2009 11th IEEE International Symposium on Multimedia

Fast Image Labeling for Creating High-Resolution Panoramic Images on Mobile Devices

Yingen Xiong and Kari Pulli
Nokia Research Center
Palo Alto, CA 94304, USA
{*yingen.xiong, kari.pulli*}*@nokia.com*

***Abstract*—We present an image labeling approach for merging a set of aligned source images into a composite image by finding optimal seams in the overlapping areas of the source images quickly and using little memory. A minimal-cost path in the overlapping area of two images is found by dynamic programming and used as an optimal seam to label images. The overlapping images are cut along the seam and merged together. A sequential image stitching procedure is integrated with the fast image labeling for producing high-resolution and high-quality panoramic images using large source images under limited computational and memory resources. The approach presents several advantages: the use of dynamic programming optimization for finding the minimal-cost path over adjacent source images guarantees finding the optimal seam and allows images to be merged quickly; ghosting and blurring problems caused by moving objects and small registration errors can be avoided by the optimal seam finding process; the combination of the sequential image stitching procedure with the fast image labeling allows processing large source images for creating high-resolution panoramic images using little memory; the fast labeling process is easy to combine with intensive blending to produce high-quality panoramic images. The method is implemented in our mobile panorama system and runs with good performance on mobile devices.**

***Keywords*-image stitching; image labeling; mobile panorama; mobile image processing; mobile computational photography; optimal seam finding; fast labeling; high resolution panorama**

I. Introduction

Mobile phones are not only communication tools, but also capable computational devices equipped with high-resolution digital cameras, high-quality color displays, and GPU hardware. Applications such as mobile augmented reality [1], [2], mobile local search [3], and mobile image matching and recognition [4] used to only work on desktop computers, but can now run on mobile phones. We are interested in creating high-resolution and high-quality panoramic images on mobile devices. A user can capture an image sequence for a wide range of scenes with a camera phone and see a panoramic image created on the mobile phone immediately, and then send it to friends or upload it onto website.

A panoramic image is created from an image sequence. Its construction process requires a lot of computation and memory. This might not be a problem for desktop computers, but mobile devices only have limited computational and memory resources. It is necessary to develop efficient panorama stitching approaches which can be applied for applications on mobile devices.

In this paper, we develop a fast image labeling approach which is capable of combining a set of aligned source images into a composite image by finding optimal seams in overlapping areas between adjacent source images and can be applied to produce high-resolution and high-quality panoramic images on mobile devices using less computation and memory than other often used methods.

A. Related Work

Our work is mainly related to image stitching. There are two main categories of image stitching approaches: transition smoothing and optimal seam finding. Combination of these two categories is also used to make full use of their advantages and avoid their disadvantages.

Transition smoothing approaches reduce color differences between source images to remove stitching artifacts. Alpha blending [5] is a widely used simple and fast transition smoothing approach. It uses a weighted combination of inputs to create a composite image. The main problem of alpha blending is that moving objects and small registration errors will cause ghosting artifacts and blurring problems. Recently, gradient domain image blending approaches [13], [7], [8], [9], [10], [11], [12] have been applied to image stitching and editing. They create a new gradient vector field by combining source image gradients to construct a Poisson equation. A composite image can be recovered from the new gradient vector field by solving the Poisson equation with boundary conditions. These algorithms can reduce color differences of source images due to changes of scene illumination and variations in camera responses during image capture and smoothen color transition for the whole composite image, producing high-quality composite images. However, their processing speed is slow, and memory costs are high. It is difficult to process large source images to create high-resolution panoramic images on mobile devices with limited computational and memory resources.

Optimal seam finding algorithms [14], [6], [15], [16], [17] search for optimal seams in overlapping areas where differences between source images are minimal. Labeling

978-0-7695-3890-7/09 $26.00 © 2009 IEEE
DOI 10.1109/ISM.2009.122

all pixels of the composite image and source images can be created using the optimal seams. The composite image is produced by copying corresponding pixels from the source images using labeling information. Agarwala et al. [6] use graph cut optimization to find contribution regions among several source images. Pixel labeling is performed by minimizing over all source images at the same time. The algorithm can be used to find different optimal seams by given different cost functions to cut the source images for creating composite images. However, computational and memory costs are high during graph cut optimization, so using it for image stitching on mobile devices may be too expensive, especially with high-resolution images.

