the boundaries. Our objective function maximizes consistency with the input images, favors boundaries aligned with strong edges, and damps large perturbations of the curves. Experimental results suggest that this method enables high-quality view synthesis with reduced matting artifacts.*

1. Introduction

Although stereo correspondence was one of the first problems in computer vision to be extensively studied, automatically obtaining dense and accurate estimates of depth from multiple images remains a challenging problem [1].

Most stereo research has been concerned solely with methods for producing accurate depth maps, so interpolated views are rarely evaluated as results. By contrast, our explicit goal is superior view synthesis from stereo. Even for easy scenes in which all objects are opaque, diffuse, and well-textured, state-of-the-art stereo techniques fail in some respects to generate high-quality interpolated views. Even if a perfect depth map were available, current methods for view interpolation share two major limitations:

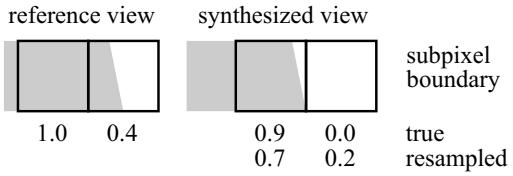


Figure 1: *View synthesis with matting. If mattes are represented at the pixel level, resampling for view synthesis can lead to blurring artifacts at object boundaries. This scene can be represented* exactly *using a sub-pixel boundary model instead.*

- **Sampling blur.** There is an effective loss of resolution caused by resampling and blending the input views.
- **Boundary artifacts.** Synthesized sprites seem to pop out of the scene, as in bad blue-screen composites, because most current methods do not perform matting to resolve mixed pixels at object boundaries into their foreground and background components. (But see some of the more recent work discussed in the next section for a few notable exceptions.)

In this paper we focus on the latter issue and propose a technique we call *boundary matting* for reducing such artifacts. Our technique, as outlined in Figs. 2–3, combines ideas from image matting and stereo to resolve mixed boundary pixels. It consists of estimating 3D curves over multiple views and uses stereo data to bootstrap this estimation.

The key feature of our approach is that occlusion boundaries are represented in 3D, leading to five contributions over the state of the art. First, compared to video matting [2] and other methods that recover pixel-level mattes for the input views [3, 4], our method is more suitable for view synthesis (Fig. 1). Second, our method can perform fully automatic matting from imperfect stereo data, for large-scale opaque objects. Third, our method exploits information from matting to refine stereo disparities along occlusion boundaries. Fourth, occlusion boundaries can be estimated

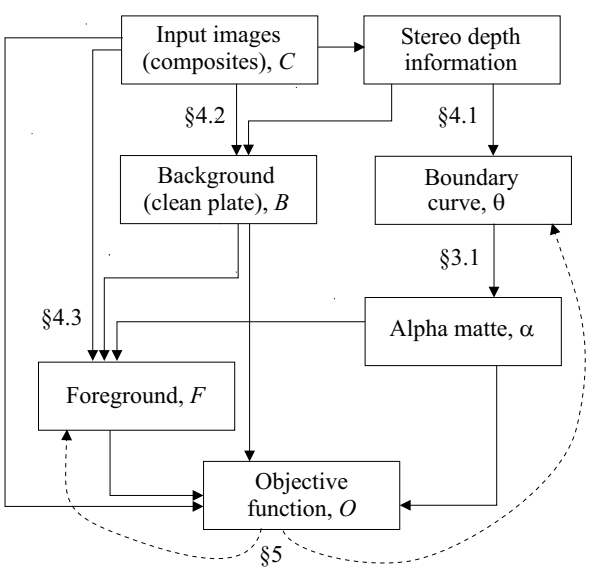


Figure 2: *Block diagram describing the system architecture. The dashed lines indicate that the objective function is used to optimize the parameters of the boundary curve and the foreground colors.*

