Mask Based Image Blending and Its Applications on Mobile Devices

Yingen Xiong\textsuperscript{a} and Kari Pulli\textsuperscript{a}

\textsuperscript{a}Nokia Research Center, 955 Page Mill Road, Palo Alto, CA 94304, USA

ABSTRACT

A mask based image blending approach is presented in this paper and implemented on mobile devices for producing mobile panoramic images. In this approach, a single channel mask is created and initialized with distributed values for each source image. It is warped and interpolated together with its source image during panorama stitching to keep related transformation information. The values of the mask provide weighting coefficients for blending images together to produce a panoramic image. By comparing with other complicated approaches such as gradient domain image blending, its computational and memory cost is low. By comparing with other simple approaches such as alpha blending, it does not need to locate boundaries of overlapping areas for determining weighting coefficient values. It can also be applied to 2D panorama stitching and its blending quality is better. The approach is implemented in a mobile panorama system to produce panoramic images for preview. It shows good performance in processing image sequences captured in both indoor and outdoor scenes. The property in low computational and memory costs as well as fast blending speed of the approach really benefits the mobile panorama system.

Keywords: Image stitching, image mosaic, image blending, mask, mobile panorama, mobile computing, mobile image processing, mobile computational photography

1. INTRODUCTION

As computational capabilities and memory of mobile devices increase and the camera quality on mobile devices improves, mobile image processing and mobile computational photography are more and more interesting. Like a computer, mobile devices can be used for photo editing, computer vision, animation, graphics, and so on. Here we are interested in creating panoramic images by combining a set of aligned images seamlessly on mobile devices. User can take an image sequence for a wide range scene with a mobile phone. After some processing, the user can see a panoramic image for the scene immediately. This can provide a very convenient application on mobile devices and a very useful tool for mobile image processing and mobile computational photography.

Image blending is the final and often very important step in creating image mosaics and panoramic images. A simple copying and pasting of overlapping areas of source images may produce visible artificial edges in the seam between images, due to differences in camera response and scene illumination, or geometrical alignment errors. Image blending can hide boundaries and reduce color differences between source images.

Existing methods for image stitching in the literature can be categorized as two main classes: \cite{1,2} optimal seam finding and transition smoothing. Each class has its own advantages and disadvantages.

Optimal seam finding algorithms such as \cite{1,3-6} search for an optimal seam in the overlapping area where the differences between source images are minimal. The composite image is produced by copying corresponding pixels from the source images using seam information. This kind of approaches works well when the source images are similar enough. However, when they are too dissimilar for the algorithms to find ideal seams, artifacts may still exist. Further processing is needed for removing them.

Transition smoothing algorithms reduce color differences between source images for hiding seams. Alpha blending\textsuperscript{7} is one of simple and fast transition smoothing approaches. It uses weighting combination to create the composite image. However, moving objects in the overlapping areas of source images or space alignment errors lead to ghosting artifacts. Recently, gradient domain image blending approaches\textsuperscript{6,8-12} are widely applied to image stitching and editing. A new gradient vector field is created with the gradients of source images and a new composite image can be recovered from the new gradient vector field by solving a Poisson equation. This
kind of algorithms can produce high quality composite images. However, the memory and computational cost is also high.

By comparison with computer, there is much less memory and low computational capability on mobile devices. In our mobile panorama system, two kinds of image blending algorithms are applied for different purposes. A simple and fast algorithm is used for producing a quick panoramic image for preview as soon as the image sequence is captured and a fine blending algorithm is used for creating a high quality image for the final result. Here we focus on the previous one.

In this paper, we are interested in developing an image blending algorithm which is capable of producing fast panoramic images with satisfying blending quality. The alpha blending algorithm is very simple and easy to implement, but its blending quality is not so good. There are some other disadvantages. Since the weighting coefficient values are dependent on distances between pixels and boundaries of the overlapping areas, the determination of these boundaries are crucial for the algorithm. When the mapping is complicated, it is not so easy to locate the boundaries. The algorithm can hardly deal with the case when there are more than two images overlapping in the same area. Further more, it is not suitable for 2D panorama stitching.

We present an image blending approach which does not need to compute boundaries of overlapping areas and can take care of color changes in any direction. In this approach, a mask is created for each source image and initialized with distributed values. It is transformed together with the source image during image stitching. The overlapped images can be combined together with the mask information. On one hand, the approach is simple and easy to be implemented on mobile devices just like the alpha blending approach. On the other hand, it can produce much better blending results than alpha blending and can be used in both 1D and 2D panorama stitching with multiple images overlapping in the same area. The detailed description for the approach will be given in following sections.