The combination of optimal seam finding and transition smoothing for image stitching is also used in panorama applications on desktop [6] and mobile devices [18]. In this case, the source images are combined together using the seams found by the optimal seam finding operation. If the seams and stitching artifacts are still visible, transition smoothing is applied to reduce the color differences between the source images to hide the seams and remove the stitching artifacts. Graph cut optimization finds the optimal seams and Poisson blending is applied for transition smoothing. High-quality panoramic images can be produced, but the computational and memory costs are high.

In our work, a fast and low memory cost image labeling approach is created. During image labeling, an error surface is constructed with squared differences of overlapped images. A low-cost path is found through the error surface by dynamic programming and used as an optimal seam to create labeling. The overlapping images are merged together along with the optimal seam. A sequential image stitching procedure is created with the fast image labeling approach. In this way, we can produce high-resolution panoramic images with large source images with low computational and memory costs. The approach is implemented in our panorama system to create high-resolution panoramic images on mobile devices. In order to compare and evaluate the performance of the approach, we also implement a widely used graph-cut-based labeling algorithm. We compare results to demonstrate advantages of the fast labeling approach in processing speed and memory usage. Good performance has been obtained for both indoor and outdoor scenes.

B. *Organization of the Paper*

In Section II, we introduce the work flow of our approach. The details of the fast image labeling approach for producing mobile panoramic images are described in Section III. Applications and result analysis are discussed in Section IV. A summary of the paper is given in Section V.

II. Summary of Our Approach

Figure 1 shows the details of the sequential image stitching procedure with the fast image labeling approach.

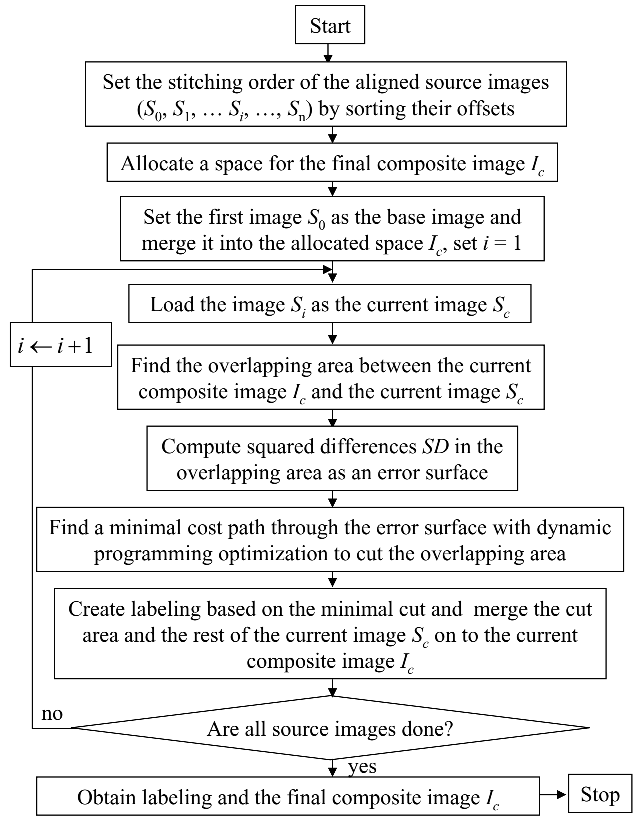

Figure 1. Work flow of the sequential image stitching procedure with the fast labeling approach.

The procedure starts with setting the stitching order $(S_0, S_1, \ldots, S_n)$ of the source images by sorting their offsets. After allocating memory space for the final composite image I_c, we set the first source image S_0 as the base image and put it into I_c as the current composite image. We continue the stitching process by inputting the next source image as the current image S_c.

We find the overlapping area between the current composite image I_c and the current image S_c and compute squared differences SD in this area as an error surface. A minimal cost path through the error surface can be found with dynamic programming. The two images match best along the minimal cost path. We use it as the optimal seam for creating labeling to cut the overlapping area. After the cut area and the remaining part of the current image are merged onto the current composite image, the process for the current image is completed. We input the next source image as the current image S_c to continue the stitching process.

After all source images are processed, we obtain the final composite image. During image stitching, we only need to keep the current composite image and the current source image in memory, which enables us to process large source images with limited computational resources in the mobile panorama system.

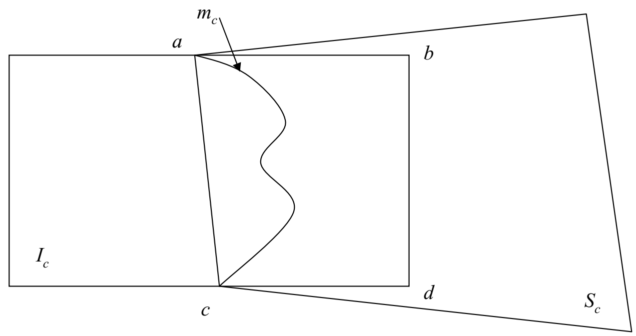

Figure 2. Find the optimal seam between the current composite image and the current source image.