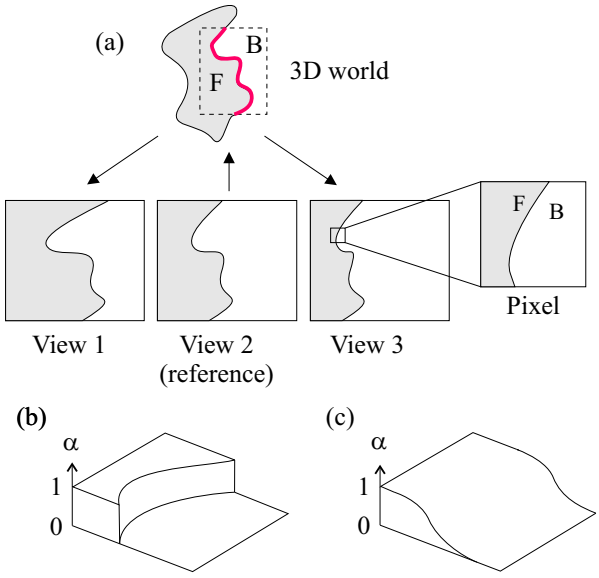


Figure 3: *Geometric view of the system. (a) Stereo depth information is used to detect an occlusion boundary in the reference view, which is backprojected to 3D as our initial curve estimate. The 3D curve is refined, along with estimates for F color, by evaluating the projections of the curve in all input views. The alpha value for a given pixel can be computed from the projected curve geometrically. (b) In the simplest case, alpha corresponds to the fraction of area on the F side of the curve. (c) Smoother blurring of the continuous alpha matte is a more realistic model.*

to sub-pixel accuracy, suitable for super-resolution or zooming. Fifth, our error metric is symmetric with respect to the input images, and so does not overly favor specific frames.

2. Previous work

Traditional matting techniques from cinema are reviewed in [5], and a *triangulation* method is presented for matting static foreground objects using multiple images taken with different backgrounds. More recent matting research has focused on *natural image matting*, where the goal is to estimate the matte from a single image, given regions hand-labelled as completely foreground and background [6, 7, 8]. These methods operate by propagating statistics of the labelled color distributions throughout the unlabelled regions, yielding impressive results. The approach in [7] was also extended using optical flow techniques to obtain a semi-automated method for matting video sequences [2]. This system for video matting also used a manually obtained clean-plate background to further constrain the background color distribution.

Several researchers have also investigated an additive transparent image formation model, useful for separating the reflections found on glass [9, 10]. Along the same lines, additive transparency has been decomposed based on parameterizing the dominant motions in the scene [11].

Few researchers, with the exception of [4, 12, 13], have tried to estimate transparency from stereo data in general terms. In [4], transparency was estimated in a volumetric fashion along with depth, using a plane-sweep algorithm, generalized to a four-dimensional $xyd\alpha$ search space. Results were mainly shown for synthetic problems, but even for those, the quality of interpolated views was limited. Similarly, the iterative voxel reconstruction approach presented in [12] gave results unsuitable for view synthesis, whereas [13] is more appropriate for volumetric scenes that are semi-transparent everywhere. Mixed pixels for stereo have also been examined as a consequence of using mixture models to estimate optical flow [14], and in developing a matching metric more robust to mixing [15].

Most closely related to our work is [3], which also estimates matting from multiple views of a scene. However, because this method is pixel-based and only estimates matting in a reference view, it is limited for synthesizing new views. As shown in Fig. 1, the naïve approach of warping and resampling the alpha matte introduces additional blurring, and makes hidden assumptions about the underlying scene. Another basic limitation of this method is that high quality stereo data is required everywhere in the image, and the extension to inaccurate stereo is unclear. Both these problems are circumvented in [3] by matting scenes with easily registered planar structures, and only demonstrating object insertion for the reference view. In contrast, since our

method is based on a 3D curve representation (see Sec. 3), the alpha matte has a well-defined geometric interpretation that is consistent across arbitrary nearby views. Moreover, we tolerate some inaccuracy in the stereo data by simultaneously estimating the matting and refining the disparity estimates (i.e., by adjusting the boundary curve).

Our geometric view of alpha values has precedence in work on (single view) user-assisted segmentation of opaque objects [16, 17]. Here, alpha is estimated from the fractional pixel coverage given by a sub-pixel parametric edge model fit to the object boundaries. Both methods require manual interaction at key frames, and neither extend readily to multiple views. By comparison, our method is fully automatic, and multiple views are fully incorporated. Along similar lines, sub-pixel edge geometry has been used to interpolate sparse point samples for rendering synthetic scenes, to better respect object and shadow boundaries [18].

3. Image formation model

To model the matting effects at occlusion boundaries, we use the well-known compositing equation [5, 19]:

$$C = \alpha F + (1 - \alpha)B \;, \tag{1}$$

which describes the observed composite color C as a blend of the foreground color F and the background color B according to the alpha value (opacity) α. The alpha matte is typically given at the pixel level, so fractional alpha-values may be due to partial pixel coverage of foreground objects at their boundaries, or due to true semi-transparency. In this work, we focus exclusively on the former case, where objects are opaque and alpha values are entirely due to the micro-geometry of partial pixel coverage.