2. ORGANIZATION OF THIS PAPER

Here is the organization of the paper. In Section 3, we introduce the work flow of the approach. The details and comparison with alpha blending approach will be described in Section 4. An implementation of the mask in the approach is explained in Section 5. Applications and result analysis is discussed in Section 6. A summary of the paper is given in Section 7.

3. SUMMARY OF OUR APPROACH

Figure 1 shows the work flow of our approach. We assume that all source images are already aligned into a composite image $I_b$. The algorithm starts with allocating memory for the composite image $I_b$ and its mask $S_{mb}$. We use the first source image $S_1$ as the base image and copy it into the composite image with a corresponding position. Then we load the next source image $S_i$ into memory as our current processing image and create a single channel mask $S_{mi}$ for $S_i$. We assign values for all pixels of the mask with a certain pattern and distribution. The source image and mask are warped and interpolated together in the stitching process. In this way, the mask can keep all transformation and boundary information of the source image. Then we blend image $S_i$ onto the composite image $I_b$ by weighting combination with weighting coefficients provided by the mask. In the mean time, we update the mask $S_{mb}$ with the current mask $S_{mi}$. At this moment, the blending process for the current
image $S_i$ has been completed. We can load the next source image $S_{i+1}$ as the current image and repeat the blending process until all source images are blended. Finally, we can obtain the final composite image $I_b$ and a fully updated mask.

In the whole blending process, we only need to keep the final composite image $I_b$ with its mask $S_{mb}$, and one source image $S_i$ with its mask $S_{mi}$ in memory. In this way, we can perform image blending for large source images on mobile devices. The use of the mask $S_{mb}$ allows us to apply the algorithm in 2D image stitching with multiple images overlapping in the same area.

4. THE MASK BASED IMAGE BLENDING APPROACH

In this section, we first review alpha blending and its disadvantages, and then develop a mask based blending approach for creating mobile panorama.

4.1 Alpha Blending Algorithm and Its Disadvantages

Figure 2(a) shows the process of the algorithm. Suppose that there is an overlapping area $abcd$ which has four borders $ab$, $bc$, $cd$, and $da$. The algorithm scans the overlapping area with horizontal lines (or vertical lines). Each new pixel value is a weighting combination of the two corresponding pixels in image $I_1$ and image $I_2$. The weighting coefficients which are also called alpha are very important parameters. As Figure 2(a) shows, the alpha varies from 0 to 1. Each value of the alpha depends on the distance between the pixel and the border. So the blending quality is dependent on determination of borders of the overlapping areas.

Alpha blending is a very simple technique. Its computational and memory cost is low. It is suitable for the case of stitching two images with one dimensional translation. However, this kind of techniques needs to compute boundaries of overlapping areas in order to compute alpha values. When the mapping process is complicated, it is not easy to obtain boundary information. If there more than two images overlapping in the same area, it is difficult to determine alpha values.
4.2 The Mask Based Blending Approach

As shown in Figure 2(b), we create a mask for each source image and assign a value to each pixel according to its position. The value is related to the distance between the pixel and the center of the mask. The pixel at the center has the largest value $a$ and the pixels on borders have the smallest value $c$. The values of intermediate pixels are assigned between $a$ and $c$ with a distribution. A detailed explanation will be given in Section 4.3 and Section 5. The top of the figure is a mask created for the source image. It changes after warping and interpolating operations shown at the bottom of the figure.

After the source image and its mask are warped and interpolated, we can perform image blending. In the blending process, the result pixel value is a weighting combination of the corresponding pixels in overlapped images. The mask provides the weighting coefficients for the blending process. In this way, we do not need an extra process to determine the weighting coefficients. Since the mask keeps all mapping and transforming information, the blending operation becomes much simple.

From experimental and application results shown in Section 6, we can see that the mask based image blending approach can take care of any color change in all directions. It can smooth color transition in overlapping areas and can take care of the case of multiple images overlapping in the same area and can be applied to 2D panorama stitching.

4.3 Design of the Masks

Distributions of mask values are important. For different kinds of source images, we can use different distributions and parameters to obtain satisfying blending results. Linear and Gaussian distributions are usually used. Mask design is one of the important steps in this approach. Different masks can give different blending effects.

In mask design, we set a large value $a$ to the center pixel and a small value $c$ to the pixels on the boundaries. From the center to the boundaries, we can design some kinds of distributions for changes of the values. For example, we can use linear or Gaussian distributions. Different distributions will give different blending effects and different parameters of the distributions will also affect the blending quality. After the mask is warped and interpolated together with the source image, it still has distributed values. We use these values as weighting coefficients for image blending. More explanations for distributions of mask values will be given in the next section.

