Paper

A *Quasi*-feature based Image Mosaic Algorithm Using Modified Block Matching Criteria

Nonmember Dae-Hyun Kim, Yong-In Yoon, and Jong-Soo Choi
Dept. of Image Eng., Graduate School of Advanced Imaging Science, Multimedia, and Film, Chung-Ang Univ.
221 Huksuk-dong, Dongjak-ku, Seoul, 156-756, KOREA

Abstract

It is important to know the projective transformation between two images in order to construct the image mosaics with multiple images. The conventional methods to obtain the projective transform use the geometrical features or the intensities of the image. These methods execute the iterative computation, so that the total amount of computation to extract the correspondence and to induce the parameters of the transform is too much.

In this paper, we propose the projective registration algorithm using the *quasi*-feature points, which are based on the intensity. We define the *quasi*-feature point as a central pixel of a block that has enough texture information. To determine these feature points, we extract the overlap image between two images using the Fourier method and divide this overlap image of the reference into the four subareas. Then we choose the central pixel of the block which has the maximum local variance in each subarea. We use the block matching algorithm, which is considered of both the camera movement and the variation of the illumination conditions, in order to match these four feature points with the four points in the target image. The proposed algorithm is applied to various images to estimate the performance and the simulation results present that this method finds the correct correspondence.

Key words : mosaics, phase correlation, *quasi*-feature, projective transform, block matching algorithm

1. Introduction

The automatic construction of large, high resolution image mosaics is an active area of research in the fields of photogrammetry, computer vision, image processing, and computer graphics. Image mosaics can be used for many different applications. The most traditional application is the construction of large aerial and satellite photographs from collections of images. More recent applications include scene stabilization and change detection, video compression and video indexing, increasing the field of view and resolution of a camera, and even simple photo editing. A particularly popular application is the emulation of traditional film-based panoramic photography with digital panoramic mosaics, for applications such as the construction of virtual environments and virtual travel [1][3].

In order to build the image mosaics with multiple images, we need to know the projective transform between two consecutive images. The conventional methods to obtain the projective transform are largely divided into two. The one is to use the geometrical information of the image, for example, edge, corner, and line [2]. After extraction of the overlap image between the reference and target image using the Fourier method, we extract points of interest from the overlap image of the reference. Then we compute all possible projective transform defined by pairs of fourtuples of points of interest and keep the best one. Even though we need four points within the overlap image, usually we overestimate this number to insure to find four good matches. Generally we take six points, but the number of point is increased as the overlap size is reduced [2]. This method can obtain the precise projective transform, but the computational cost is very high. Therefore the more images are used, the higher cost is needed. The other is to directly minimize the discrepancy in intensities between pairs of images after applying the transformation we are recovering [1][3]. This method has the advantage of not requiring any easily identifiable feature points. To perform the minimization, we use the Levenberg-Marquardt iterative nonlinear minimization algorithm [1][3][11]. These iterative calculations increase the total amount of the computation and this is not appropriate for real time implementation.

In this paper, we propose the projective registration algorithm that extracts *quasi*-feature points, which are based on the image intensity, and extracts their correspondence using the block matching algorithm. The *quasi*-feature point is defined as a central pixel of a block that has enough texture information. The more texture information is used the more precise match is achieved. To determine these feature points, we extract the overlap image between two images using the Fourier method and divide this overlap image of the reference into the four subareas. Then we choose the central pixel of the block which has the maximum local variance in each subarea as the *quasi* feature points. We use the block matching algorithm [9][10], which is usually used in motion compensation, to match these selected four feature points with the four points in the target image. The proposed algorithm uses the modified block matching criteria which is considered of the image distortion due to the camera movement and the variation of the illumination conditions when acquiring the images.

Once we have found the projective transform between two images, we can warp image into the reference frame using the transform and then blend the two images together. To reduce visible artifacts, we weight images being blend together more heavily towards the center, using a bilinear weighting function.

2. Extraction of the overlap image between two images

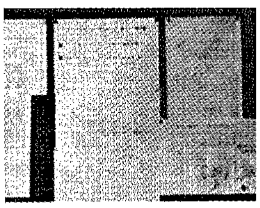
The sequence of images, which are acquired from a camera, has common information. Since the overlap area between two consecutive images has common information, we can induce the projective transform using the correspondence in the two overlap areas. Generally to extract the overlap area between two images, we transform the images into the frequency domain. Then we can extract the overlap area of each image using the phase information.

Fourier transform searches for the optimal match according information in the frequency domain. This is less computational cost and more robust to the noise rather than others [4][5][6].

