Panorama Mosaic Optimization for Mobile Camera Systems

Seong Jong Ha, Hyung Il Koo, Sang Hwa Lee, Nam Ik Cho, Soo Kyun Kim, *Member,* IEEE

Abstract — *In order to enable the real-time image mosaic in mobile devices that have very limited computing power and memory, we propose an efficient image mosaic algorithm with integer arithmetic. The proposed algorithm is focused not only on the computational efficiency, but also on the performance improvement of image mosaic. For the efficient projection without transform estimation, we obtain successive images through the semitransparent view finder which shows the fixed amount of right part of previously captured image. We project each images onto cylindrical surface by the proposed projection equation and integer arithmetic. We align the projected images by hierarchical hexagon search and a matching measure that is robust to illumination changes caused by different exposures and vignette distortion. Finally, the images are naturally blended using color compensation and dynamic programming based stitching. The proposed mosaic algorithm is embedded and tested in mobile phone systems. According to various tests and experiments, the proposed algorithm shows good panorama composition in real-time compared with the other PC-based methods*[1].

Index Terms — Embedded panorama mosaic, color compensation, integer arithmetic, stitching

I. INTRODUCTION

When we see splendid scenes, such as sea horizons, mountains, and prospects from the top, we wish to capture them in a single photograph. However, the usual cameras have relatively narrow view angles to capture the scenes at once. We have no choice but to take a picture as small as the scenes are included in a camera view, which diminishes the objects in size to be indistinguishable and results in reduced image resolution. Panoramic image mosaic is a solution to capture the scenes in a single image of high resolution [1]. It composes wide view images from multiple images which include the same scenes partially at the slightly different viewpoints. The image mosaic needs many image processing topics, such as feature extraction, transform estimation between images, image warping onto mosaic surfaces, local alignment of warped images, boundary stitching, blending, and so on.

Many algorithms on image mosaic have been developed so far. The most important and classical works are related with feature extraction and transform estimation between images. Szeliski considered spatial transform between images as an 8-parameter perspective motion model, and used Levenberg-Marquardt algorithm to find image correspondences [4]. Some researchers thought spatial transform as a 3-parameter rotational motion model. Szeliski and Brown estimated the spatial relation between images by updating rotational matrices [8], [9]. Lowe used scale invariant feature transform (SIFT) features to estimate the transformation between images [10], [14]. After obtaining the motion model parameters and transformations between images, each image is transformed and projected onto the common mosaic surface. Most of the image mosaic algorithms select cylindrical or spherical surfaces as the mosaic surface, and project images onto the surface by the estimated transformations and motion models. Then, the projected images are composed into a single image seamlessly. To synthesize multiple images into a single large one, Freeman stitched the images using dynamic programming on the mosaic surface [7]. Feathering [6], multi-band blending [3], [10], and image gradients [11], [12] were used to make the boundaries between the projected images smooth. Some approaches to get rid of exposure difference and radiometric distortion were also proposed. Uyttendaele compensated the exposure differences using block-based exposure adjustment [6]. Goldman eliminated vignette aberration by introducing anti-vignette functions [13].

The mosaic algorithms presented above show good performances in the usual scenes. However, since all of the works on image mosaics are developed and implemented on general personal computer (PC) environments, we have to transmit the captured images to a PC for generating mosaic images.

This paper proposes and optimizes an embedded panorama mosaic algorithm for mobile camera systems like digital cameras and mobile phone cameras. The PC-based mosaic algorithms described before do not work well on the mobile camera systems because of the limitations of computational power and memory of hand-held devices. The proposed mosaic algorithm operates on the usual mobile camera systems in real-time. We can verify the mosaic results right after all the images are acquired by the mobile imaging systems.

In order to enable the real-time panoramic mosaic in mobile camera systems, we propose an efficient image mosaic algorithm with integer arithmetic and programming optimization. The proposed algorithm is focused not only on the computational efficiency, but also on the performance improvement of each step needed for image mosaic. We consider spatial transform between images as 3-parameter

[1] Seong Jong Ha, Hyung Il Koo, Sang Hwa Lee, and Nam Ik Cho are with School of Electrical Engineering and Computer Science, Seoul National University, Kwanak-gu, Seoul, 151-742, South Korea (emails: oanchovy@ispl.snu.ac.kr, hikoo@ispl.snu.ac.kr, lsh@ipl.snu.ac.kr, nicho@snu.ac.kr).

Soo Kyun Kim is with Samsung Electronics Co., Ltd., South Korea (email: sookyun.kim@samsung.com) .

Sang Hwa Lee is the corresponding author (e-mail: lsh@ipl.snu.ac.kr).

Contributed Paper
Manuscript received August 31, 2007

rotational model, and use semitransparent panoramic viewer to capture the successive images. Since the images acquired through the panoramic view finder have a predetermined rotational transformation, we don't have to perform the estimation of correspondences and transforms which need much computational load. For the fast projection of images onto the mosaic surface, we do not use the conventional transforms which contain trigonometric functions, but modify the transformations into integer functions. The projected images onto the mosaic surface are well aligned by a robust error metric and fast hexagon-based search in the integral image space. The exposure differences and color distortions are also compensated in synthesizing projected images. Finally, we stitch the images by using dynamic programming and blend the image boundaries linearly. The proposed technique optimizes the mathematical operations for image mosaic with integer arithmetic, and is operated real-time in portable camera systems without loss of performance. The users can check the mosaic results as soon as images are acquired using the proposed mosaic system.