III. Image Labeling with Dynamic Programming Optimization

We aim to find optimal seams to merge the source images together quickly and using little memory, so that it can be applied for producing high-resolution panoramic images on mobile devices. Here we describe the approach in detail. In order to compare and evaluate performance, we also implemented a labeling approach using graph cut optimization.

A. Labeling with Dynamic Programming

We want to merge the images on places where they differ the least. As shown in Figure 2, suppose that $abcd$ is the overlapping area between the current composite image I_c and the current source image S_c. I_c^o and S_c^o are the overlapping images in the area $abcd$ of I_c and S_c respectively. We compute squared differences d between I_c^o and S_c^o as an error surface,

$$d = (I_c^o - S_c^o)^2. \tag{1}$$

We apply dynamic programming to find a minimal cost path through this surface. We scan the error surface row by row and compute the cumulative minimum squared difference D for all paths,

$$\begin{aligned} D(h, w) = d(h, w) + \min(&D(h-1, w-1), \\ &D(h-1, w), D(h-1, w+1)) \end{aligned} \tag{2}$$

where $h = 2, \ldots, n_r$ and $w = 2, \ldots, n_c$ are the indices of the row and column of the error surface respectively.

The optimal path m_c can be obtained by tracing back the paths with a minimal cost from bottom to top.

For the last row, the minimum value can be used to determine the end (h_o, w_o) of the optimal path. For the next upper row, if $D(h_o - 1, w) = D(h_o, w_o) - d(h_o, w_o), w \in \{w_o - 1, w_o, w_o + 1\}$, then the position of the optimal path in this row is $(h_o - 1, w)$. We repeat the process until all rows have been traced.

Figure 3 shows the process of optimal seam finding with dynamic programming optimization. Figures 3 (a) and (b) are the overlapping images in the overlapping area of $abcd$ of I_c and S_c respectively. The error surface shown in Figure 3 (c) is computed through the squared differences between the two images shown in Figure 3 (a) and (b). After that, the cumulative squared difference D is computed with the error surface d and is shown in Figure 3 (d). Meanwhile, we also obtain all possible paths shown in Figure 3 (e). After tracing back with dynamic programming, we obtain an optimal path shown in Figure 3 (f) along which the two images in (a) and (b) match best. We use the optimal path as an optimal seam to create labeling and cut the overlapped images.

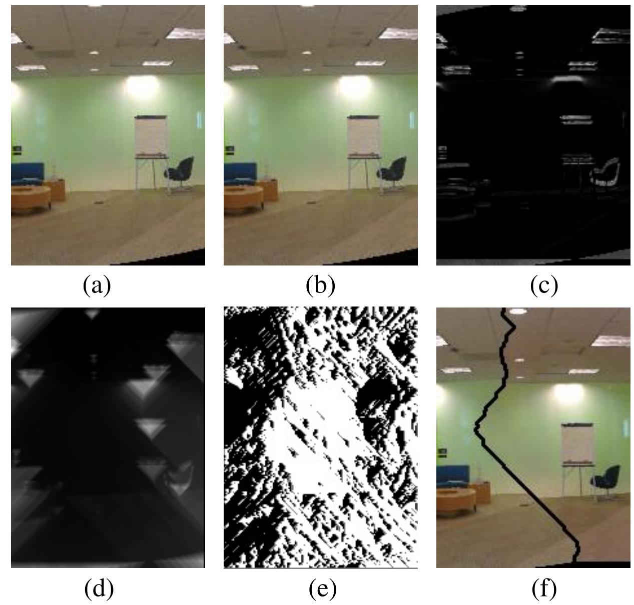

Figure 3. Process of finding the optimal seam with dynamic programming optimization.

We update the current composite image I_c by merging the current image S_c with the labeling information and continue the labeling process with the next source image. After all source images are processed, we obtain the final composite image.

B. Labeling with Graph Cut Optimization

Figure 4 shows an image sequence with aligned source images $S_0, S_1, \ldots, S_n$ and the composite image I_c. We apply graph cut optimization to find optimal seams $m_0, m_1, ..., m_{n-1}$ in the overlapping areas and create mapping or labeling between pixels in image I_c and the source images. With the labeling, we can copy corresponding pixels from the source images to the composite image I_c.