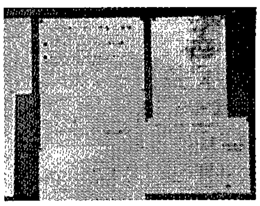
Phase correlation relies on the translation property of the Fourier transform. Given two images $f_1(x,y)$, $f_2(x,y)$ which differ only by a displacement (x_0, y_0) and their corresponding Fourier transforms $F_1(u,v), F_2(u,v)$, their cross-power spectrum is defined as equation (1)

$$\frac{F_1(u,v)\cdot F^*{}_2(u,v)}{\left|F_1(u,v)\cdot F_2(u,v)\right|} = e^{(-j2\pi(ux_0+vy_0))} \qquad (1)$$

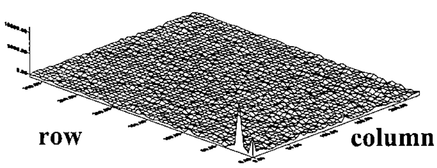
where F^* is the complex conjugate of F. If we take the inverse Fourier transform of the equation (1), then we will have a function which is an impulse, that is, it is approximately zero everywhere except at the displacement which is needed to optimally register the two images [4]. The figure 1 presents this result. (a) is the reference image and (b) is the translated image. (c) is the result that we apply these two image into the equation (1) and take the inverse transform. As we can know from the figure, the location of the impulse represents the displacement.



(a) Reference image



(b) Translated image



(c) Phase correlation result

Fig. 1 The result that apply the Fourier method

3. Algorithm that Extract the *Quasi*-feature and its Correspondence

The previous method demands high computational cost in order to induce the projective transform due to the iterative calculation. In this paper we propose the simple algorithm that selects the *quasi*-feature points, which are based on the intensities of images, and extracts their correspondence using the block matching algorithm.

3.1 Extraction of the *Quasi*-feature point

The *quasi*-feature is not the geometrical feature of the image but the feature based on the intensities of the image. So the *quasi*-feature point is defined as a central pixel of a block that has enough texture information. The four *quasi*-feature points are selected in the overlap area of the reference image. Because the selected four feature points don't have to exist in the same line, we divide the overlap area into four subarea. Next we search a block that has maximum local variance in each subarea. Then the central pixel of the searched block is defined as a *quasi*-feature point.

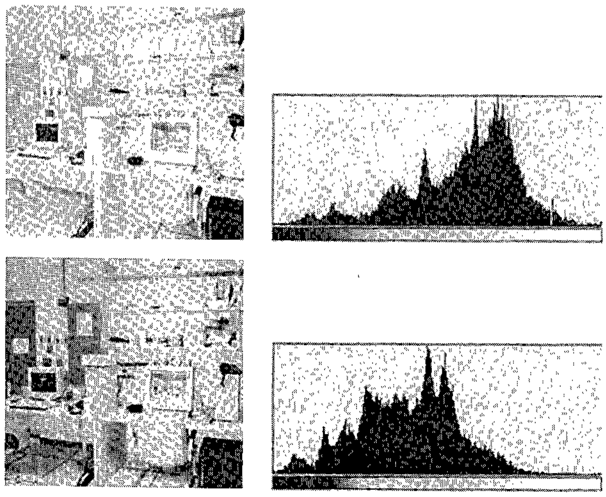
3.2 Histogram equalization being considered luminance variation

As the illumination conditions are changed, both the brightness and the contrast of the images fed from the camera are also changed. These variations become the causes which we mismatch the corresponding points when we apply the block matching algorithm. Therefore we have to consider the luminance variation for robust matching.

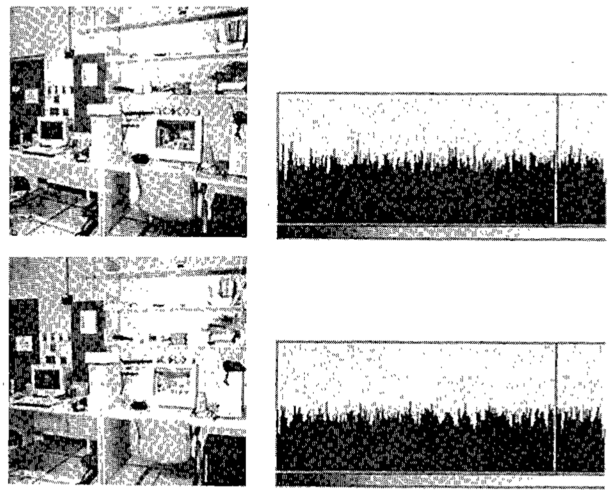
Since the histogram equalization distributes an image's graylevels uniformly about the range of graylevels, all images will have approximately the same brightness and contrast, hence allowing images to be compared equally without a bias due to perceived contrast and brightness differences [7]. This equalization is applied into the two overlap areas. The figure 2-(a) presents the images and their histogram before histogram equalization, (b) presents the images and their histogram after histogram equalization. We can easily know that the brightness and the contrast of two images are different from each histogram. As we apply the histogram equalization into each image, however, the different brightness and the contrast of the images would be similar each other.

