Improved Poisson Matting for a Real Time Tele-presence System Using GPU

Xiaozhou Zhou
Department of Computing Science
University of Alberta
xzhou3@cs.ualberta.ca

Pierre Boulanger
Department of Computing Science
University of Alberta
pierreb@cs.ualberta.ca

Abstract

In this paper, an improved Poisson matting method is proposed to segment participants in real-time at a tele-presence session from their background. In order to improve the matting process, we introduce the concept of color distance and extend the standard Poisson matting using patch matching. The idea of patch based matching algorithm which is widely used in texture synthesis is adopted here to estimate the foreground and background color more precisely in complex scenes. A set of experimental results demonstrate the accuracy and robustness of the proposed method. We also present a GPU (Graphics Processing Unit) implementation of the algorithm capable of an average speed-up of 25 times compared to its CPU implementation.

1. Introduction

Tele-presence is one of the most important applications of computer vision and image processing nowadays. Tele-presence not only provides a shared virtual meeting environment, but also supports eye contact and non-verbal communication that are so important in real-world conversations. Ultimately tele-presence system will be able to create the illusion that participants are in the same meeting room and talking face to face.

In this paper, we focus on the first step of a real time tele-presence system where each participants need to be extracted from their background using a robust foreground-background segmentation technique. The extracted foregrounds (participants) are then processed in the next step of a tele-presence pipeline where it is inserted in the virtual meeting room. Examples of such systems can be found in the literature for virtual meeting [1][2][3][4] or for direct eye contact conferencing systems [2][3]. This segmentation process requires that the system must automatically determine if each pixel are either foreground or background. In many algorithms, some foreground parts will be miss-classified if the color of the background is close to its foreground. Many of the algorithms found in the literature do not work well for regions with subtle artifacts, such as hair and glasses and need to be improved in order to create high-quality video avatars.

Image matting is a process of image segmentation where pixels are classified as either foreground or background. The main different with other image segmentation techniques is that it allows the pixels to belong at the same time to the foreground or background categories and is frequently called “soft segmentation”. Image matting can be described mathematical by:

$$\mathbf{I}(i,j) = \boldsymbol{\alpha}_{i,j} * \mathbf{F}(i,j) + (1-\boldsymbol{\alpha}_{i,j}) * \mathbf{B}(i,j) \qquad (1)$$

(i,j) is the 2D image coordinates of a pixel and $\mathbf{I}(i,j)$ represents the color of the pixel (i,j). Similarly, $\mathbf{F}(i,j)$ and $\mathbf{B}(i,j)$ are the foreground and background color of pixel (i,j) respectively. The matrix $\boldsymbol{\alpha}$ is a weighing value for each pixel between the foreground and the background and is set to be between [0, 1]. When $\boldsymbol{\alpha}_{i,j} = 1$ this means that the pixel is definitely foreground, and when $\boldsymbol{\alpha}_{i,j} = 0$ this means that the pixel is definitely background. Image matting starts by defining a tri-map that can be initialized manually or automatically using IR imaging as in [6]. A tri-map image consists of pixels with three states: definite foreground (*DF*), definite background (*DB*), and unknown region.

Among all the image matting methods one can find in the literature, Poisson matting [5] has the most potential to be implemented for real-time applications because solving Poisson equation is the same at each pixel which is ideal for a GPU implementation.

Although the GPU processing power can help us reach our requirement for real time (30FPS), there are some constrains that must be imposed that will limit the quality of the matting results. First, the closest pixel (measured by the smallest Euclidean distance in color space) is chosen as the estimation of the foreground and background color in the unknown region. However, in real cases, the closest pixel color requirement is sometimes not the correct one. Moreover, this distance metric is not robust without neighbor and color information. In this paper, two algorithms are presented to overcome the shortcomings of the standard Poisson matting algorithm. The first improvement is based on a texture comparison technique of local patches [8], inspired by texture synthesis [7][8]. It takes advantage of the neighbor information and is employed to improve the estimation of the foreground and background. The other improvement is the use of color distance [10] to calculate the patch similarity. Color distance collects color information from RGB channels and simulates the YUV space. Experiment results demonstrate that the proposed methods enhance the accuracy of Poisson matting and works very well in the context of the real-time requirements for tele-presence.