Objective functions are important for graph cut optimization. With different kinds of objective functions, we can obtain different results. For optimal seam finding or labeling, an objective function O is a function of a pixel labeling L, i.e., $O(L)$.

Considering implementation on mobile devices, we create

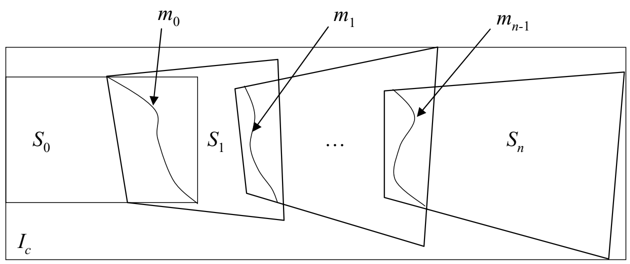

Figure 4. Find optimal seams with graph cut optimization.

a simple and efficient objective function which includes two items: pixel property P_p and color differences D_p between neighboring pixels.

$$O(L) = \sum_{k} P_p(k, L(k)) + \sum_{k,j} D_p(k, j, L(k), L(j)) \quad (3)$$

where

$P_p(k, L(k))$ depends on the property of pixel k;
$D_p(k, j, L(k), L(j))$ is the color difference over all pairs of neighboring pixels k, j.

For invalid pixels, we set the pixel property item $P_p(k, L(k))$ to a very large number, which means that the seams may not go to invalid areas. Otherwise, we set it to zero, i.e.,

$$P_p(k, L(k)) = \begin{cases} N & \forall k \in \Phi \\ 0 & \text{others} \end{cases} \quad (4)$$

where

N is a large number;
Φ is an invalid area.

The invalid areas are created in image mapping process after spatial alignment.

For color differences D_p, we compute Euclidean distances in RGB space over all pairs of neighboring pixels k, j, i.e.,

$$\begin{aligned} &D_p(k, j, L(k), L(j)) \\ &= \|S_{L(k)}(k) - S_{L(j)}(k)\| + \|S_{L(k)}(j) - S_{L(j)}(j)\| \end{aligned} \quad (5)$$

where

S_i is source image i;
$L(k)$ is the label of pixel k.

Of course, we can add other items into the objective function to consider other properties when cutting the source images. For example, we can add an item to consider edge information, and so on, however, it will use more computational and memory.

The optimal seam finding or labeling problem can be cast as a binary graph cut problem. The procedure of "alpha expansion" [19] can be used to minimize the objective function and obtain an optimal solution globally.

However, the optimization process using graph cuts needs to keep all source images in memory. For implementation on a desktop computer, it may not be a big problem, but for implementation on mobile devices which have more limited resources, it will be a crucial problem when processing large source images. Besides, the application of multi-label graph cuts requires a potentially large number graph cut iterations. It is very slow for applications on mobile devices. These problems are solved in the fast image labeling approach presented in the previous section.

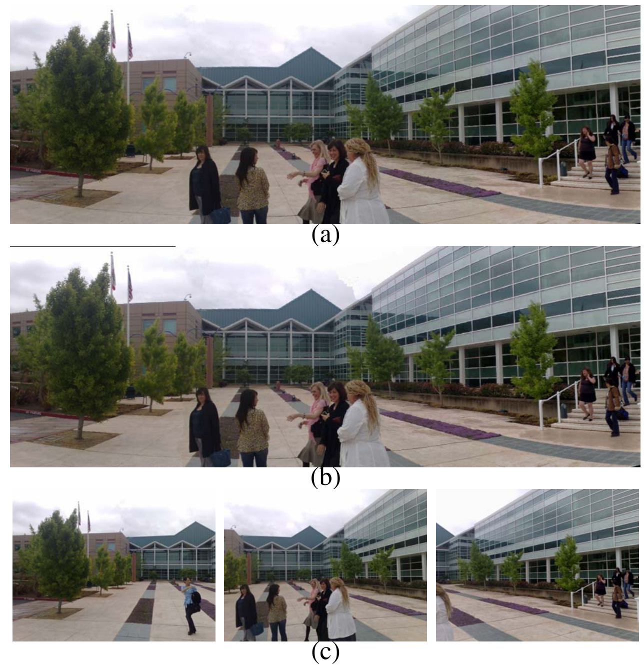

Figure 5. Panoramic images produced by the fast labeling (a) and graph cut optimization (b) approaches with three 1024×768 source images (c) of similar color and luminance.