The rest of paper is organized as follows. We describe the camera motion model and panoramic view finder in Section 2. Section 3 explains the proposed image mosaic algorithms and optimization methods in detail. Experimental results with mobile phones are shown in Section 4. Finally, we conclude the paper in Section 5.

II. MOTION MODEL OF CAMERA

For fast and stable mosaic results, we have to simplify some procedures in the mosaic algorithm. Especially, since feature points extraction and transform estimation require much nonlinear floating point computation and are unstable in some cases, it is desirable to omit the processes under reasonable constraints. We assume that the users move their cameras to capture overlapped scenes only in one direction. In other words, the mosaic system generates the horizontal (or vertical) panoramic views. The camera motion and transformations between successive images can be modeled as a specific form under the assumption. Then, the camera motion is described as horizontal (or vertical) rotation with fixed angle θ and focal length f. The following subsections relate the camera motion with transform function in the image coordinates.

A. Camera Motion and Panoramic View Finder

The successive images for image mosaic should be captured with some overlapped regions. The correspondences between overlapped regions are exploited to estimate transformations of images and to align the warped images. When there is no overlapped region or correspondences between images, the mosaic is not performed. To get the constant overlapped area between captured images in the case of horizontal panorama composition, the camera must be rotated by a fixed angle along the y-axis. We set the rotation angle along the y-axis using image dimension.

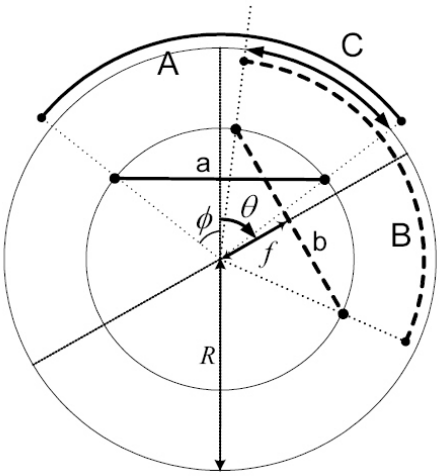


Fig. 1. Top view of geometry of camera motion. Two successive images are captured by rotating a camera along the y-axis. When we fix the amount of overlapped area between two images, the rotation angle θ is determined by the focal length of camera and image width.

Let a and b in Fig. 1 be the first and second images respectively. The rotation angle of camera is θ and the focal length is f. Since the camera rotates only in the horizontal direction, the cylindrical surface is the most suitable one for the projection and composition of panoramic images. The images are projected onto the cylindrical surface that has a radius of R, and the projected images, A and B are generated. The arc C in Fig. 1 is the overlapped part of two projected images. Let ϕ be the half of view angle. Then, it is determined by camera parameters, W and f,

$$\phi = \arctan(\frac{W/2}{f}),$$

where W is the width of image. Let τ be the ratio of overlapped area in the projected images. Since the angle of C becomes $2\phi - \theta$ and the arc length of C is $R(2\phi - \theta)$, τ is derived as

$$\tau = \frac{2R\phi - R\theta}{2R\phi} = 1 - \frac{\theta}{2\phi}. \quad (1)$$

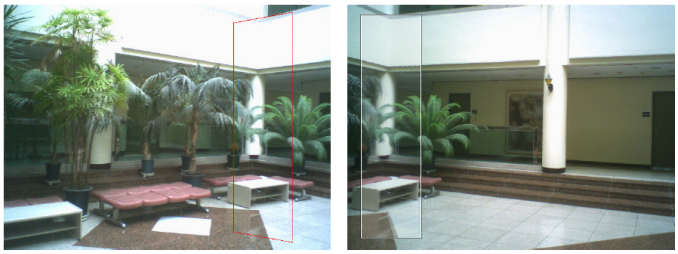
When the ratio τ is set, the angle of camera rotation θ is determined as

$$\theta = 2\phi(1-\tau), \quad (2)$$

which is a main parameter of perspective projection matrices between successive images. Thus, we can obtain transformations without feature correspondences when we know the rotation angle θ. We make a panoramic view

finder to acquire the successive images of horizontal rotation with fixed overlapped area. As shown in Fig. 2, we warp a part of previously acquired image onto LCD to guide users to take pictures with horizontal rotation of fixed angle. Users acquire the next image when the real image through camera view finder is almost matched to the warped image part. The warping is performed by the specific transformation matrix described in the next subsection. By the panoramic view finder, we can omit the process of feature extraction, estimation of correspondences and transformations, which need large amount of computation.

It should be noted that the part of previously captured image on LCD is the projected (rotated) image along the camera motion, not the original part as it was. Since the same scene becomes warped and transformed with respect to the different viewpoints, we should consider the current and previous images at the same viewpoint when we match them. Thus, the previous image part is transformed with respect to the camera motion so that the viewpoints of images are equivalent.



(a) (b)

Fig. 2. Panoramic view finder. (a) part of previously acquired image to be warped (red quadrangle region), (b) next image to be taken using panoramic view finder. The white rectangle region is compared with the warped red quadrangle region in (a).