Our method for inverting Eq. (1) exploits stereo information, in the spirit of the triangulation method [5]. Rather than viewing a static object with a single camera and manually changing the background, we obtain simultaneous images of the object from multiple viewpoints. Under the assumption that foreground color varies little over nearby views, multiple views can also provide us with images of the same foreground region against different backgrounds [3]. While triangulation approaches require that the background be known, this can theoretically be obtained from multiple views using stereo. *Note that we do not restrict our calculations to a reference image*, as in [3], so our 3D boundary curves in general lead to different alpha values for corresponding pixels in each view. This is a useful feature because it helps us resolve the ambiguity where the background is constant over all views.

3.1. Boundary curves in 3D with blurring

We model the occlusion boundary of a foreground object as a single (possibly open) 3D curve. For such a curve to be globally consistent with all of its projections, we assume that the occluding contours of the foreground objects are sufficiently sharp relative to both the closeness of the views and the standoff distance of the cameras (unlike, e.g., [20], which assumes that the object surface may be smoothly curved). Even for relatively smoother contours, although the boundary curve only approximates a path through the swept occlusion surface, this approximation may still be accurate enough to improve our estimation of alpha. After refinement, our method localizes this curve to sub-pixel.

In our approach, we model the 3D curve as a spline parameterized by control points, θ. For now, we simply take this curve to be piecewise linear, parameterized using the (metric) 3D coordinates of the control points; the extension to higher-order splines should be straightforward. The control points are spaced on the order of several pixels apart, and so do not constitute direct geometric models for such extremely fine-scale objects as hair or foliage. Rather, for such objects, these splines can only approximate partial pixel coverage along occlusion boundaries.

Given the camera projection matrix, M, for a particular view, we construct a signed distance function from the projected curve, $d(M, \theta)$, defined to be positive on the foreground side. In the ideal case, with a Dirac point spread function, the continuous alpha matte for the i-th view is:

$$\alpha_i(\theta) = \begin{cases} 1, & d(M_i, \theta) > 0 \\ 0, & \text{otherwise} \end{cases} \;. \tag{2}$$

This is a simple 2D step function of the curve parameters (Fig. 3(b)).

We simulate image blurring due to camera optics and motion by convolving α with an isotropic 2D Gaussian function $G(0, \sigma)$,

$$\begin{aligned} \alpha_i(\theta, \sigma) &= \alpha_i(\theta) * G(0, \sigma) \\ &= \tfrac{1}{\sigma\sqrt{2\pi}} \textstyle\int_{-\infty}^{d(M_i,\theta)} \exp\left(\frac{-t^2}{2\sigma^2}\right) dt \;. \end{aligned} \tag{3}$$

This modified model gives us a smoothed step function for alpha (Fig. 3(c)), parameterized using a single additional variable, σ.

For a given pixel j, we can generate the resulting pixel-level alpha value by integrating either of the continuous alpha functions proposed in Eqs. (2–3) over the footprint of that pixel. For view i, this gives $\alpha_{ij} = \iint_j \alpha_i$. For the ideal case of Eq. (2), this is equivalent to computing the area on the foreground side of the projected curve, which has a simple form when the curve is piecewise linear. On the other hand, the blurred model of Eq. (3) is more complicated, so we approximate the integral using supersampling.

3.2. Objective function

We formulate boundary matting as finding the 3D boundary curve and foreground colors that best fit the V input images. Our primary goal is to minimize inconsistency with

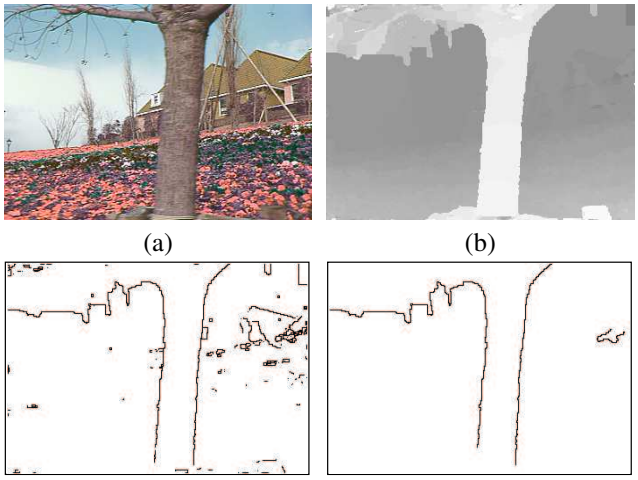


Figure 4: *Boundary initialization. For a well-known stereo sequence, we show (a) the reference image, (b) the depth map calculated using [21], (c) the depth discontinuity map, corresponding to the thresholded gradient of depth, and (d) the initial occlusion boundaries extracted from (c).*

