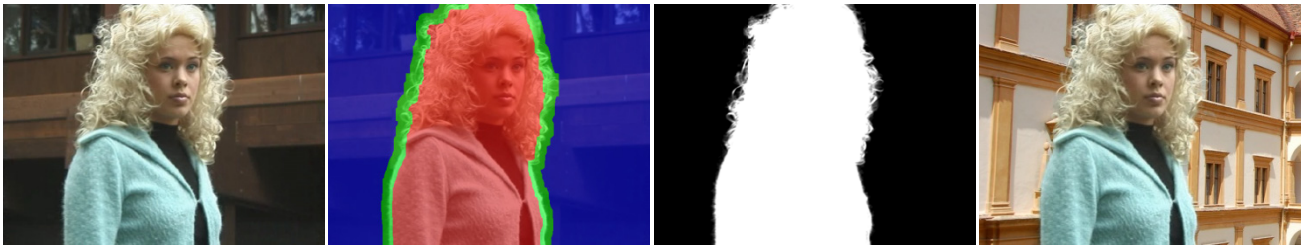
Real-Time Video Matting using Multichannel Poisson Equations

Minglun Gong
Memorial U. of Newfoundland
St. John's, NL, Canada
gong@cs.mun.ca

Liang Wang
Univ. of Kentucky
Lexington, KY, USA
lwangd@cs.uky.edu

Ruigang Yang
Univ. of Kentucky
Lexington, KY, USA
ryang@cs.uky.edu

Yee-Hong Yang
Univ. of Alberta
Edmonton, AB, Canada
yang@cs.ualberta.ca



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

Figure 1: Real-time video matting results on the video sequence captured by Chuang et al. [3]: (a) input frame; (b) automatic generated trimap using bilayer segmentation; (c) estimated alpha matte; (d) composite result.

ABSTRACT

This paper presents a novel matting algorithm for processing video sequences in real-time and online. The algorithm is based on a set of novel Poisson equations that are derived for handling multichannel color vectors, as well as the depth information captured. A simple yet effective approach is also proposed to compute an initial alpha matte in the color space. Real-time processing speed is achieved through optimizing the algorithm for parallel processing on the GPUs. To process live video sequences online and autonomously, a modified background cut algorithm is implemented to separate foreground and background, the result of which guides the automatic trimap generation. Quantitative evaluation on still images shows that the alpha mattes extracted using the presented algorithm is much more accurate than the ones obtained using the global Poisson matting algorithm and are comparable to that of other state-of-the-art offline image matting techniques.

1 INTRODUCTION

Matting studies how to extract foreground objects with per-pixel transparency information from still images or video sequences. Generally speaking, it tries to solve the following ill-posed problem:

Given a color image $\mathbf{I}$, which contains both foreground and background objects, calculate the matte α, foreground color $\mathbf{F}$, and background color $\mathbf{B}$, so that the following alpha compositing equation is satisfied:

$$\mathbf{I} = \alpha\mathbf{F} + (1-\alpha)\mathbf{B} \quad (1)$$

Graphics Interface Conference 2010
31 May - 2 June, Ottawa, Ontario, Canada

A variety of techniques have been developed in the past decade for still image matting [1, 4, 6, 7, 9, 13, 14, 18, 19] and several algorithms are also proposed for handling video sequences [1, 3, 8, 10-12, 16]. However, due to the high computational cost involved, so far real-time video matting for dynamic scenes can only be achieved under studio settings using specially designed optical devices [12]. To the best of our knowledge, technique for real-time video matting based on color observation only is not yet available.

On another front, impressive results have been reported for separating foreground objects from live videos using bilayer segmentation [5, 15]. Using just color information, their algorithm can extract the moving foreground object in real-time, making it a powerful technique for video conferencing and live broadcasting. However, bilayer segmentation cannot capture the fuzzy boundaries surrounding the foreground object caused by hair, fur, or even motion blur. Although the border matting technique [13] is applied to alleviate the aliasing problem along object boundaries, the strong constraint used in border matting limits its capability of handing objects with complex alpha matte, such as the one shown in Figure 1. Applying real-time video matting to extract alpha matte in fuzzy areas is the next logical step.

A real-time matting algorithm is presented in this paper to achieve this goal. The algorithm is able to perform matting using only color cue at real-time speed and to produce alpha mattes with qualities comparable to existing offline image matting approaches. When additional information, such as background color and/or scene depth, is available, the algorithm can utilize the information to further improve the matting accuracy.

2 RELATED WORK

Here we limit the discussion to video matting approaches. Readers are referred to [17] for a nice survey on related image matting techniques.

2.1 Video Matting

Video matting is pioneered by Chuang et al. [3]. In their Bayesian video matting approach, users are required to manually

specify trimaps for some key frames. These trimaps are then propagated to all frames using the estimated bidirectional optical flows. Finally the alpha matte for each frame is calculated independently using Bayesian matting [4].

Trimaps are generated from binary segmentations in two video object cutout approaches [10, 16]. Individual frames are over-segmented into homogenous regions, based on which a 3D graph is constructed. The optimal cut that separates foreground and background regions are found using 3D graph cuts. Pixels within a narrow band of the optimal cut are labeled as unknown regions, with their alpha values estimated using image matting techniques.

No over-segmentation is required in the geodesic matting algorithm, which treats the video sequence as a 3D pixel volume [1]. Each pixel is classified into foreground or background based on its weighted geodesic distances to the foreground and background scribbles that users specified for a few key frames. The alpha values for pixels within a narrow band along the foreground/background boundaries are explicitly computed using geodesic distances.