B. Transforms between Images

We assume that the focal length f of camera is fixed for all of the images. If the focal length varies for each image, we have to estimate correspondences and transforms of images. This is the usual case in the PC-based mosaic algorithms which do not work well in the mobile camera systems. The transformation matrix between two images (i^{th} and j^{th} images) by the camera motion [9] is defined as

$$\mathbf{H}_{ij} = \mathbf{K}_i \mathbf{R}_i \mathbf{R}_j^{-1} \mathbf{K}_j^{-1}, \tag{3}$$

where the intrinsic camera matrix $\mathbf{K}_i$ for the i^{th} image is defined as

$$\mathbf{K}_i = \begin{bmatrix} f_i & 0 & 0 \\ 0 & f_i & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The rotation matrix is presented in the exponential form,

$$\mathbf{R}_i = e^{\mathbf{\Theta}_i}, \quad \mathbf{\Theta}_i = \begin{bmatrix} 0 & -\theta_{i3} & \theta_{i2} \\ \theta_{i3} & 0 & -\theta_{i1} \\ -\theta_{i2} & \theta_{i1} & 0 \end{bmatrix},$$

where θ_{i1}, θ_{i2}, θ_{i3} are three rotation angles. By the assumption that camera rotates only in the y-axis and the focal length is fixed, we set $f_i = f$ in $\mathbf{K}_i$ and $\theta_{i1} = \theta_{i3} = 0$ in $\mathbf{R}_i$. Because the angle between successive images is set to be θ in (2), the transformation matrix (3) between successive images is simplified as

$$\mathbf{H}_{ij} = \begin{bmatrix} \cos\theta & 0 & f\sin\theta \\ 0 & 1 & 0 \\ -f^{-1}\sin\theta & 0 & \cos\theta \end{bmatrix}. \tag{4}$$

III. Procedure of Generating Mosaic Images

A. Projection onto Mosaic Surface

The acquired images are transformed using (4) before projecting the images directly onto the cylindrical surface. To project the transformed images onto the cylindrical surface, we convert 3 dimensional (3D) real coordinates into 2D image coordinates through the pinhole camera model

$$(x, y) = (f\frac{X}{Z}, f\frac{Y}{Z}), \tag{5}$$

where (X, Y, Z) and (x, y) are points in 3D world coordinates and 2D image coordinates, respectively. Then, we project the points in the image coordinates onto the cylindrical surface. Let (u, v) be a point on the cylindrical surface with radius R as shown in Fig. 3. The coordinate (u, v) is equivalent to $(R\varphi, h)$ in the cylindrical coordinates,

$$(u, v) = (R\varphi, h),$$

where φ is the clockwise angle from the center axis and h is a height from the origin. The world coordinates (X, Y, Z) is represented as

$$(X, Y, Z) = (R\sin\varphi, h, R\cos\varphi). \tag{6}$$

From (5) and (6), the transformation of (u, v) into (x, y) in 2D image coordinates becomes

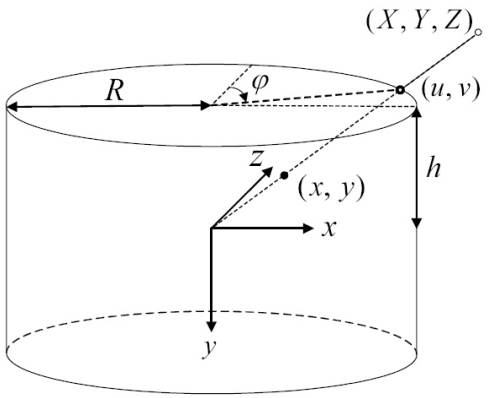


Fig. 3. Cylindrical mosaic surface and its 3D coordinate system. Each image captured by a camera is projected onto the cylindrical surface. The real world coordinate is expressed as (X,Y,Z), image coordinate as (x,y), and cylindrical one as (u,v).

$$(x,y)=\left(f\frac{R\sin\varphi}{R\cos\varphi},f\frac{h}{R\cos\varphi}\right)=\left(f\tan(u/R),f\frac{v}{R\cos(u/R)}\right). \quad (7)$$

Equation (7) is the conventional way to project an image onto the cylindrical surface. However, it is difficult to perform the projection (7) with integer operations due to the trigonometric functions. We modify the projection equation (7) for integer arithmetic as

$$\begin{aligned}(x,y)&=\left(f\tan(u/R),f\frac{v}{R\cos(u/R)}\right)\\&=\left(f\frac{R\sin(u/R)}{\sqrt{R^2-R^2\sin^2(u/R)}},f\frac{v}{\sqrt{R^2-R^2\sin^2(u/R)}}\right)\\&=\left(f\frac{u}{\sqrt{R^2-u^2}},f\frac{v}{\sqrt{R^2-u^2}}\right),\end{aligned} \quad (8)$$

where $\sin(u/R)$ is approximated to u/R. We observe that $u \ll R$ is usually satisfied in our mosaic system (640x480 images and $R \gg 320$). Based on the observation of our mosaic system, we modify the usual projection equation (7) into (8) for fast operation. The proposed equation eliminates the trigonometric functions which consume much computational time in the mobile devices. The square root function $\sqrt{R^2-u^2}$ in (8) can be calculated by the integer programming using fast Bresenham algorithm, which is developed for drawing circles [2]. In the experiments, we will show that the proposed projection equation (8) has little geometric distortion, while requiring very little computational complexity than the conventional equation (7).