the images, according to the matting equation, Eq. (1). This leads to a basic objective function encoding the total cost of matting inconsistency:

$$\mathcal{O}(\theta, F) = \sum_{i=1}^{V} \sum_{j=1}^{N} [\, C_{ij} - \alpha_{ij}(\theta)\, F_j - (1 - \alpha_{ij}(\theta))\, B_{ij} \,]^2 \,, \quad (4)$$

where N is the number of pixels along the curve in each view. In practice, we evaluate this objective function over all pixels in a conservatively wide band around the boundary curve (e.g., Fig. 6(b)), to ensure that every mixed pixel contributes to Eq. (4).

If we are using the blurred image formation model from Eq. (3), we also need to determine the optimal value for the blur parameter, σ. Currently, we estimate this parameter using a coarse exhaustive search, separately from the rest of the optimization.

4. Initialization using stereo data

The starting point for boundary matting is an initialization derived from stereo. We use [21] to compute stereo, which assumes that camera calibration is known. While initialization depends on the accuracy of the stereo data, we later refine the matting using an optimization based on Eq. (4).

In this section, we describe how to extract occlusion boundaries from the stereo data, how to estimate the *clean-plate* background B for pixels near the occlusion boundary, how to initialize our estimate of foreground color F, and how to construct a prior favoring strong edges at the boundary that can be used to tweak the initial guess.

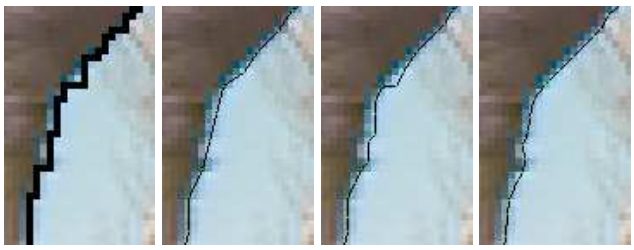


Figure 5: *Spline fitting to extracted occlusion boundaries. (a) Pixel-level occlusion boundary extracted from a region at the top-middle of Fig. 4(a). The piecewise-linear spline is shown overlaid at sub-pixel resolution (b–d). (b) Initial fit to the extracted boundary. (c) Adaptive subdivision in regions of poor matting. (d) Snapping to the strongest nearby edge within a 1.0-pixel radius.*

4.1. Boundary initialization

To extract the inital curve corresponding to an occlusion boundary, we first form a depth discontinuity map by applying a threshold to the gradient of the depth map for the reference view (Fig. 4(b–c)). Next, we separate this image into multiple contours, greedily removing the longest four-connected curves until no depth discontinuities of some minimum length remain (Fig. 4(d)). By growing the longest curves possible, we eliminate the small spurs and loops that are mainly due to inaccurate stereo. We also ensure that our discontinuity curves do not cross other depth discontinuity boundaries, as in T-junctions between three regions of different depths.

Next, we backproject the points along the discontinuity curve into 3D using the stereo (foreground) depth (Fig. 3), and fit a 3D curve to these points. Our current fitting technique is very simple, with control points sampled uniformly, at every fifth point along the discontinuity curve (Fig. 5(b)). We connect the control points with line segments to obtain a piecewise-linear spline in 3D.

After initial boundary extraction, we evaluate the curve for consistency with the matting equation (see Sec. 5 for more details). In regions with high matting error, we subdivide the curve, but only once (Fig. 5(c)). While we have experimented with a general adaptive subdivision scheme, the four-connected boundary gives undesired staircase artifacts with smaller stopping criteria.