The above approaches are all designed to handle pre-captured video sequences offline, both of which utilize temporal coherence for more accurate results. Our approach, on the contrary, is designed for handling live captured videos in real-time. When processing a given frame, only the previous frames are available to us.

2.2 Online Video Matting

There are several online video matting techniques available. For example in defocus matting [11], the scene is captured using multiple optically aligned cameras with different focus/aperture settings. The trimap is automatically generated based on the focused regions of captured images. The alpha matte is calculated by solving an error minimization problem, which takes several minutes per frame.

Automatic video matting can also be done using a camera array [8]. The images captured are aligned so that the variance of pixels reprojected from the foreground is minimized whereas the one of pixels reprojected from the background is maximized. The alpha values are calculated using a variance-based matting equation. The computational cost is linear with respect to the number of cameras and near-real-time processing speed is achieved.

Real-time video matting has been achieved in studio setting [12]. In their approach, the background screen is illuminated with polarized light and the scene is captured by two cameras each with a different polarizing filter. Since the background has different colors in the two captured images, the simple blue screen matting can be applied to extract the alpha matte in real-time.

All the matting approaches above require images captured from multiple cameras and utilize additional cues, such as focus and polarization settings or viewpoint changes. Our approach uses one camera only and can generate the alpha matte in real-time using color observation only.

2.3 Other Related Work

Our approach is inspired by the Poisson matting algorithm [14]. In Poisson matting, alpha matte is estimated using a selected color channel k. An approximate gradient of alpha is calculated by taking gradient on both sides of Equation (1) and omitting the gradients of the foreground and the background by assuming that they are both smooth across the image:

$$\begin{aligned} &\nabla \mathbf{I}_k = (\mathbf{F}_k - \mathbf{B}_k)\nabla\alpha + \alpha\nabla\mathbf{F}_k + (1-\alpha)\nabla\mathbf{B}_k \\ &\Rightarrow \nabla\alpha \approx \frac{1}{\mathbf{F}_k - \mathbf{B}_k}\nabla\mathbf{I}_k \end{aligned} \tag{2}$$

The global Poisson equation is then set up by taking divergence on both sides:

$$\Delta\alpha \approx \operatorname{div}\left(\frac{\nabla\mathbf{I}_k}{\mathbf{F}_k - \mathbf{B}_k}\right) \tag{3}$$

Poisson matting is computationally efficient and easy to implement. However, it tends to yield large errors when the trimap is imprecise and/or the background is not smooth. Manual editing using local Poisson equations can be applied to correct the errors [14], but is impractical when handling video sequences. Recent research shows that the accuracy of Poisson matting result can be improved when the background color **B** for the unknown region is known [6]. However, this technique does not help when the true background color is unavailable.

2.4 Contributions

This paper presents several important and new improvements over the original Poisson matting technique. First, a novel set of Poisson equations are derived, which compute the gradient of the alpha using all color channels, instead of a selected color channel. This not only avoids the complex channel selection process [14], but also improves the matting accuracy — the experiments show that the alpha mattes obtained using all color channels are better than the one obtained using any of the three channels.

Secondly, a new way of generating the initial alpha matte solution in the RGB color space is presented. We show that, when the Poisson equation is solved numerically using inaccurate gradient information, using the color-space matte initialization step not only helps to accelerate the convergence, but also improves the robustness of the Poisson matting technique against imprecise trimap specification.

Finally, the presented algorithm is implemented on the GPUs and can achieve a real-time processing speed for video of 640×480 resolution. To our best knowledge, this is the first approach that achieves real-time video matting using observed color only.

3 Multichannel Poisson Equations

In this section, we first derive a general multichannel Poisson equation for matting based on color information only. Two variants of the equation are then discussed for handling cases where the background or the depth of the scene can be captured or recovered. Finally, a multichannel equation for alpha matte initialization is presented.

3.1 Equation for General Color Image Matting

Unlike the global Poisson matting equation used by Sun et al. [14], the multichannel Poisson equations are derived using all color channels. This is done by first rearranging Equation (1) into:

$$\mathbf{I} - \mathbf{B} = \alpha(\mathbf{F} - \mathbf{B}) \tag{4}$$

Taking gradient on both sides and applying the Leibnitz's law gives us:

$$\nabla \otimes (\mathbf{I} - \mathbf{B}) = (\nabla\alpha) \otimes (\mathbf{F} - \mathbf{B}) + \alpha\big(\nabla \otimes (\mathbf{F} - \mathbf{B})\big) \tag{5}$$

where $\nabla \otimes \mathbf{I}$ represents the tensor product between the gradient operator and color image **I**. That is:

$$\nabla \otimes \mathbf{I} = \begin{bmatrix} \partial \mathbf{I}_r/\partial x & \partial \mathbf{I}_g/\partial x & \partial \mathbf{I}_b/\partial x \\ \partial \mathbf{I}_r/\partial y & \partial \mathbf{I}_g/\partial y & \partial \mathbf{I}_b/\partial y \end{bmatrix} \tag{6}$$