The paper is organized as following. Section 2 starts by presenting the so called "standard" Poisson matting algorithm and then illustrates how to extend this algorithm. Section 3 presents experiment results comparing the two methods and its speed-up using GPU. We will then conclude and describe our future work plan.

2. Improved Poisson Matting

2.1. Standard Poisson Matting

There exist two different types of Poisson matting algorithms: global Poisson matting and local Poisson matting. Since local Poisson matting necessitates manual operation to post-process the matting result, in this paper, we will only deal with global Poisson matting as tele-presence require fully automatics segmentation. The main steps of Poisson matting as described in [5] are as following:

(1) *Foreground and background initialization:* create a tri-map classification either manually or using IR illumination as described in [6] or range data is in [9] .

(2) *Fill unknown regions:* for each pixel in the unknown region, find the nearest pixel in the definite foreground or definite background, and then copy the color from this pixel in two images *F* and *B*.

(3) α *Reconstruction:* according to the matting equation (1) described in [5], the partial derivatives of the original image is a combination of the derivative of the foreground image F and background image B and the weighing function $\boldsymbol{\alpha}$:

$$\nabla \mathbf{I} = (\mathbf{F}-\mathbf{B})\nabla\boldsymbol{\alpha}_{i,j} + \boldsymbol{\alpha}_{i,j}\nabla\mathbf{F} + (1-\boldsymbol{\alpha}_{i,j})\nabla\mathbf{B} \qquad (2)$$

where $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}\right)$.

Assuming the foreground and background are smooth where $\nabla\mathbf{F}$ and $\nabla\mathbf{B}$ close to zero and by ignoring the two terms $\boldsymbol{\alpha}_{i,j}\nabla\mathbf{F}$ and $(1-\boldsymbol{\alpha}_{i,j})\nabla\mathbf{B}$, Equation (2) can be rewritten as:

$$\nabla\boldsymbol{\alpha} = \frac{\nabla\mathbf{I}}{(\mathbf{F}-\mathbf{B})} \qquad (3)$$

Under Dirichlet boundary condition, the partial derivative of both sides of (3), matting equation is transformed into the form of a Poisson equation:

$$\Delta\boldsymbol{\alpha} = div\left(\frac{\nabla\mathbf{I}}{(\mathbf{F}-\mathbf{B})}\right) \qquad (4)$$

where $\Delta = \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)$ (5)

(4) *Refinement:* depending on the solution of Poisson equation, divide those pixels in the unknown region with $\boldsymbol{\alpha}$ values greater than 0.95 to the definite foreground. The same to the background, those pixels whose $\boldsymbol{\alpha}$ values less than 0.05 are added to the definite background.

(5) Iterate step (3) and (4) until convergence.

Please refer to [5] for more details about global Poisson matting.

2.2.Matching Using Patch Related Information

Besides the tri-map, the image is divides into three parts: target foreground (*TF*), target background (*TB*) and unknown region. The unknown region is the same with the tri-map. Target foreground is composed by those pixels whose neighboring pixels in the same patch are also in the definite foreground. Target background has the similar definition.

Suppose $\Psi(i,j)$ is a n x n patch centered at (i,j), the pixels in patch $\Psi(i,j)$ can be defined as:

$$P(i,j) = \{(m,n) \mid (m,n) \in \Psi(i,j)\} \qquad (6)$$

Therefore, the target foreground is:

$$TF = \{(s,t) \mid (s,t) \in DF, P(s,t) \subset DF\} \qquad (7)$$

Similarly, the target background is:

$$TB = \{(s,t) | (s,t) \in DB, P(s,t) \subset DB\} \quad (8)$$

The default patch size set in this paper is at a neighborhood of 9 x 9.