5. IMPLEMENTATION

Figure 3 (a) shows implementation of the mask in real applications. We create a 32 bit image to store the source image and mask information. As shown on the top of the figure, for each pixel, channel R, G, and B needs 8 bits respectively. This leaves 8 bits. We can use the 8 bits to store the mask value for the pixel.
The bottom left of Figure 3 (a) shows initial values in different pixels of the mask. In the center, the mask has a value of 255 and on boundaries it has a value of 1. Each value can be stored with 8 bits. Together with RGB values, every pixel has 32 bits. The mask is warped and mapped together with the source image. The bottom right of Figure 3 (a) shows the transformed mask.

Figure 3 (b) shows an example how the mask changes after it is wrapped with the source image. The top left of Figure 3 (b) shows the source image and the bottom left of Figure 3 (b) is a mask which is created for this image. In the center, the value is 255, and on the borders, the value is 1. We use linear distribution to set the values between the center to the borders. The top right of Figure 3 (b) shows the warped image and the bottom right of Figure 3 (b) shows the warped mask. From the figure we can see that the distorted mask keeps all information of the warped image.

We can use different distributions to set the mask values. For example, we can use linear and Gaussian distributions. Also we can use different patterns to set up the value distribution. In Figure 3 (a), the values on same rectangle in the mask are set to be same. Of course, we can also use other patterns such as circles and ellipses. Different distributions and patterns will cause different effects in blending operation. We also can use different parameters in distributions to get different results. For example, by changing parameters of Gaussian distribution, we can make its energy more concentrate to the center. In this case, the weighting coefficients will be smaller on borders during blending operation. Figure 4 shows some patterns and distributions which are usually used for setting masks. They include rectangular pattern with linear distribution, rectangular pattern with Gaussian distribution, circular pattern with Gaussian distribution, and circular pattern with linear distribution.

6. APPLICATIONS AND RESULT ANALYSIS

The approach is implemented on mobile devices for image stitching. It has been tested with different image sequences captured in different conditions. It shows satisfying performance in both indoor and outdoor scenes. In this section, we show some example applications and results.

6.1 Applications in Viewfinder Image Stitching

First, we apply the approach to a stitching process of viewfinder images. Figure 5 (a) and (b) shows some example results. In each figure, the top shows the blending result and the bottom shows the source images. From the results we can see that the approach can handle the color changes in both x and y directions. It can smooth color transition between the adjacent images and merge them together seamlessly.
From Figure 5(a) we can also see another fact. Although the color and luminance of the two source images are very different, the blending result is still satisfying.

6.2 Applications in Website Image Stitching

We apply the approach to image sequences downloaded from website. Figure 6 shows some example results. The image size is much larger than the viewfinder image size. Like applications to viewfinder image stitching, the approach still can obtain satisfying blending results for these image sequences. Although the color and luminance of images in the image sequence of Figure 6 (a) is different, the approach can still obtain the blending result with satisfying smooth color transition. The results also show that the approach works fine when it is applied to image sequences.
6.3 Applications in the Case of Multiple Images Overlapping in the Same Area

In the actual applications, we require that the blending approach can handle the case of 2D panorama stitching and multiple images overlapping in the same area. Figure 7 shows an example of the case. In the figure, the bottom shows the source images and the top shows the result. There are 6 source images in the sequence which is captured with 2D motion. Some areas are overlapped by more than two images. From the result we can see that the approach can be used in 2D panorama stitching and it does not care the stitching order. The result also shows that the approach can handle the case which there are multiple images overlapping in the same area.

Besides, we have also applied the approach to the case of indoor scenes. Figure 8 shows an example result. From the result we can see that the approach can produce satisfying result for the scene. It shows good performance.
7. CONCLUSIONS AND DISCUSSION

In this paper, a mask based image blending approach for image stitching is developed. Since the computational and memory cost is low, it is suitable for implementation on mobile devices. It does not need to compute boundaries of overlapping areas. The blending quality is satisfying.

In this approach, a single channel mask is created for each source image and initialized with distributed values. It is warped and interpolated together with the source image. The values of the mask provide weighting coefficients for blending the overlapped images together to produce panoramic images. The approach does not need to compute boundaries of overlapping areas. All needed information is provided by the mask. Different distributions of the pixel values in the mask will give different blending effects. The approach can take care of color changes in all directions in overlapping areas during blending operations. It does not care whether the mapping process is simple or complicated. It is also suitable to the case of multiple images overlapping in the same area. The approach is implemented on mobile devices to produce quick panoramic images for preview. The performance and results are satisfying.

Future work includes further speeding up the algorithm. OpenGL will be considered to be used.

REFERENCES