We also modify our initial guess to reflect the fact that occlusion boundaries tend to coincide with strong edges, perturbing the control points in the reference image to the local peak of an edge potential field (Fig. 5(d)). We first apply a multiscale difference-of-Gaussians edge detector to each image, localizing edgels to sub-pixel, and use this to pre-compute edge potential fields, $\{E_i\}$, quantized to 0.25 pixels. We define these fields as the sum of "forces" pro-

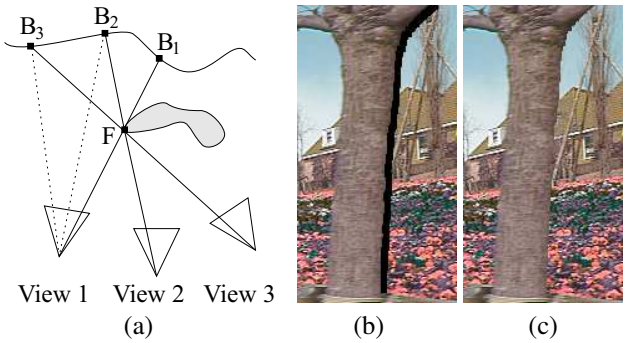


Figure 6: *Estimation of the clean-plate background. (a) The region labelled F is a mixed pixel in all views. The background colors B_2 and B_3 can be obtained from view 1, by following the dashed lines. However, B_1 is occluded in all views. (b–c) A region of Fig. 4(a) is shown, (b) with pixels near the boundary highlighted, and (c) with these pixels filled in using our clean-plate background estimate.*

portional to edgel strength and inversely proportional to squared edgel distance. Though edges are a strong cue for occlusion boundaries in many scenes, this heuristic can also be distracted by spurious internal texture, so we limit the perturbation to a 1.0-pixel radius neighborhood.

4.2. Background (clean plate) estimation

As discussed in Sec. 3, using stereo data to triangulate the matting problem requires that the background be known. A clean plate background refers to an image where foreground pixels are replaced with (unmixed) background colors, and is specified in many systems using manual interaction at keyframes [2, 3]. However, this process can in theory be made automatic by exploiting stereo information to grab corresponding background colors from nearby frames in which the background is exposed (Fig. 6). Note that aside from specifying the initial guess for the 3D boundary curve, the only place our approach relies on accurate stereo is in warping the background from nearby views.

For a given boundary pixel, we find potentially corresponding background colors by forward-warping that pixel to all other views. This warping is performed according to the depth on the background side of the boundary, as given by stereo. We use nearest-neighbor sampling to avoid blurring mixed pixels together, assuming that the background texture is slowly varying. If a forward-warped pixel has background depth in the new view, it becomes a candidate source from which to grab the background.

We use a *color consistency test* to select the exposed background pixel, to avoid selecting mixed pixels labelled with background depths. From all views, we choose the corresponding pixel at background depth that is the most color-consistent with its immediate neighbors also at background depth. This heuristic assumes slowly varying background texture, but seems to work well in most of our cases.

If a corresponding background pixel cannot be estimated (e.g., it is occluded by the foreground object in all views), then it is marked as such, and this pixel is not used directly in the optimization. In rendering the results, we either highlight these pixels as unknown, or use the naïve non-matting approach to determine color (i.e., $F = B = C$) and estimate alpha from the curve.

4.3. Foreground estimation

Given an estimate for the parameters of the curve (which determine alpha), the clean-plate background, B, and the input images, C, we can obtain a reasonable guess for the foreground colors, F. For a given pixel we can simply invert the matting equation, Eq. (1), to obtain:

$$\hat{F}(\alpha) = (C - (1-\alpha)B)/\alpha \; . \tag{5}$$

For robustness, we aggregate $\hat{F}$ estimates over all V views, with the correspondence between foreground pixels determined from stereo. Analogous to clean plate background estimation, we use the foreground-side depth from stereo to warp the boundary pixels in the reference view to all other views. To aggregate the $\hat{F}$ estimates of Eq. (5), we take the weighted average,

$$\hat{F} = \frac{\sum_{i=1}^{V} \alpha_i^2 \hat{F}(\alpha_i)}{\sum_{i=1}^{V} \alpha_i^2} \; , \tag{6}$$

with the weights constructed to favor F color information from pixels posited to contain more foreground, based on the curve estimate. Note that this formula is also the statistically optimal least-squares estimate for F given the set of V noise-contaminated composite color pixels, $C_i = \alpha_i F + (1-\alpha_i)B_i$.

5. Parameter optimization

Now that we have constructed a clean-plate background (Sec. 4.2), specified correspondences between pixels across the different views (Sec. 4.3), and provided initial estimates for the curve parameters and the foreground colors (Sec. 4.1 and 4.3), we are in a position to modify these estimates to better fit the images.

Because the objective function, Eq. (4), is highly nonlinear, involving bilinearity in the variables, perspective projection, and a complicated form for alpha as the partial pixel coverage of a projected spline (possibly convolved some blurring), we resort to Levenberg-Marquardt nonlinear least-squares optimization with the Jacobian calculated using the forward difference approximation [22]. The stopping criteria and step size for this optimization are both related to a parameter encoding the predicted accuracy of the objective function.