IV. Applications and Result Analysis

The fast image labeling approach is integrated into a sequential image stitching procedure and implemented in our mobile panorama system for producing high-resolution panoramic images on mobile devices. It has been tested under different conditions, and it yields good performance for both indoor and outdoor scenes. We compare the results obtained by the fast labeling approach and graph cut optimization to demonstrate advantages of the fast labeling in processing speed and memory consumption. In this section, we present some results obtained by running the method on Nokia N95 8GB mobile phones with an ARM 11 332 MHz processor and 128 MB RAM. It can also be run on other mobile devices. In these applications, the size of source images is 1024×768. We have also applied the approach to larger source images. It works fine and performance is satisfying.

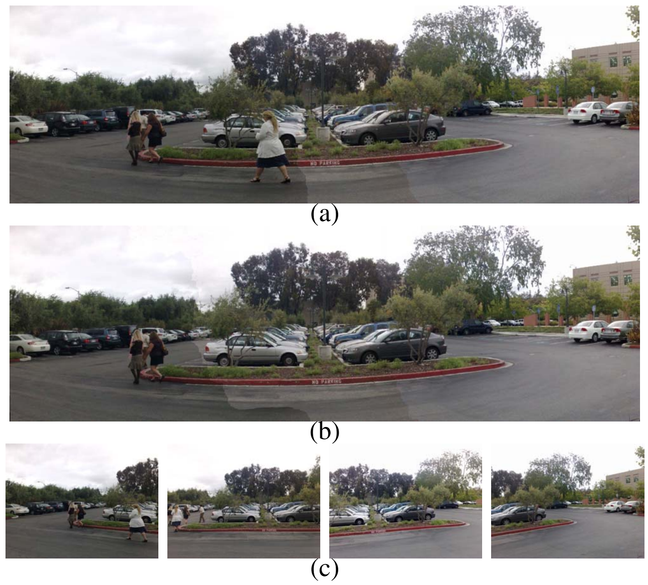

Figure 6. Panoramic images produced by the fast labeling (a) and graph cut optimization (b) approaches with four 1024×768 source images (c) of different color and luminance.

A. Applications to image stitching for source images with similar color and luminance

We apply our approach to cases in which the color and luminance in captured source images are similar. Figure 5 shows an example. The image sequence includes three source images shown in Figure 5 (c). After image alignment, the source images are stitched together to create a panoramic image. Figures 5 (a) and (b) show the results created by the fast image labeling and graph cut optimization approaches, respectively. The fast labeling takes 3.44 seconds and the graph cut optimization takes 121.88 seconds to label the images, thus fast labeling is about 35 times faster.

We can also observe that the seams created by fast labeling are better than the ones created by the graph cut approach. They are almost invisible in the panoramic image shown in Figure 5 (a). In this case we can use it as the final result. However, the seams in Figure 5 (b) can still be seen. Further processing is needed to reduce the color differences between the source images for hiding the seams. Notice also that while the image sequence is captured, people are moving in the scene. Both approaches can find seams that avoid ghosting or tearing problems across the seams.

B. Applications to image stitching for source images with different color and luminance

Figure 6 shows an example of applying our approach to stitching source images with different color and luminance levels. The image sequence includes four source images captured in an outdoor scene. Figures 6 (a) and (b) show the results produced by the fast labeling and graph cut optimization approaches, respectively. In the panorama stitching, the fast image labeling takes 5.92 seconds and the graph cut approach takes almost 30 times longer, 165.92 seconds

Like the previous application, people in the scene are moving while the image sequence is captured. Both approaches can create proper seams to avoid ghosting problems, but the fast labeling approach also keeps one more moving object in the panoramic image. Since the luminance in first two images is very different from the third and fourth images, neither approach could find invisible seams. Stitching artifacts can still be seen. Further blending is needed to reduce the color differences to hide the seams in the panoramic images.

C. Applications to image stitching for indoor scenes

We apply our approach to panorama stitching for source images captured in indoor scenes. Figure 7 shows an example with seven source images. The results created by the the fast labeling and graph cut optimization approaches are shown in Figures 7 (a) and (b), respectively. In the panorama stitching, the fast image labeling approach takes 17 seconds and the graph cut optimization approach takes about 35 times longer, 595 seconds.

From Figure 7 (c) we can see that the luminance among the source images in the image sequence is very different, especially between the first four and the others. From the results shown in Figures 7 (a) and (b) we can see that the seams which are found by the fast labeling approach are much better than those found by the graph cut optimization approach. The previous are almost invisible and the latter can still be seen.