Instead of relying on the smoothness assumption to omit the unknown α term, here we multiply a column vector $(\mathbf{F}-\mathbf{B})$ on both sides of Equation (5), which yields:

$$\begin{aligned} &\big(\nabla \otimes (\mathbf{I}-\mathbf{B})\big)(\mathbf{F}-\mathbf{B}) \\ &= \big((\nabla\alpha) \otimes (\mathbf{F}-\mathbf{B})\big)(\mathbf{F}-\mathbf{B}) + \alpha\big(\nabla \otimes (\mathbf{F}-\mathbf{B})\big)(\mathbf{F}-\mathbf{B}) \\ &= \nabla\alpha\big((\mathbf{F}-\mathbf{B})\cdot(\mathbf{F}-\mathbf{B})\big) + \big(\nabla \otimes (\mathbf{F}-\mathbf{B})\big)\alpha(\mathbf{F}-\mathbf{B}) \end{aligned} \tag{7}$$

Now the unknown α can be removed by substituting Equation (4) into Equation (7), which yields:

$$\begin{aligned} &\big(\nabla \otimes (\mathbf{I}-\mathbf{B})\big)(\mathbf{F}-\mathbf{B}) \\ &= \nabla\alpha\big((\mathbf{F}-\mathbf{B})\cdot(\mathbf{F}-\mathbf{B})\big) + \big(\nabla \otimes (\mathbf{F}-\mathbf{B})\big)(\mathbf{I}-\mathbf{B}) \end{aligned} \tag{8}$$

Therefore, the gradient of alpha can be calculated using:

$$\begin{aligned} \nabla\alpha &= \frac{\big(\nabla \otimes (\mathbf{I}-\mathbf{B})\big)(\mathbf{F}-\mathbf{B}) - \big(\nabla \otimes (\mathbf{F}-\mathbf{B})\big)(\mathbf{I}-\mathbf{B})}{(\mathbf{F}-\mathbf{B})\cdot(\mathbf{F}-\mathbf{B})} \\ &= \frac{(\nabla \otimes \mathbf{I})(\mathbf{F}-\mathbf{B}) - (\nabla \otimes \mathbf{F})(\mathbf{I}-\mathbf{B}) - (\nabla \otimes \mathbf{B})(\mathbf{F}-\mathbf{I})}{(\mathbf{F}-\mathbf{B})\cdot(\mathbf{F}-\mathbf{B})} \end{aligned} \tag{9}$$

It is noteworthy that the above equation is derived without any approximation. Therefore, if both foreground and background colors are known and different (the Smith-Blinn assumption), the gradient of alpha can be precisely calculated. When they are both unknown, however, we need to assume they are smooth and omit their gradients. This gives us the following multichannel Poisson equation:

$$\begin{aligned} &\nabla \otimes \mathbf{F} \approx \mathbf{0}, \nabla \otimes \mathbf{B} \approx \mathbf{0} \\ &\Rightarrow \Delta\alpha \approx \mathrm{div}(\mathbf{G}) = \mathrm{div}\left(\frac{(\nabla \otimes \mathbf{I})(\mathbf{F}-\mathbf{B})}{(\mathbf{F}-\mathbf{B})\cdot(\mathbf{F}-\mathbf{B})}\right) \end{aligned} \tag{10}$$

where $\mathbf{G}$ is the approximate gradient of matte.

Please note that, if only one color channel is used, the above multichannel Poisson equation degenerates into Equation (3):

$$\Delta\alpha \approx \mathrm{div}\left(\frac{\nabla \mathbf{I}_k(\mathbf{F}_k - \mathbf{B}_k)}{(\mathbf{F}_k - \mathbf{B}_k)^2}\right) = \mathrm{div}\left(\frac{\nabla \mathbf{I}_k}{\mathbf{F}_k - \mathbf{B}_k}\right) \tag{11}$$

3.2 Matting Equation for Known Background

Previous work has shown that the background information estimated from input video sequence can help to improve matting quality [3, 6]. Under our multichannel Poisson matting derivation, the known background information can be easily incorporated — instead of omitting both foreground and background gradients, we now only need to omit the gradient of unknown foreground. Hence, the following Poisson equation can be derived from Equation (9):

$$\begin{aligned} &\nabla \otimes \mathbf{F} \approx \mathbf{0} \Rightarrow \Delta\alpha \approx \mathrm{div}(\mathbf{G}_B) \\ &= \mathrm{div}\left(\frac{(\nabla \otimes \mathbf{I})(\mathbf{F}-\mathbf{B}) - (\nabla \otimes \mathbf{B})(\mathbf{F}-\mathbf{I})}{(\mathbf{F}-\mathbf{B})\cdot(\mathbf{F}-\mathbf{B})}\right) \end{aligned} \tag{12}$$

where $\mathbf{G}_B$ is the approximate gradient of matte under known background, which is a more accurate approximation than $\mathbf{G}$.

3.3 Matting Equation for Known Depth

It has also been shown that additional depth information captured using depth sensor helps to improve matting qualities for both Bayesian matting and Poisson matting [20]. In [20], the depth information is integrated into the Bayesian matting equation as an additional color channel. However for Poisson matting, the depth information is used for validation only, since the original Poisson equation can only handle one color channel.

Under our multichannel Poisson matting formulation, the depth information can be integrated naturally into the matting equation. Same as in [20], here we assume depth readings in fuzzy areas follow the same alpha compositing rule as color does. Whether or not this assumption holds depends on the mechanism used for capturing depth (it appears to be valid for the depth images we captured, as shown in Figure 7(b)). Therefore, we have:

$$\begin{cases} \mathbf{I}-\mathbf{B} = \alpha(\mathbf{F}-\mathbf{B}) \\ \lambda(I_d - B_d) = \alpha\lambda(F_d - B_d) \end{cases} \tag{13}$$

where $F_d, B_d, \& I_d$ are the foreground, background, and observed depths, respectively. Parameter λ controls the contribution of the depth information.