3.3 Modified Block Matching Criteria

To search the correspondence for the selected four *quasi*-feature points in the reference image, we use the block matching algorithm that is generally used for motion compensation [8][9][10]. The previous algorithm simply compares the intensity in the block. However as the camera moves the image is distorted, and as the illumination conditions are changed the intensities of the image is also changed. Therefore if we don't consider these variations we cannot obtain the correct correspondences.



(a) Input images and their histogram before equalization



(b) Input images and their histogram after equalization

Fig. 2 Input images and their histogram before and after histogram equalization

Consequently, in this paper, we define the modified block matching criteria. The fundamental criteria is the minimum MAD (Mean Absolute Difference)[8] and the weight function and the BVD (Block Variation Difference) are added to the minimum MAD. Equation (2) presents this modified block matching criteria.

$$
\begin{aligned}
E(d_x, d_y) = [&\sum_{(x,y)\in B} \left| s(x,y,k) - s(x+d_x, y+d_y, k+1) \right| \cdot w_{i,j} \\
&+ \left| \mathrm{var}(x,y,k) - \mathrm{var}(x+d_x, y+d_y, k+1) \right|]^{1/2} \\
\left[\hat{d}_x, \hat{d}_y\right]^T &= \arg \min_{(d_x, d_y)} E(d_x, d_y)
\end{aligned}
\tag{2}
$$

$s(x,y,k)$ is the intensity of (x,y) in the k_{th} frame, $\mathrm{var}(x,y,k)$ is the variation of the block that (x,y) is center in the k_{th} frame.

The movement of the camera would distort an image, hence we have to consider this distortion for the robust match. The image distortion increases monotonically as the searched location moves away from the direction of minimum distortion. In other words, the distortion is small around the feature point, but the distortion increases monotonically as the distance from the feature is increased. Therefore we apply the weight function into the block so that this weight decreases the error in the small distortion and increases the error in the large distortion. This enables us to get the optimal match. The weight function that is proportional to the distance from the center of the block is defined as equation (3)

$$
\begin{aligned}
w_{i,j} &= d_{i,j} / D, \; -7 \le i, j \le 7 \\
d_{i,j} &= \sqrt{i^2 + j^2}, \;\; D: \text{maximum distance}
\end{aligned}
\tag{3}
$$

Since the *quasi*-feature point is defined as a maximum local variance, the corresponding block has to be a maximum local variance. So we include the variation term in the matching criteria. As that result, the matching correctness increases. The figure 3 presents the result that we apply the proposed algorithm. The four marked points in (a) are the selected *quasi*-feature points and the marked points in (b) are the corresponding points which are found with our proposed algorithm. As we see from the experimental results, the performance of the proposed algorithm is better than the general block matching criteria.

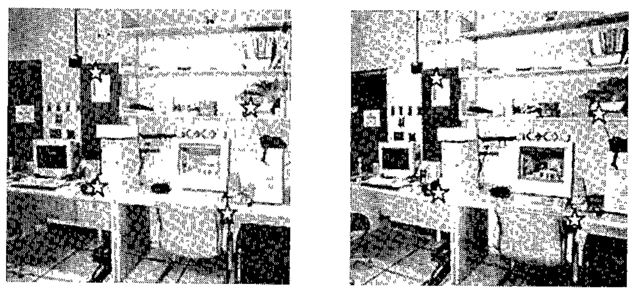


Fig. 3 Results that applied the proposed algorithm *Quasi*-feature points (left) and their correspondence (right) are marked with ☆

4. 2D Projective Mapping

The projective mapping, also known as the perspective or homogeneous transformation, is a projection from one plane through a point onto another plane. Homogeneous transformations are used extensively for 3D affine modeling transformations and for perspective camera transformations. The 2D projective mappings studied here are a subset of these familiar 3D homogeneous transformations.