D. Applications for creating $360°$ *panorama with very long image sequences*

We apply our image stitching method for creating $360°$ panoramic images with very long image sequences. Figure 8 shows an example. In this application, there are 17 source images in the image sequence. In this case, the graph cut optimization approach can not be run on the mobile phone since there is not enough memory available. However, the fast labeling approach still can. We compare their performance by running them on a desktop computer. The results created by the fast labeling and graph cut optimization approaches are shown in Figure 8 (a) and (b) respectively. The fast labeling approach is about 92 times faster than the graph cut optimization approach. According to our tests, the longer the image sequences, the more advantage the fast labeling approach provides. Besides, it can process more and larger source images than the graph cut optimization approach can on mobile devices with limited resources.

When running on a mobile phone, the fast image labeling approach takes 34 seconds and the result is the same as in Figure 8 (a). Since the fast labeling approach is integrated in the sequential image stitching procedure, during the panorama stitching, it only needs to keep the panoramic

Figure 7. Panoramic images produced by the fast labeling (a) and graph cut (b) with 7 1024×768 source images (c) in an indoor scene. The seams found by the fast labeling approach are much better than the ones by the approach with graph cut optimization. They are almost invisible. However, the graph cut seams can be seen clearly.

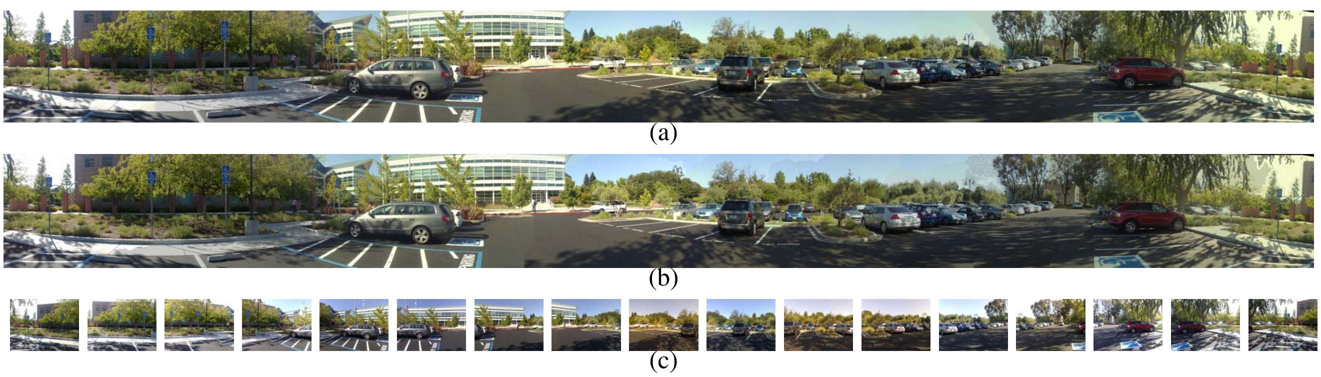

Figure 8. Panoramic images produced by image stitching using the fast image labeling (a) and graph cut optimization (b) with 17 1024×768 aligned source images (c). The previous is more than 92 times faster than the latter and only needs to keep one source image in memory during the stitching while graph cuts needs to keep all source images.

image and the current source image in memory. As long as there is enough memory for the final panoramic image and the current source image, the approach does not care how many source images are processed during stitching. However, the graph cut optimization approach needs to keep all source images in memory to find all optimal seams globally for the panoramic image at one time. It needs much more memory than the fast labeling approach for processing the same number of source images. The fast processing speed and low memory consumption are two main advantages of the fast labeling approach for mobile implementation.

Our approach has been tested with many image sequences with different cases. Figure 9 shows more results produced by the fast image labeling approach with long image sequences. All these results were obtained on mobile phones. In order to reduce differences of color and luminance and to hide the seams, we perform color and luminance compensation for source images before stitching them onto panoramic images and simple linear blending along with the seams in these applications. We will report these details in a future paper. From Figure 9 we can see that the panoramic images have good color and luminance transition. The fast labeling approach presents good performance in these different cases.