Following the same derivation from Equation (5–10) gives us the following Poisson equation:

$$\begin{aligned} &\Delta\alpha \approx \mathrm{div}(\mathbf{G}_D) \\ &= \mathrm{div}\left(\frac{(\nabla \otimes \mathbf{I})(\mathbf{F}-\mathbf{B}) + \lambda^2\nabla I_d(F_d - B_d)}{(\mathbf{F}-\mathbf{B})\cdot(\mathbf{F}-\mathbf{B}) + \lambda^2(F_d - B_d)(F_d - B_d)}\right) \end{aligned} \tag{14}$$

where $\mathbf{G}_D$ is the approximate gradient of alpha with known depth.

3.4 Equation for Alpha Matte Initialization

Once the Poisson equation is established, a unique solution can be computed, which minimizes the following variational problem:

$$\alpha^* = \arg\min_{\alpha} \int_{p\in\Omega} \|\nabla\alpha(p) - \mathbf{G}(p)\|^2 dp \tag{15}$$

where Ω is the unknown region in the trimap.

Since the gradient of alpha obtained using estimated foreground and background may be inaccurate, the solution found by solving the Poisson equation may not be optimal. Figure 2 illustrates this problem using a 1D example.

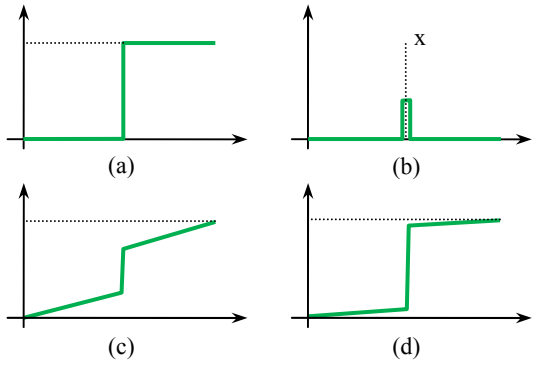


Figure 2: An example of recovering an unknown function (a) from inaccurate gradient information (b). Note that the gradient value at *x* is lower than the true value. While the reconstructed function (c) offers smaller mean square error in the gradient domain, the function (d) is a closer solution to the true function.

In practice, we found that the above problem can be alleviated by a simple yet very effective approach. The key idea is based on the following observation — when solving the Poisson equation numerically under limited precision, e.g., the alpha values are represented using integers within [0, 255] on the GPU, the final solution may not converge to the global optimum. As a result, when

a good initial alpha matte is provided, the solution found can actually be better than the one that minimizes Equation (15).

To generate a good initial solution, here we compute the initial alpha matte directly in the color space, before using the Poisson equation to solve the matte in the gradient space. This way, the ambiguities associated with inferring colors from inaccurate gradients can be resolved using the initial solution obtained from the color space.

The equation for alpha initialization is derived by applying dot product with $(\mathbf{I}-\mathbf{B})$ on both sides of Equation (4):

$$(\mathbf{I}-\mathbf{B})\cdot(\mathbf{I}-\mathbf{B}) = \alpha(\mathbf{F}-\mathbf{B})\cdot(\mathbf{I}-\mathbf{B})$$
$$\Rightarrow \alpha = \frac{(\mathbf{I}-\mathbf{B})\cdot(\mathbf{I}-\mathbf{B})}{(\mathbf{F}-\mathbf{B})\cdot(\mathbf{I}-\mathbf{B})} \quad (16)$$

where **F** and **B** are estimated foreground and background colors in the unknown region.

Similar equations can be derived by computing dot-products with other color difference vectors, such as $(\mathbf{F}-\mathbf{B})$ or $(\mathbf{F}-\mathbf{I})$. However, we found that using $(\mathbf{I}-\mathbf{B})$ generally works better since **I** is known and **B**, if unknown, can be more accurately estimated than **F** can be.

Our experiments show that using the above alpha matte initialization approach not only makes the Poisson equation solving process converge faster, but also yield more accurate alpha matte, especially when the trimap contains large unknown regions.

4 Real-time Video Matting Algorithm

Here we first explain how the proposed algorithm extracts alpha mattes for still images. This is followed by discussions on how to automatically generate trimaps for online video matting.

4.1 Matte Extraction for Still Images

Figure 3 shows the procedures involved for extracting alpha matte based on a given trimap. For real time performance, all these procedures are optimized for parallel execution on the GPUs. In addition, the early Z-kill feature is utilized to limit the computations to the unknown region of the trimap only. This is done by initializing the depth buffer in which known foreground and background pixels are assigned with zero depth values. Therefore, these pixels will not be processed when rendering image-sized quad positioned further away. Our tests show that enabling early Z-kill greatly improves the processing speed.

Using the source image and a given trimap as inputs, the first step estimates the colors of both foreground and background in the unknown region. Based on the smoothness assumption, an unknown pixel's foreground/ background color can be approximated using the color of the nearest pixel in the known foreground/background region [14]. Following this idea, an image morphology based procedure is used. The procedure fills an unknown foreground (background) pixel with the average color of its neighbors *iff.* at least one of its four neighbors has known foreground (background) color. Once a pixel is filled, the depth buffer used for early Z-kill is updated accordingly so that the pixel will not be processed again.