Inspired by [7], the estimated foreground colors are not just copied from the nearest pixels in the definite foreground. Instead, for each patch $\Psi(i,j)$ in the unknown region, we look at the closest patch $\Psi(p,q)$ in the target foreground (with the smallest distance). Then, the estimated foreground color of pixel (i,j) is determined by the average color of a 3 x 3 $\Psi'(p,q)$.The estimated background is done in the same way.

2.3. Color Distance

Before the definition of the distance between two patches, we introduce a Euclidian distance for color. In traditional Poisson matting, the color images are converted to gray value where color information is lost. The same value in gray scale does not means they come from the same 3- channel color. In other words, the smallest color distance in gray scale level is not equal to the smallest color distance in color space. Therefore, the measure of distance of two patches has to be processed in color space.

A simple way to calculate the color difference is the sum of squared difference (SSD) of 3 channels in RGB space. However, RGB does not model how human perceive color. Ideally one should convert the color to the YUV space but that would increase the computation load during execution.

As suggested in [10], a weighted Euclidean distance in RGB is described as:

$$d = \sqrt{2*(R_1 - R_2)^2 + 4*(G_1 - G_2)^2 + 3*(B_1 - B_2)^2} \quad (9)$$

This function is a simulation of the color distance in YUV space while the calculation depends on the RGB values directly. In this way, the distance of two patches is defined as the sum of color distance between all pairs of corresponding pixels using Equation 10.

2.4.Estimation of Foreground and Background

In addition, the construction of estimated foreground and background has a certain order. We give a confidence value to each pixel in the unknown region. For pixel (i,j), the confidence value is defined by how many pixels in $\Psi(i,j)$ belong to the definite foreground or background regions. The pixel with the highest confidence value has the highest priority to choose the closest patch. All the confidence values are updated when the pixels with the highest confidence values have found the closest patches.

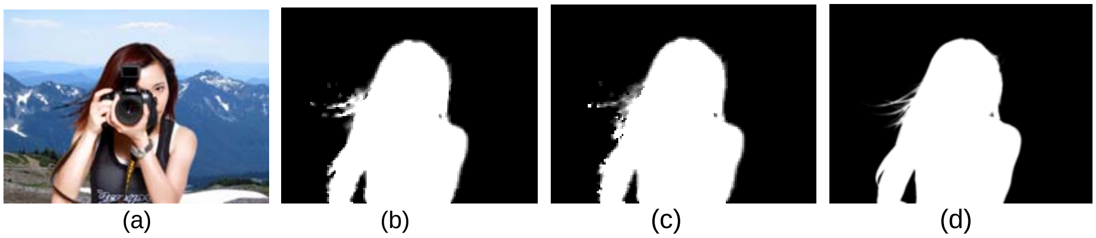

Figure 1. Comparison between improved method and standard Poisson matting (Image 1): (a) original image (b) proposed method (c) standard Poisson matting (d) ground truth

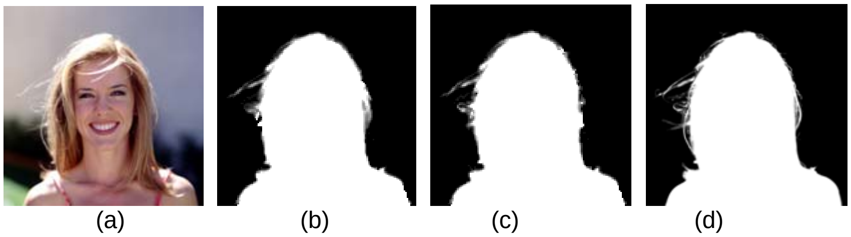

Figure 2. Comparison between improved method and standard Poisson matting (Image 2): (a) original image (b) proposed method (c) standard Poisson matting (d) ground truth

3. Experiment Results

We tested our algorithm compared to the standard Poisson matting technique and to a ground truth segmentation. Figure 1 and Figure 2 show the results obtained for both methods. From left to right, (a) is the original color image, (b) is the matting obtained from the proposed method, (c) is the matting obtained using standard Poisson matting and (d) is the ground truth. One can observe that the improved method can segment more precisely the image near subtle changes than the standard method.