Now we consider the planar projective transform that a point in a 3D space is projected into two different image planes which have different view points. The figure 4 presents this relationship.

For a camera centered at the origin, the relationship between a 3D point $P(X,Y,Z)$ and its image coordinates $p(x,y,1)$ can be described by

$$p = V \cdot P \tag{4}$$

where V represents the camera intrinsic parameters. The

3D direction corresponding to a screen pixel p is given by $P = V^{-1} \cdot p$.

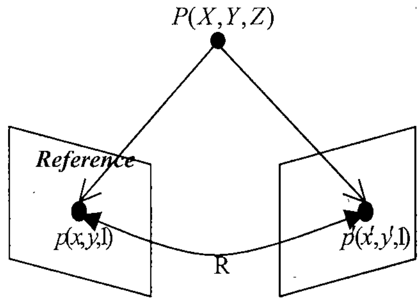


Fig. 4 Example of the planar projective transform

For a camera rotating around its center of projection, the perspective projection between two images p and p' is therefore given by equation (5),

$$p' = V \cdot R \cdot P = V \cdot R \cdot V^{-1} \cdot p = M_{projective} \cdot p \quad (5)$$

where R represents the 3D rotation matrix of a camera.

$$M_{projective} = \begin{bmatrix} m_0 & m_1 & m_2 \\ m_3 & m_4 & m_5 \\ m_6 & m_7 & m_8 \end{bmatrix} \quad (6)$$

$M_{projective}$ is the projective matrix to induce ultimately. Generally if $m_8 \neq 0$, m_8 is normalized with 1. If we use the equation (4), (5), and (6) then we can obtain the corresponding relation between two points like (7). Then we apply four *quasi*-feature points pairs into the equation (7) to obtain the 8 equations like (8).

$$x'_k = \frac{m_0 x_k + m_1 y_k + m_2}{m_6 x_k + m_7 y_k + m_8}$$
$$y'_k = \frac{m_3 x_k + m_4 y_k + m_5}{m_6 x_k + m_7 y_k + m_8} \quad (7)$$

$$m_0 x_k + m_1 y_k + m_2 - m_6 x_k x'_k - m_7 y_k x'_k = x'_k \quad (8)$$
$$m_3 x_k + m_4 y_k + m_5 - m_6 x_k y'_k - m_7 y_k y'_k = y'_k, k = 1,2,3,4$$

If we solve these equations with four corresponding pairs that are searched, then we can obtain the projective transform between two consecutive images. When we solve these equations we need not any iterative calculation.

5. Experimental results

We experiment the proposed algorithm with the five image pairs, which are presented in the figure 5, to estimate the performance of the proposed algorithm. The image size is 320×240 and block size is 15×15, and the search range is 32×32. As the figure 6 presents, the proposed algorithm is composed of largely two parts. Fig. 6-(a) and (b) show the global and local matching procedure, respectively.

Once we have found the projective transform, $M_{projective}$, between two images, we can warp image into the reference frame using the transform, $M_{projective}$ and then blend the two images together. When blending multiple images, the projective transform of the k_{th} frame is represented as a form of multiplied transform. To reduce visible artifacts, we weight images being blend together more heavily towards the center, using a bilinear weighting function. These equations are presented in (9) and (10).

$$M_k = M_{k-1} \cdot M_{k-2} \cdot M_{k-1} \cdot \ldots \cdot M_1 = \prod_{i=1}^{k} M_i \quad (9)$$

$$I_{mosaic}(x,y) = \frac{I_{reference}(x,y) \cdot W_B + I_{target}(x,y) \cdot W_B}{W_A + W_B} \quad (10)$$

$$W_A = \sqrt{\left(x - \frac{I_{ref\_width}}{2}\right)^2 + \left(y - \frac{I_{ref\_height}}{2}\right)^2}$$
$$W_B = \sqrt{\left(x - \frac{I_{tar\_width}}{2}\right)^2 + \left(y - \frac{I_{tar\_height}}{2}\right)^2}$$

where I_{mosaic} is the blended intensity at (x, y), and both W_A and W_B are the weight functions of the reference and target image, respectively.

The experimental results are compared with previous two methods. The first method is to directly match by human intervention and the second is to use the Levenberg-Marquardt iterative nonlinear method. The evaluation function is presented as equation (11) [1][2][3], and in ideal case its value is zero.

$$E = \frac{1}{N}\left\{ \sum_{(x,y)\in R} \left[I_k(x,y) - I'_{k-1}(x,y)\right]^2 \right\}^{1/2} \quad (11)$$
R : overlap region

Each I_k, I'_{k-1} is the intensity of the reference and warped target image and N is the number of the pixel in the overlap region R.