The estimated foreground and background colors, as well as the source image, are then used as inputs for initializing the alpha matte based on Equation (16). The initial solution obtained, shown in Figure 3(e), is quite accurate even when both foreground and background contain detailed textures. However close inspection shows that, due to imprecise foreground/background color estimation, artifacts do exist in areas such as the one highlighted with a red rectangle.

The same inputs are also used for calculating the approximate Laplacian of alpha. Depending on whether and what additional information is available, one of the Equations (10), (12), or (14) is selected. In this case, Equation (10) is used since no additional information is available. As shown in Figure 3(f), the Laplacian of alpha provides details on how the alpha matte changes locally, which helps to correct errors in the initial solution.

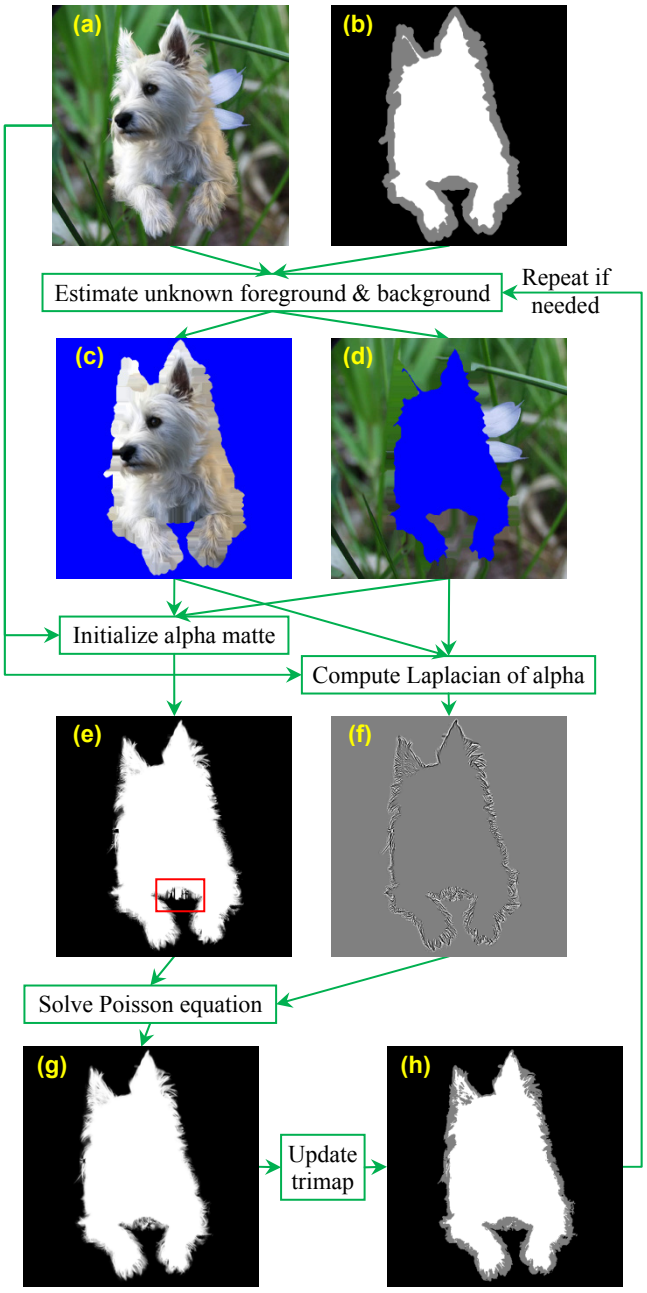


Figure 3: The flowchart of the presented algorithm and the intermediate results obtained for the "dog" dataset. (a) input image; (b) trimap; (c) estimated foreground; (d) estimated background; (e) initial alpha matte (f) approximate Laplacian of alpha; (g) estimated alpha matte; (h) updated trimap.

With the initial alpha matte and the Laplacian of alpha computed, the Poisson equation is now established. To facilitate

parallel implementation on GPUs, the Jacobi method is used here to solve the Poisson equation. Although the Jacobi method has the reputation of slow convergence, we found that it takes about 50 iterations only to converge in our experiments, thanks to the accurate initial solution obtained. To err on the safe side, the number of iterations is fixed at 64.

The estimated alpha matte allows us to generate a more accurate trimap, which helps to obtain better foreground/ background estimations and in turn better alpha matte. Similar to Sun et al.'s approach [14], here a new trimap is obtained through applying multilevel thresholding on the estimated alpha matte. That is, a pixel is labeled as foreground if the estimated alpha value is higher than T_{high} and as background if the alpha value is lower than T_{low}. Here the two thresholds are fixed at 0.95 and 0.05, respectively.

The updated trimap is used to repeat the matte extraction process. Ideally the process should repeat until it converges, i.e., the updated trimap is the same as the previous version. However, to ensure constant processing speed, a user-specified (fixed) number of iterations, N, are used. Through experiments, we found that just two iterations are sufficient in most cases and hence we set $N = 2$ for all the experiments shown in this paper.

4.2 Trimap Generation for Video Matting

In order to process video sequences online, a bilayer segmentation procedure is implemented, whose output is used to generate trimaps automatically for each frame.