5.1. Two-stage estimation

In our experience, it is faster and more stable to first optimize only the curve parameters, while estimating the foreground colors based on the alpha values derived from the curve (Sec. 4.3). We thus suggest a two-stage approach, where the optimized curve from the first stage is used as an initial guess for the joint estimation of both the curve and foreground colors.

Assuming that the starting point is reasonably accurate, this optimization should refine our curve, and separate mixed pixels into background and foreground components. If our initial estimate is further than one pixel away from the curve, we may get trapped in a local minimum, because differential changes to the curve parameters may not improve matting consistency. Hopefully, even in the case of gross stereo errors, the control points may have wide enough support so that some pixels are indeed affected by their movement, and these pixels will gradually guide the curve closer to the true solution. The blurred image formation model, Eq. (3), may be more resilient to stereo errors than the basic model, because the control points have a larger support.

5.2. Adding edge snapping and state damping

We also created a penalty function to bias the optimization to areas with stronger edges, so the overall optimization can be considered a kind of 3D snake [23]. This function reuses the edge potential fields, $\{E_i\}$, described in Sec. 4.1, normalized to have a maximum of one. We project all $n(\theta)$ control points, denoted $\{\theta_p\}$, into each view using the camera matrices $\{M_i\}$, then calculate a penalty term proportional to inverse edge strength,

$$P_1(\theta, F) = \sum_{i=1}^{V} \sum_{p=1}^{n(\theta)} \left[\, 1 - E_i(M_i\theta_p) \,\right]^2 , \quad (7)$$

for the control points over all views.

An additional penalty function is used to discourage the control points from being displaced too far from their starting positions,

$$P_2(\theta, F) = \sum_{i=1}^{V} \sum_{p=1}^{n(\theta)} \left[\, \max\left(0,\ ||M_i\theta_p - M_i\theta_{p0}||^2 - 4.0\right) \,\right]^2 , \quad (8)$$

where θ_{p0} is the initial location of the p th control point. The penalty is zero for displacements of 2.0 pixels or less, but increases rapidly after that. This function helps avoid degenerate configurations where the curve collapses on itself.

We add these penalty terms to the original objective function (Eq. 4), and express this succinctly as:

$$\mathcal{O}_{\text{new}} = \mathcal{O} + \lambda_1 P_1 + \lambda_2 P_2 \, . \quad (9)$$

Moderate values of λ_1 and λ_2 ensure that neither edges nor initial positions exert too much influence over the optimization. In practice, the optimization did not seem overly sensitive to these parameters, so the same setting was used across all datasets. These parameters were normalized by $k = N/n(\theta)$, and set to $\lambda_1 = 0.015k$ and $\lambda_2 = 0.053k$.

6. Results

For all datasets, we used five input views, with the middle view designated as the reference view for initialization. For our first experiment, we applied boundary matting to the dataset shown in Fig. 4 to insert a new object between the foreground and background layers (Fig. 7). The results show that not only is the matting greatly improved, but the extracted boundaries can even be sharpened, by rendering the curves at sub-pixel resolution (Fig. 7(d)).

The same dataset was also used for a view synthesis task, for matting both an input view and a novel interpolated view (Fig. 8). In both cases, boundary matting produces a significant improvement over naïve view synthesis (i.e., forward warping with a fixed footprint, then feathering between the warped images). These results also demonstrate some tolerance to inaccurate stereo, since the initial stereo estimate in the region shown (Fig. 8(b–e)) was up to two pixels off. We also experimented with a variety of settings for the blur parameter. While the addition of blur did not appear to improve the matting for this case, the optimal blurred boundary better matches the appearance of the input images.

