7.2-4

in *Consumer Electronics (ICCE), 2010 Digest of Technical Papers International Conference on*, 9-13 2010, pp. 319 –320.

# Fast Panorama Stitching on Mobile Devices

Yingen Xiong and Kari Pulli, *Member, IEEE*

Fig. 1 Panoramic images created by the fast panorama stitching (top) and graph cut optimization (middle) with 17 1024×768 source images. The previous is 92 times faster than the latter.

***Abstract*—We propose a fast image stitching approach which can combine a set of aligned images into a composite with low memory consumption and implement it in a mobile panorama system for creating high-resolution and high-quality panoramic images on mobile devices. It applies color correction to smooth color transition, dynamic programming to find optimal seams in overlapping areas between adjacent images, and simple linear blending to hide the seams after stitching. A sequential procedure is created with the fast stitching approach for constructing panoramic images. The approach presents two main advantages. The use of dynamic programming for optimal seam finding guarantees fast processing speed. The sequential panorama stitching allows saving memory resources. The approach has been tested with different image sequences and applied to different scenes. Good performance has been obtained.**

## I. INTRODUCTION

We are building high-resolution and high-quality panoramic images on mobile phones. However, construction of panoramic images needs a lot of computation. Mobile phones have limited computational resources. We need to develop proper stitching approaches to fit mobile applications.

There are two kinds of image stitching methods: transition smoothing and optimal seam finding. Transition smoothing approaches [1] reduce color differences between source images for hiding seams. Optimal seam finding approaches [2] search for optimal seams in overlapping areas. Labels for pixels of the composite can be created according to the optimal seams. The composite is produced by copying corresponding pixels from source images using labeling. In our work, a fast stitching approach is created and integrated into a sequential procedure. It can be used for creating high-resolution and high-quality panoramas on mobile devices with fast speed and low memory consumption.

## II. FAST PANORAMA STITCHING

Our goal is to create a fast and low memory cost approach for image stitching and integrate it into a sequential procedure for creating high-resolution panoramic images.

Differences in color and luminance of source images may cause artificial edges in result panoramic images. We perform color correction in the gamma-corrected RGB color space for each source image before stitching it to the panoramic image to reduce stitching artifacts. Color correction coefficients for each source image $S_i$ $(i = 1, 2, \ldots, n)$ can be computed by

$$\alpha_{c,i} = \frac{\sum_{p} (P_{c,i-1}(p))^{\gamma}}{\sum_{p} (P_{c,i}(p))^{\gamma}} \qquad c \in \{R, G, B\}\ (i = 2,3,\cdots,n) \cdot \tag{1}$$

Where, $P_{c,i}(p)$ is the color value of pixel $p$ in image $S_i$ in the overlapping area between $S_i$ and $S_{i-1}$ in color channel $c$ and $\gamma$ is a gamma coefficient. With the color correction coefficients, we can perform color correction for source image $S_i$ by

$$P_{c,i}(p) \leftarrow \alpha_{c,i}^{\frac{1}{\gamma}} P_{c,i}(p) \quad (i = 1,2,\cdots,n). \tag{2}$$

Where, $\alpha_{c,1} = 1$. In practical applications, we perform global adjustment for the coefficients for reducing cumulative errors. Besides, we search for an image with best color distribution in the image sequence and use it to adjust color for the first image before color correction for other images.

In order to stitch source image $S_i$ to the panoramic image $I_c$, we need to find an optimal seam in the overlapping area and create labeling for all new pixels of $I_c$.

We want to merge two images on pixels where they match best. Suppose $I_c^o$ and $S_i^o$ are the overlapping images of the current panoramic image $I_c$ and the current source image $S_i$ respectively. We compute a squared difference $e$ between $I_c^o$ and $S_i^o$ as an error surface,

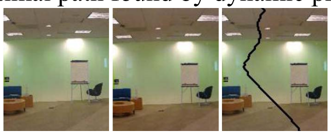
$$e = (I_c^o - S_i^o)^2, \tag{3}$$

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

$$E(h,w) = e(h,w) + \min(E(h-1,w-1), E(h-1,w), E(h-1,w+1)). \tag{4}$$

978-1-4244-4316-1/10/$25.00 ©2010 IEEE

The optimal path can be obtained by tracing back 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 $E(h_o\text{-}1, w) = E(h_o\, w_o) - \mathrm{d}(h_o, w_o)$, $w \in \{w_o\text{-}1, w_o, w_o\text{+}1\}$, the position of the optimal path in this row is ($h_o$-1,$w$). Repeat the process until all rows are traced. Fig. 2 shows an example of finding the optimal path. Fig. 2 (a) and (b) are two overlapping images and (c) is the optimal path found by dynamic programming.