Bilayer segmentation has been extensively studied [5, 15]. The problem can be formulated as finding a binary labeling that minimizes an energy function of the form:

$$\mathrm{L}^* = \arg\min_{\mathrm{L}} \sum_{\mathrm{p}\epsilon\mathrm{I}} \mathrm{D}\big(\mathrm{L}(\mathrm{p})\big) + \mu \sum_{(\mathrm{p},\mathrm{q})\epsilon\Gamma} \mathrm{C}\big(\mathrm{L}(\mathrm{p}), \mathrm{L}(\mathrm{q})\big) \qquad (17)$$

where $D\big(L(p)\big)$ is the data term that measures how well the labeling agrees with the measurements. $\mathrm{C}\big(L(p), L(q)\big)$ is the contrast term that encourages neighboring pixels to have the same label if they have similar colors (and depths). μ is a constant for balancing the two terms. Γ denotes the set of all 4- or 8-connected neighboring pixel pairs.

A widely adopted contrast term is used here, which is defined as:

$$C\big(L(p), L(q)\big) = |L(p) - L(q)| \cdot \mathrm{e}^{\frac{-\|\mathbf{I}_p - \mathbf{I}_q\|^2}{\gamma}} \qquad (18)$$

where $\|\mathbf{I}_p - \mathbf{I}_q\|$ is the L2 norm of the color difference between pixels p and q. γ is set to $2\langle\|\mathbf{I}_p - \mathbf{I}_q\|^2\rangle$, where $\langle . \rangle$ indicates expectation over all connected pixel pairs in an image.

While bilayer segmentation is not the primary focus of this paper, we did extend existing algorithms to achieve real-time foreground extraction. When handling videos sequences with known background appearance information, a variant of the background cut [15] is applied. In this scenario we assume that the camera is mostly stationary and the background is not cluttered. The background likelihood model does not change over time and can be modeled using a similar method as that proposed in [15]. For the foreground color model, instead of modeling the color likelihood using Gaussian Mixture Models and learning the foreground mixtures via Expectation Maximization (EM), our implementation models the foreground color likelihood non-parametrically using histogram in the RGB space. This simplification greatly reduces the computational costs and negates the need for EM initialization. Similar to [5], the foreground color likelihood model is learned over successive frames. The color histogram for foreground pixels is updated dynamically according to the segmented foreground image data from previous frame. A moderate amount of smoothing is applied to the 3D color histogram to avoid overlearning.

In addition, when depth is captured using a depth sensor, the data term is computed using both color and depth cues. The foreground and background depth likelihoods are modeled using two Gaussian models learned from the depth values of labeled foreground/background pixels in previous frame. This learning process ensures that the most recent depth variations in the scene are captured. Note that depth information is usually more stable than appearance based cues in challenging cases such as scenes that contain dynamic background, sudden illumination change etc. Therefore when scene depth is available we no longer assume the background appearance is previously known and the per-pixel background color model defined in [15] is abandoned. The global foreground/background color likelihoods are both modeled dynamically with color histograms. By combining color and depth cues the binary segmentation process is less sensitive to lighting variation, moving objects and camera shaking, making our system more flexible and robust in various scenarios.

Once the per-pixel labeling costs are calculated using the data term, the optimal labeling that minimizes Equation (16) can be found efficiently using the min-cut algorithm [2]. The trimap is then generated from the binary segmentation result by eroding both foreground and background regions and marking the in-between area as unknown.

5 Experimental Results

The presented algorithm is tested using a variety of still images and video sequences. Some representative results are shown here.

5.1 Quantitative Evaluation using Still Images

The algorithm is evaluated using datasets with ground truth alpha mattes presented by Wang and Cohen [19]. Each dataset contains ten different trimaps of different level of accuracies, with T0 being the most accurate trimap and T9 the least accurate one. The result for the "dog" dataset shown in Figure 3 is generated using the fifth trimap. The results for the rest four datasets obtained using the second trimap (not necessarily the best choice) are shown in Figure 4. These results confirm that the proposed algorithm can produce visually appealing alpha mattes for complex scenes.

For quantitative evaluation, an alpha matte is generated using each of the ten trimaps and its accuracy is evaluated using the mean square error (MSE). The lowest MSE value among the ten results (E_{min}) and the difference between the highest and the lowest MSE values (E_{diff}) are shown in Table 1. The latter measurement gives a good indication of the robustness of a given algorithm [19].

As shown in the table, when compared to the global Poisson matting, our multichannel Poisson algorithm reduces E_{min} value by 65~90% and the E_{diff} value by 70~90%. This suggests that our approach is not only more accurate than the original Poisson matting, but also more tolerant to imprecise trimaps, which is an important property for video matting since automatically generated trimaps are generally not as accurate as manually labeled ones. It is also noteworthy that the performance gain is achieved without using any additional information. The results can be further improved if the background or depth information is available.

Figure 4: Results of our algorithm on “hair”, “camera”, “bird”, and “child” datasets. From top to bottom: source images, input trimaps, ground truth alpha mattes generated by Wang and Cohen [19], estimated alpha mattes using the proposed method, and composite results.

Table 1: Quantitative comparison with existing algorithms (measurements for existing approaches are reported by [19]). E_{min} : the minimum MSE value obtained using 10 different trimaps; E_{diff} : the difference between the maximum and the minimum MSE values.