We can also save much memory and computations by using the symmetry of image coordinates. As shown in Fig. 4, the center of image is considered as the origin, and a quarter of input image is projected using (8) and integer arithmetic. Then, the rest 3 quarters of image are projected using the spatial symmetry.

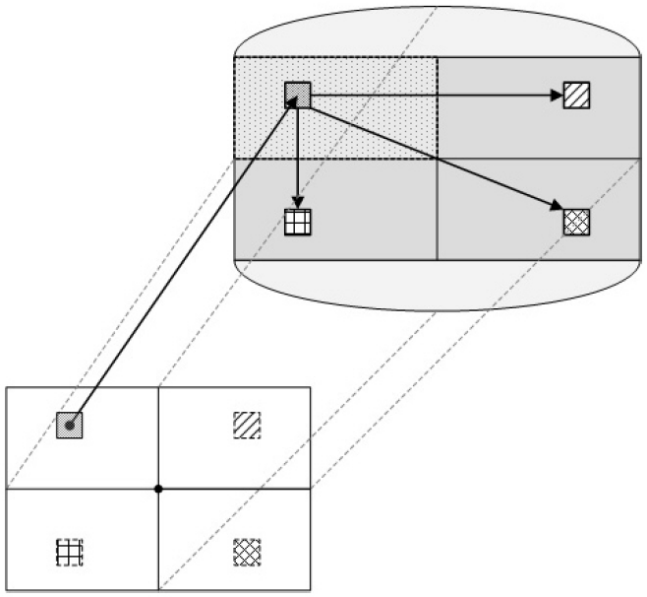


Fig. 4. Symmetry of image coordinates when projecting a rectangle onto the cylindrical surface. We can save the computations and memories using the coordinates symmetry. The left upper quarter of image is first transformed using (8), and the other quarters are transformed by the spatial symmetry.

B. Images Alignment

Though the successive images are acquired by the panoramic view finder, the users are prone to make errors in rotational motion and fitting the objects into the view finder. The errors cause the discrepancy between real camera motion and the fixed model of our assumption, which results in misalignment of successive images on the cylindrical mosaic surface. Hence, we need to align the projected images locally on the mosaic surface. The local alignment is performed by block-based motion search in the overlapped regions. We introduce a robust error measure for the better image alignment, and use hexagon-based search and an integral image [15] for fast operation.

The integral image at $\mathbf{x}=(x,y)$ represents the sum of all the pixel values in a rectangular region such that $I_{\text{int}}(\mathbf{x})=\sum_{i\le x,j\le y} I(i,j)$ (see Fig. 5 (a)). Once the integral image is generated for every pixel in the image, the sum of pixels in a shaded block in Fig. 5 (b) can be calculated with four additions, $I_{\text{int}}(\mathbf{x}_4)-I_{\text{int}}(\mathbf{x}_2)-I_{\text{int}}(\mathbf{x}_3)+I_{\text{int}}(\mathbf{x}_1)$. The integral image calculates block-based error metric much faster.

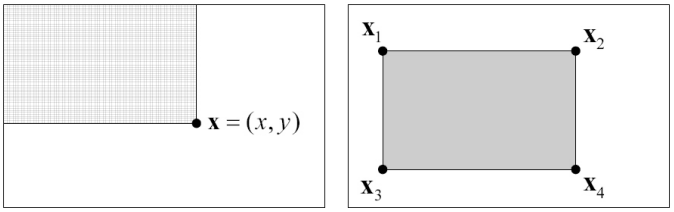


Fig. 5. Integral image. (a) Definition, (b) sum of pixels in the shaded region by using integral image.

The proposed error metric considers the illumination difference due to exposure changes and vignette. It exploits illumination means of overlapped image regions,

$$E(\mathbf{u}) = \sum_{\mathbf{x}_i \in \mathbf{R}^{ov}} [(I_1^{ov}(\mathbf{x}_i + \mathbf{u}) - \bar{I}_1^{ov}(\mathbf{u})) - (I_0^{ov}(\mathbf{x}_i) - \bar{I}_0^{ov})]^2, \quad (9)$$

where I_0^{ov} and I_1^{ov} are the overlapped image area in the successive two images I_0 and I_1. $I_0^{ov}(\mathbf{x}_i)$ is a luminance value at $\mathbf{x}_i$, and $\bar{I}_0^{ov}$ and $\bar{I}_1^{ov}$ are luminance means in the overlapped area. Note that I_0^{ov} is a part of I_0, and moves in I_1. Thus, the mean $\bar{I}_0^{ov}$ is constant whereas $\bar{I}_1^{ov}$ varies with the motion vector $\mathbf{u}$. We search for the optimal locations of projected images such that $\mathbf{u}$ minimizes the error metric,

$$\arg\min_{\mathbf{u}} E(\mathbf{u}). \quad (10)$$

$\bar{I}_1^{ov}(\mathbf{u})$ must be calculated whenever $\mathbf{u}$ varies, which requires too much computation. We use the integral image to reduce the computational complexity in motion search. And we utilize hexagon-based search for faster operation [5]. The integral image and hexagonal search enable the local alignment process faster with integer arithmetic.