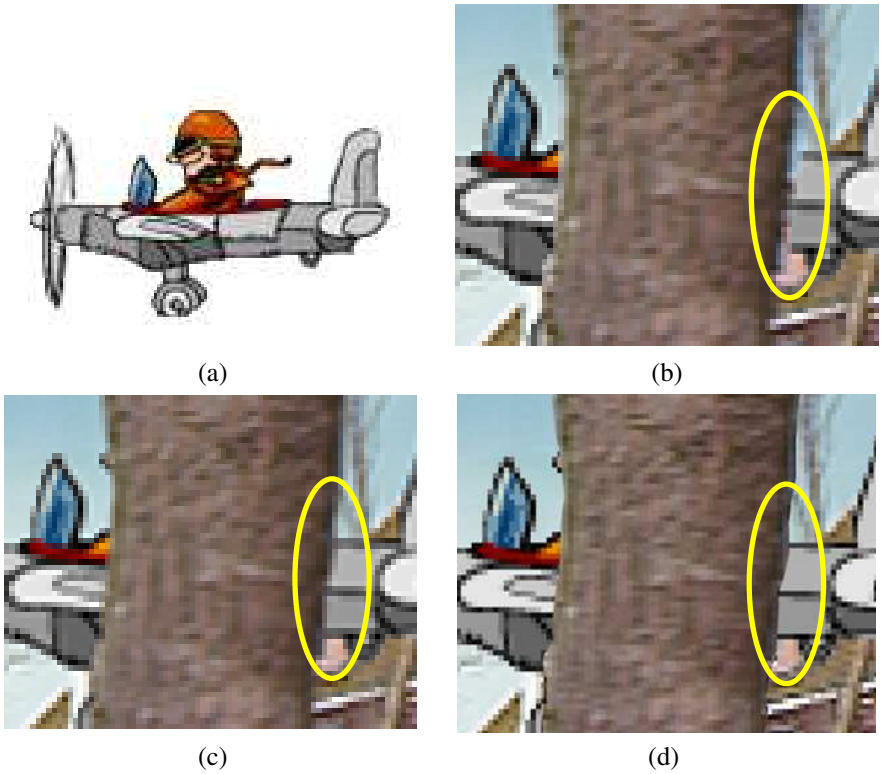
Our method broke down for the upper-left region of the tree with many twigs, yet performed no worse here than naïve view synthesis ignoring matting. The reason for failure was not the inability to localize a consistent 3D curve, but rather that inaccurate stereo led to an initial boundary estimate up to ten pixels off. Without additional color-based priors, our matting method is content to accept the possibility of depth discontinuities occurring across untextured regions of sky, trapping the optimization in local minima with $F = B$.

Finally, for a synthetic dataset consisting of a planar ellipse-shaped sprite, boundary matting is visually indistinguishable from the ground truth (Fig. 9(c)), and demonstrates further resilience to artificial noise (Fig. 9(c–e)).

7. Concluding Remarks

For seamless view interpolation, mixed boundary pixels must be resolved into foreground and background components, and boundary matting appears to be a useful tool for addressing this problem in an automatic way. Using 3D curves to model occlusion boundaries is a natural representation that provides several benefits, including the ability to super-resolve the depth maps near occlusion boundaries.

A current limitation of our approach is its lack of reasoning about color statistics, which has proven very useful in natural image matting [7, 2]. Such an ability might enable us to resolve boundaries even in areas where stereo



(a) (b) (c) (d)

Figure 7: *Object insertion. The object shown in (a) is inserted behind the tree from Fig. 4(a), with one region of interest highlighted. (b) Naïve object insertion, with no matting, leads to background spill and haloing artifacts at the edges of the tree. (c) Using our boundary matting method leads to an improved composite. (d) Boundary matting shown at super-resolution demonstrates sharpening of the boundaries, although the mismatch in resolutions may appear artificial.*

gives grossly incorrect depths, as in the upper-left region of the tree in Fig. 4(a). By integrating boundary matting with complementary aspects of pixel-based matting methods [7, 3], we hope to extend the generality of boundary matting while retaining its superior view synthesis. In the future, we would also like to adapt boundary matting to a dynamic stereo framework, where disocclusions over time may reveal additional information to improve the matting.

References

[1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Intl. Journal of Comp. Vision*, vol. 47, pp. 7–42, May 2002.

[2] Y.-Y. Chuang, A. Agarwala, B. Curless, D. H. Salesin, and R. Szeliski, "Video matting of complex scenes," in *SIGGRAPH*, pp. 243–248, 2002.

[3] Y. Wexler, A. W. Fitzgibbon, and A. Zisserman, "Bayesian estimation of layers from multiple images," in *ECCV*, vol. 3, pp. 487–501, 2002.

[4] R. Szeliski and P. Golland, "Stereo matching with transparency and matting," in *ICCV*, pp. 517–526, 1998.

[5] A. R. Smith and J. F. Blinn, "Blue screen matting," in *SIGGRAPH*, pp. 259–268, 1996.

[6] M. Ruzon and C. Tomasi, "Alpha estimation in natural images," in *CVPR*, pp. 18–25, 2000.