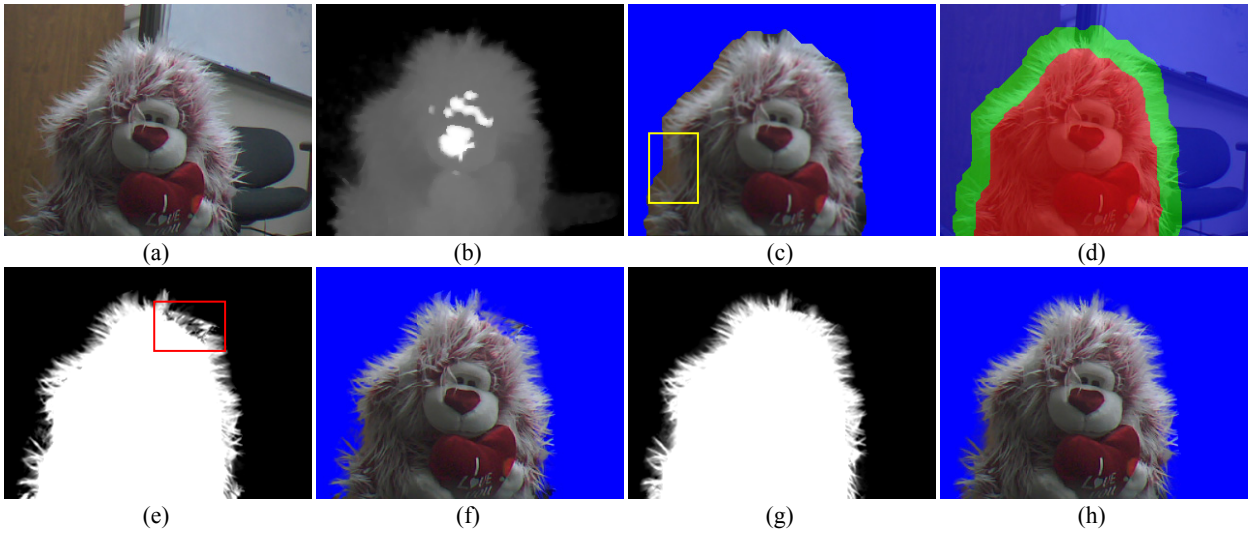
	Dog		Hair		Camera		Bird		Child	
	E_{min}	E_{diff}	E_{min}	E_{diff}	E_{min}	E_{diff}	E_{min}	E_{diff}	E_{min}	E_{diff}
Poisson [14]	340	1330	359	1830	451	2891	879	3174	832	2442
Random walk [7]	198	307	274	401	151	393	279	638	1732	1795
Knockout2	154	596	150	516	33	336	338	1387	435	888
Bayesian [4]	82	724	69	406	28	687	194	938	120	4994
Iterative BP [18]	69	356	78	362	27	227	207	903	214	553
Closed-form [9]	59	137	77	143	23	356	157	237	503	582
Robust [19]	41	95	31	165	10	155	69	381	114	394
Ours (rank)	78.1 (4)	234 (3)	67.2 (2)	317 (3)	48.3 (6)	292 (3)	188 (3)	912 (5)	287 (4)	509 (2)

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

Figure 5: Artifacts caused by imprecise trimap and unsmooth background: (a) source image, (b) trimap that has larger unknown areas than the one shown in Figure 4, (c) estimated alpha matte, (d) composite result.

(a) (b) (c)

Figure 6: Result comparison for the input frame shown in Figure 1: (a) result of our algorithm without using background knowledge (b) result of our algorithm using background knowledge (c) Bayesian video matting result, which also uses background.



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

(e) (f) (g) (h)

Figure 7: Result for a frame in the "toy" sequence: (a) color input (b) depth captured (c) bilayer segmentation result (d) automatically generated trimap (e-f) results obtained using color information only (g-h) results obtained using the additional depth information.

The comparison also suggests that our approach is comparable to other state-of-the-art matting approaches. It ranks on average 3.8 out of 8 on E_{min} measure and 3.2 out of 8 on E_{diff} measure. Considering our algorithm is designed for handling video sequences in real-time whereas others require seconds/minutes on a single image, this result is very encouraging.

Nevertheless, being a Poisson-matting-based approach, the presented algorithm inherits the assumption that the foreground and the unknown background are smooth. Otherwise, artifacts may occur. Figure 5 illustrates this problem, where sharp edges in background result in erroneous alpha matte. This problem becomes more noticeable when the unknown regions get larger, as the foreground/background colors inferred for the unknown regions based the smoothness assumption get more unreliable. Luckily, under controlled environments, this problem can be addressed by pre-capturing an image of the unsmooth background. Equation (12) can then be used for setting up the Poisson equation, which can accommodate the gradient changes within background itself.

5.2 Video Matting Results

Figure 1 and Figure 6 show the results on a frame of the "walk" sequence, which is originally presented by Chuang et al. [3]. The video is captured by a panning camera that follows a walking actress. Since there is no moving object in the background, once the foreground and background are separated, it is possible to fill in the missing background pixels using nearby frames [3]. Hence, we can evaluate the performance of our algorithm under both known and unknown background settings.

Figure 6(a) shows the result obtained without using the background information. While most of the hair details are extracted, artifacts do exist in the red rectangular area of the zoomed in view, where the window's reflection (observable in Figure 6(a)) is not recovered in the background estimated based on the smoothness assumption. These artifacts are removed when the known background reconstructed using multiple frames is used for setting up the Poisson equation. The final result is visually comparable to the one generated by the offline Bayesian video matting algorithm, which also uses the background.

A screen captured video is submitted. The different regions in the video are as follow: top left: input frame; top right: alpha matte extracted using the presented algorithm; bottom left: composite image with a constant color background; and bottom right: composite image with a natural image background.