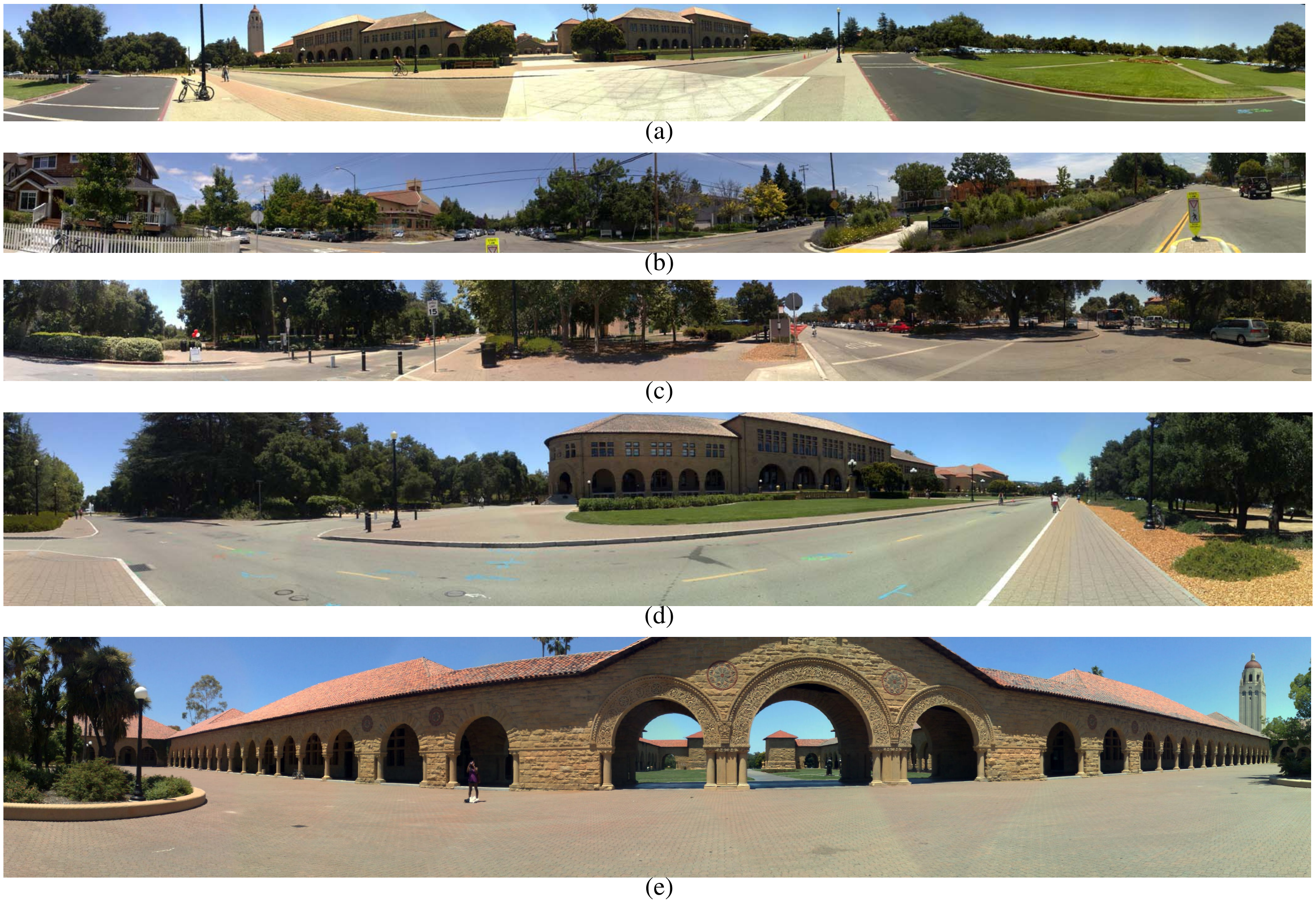

Figure 9. More examples of panoramic images produced by image stitching using the fast image labeling with long image sequences.

V. Conclusions and Discussion

W have presented a fast image labeling approach and implemented it in our mobile panorama system to produce high-resolution panoramic images. Compared to graph cut optimization, it can produce panoramic images much faster and using much less memory. When it is applied to scenes of from 5 to 17 images, it is from 30 to 92 times faster than the graph cut approach, respectively, and when the scene illumination does not change much while the image sequence is captured, it may find invisible seams for the panoramic image. The approach can be applied to create very large high-resolution panoramic images with large source images as long as the system has enough memory for the final panoramic image and the current processing source image.

The approach is very simple. After the overlapping area between two images is located, an error surface created by computing the squared differences of color in the overlapping area. A low-cost path is found by dynamic programming optimization. Along with the low-cost path, the two images match best. The path is used as the optimal seam to create labeling. The two images can be cut along the seam. A sequential image stitch procedure is also created and integrated with the fast labeling approach to create panoramic images. The use of the integration allows us to produce high-resolution panoramic images from large source images with fast speed and low memory consumption.

There are two main advantages of the approach. It can perform image labeling in panorama stitching with very fast speed. The stitching quality is as good, and sometimes better, as the graph cut optimization approach. The combination of the sequential image stitching procedure and the fast labeling approach allows us to create large panoramic images with limited resources. These advantages are crucial for implementation on mobile devices.