According to [11], a MSE (Mean Square Error) estimate can be used as an efficient and objective criterion to give a more rigorous comparison between the two methods. Table 1 shows the MSE between the matting results and ground truths. On average the segmentation error was reduced by 5% to 10%.

Table 1. MSE comparison (Original Poisson matting/Improved method)

	Image 1	Image 2
Original	381.46	376.63
Improved	363.22	344.62
% of Improvement over original method	4.78%	8.5 %

3.2. Comparison of GPU and Implementation

We implement the improved Poisson matting on both CPU and GPU. The GPU version runs on an NVIDIA GeForce 9800 X2 display card, which provides CUDA (Compute Unified Device Architecture) to support parallel computation. The CPU implementation runs on an Intel (R) Core (TM) 2 Extreme CPU X9770 @ 3.02GHz. The size of Image 1 is 400*300 and the size of Image 2 is 400* 400.

Table 2. Running time comparison between CPU and GPU implementation (in seconds)

	Image1	Image2
CPU	97.4	96.9
GPU	3.8	4.7
Speed-up	26	20.6

Although we only have on average a 25 times speedup on one GPU. In the future, we will continue to optimize our GPU code and use multi-GPU to further improve the speed to the hard constraints of 30FPS.

4. Conclusion and Future Work

In this paper, we propose a new and improved matting algorithm that can be used in tele-presence system. By performing patch based comparison and by using color distance for a similarity metric we were able to improve significantly the matting process especially near complex borders. The experiments demonstrate that the improved Poisson matting works more accurate and robust than global Poisson matting. We also demonstrated that 25 times speedup are possible on a standard GPU. We are currently implementing the same algorithm on a 4 NVIDIA 5800 GPUs cluster where we hope to reach 30 FPS.

5. Acknowledgement

Thanks Dr. Jue Wang for providing the dataset.

Reference

[1] Gibbs SJ, Arapis C and Breiteneder CJ. TELEPORT – towards immersive co- presence. Multimedia Systems, 7(4): 214-221, 1999.

[2] Schreer O, Hendriks E,and *et al*. Virtual team user environment (VIRTUE) – a key application in telecommunication. In Proceedings of eBusiness and eWork, Prague, Cech Republic, CD-ROM.

[3] R.Tanger, P.Kauff and O.Schreer. Immersive meeting point (im.point) – An approach towards immersive media portals. In Proceedings of the Pacific – Rim Conference on Multimedia, Springer, Berlin, 2004.

[4] H.Harlyn Baker, Nina Bhatti and et.al. Computation and performance issues in Coliseum: an immersive videoconference system. In Proceedings of the eleventh ACM International Conference on Multimedia, November, Berkeley, CA, USA, 2003.

[5] J.Sun, J.Jia, C-K.Tang and H-Y. Shum. Poisson Matting. In Proceedings of ACM SIGGRAPH, 315-321, 2004.

[6] Wu, Q., Boulanger, P. and Bischof, W. F. Automatic bi-layer video segmentation based on sensor fusion. In Proceedings of International Conference on Pattern Recognition, Tampa, USA, December 8-11, pp. 1-4.

[7] A. Criminisi, P. P'erez, and K. Toyama. Object Removal by Exemplar-based Inpainting. In Proceedings of Computer Vision and Pattern Recognition, 2003.

[8] A.Efros and T.Leung, Texture synthesis by non-parametric sampling. In Proceedings of International Conference on Cpomputer Vision, 1033- 1038, 1999.

[9] G. Iddan and G. Yahav, “3D Imaging in the studio (and elsewhere)”, Proc. SPIE, 2001, pp. 48-55.

[10] http://www.compuphase.com/cmetric.htm

[11] J. Wang and M. Cohen, Optimized color sampling for robust matting. In Proceedings of Computer Vision and Pattern Recognition, 2007.