The second sequence captures an indoor scene using the 3DV System's Z-Cam. While this camera can capture both color and depth information, the color image quality is not ideal. As shown in Figure 7(c), the poor color contrast and the complex shape of the furry toy cause the bilayer segmentation approach to incorrectly label some background pixels as foreground, especially in the area highlighted by the yellow rectangle. Most errors are corrected by the proposed matting algorithm by treating a wide band of pixels along the foreground/background boundaries as unknown. Nevertheless, as shown in Figure 7(e), when the Poisson equation is set up using color information only, artifacts exist in the area highlighted by the red rectangle. The artifacts are again caused by the violation of the background smoothness assumption. Fortunately, the strong gradient in the captured color data does not appear in the captured depth data. Hence, the artifacts are removed when the additional depth information is used for setting up the Poisson equation.

A video generated using this sequence is also submitted. The different regions are: top left: captured color channels; top center: captured depth channel; top right: composite image generated using bilayer segmentation results; bottom left: trimap generated from bilayer segmentation; bottom center: alpha matte extracted using the presented algorithm; and bottom right: composite image generated using estimated alpha matte.

In terms of processing speed, the algorithm is tested on a Lenovo S10 workstation with Intel 3GHz Core 2 Duo CPU and NVIDIA Quadro FX 1700 GPU. The presented video matting algorithm runs on the GPU at 40fps for video sequence of resolution 640×480. The bilayer segmentation part is implemented on the CPU using a separate procedure and runs at 21fps.

6 Conclusions and Future Work

A novel real-time online video matting algorithm is presented in this paper. The algorithm is based on the Poisson matting, which makes it suitable for parallel implementation on the GPUs. The matting results generated by our algorithm are much more accurate and robust than those of the original Poisson matting approach, thanks to the multichannel Poisson equations and the color-space matte initialization. The quantitative evaluation on still images shows that our results are comparable to state-of-the-art offline image matting techniques.

Much future work can be done along this direction. For example, in our current implementation, the temporal coherence in a video is used for background modeling only and hence is not fully utilized. How to better enforcing temporal coherence is certainly worthy for more investigation. In fact, we tried to use existing optical flow techniques to enforce the consistency among the alpha mattes of adjacent frames, but found that the estimated flows are unreliable along the fuzzy boundaries where they are needed the most. Hence a better way of enforcing temporal coherence is needed.

Acknowledgments:

The authors would like to thank the anonymous reviewers for their constructive and detailed comments. Fundings from NSERC, RDC IRIF fund are gratefully acknowledged.

References:

[1] X. Bai and G. Sapiro, "A Geodesic Framework for Fast Interactive Image and Video Segmentation and Matting," *Proc. ICCV*, 2007.

[2] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE TPAMI*, vol. 23, no. 11, pp. 1222-1239, 2001.

[3] Y.-Y. Chuang, A. Agarwala, B. Curless, D. H. Salesin, and R. Szeliski, "Video Matting of Complex Scenes," *Proc. Siggraph*, pp. 243-248, 2002.

[4] Y.-Y. Chuang, B. Curless, D. Salesin, and R. Szeliski, "A Bayesian Approach to Digital Matting," *Proc. CVPR*, pp. 264-271, 2001.

[5] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov, "Bilayer Segmentation of Live Video," *Proc. CVPR*, 2006.

[6] M. Gong and Y.-H. Yang, "Near-real-time image matting with known background," in *CRV*. Kelowna, BC, Canada, 2009.

[7] L. Grady, T. Schiwietz, S. Aharon, and R. Westermann, "Random Walks for Interactive Alpha-Matting," *Proc. VIIP*, pp. 423-429, 2005.

[8] N. Joshi, W. Matusik, and S. Avidan, "Natural Video Matting using Camera Arrays," *Proc. Siggraph*, 2006.

[9] A. Levin, D. Lischinski, and Y. Weiss, "A Closed Form Solution to Natural Image Matting," *IEEE TPAMI*, vol. 30, no. 2, pp. 228-242, 2008.

[10] Y. Li, J. Sun, and H.-Y. Shum, "Video Object Cut and Paste," *Proc. Siggraph*, pp. 595-600, 2005.

[11] M. McGuire, W. Matusik, H. Pfister, J. F. Hughes, and F. Durand, "Defocus Video Matting," *Proc. Siggraph*, 2005.

[12] M. McGuire, W. Matusik, and W. Yerazunis, "Practical, Real-time Studio Matting using Dual Imagers," *Proc. Eurographics Symposium on Rendering*, 2006.

[13] C. Rother, V. Kolmogorov, and A. Blake, ""GrabCut": interactive foreground extraction using iterated graph cuts," *Proc. Siggraph*, pp. 309-314, 2004.

[14] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum, "Poisson matting," *Proc. Siggraph*, pp. 315-321, 2004.

[15] J. Sun, W. Zhang, X. Tang, and H.-Y. Shum, "Background Cut," *Proc. ECCV*, pp. 628-641, 2006.

[16] J. Wang, P. Bhat, R. A. Colburn, M. Agrawala, and M. F. Cohen, "Interactive video cutout," *Proc. Siggraph*, pp. 585-594, 2005.

[17] J. Wang and M. Cohen, "Image and Video Matting: A Survey," *FTCGV*, vol. 3, no. 2, 2007.

[18] J. Wang and M. Cohen, "An iterative optimization approach for unified image segmentation and matting," *Proc. ICCV*, pp. 936-943, 2005.

[19] J. Wang and M. Cohen, "Optimized Color Sampling for Robust Matting," *Proc. CVPR*, 2007.

[20] O. Wang, J. Finger, Q. Yang, J. Davis, and R. Yang, "Automatic Natural Video Matting with Depth," *Proc. PG*, 2007.