(a) (b) (c)

Fig. 2 Find an optimal path in overlapping images with dynamic programming.

We use the optimal path as an optimal seam to stitch image $S_i$ to $I_c$. In order to further smooth color transition across the seam, we perform image blending in the area along the seam with a width $\delta$ in each side. The new pixel value can be calculated by weighted combination of the corresponding pixels,

$$P_{I_{c,new}}(p) = \frac{d_1^n P_{I_c}(p) + d_2^n P_{S_i}(p)}{d_1^n + d_2^n}\ . \qquad (5)$$

Where $d_1$ and $d_2$ are distances between current pixel $p$ to the boundaries. Different $n$ results in different color transition.

Fig. 3. Panoramic images produced by the fast stitching and graph cut approaches with 4 source images in an outdoor scene with moving objects

## III. APPLICATIONS AND RESULT ANALYSIS

The fast panorama stitching approach is implemented in our mobile panorama system for producing high-resolution panoramic images on mobile devices. It has been tested and applied in different scenes. We present some example applications and results which are obtained on Nokia N95 8GB mobile phones with an ARM 11 332 MHz processor and 128 MB RAM. The size of source images is 1024×768. In order to evaluate performance, we compare the results obtained from the fast stitching approach and a commonly used graph cut approach [2, 3].

Fig. 1 shows an application for an outdoor scene. In this application, a $360^o$ panorama is created with 17 source images shown at the bottom. The top and the middle are the results of the fast stitching and graph cut approaches respectively. The previous is about 92 times faster than the letter. According to our tests, the longer the image sequences are, the more advantages the fast approach expresses.

Fig. 3 shows an application for an outdoor scene with moving objects. The top and the middle of the figure are the results of the fast stitching and graph cut approaches. During panorama stitching, the fast approach takes 5.92 seconds and the graph cut approach takes 165.92 seconds. The previous is about 28 times faster than the latter. Both approaches can create proper seams to avoid ghosting artifacts.

Fig. 4 shows an application for an indoor scene with 7 source images. The top and middle shows the results produced by the fast stitching and graph cut approaches respectively. In panorama stitching, the fast approach takes 17 seconds and the graph cut approach takes 595 seconds. The previous is about 35 times faster than the latter.

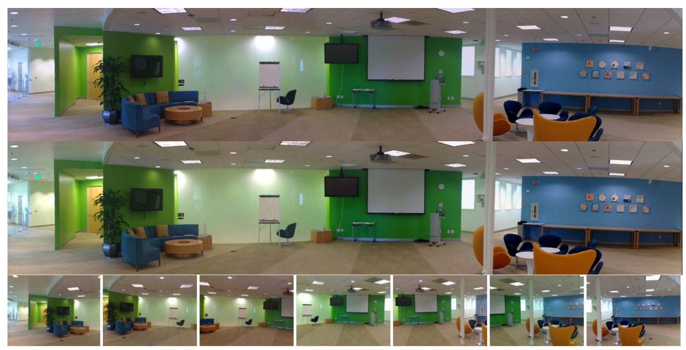


Fig. 4. Panoramic images produced by the fast stitching and graph cut approaches with 7 source images in an indoor scene

## IV. CONCLUSIONS AND DISCUSSION

We presented a fast panorama stitching approach and implemented it in a mobile panorama system for creating high-resolution and high-quality panoramic images on mobile devices. By comparing with the commonly used graph cut approach, the fast stitching approach has properties of fast processing speed and low memory consumption, which is very important for implementation on mobile devices. It has been tested with different image sequences and applied to different scenes. Good performance has been obtained.

## REFERENCES


[1] P. P´erez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. Graph.*, 22(3):313–318, 2003.

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

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