C. Color and Luminance Compensation

If the exposure is fixed when taking pictures of the same scene, the overlapped areas of images have the same luminance and colors. However, since the exposure usually changes at the different viewpoints, there are brightness and color difference between images. For color and luminance compensation, we assume that the objects in the scene have Lambertian surfaces [18]-[20]. The radiance reflection from the object surface decides pixel color irrespective of camera viewpoints as

$$P_C = f(L_C). \quad (11)$$

We model the pixel color to be proportional to the light radiance as [17], [18]

$$P_C = \alpha \times L_C, \quad (12)$$

where L_C is the radiance of a color channel C (R, G, and B), and α is a proportional coefficient. Given an overlapped area of two images, the pixel color of $\mathbf{x}$ can be described as

$$P_{0,C}(\mathbf{x}) = \alpha_0 \times L_C, \quad P_{1,C}(\mathbf{x}) = \alpha_1 \times L_C, \quad (13)$$

where $P_{i,C}(\mathbf{x})$ is the pixel color at $\mathbf{x}$ in the overlapped area of the i^{th} image. Hence, we can assume that two pixel values at the same position have a linear relation. Specifically, due to noise and erroneous local alignment, we use the mean in the overlapped area for the linear relation,

$$\bar{P}_{1,C} \; ; \; \frac{\alpha_1}{\alpha_0} \times \bar{P}_{0,C}, \quad (14)$$

where $\bar{P}_{i,C}$ is the mean value of the i^{th} overlapped image for channel C.

The linear model is defined in the RGB color space whereas the color space of portable cameras is usually the YCbCr. In the case of Y component, it is the linear combination of R, G, and B components, and (14) is still satisfied, i.e.,

$$\bar{P}_{1,Y} \; ; \; \frac{\alpha_1}{\alpha_0} \times \bar{P}_{0,Y}. \quad (15)$$

In the case of color components, Cb and Cr, they are not the linear combinations of RGB, but their AC components are linear combination of RGB. We modify (14) and propose a color compensation technique in the YCbCR color space using luminance means as follows:

1) Calculate $\bar{P}_{0,Y}$ and $\bar{P}_{1,Y}$,

$$\frac{\alpha_1}{\alpha_0} (= \frac{\bar{P}_{1,Y}}{\bar{P}_{0,Y}}). \quad (16)$$

2) If the 1^{st} image is the reference and the 0^{th} image is compensated, compensate the luminance values in the 0^{th} image,

$$P'_{0,Y}(\mathbf{x}) = \frac{\alpha_1}{\alpha_0} \times P_{0,Y}(\mathbf{x}). \quad (17)$$

3) Compensate two chrominance components,

$$P'_{0,C'}(\mathbf{x}) = \frac{\alpha_1}{\alpha_0} \times (P_{0,C'}(\mathbf{x}) - 128) + 128, \; C' \in \{Cb, Cr\}. \quad (18)$$

The color compensation process is performed in the whole image, not only in the overlapped region. Thus, the luminance and colors of images are globally regularized, which reduces abrupt changes between images and makes the image boundaries unseen.

D. Stitching Using Dynamic Programming

After aligning the projected images locally, we synthesize the overlapped areas to make a mosaic image. If the pixels in the overlapped areas are synthesized by weighted sums as in the conventional panorama methods, there are some blurring and boundary artifacts as shown in Fig. 6 (a). This is caused by the misalignment in the subpixel accuracy in spite of the local alignment process. In order to alleviate the blurring and artifacts, we find a plausible boundary between images in the overlapped region by using dynamic programming method.

The resulting stitched image is shown in Fig. 6 (b). Dynamic programming consists of 3 steps, initialization, matrix filling (scoring), and trace back (alignment) as follows:

1) Initialization

Initialize the cost matrix $C(x,y)$ for $\forall(x,y)$ as

$$C(x,y)=\left|\tilde{I}_0^{ov}(x,y)-\tilde{I}_1^{ov}(x,y)\right|, \qquad (19)$$

which stores the difference of two color-compensated images in the overlapped region.

2) Matrix filling (Scoring)

Make a matrix with the same size of the cost matrix and fill it with the minimum cumulative difference along several paths from top to bottom.

- $y=0:\ C'(x,y)=C(x,y)$
- $y=1,2,3,\cdots:$

$$C'(x,y)=\min[C'(x-1,y-1),C'(x,y-1),C'(x+1,y-1)]+C(x,y).$$

3) Trace back (Alignment)

The optimal boundary is found by tracing back the path with minimum cost from bottom to top.

- Last row

$$(x_{\text{opt}},y_{\text{opt}})=\arg\min_{x,y}C'(x,y).$$

- The next upper row

If $C'(x,y_{\text{opt}}-1)=C'(x_{\text{opt}},y_{\text{opt}})-C(x_{\text{opt}},y_{\text{opt}}),$

where $x\in\{x_{\text{opt}}-1,x_{\text{opt}},x_{\text{opt}}+1\}$,

then, $(x_{\text{opt}},y_{\text{opt}})=(x,y_{\text{opt}}-1).$

- Repeat the next upper row until the first row

(a) (b)

Fig. 6. Comparison of overlapped region. (a) weighted average of two overlapped images, (b) separating two overlapped images by image stitching. We see clearer and sharper ridges of mountain in (b).