[7] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski, "A Bayesian approach to digital matting," in *CVPR*, pp. 264–271, 2001.

[8] P. Hillman, J. Hannah, and D. Renshaw, "Alpha channel estimation in high resolution image and image sequences," in *CVPR 2001*, pp. 1063–1068, December 10–13 2001.

[9] R. Szeliski, S. Avidan, and P. Anandan, "Layer extraction from multiple images containing reflections and transparency," in *CVPR*, pp. 246–253, 2000.

[10] Y. Tsin, S. Kang, and R. Szeliski, "Stereo matching with reflections and translucency," in *CVPR*, pp. 702–709, 2003.

[11] M. Irani, B. Rousso, and S. Peleg, "Computing occluding and transparent motions.," *Intl. Journal of Comp. Vision*, vol. 12, pp. 5–16, Feb. 1994.

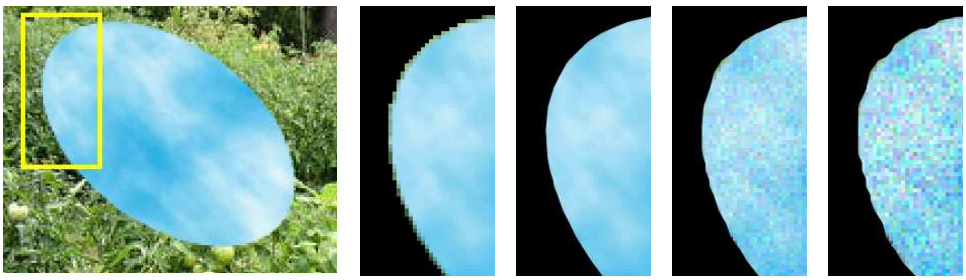
[12] J. D. Bonet and P. Viola, "Roxels: Responsibility weighted 3D volume reconstruction," in *ICCV*, pp. 418–425, 1999.

[13] S. W. Hasinoff and K. N. Kutulakos, "Photo-consistent 3D fire by flame-sheet decomposition," in *ICCV*, pp. 1184–1191, 2003.

[14] S. Ju, M. J. Black, and A. D. Jepson, "Skin and bones: Multi-layer, locally affine, optical flow and regularization with transparency," in *CVPR*, pp. 307–314, 1996.

(a) (b) (c) (d) (e)

Figure 8: *Boundary matting for view synthesis. The first row corresponds to the reference view, and the second row corresponds to an interpolated view. (a) Input image (ground truth interpolated view). (b) Zoomed-in region. (c) Naïve foreground separation without matting shows significant spill from the background layer. (d) Using the boundary matting method reduces this artifact. (e) Boundary matting with a blurred edge model ($\sigma = 0.4$ pixels) better matches the blur in the input images.*



(a) (b) (c) (d) (e)

Figure 9: *Boundary matting synthetic data, with the addition of noise. (a) Reference image of a planar textured ellipse, with zoomed region indicated. (b) Foreground object extracted without matting. (c) Boundary matting (super-resolution). (d–e) Boundary matting with zero-mean Gaussian noise, (d) $\sigma_{\text{noise}} = 5\%$, and (e) $\sigma_{\text{noise}} = 10\%$.*

[15] S. Birchfield and C. Tomasi, "A pixel dissimilarity measure that is insensitive to image sampling," *IEEE Trans. PAMI*, vol. 20, no. 4, pp. 401–406, 1998.

[16] T. Mitsunaga, T. Yokoyama, and T. Totsuka, "Autokey: Human assisted key extraction," in *SIGGRAPH*, pp. 265–272, 1995.

[17] E. N. Mortensen and W. A. Barrett, "Toboggan-based intelligent scissors with a four parameter edge model," in *CVPR*, pp. 2452–2458, 1999.

[18] K. Bala, B. Walter, and D. Greenberg, "Combining edges and points for interactive anti-aliased rendering," in *SIGGRAPH*, pp. 631–640, 2003.

[19] T. Porter and T. Duff, "Compositing digital images," in *SIGGRAPH*, pp. 253–259, July 1984.

[20] J. J. Koenderink, "What does the occluding contour tell us about solid shape?," *Perception*, vol. 13, pp. 321– 330, 1984.

[21] S. B. Kang, R. Szeliski, and J. Chai, "Handling occlusions in dense multi-view stereo," in *CVPR*, pp. 103–110, 2001.

[22] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer, 1999.

[23] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Intl. Journal of Comp. Vision*, vol. 1, pp. 321–331, January 1988.