Our approach is implemented in a mobile panorama system and can be run on mobile devices. It has been tested and applied to different scenes with different conditions. From the example applications and tests, we can analyze the performance of the approach. The approach has these properties: in the case of source images with similar color and luminance, the approach may find invisible seams; like other optimal seam finding approaches, it can find optimal seams to avoid ghosting and blurring problems caused by moving objects and small spatial registration errors; the property of fast speed and low memory consumption allows us to create high-resolution and high-quality panoramic images on mobile devices efficiently; and it can be applied to quickly create 360° panoramic images from long image sequences on mobile devices.

Future work includes color and luminance compensation before image labeling and combination with fast transition smoothing for hiding visible seams and removing stitching artifacts.

REFERENCES

[1] W.-C. Chen, Y. Xiong, J. Gao, N. Gelfand, and R. Grzeszczuk, "Efficient extraction of robust image features on mobile devices," in *ISMAR '07: Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 1–2.

[2] D. Wagner, T. Langlotz, and D. Schmalstieg, "Robust and unobtrusive marker tracking on mobile phones," *Mixed and Augmented Reality, IEEE / ACM International Symposium on*, vol. 0, pp. 121–124, 2008.

[3] G. Takacs, V. Chandrasekhar, N. Gelfand, Y. Xiong, W.-C. Chen, T. Bismpigiannis, R. Grzeszczuk, K. Pulli, and B. Girod, "Outdoors augmented reality on mobile phone using loxel-based visual feature organization," in *MIR '08: Proceeding of the 1st ACM international conference on Multimedia information retrieval*. New York, NY, USA: ACM, 2008, pp. 427–434.

[4] D. Chen, S. S. Tsai, V. Chandrasekhar, G. Takacs, J. Singh, and B. Girod, "Robust image retrieval using multiview scalable vocabulary trees," in *Visual Communications and Image Processing 2009*, M. Rabbani and R. L. Stevenson, Eds., vol. 7257. SPIE, 2009, p. 72570V. [Online]. Available: http://link.aip.org/link/?PSI/7257/72570V/1

[5] Y. Xiong and K. Pulli, "Mask based image blending approach and its applications on mobile devices," in *the Sixth International Symposium on Multispectral Image Processing and Pattern Recognition*. SPIE, 2009.

[6] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen, "Interactive digital photomontage," *ACM Trans. Graph*, vol. 23, pp. 294–302, 2004.

[7] A. Levin, A. Zomet, S. Peleg, and Y. Weiss, "Seamless image stitching in the gradient domain," in *In Eighth European Conference on Computer Vision (ECCV 2004*. Springer-Verlag, 2004, pp. 377–389.

[8] R. Szeliski, M. Uyttendaele, and D. Steedly, "Fast poisson blending using multi-splines," in *Technical Report MSR-TR-2008-58, Microsoft Research*, 2008.

[9] M. Kazhdan and H. Hoppe, "Streaming multigrid for gradient-domain operations on large images," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 1–10, 2008.

[10] R. Fattal, D. Lischinski, and M. Werman, "Gradient domain high dynamic range compression," in *SIGGRAPH '02: Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 2002, pp. 249–256.

[11] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 313–318, 2003.

[12] J. Jia, J. Sun, C.-K. Tang, and H.-Y. Shum, "Drag-and-drop pasting," in *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*. New York, NY, USA: ACM, 2006, pp. 631–637.

[13] Z. Farbman, G. Hoffer, Y. Lipman, D. Cohen-Or, and D. Lischinski, "Coordinates for instant image cloning," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 1–9, 2009.

[14] N. Gracias, M. Mahoor, S. Negahdaripour, and A. Gleason, "Fast image blending using watersheds and graph cuts," *Image Vision Comput.*, vol. 27, no. 5, pp. 597–607, 2009.

[15] D. L. Milgram, "Computer methods for creating photomosaics," *IEEE Trans. Comput.*, vol. 24, no. 11, pp. 1113–1119, 1975.

[16] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 2001, pp. 341–346.

[17] J. Davis, "Mosaics of scenes with moving objects," in *CVPR '98: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Washington, DC, USA: IEEE Computer Society, 1998, p. 354.

[18] Y. Xiong and K. Pulli, "Gradient domain image blending and implementation on mobile devices," in *MobiCase '09: Proceedings of The First Annual International Conference on Mobile Computing, Applications, and Services*. Springer Berlin / Heidelberg, 2009.

[19] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 65–81, 2004.