E. Linear Blending

The last step of panorama mosaic is to blend stitched boundaries. Because the color compensation and stitching reduce blurring and boundary artifacts enough, we use just linear blending. 16 pixels of each side from the stitched boundary are used for linear blending.

TABLE I
TEST DEVICE SPECIFICATION

CPU	Intel PXA272 520MHz
OS	Windows Mobile 2003 SE Edition
MEMORY	128MB ROM, 64MB RAM
DISPLAY	2.8" 65K TFT LCD
CAMERA	1.3 megapixel CMOS

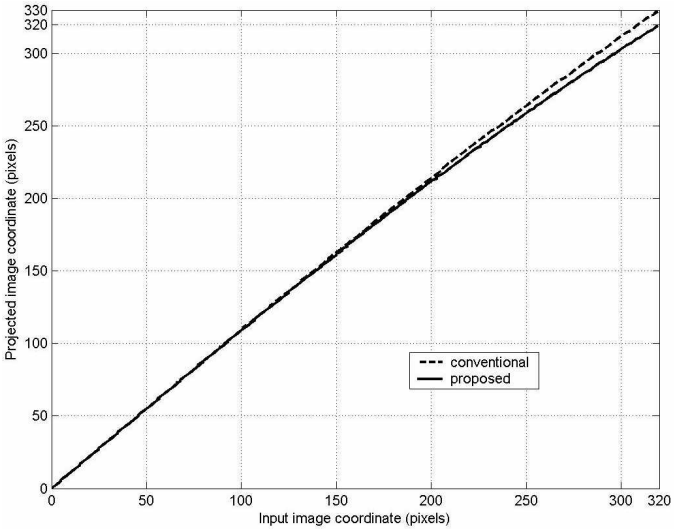


Fig. 7. Comparison of projection errors. (dash curve) conventional equation, (solid curve) proposed projection equation.

IV. EXPERIMENTAL RESULTS

The proposed image mosaic algorithm is applied to a PDA phone with the specification in Table I. The image size is 640×480, and the focal length of test images is 690 pixels. The radius of cylindrical mosaic surface is 760 pixels, which is related with the focal length. First, we evaluate the proposed projection equation (8) which is the approximation of exact one (7). The projection error is shown in Fig. 7, where dashed and solid lines represent the projections by (7) and (8) respectively. The x-axis in Fig. 7 represents x coordinates of the captured original image, and the y-axis indicates the corresponding location in the projected image. Fig. 7 shows that the proposed projection equation causes errors by 10 pixels at maximum when 640×480 image is projected. The proposed method approximates the usual conventional formula based on the assumption of $R\square u$. Thus, as u increases, the projection error also increases. However, this distortion is not visible in the projected images. Fig. 8 shows the images projected by the conventional one and the proposed one with a test image. As shown in Fig. 8, the proposed method projects the image without noticeable distortions compared with (7). Because the error is not noticeable in the VGA images, there is no problem for the smaller images. Therefore, the proposed equation (8) is a reasonable approximation for fast projection. The proposed projection

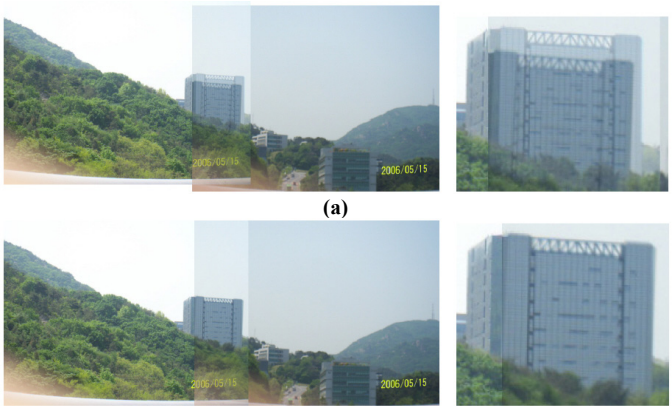
takes 0.039 seconds whereas the conventional one requires 2.696 seconds on the same test device.

(a) (b) (c)

Fig. 8. Projected images. (a) Original image (640×480), (b) conventional projection, (c) proposed projection.

(a) (b)

Fig. 9. Test images for the comparison of performances of error metrics.



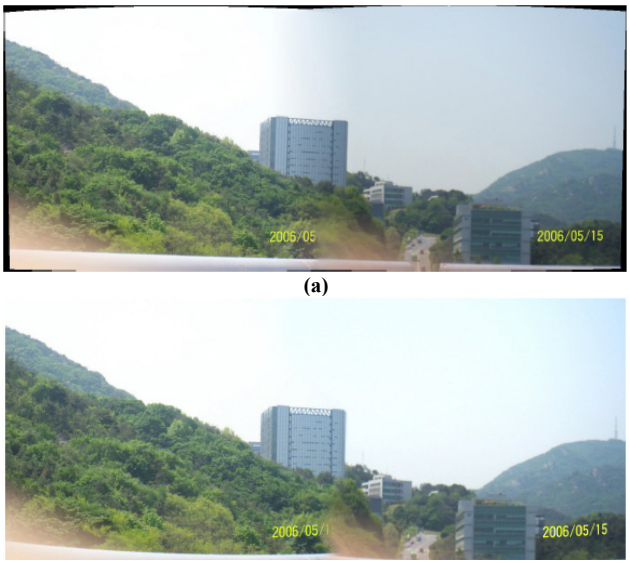
(a)

(b)

Fig. 10. Local alignment results and their zooming. (a) SAD, (b) the proposed metric.

Second, we evaluate the proposed error metric (9). Fig. 9 shows test images with prominent exposure difference, and Fig. 10 shows the results of local alignment by the error metrics. As shown in Fig. 10, if images have exposure and brightness differences, they are not well aligned by SAD or SSD metric, whereas the proposed error measurement aligns the images better by the luminance compensation.

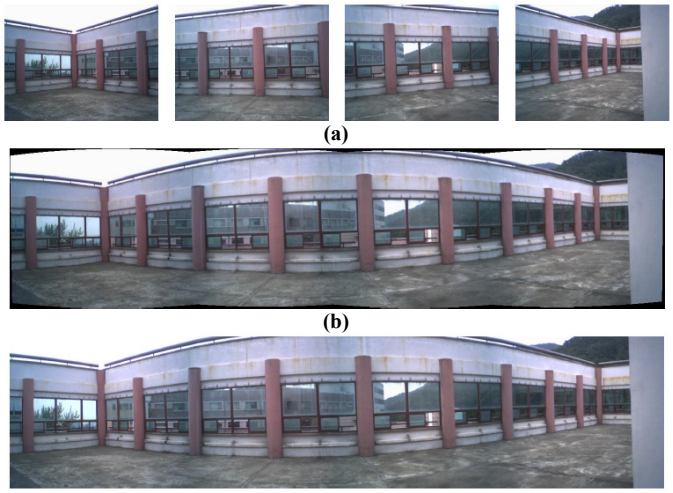
Third, we evaluate the proposed color compensation with test images in Fig. 9. Fig. 11 compares the proposed method with multi-band blending [16]. Fig. 11 (a) shows natural connection in the overlapped areas, but still some exposure difference in the non-overlap areas. Fig. 11 (b) shows natural connection in the overlapped regions with less exposure and color differences.



(a)

(b)

Fig. 11. Evaluation of color compensation. (a) multi-band blending [16], (b) the proposed color compensation followed by linear blending.



(a)

(b)

(c)

Fig. 12. Comparison of mosaic image results by (a) autostitch [16] and (b) proposed mosaic system.

Finally, the mosaic performance of proposed algorithm is compared with that of autostitch program which is a PC-based mosaic algorithm using SIFT features and multi-band blending [16]. Fig. 12 and Fig. 13 show the mosaic results for outdoor and distant indoor scenes. According to various tests, the proposed system shows good performance both for indoor and outdoor scenes. The proposed mosaic system takes 1.2 seconds for four 640×480 images within 4M bytes memory, whereas the autostitch algorithm at 3GHz Pentium PC takes 5-6 seconds with much more memory. Furthermore, the proposed mosaic algorithm is robust for such simple scenes that there are little feature correspondences. Fig. 13 shows the robustness of proposed mosaic system, where the autostitch program does not work well because of no SIFT feature matching. Fig. 14 shows some mosaic results by the proposed system.

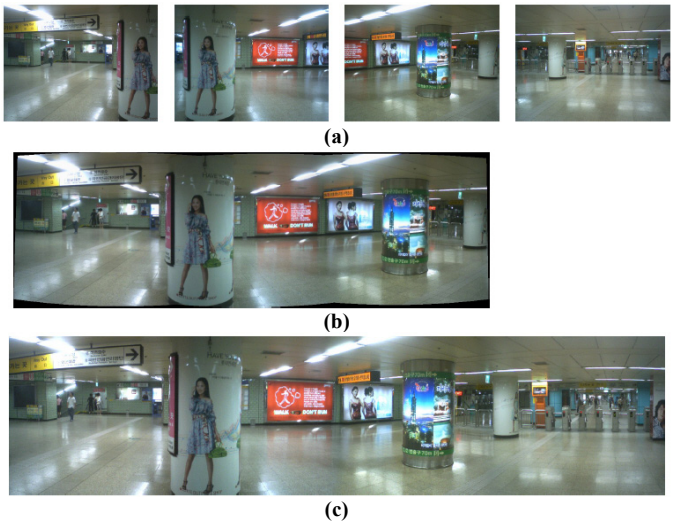


Fig. 13. Comparison of mosaic images. (a) Four original images, (b) autostitch [16], (c) proposed mosaic system. The fourth image in (a) is not stitched by autostitch [16] because of no SIFT feature matching.

Fig. 14. Panorama results by the proposed mosaic system.

V. Conclusion

This paper has proposed embedded panorama mosaic algorithms for mobile camera systems. The algorithms for mosaic processes are optimized by integer programming so that they operate real-time in the usual potable camera systems. The proposed system consists of panoramic view finder, approximated projection onto cylindrical surface, hexagon-based image alignment using robust error measurement and integral image, color compensation, image stitching based on dynamic programming, and linear blending. The proposed mosaic system is embedded and tested in actual mobile phone systems. According to various tests and experiments, the proposed algorithm shows fast and good panorama composition compared with the other PC-based and embedded algorithms. The proposed mosaic system is expected to be embedded in the commercial mobile phones and camera systems.

References

[1] R. Szeliski, "Image alignment and stitching: A tutorial," Preliminary draft, Jan. 2005.

[2] J. E. Bresenham, "A linear algorithm for incremental digital display of circular arcs," *Commun. ACM*, vol. 20, pp. 100-106, Feb. 1977.

[3] Y. Xiong and K. Turkowski, "Registration, calibration and blending in creating high quality panoramas," *IEEE Workshop on Applications of Computer Vision*, pp. 69-74, Oct. 1998.

[4] R. Szeliski, "Video mosaics for virtual environments," *IEEE Computer Graphics and Applications*, pp. 22-30, Mar. 1996.

[5] C. Zhu, X. Lin, and L. Chau, "Hexagon-based search pattern for fast block motion estimation," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 12, no. 5, May 2002.

[6] M. Uyttendaele, A. Eden, and R. Szeliski, "Eliminating ghosting and exposure artifacts in image mosaics," *Proc. Computer Society Conf. CVPR*, vol. 2, pp. 509-516, Dec. 2001.

[7] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," *ACM SIGGRAPH*, pp. 341-346, Aug. 2001.

[8] H. Shum and R. Szeliski, "Systems and experiment paper: Construction of panoramic image mosaics with global and local alignment," *Int. Journal of Computer Vision*, vol. 36, pp. 101-130, Feb. 2000.

[9] R. Szeliski and H. Shum, "Creating full view panoramic image mosaics and environment maps," *ACM SIGGRAPH*, pp. 251-258, Aug. 1997.

[10] M. Brown and D. Lowe, "Recognising panoramas," *Int. Conf. Computer Vision*, vol. 2, pp. 1218-1225, Oct. 2003.

[11] A. Levin, A. Zomet, S. Peleg, and Y. Weiss, "Seamless image stitching in the gradient domain," *Eur. Conf. Computer Vision*, vol. IV, pp. 377-389, May 2004.

[12] A. Zomet, A. Levin, S. Peleg, and Y. Weiss, "Seamless image stitching by minimizing false edges," *IEEE Trans. Image Processing*, vol. 15, no. 4, Apr. 2006.

[13] D. B. Goldman and J. Chen, "Vignette and exposure calibration and compensation," *Int. Conf. Computer Vision*, vol. 1, pp. 899-906, Oct. 2005.

[14] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. Journal of Computer Vision*, vol. 60, pp. 91-110, Nov. 2004.

[15] P. Viola, M. Jones, "Rapid object detection using a boosted cascade of simple features," *IEEE Computer Society Conf. on CVPR*, vol. 1, pp. 511-518, Dec. 2001.

[16] M. Brown and D. Lowe, Autostitch home page, http://www.autostitch.net, 2005.

[17] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, Prentice Hall, 2003.

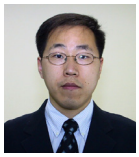
[18] Wikipedia: Lambertian reflectance, http://en.wikipedia.org/wiki/Lambertian_reflectance, Oct. 2006.

[19] S. Lin and S. W. Lee, "Estimation of diffuse and specular appearance," *Int. Conf. Computer Vision*, 1999.

[20] S. Shafer, "Using color to separate reflectance components," *Color Research and Application*, vol. 10, pp. 210-218, 1985.

Seong Jong Ha was born in Seoul, Korea, on December 1, 1978. He received the B.S. and M.S. degrees in electrical engineering from Seoul National University, Seoul, Korea, in 2005 and 2007, respectively. He is currently a researcher in Institute of New Media & Communication, Seoul, Korea. His research interests include panorama and super-resolution.

Hyung Il Koo was born in Daegu, Korea, on October 27, 1977. He received the B.S. and M.S. degrees in electrical engineering from Seoul National University, Seoul, Korea, in 2002 and 2004, respectively. He is currently working toward the Ph.D. degree in electrical engineering at Seoul National University. His research interests include computer vision, and speech enhancement.



Sang Hwa Lee received the B.S., M.S., and Ph.D. in electrical engineering from Seoul National University, Seoul, Korea, in 1994, 1996, and 2000, respectively. His research interests include image processing, video coding, computer vision, computer graphics, and pattern recognition.

Nam Ik Cho received the B.S., M.S., and Ph.D. degrees in control and instrumentation engineering from Seoul National University, Seoul, Korea, in 1986, 1988, and 1992, respectively. From 1994 to 1998, he was with the University of Seoul, Seoul, Korea, as an Assistant Professor of Electrical Engineering. He joined the School of Electrical Engineering, Seoul National University, in 1999, where he is currently a Professor. His research interests include speech, image, video signal processing, and adaptive filtering.

Soo Kyun Kim received Ph. D. in Computer Science & Engineering from Korea University, Seoul, Korea, in 2006. He joined Telecommunication Network Business at Samsung Electronics Co., Ltd. from 2006. His research interests include mobile graphics, geometric modeling, and interactive computer graphics.