The figure 7 presents the results of the figure 5-(1) using the methods above mentioned. As we can see, the all of results are good, but we still can see some artificial in (b), for example, the bottom stair is curved. It means that (b) doesn't reflect geometrical feature completely. The figure 8 presents the comparison between the general block matching criteria and the proposed block matching criteria. As we use the proposed matching criteria, the two lines, which are marked with dotted circles, are well aligned and the overall blurring is reduced. Therefore, we can obtain the improved mosaic image. The quantitative analysis is presented in the table 1, 2, and 3.

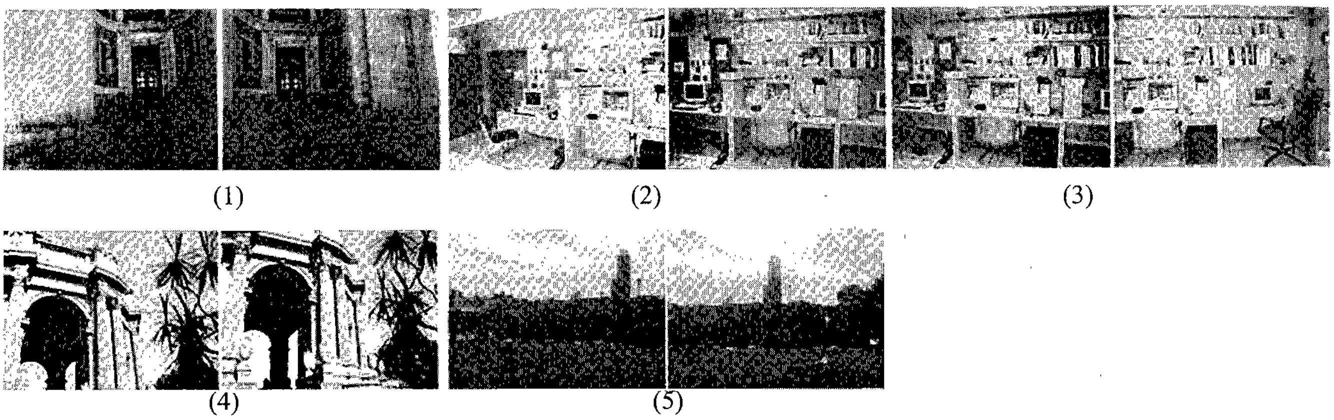
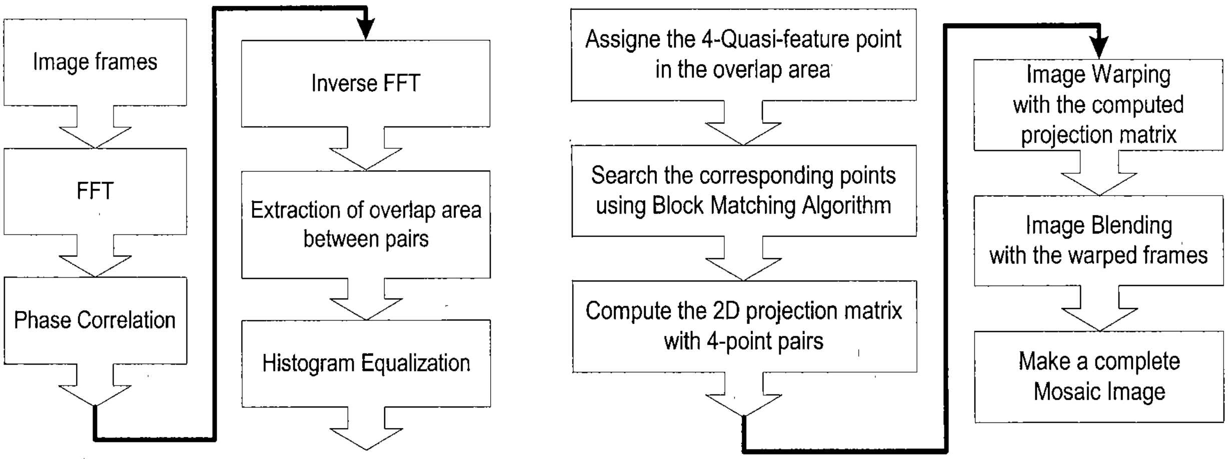


Fig. 5. Five image pairs to be used for experimentation



(a) Procedure of the global matching

(b) Procedure of the local matching

Fig. 6. Overall procedure of the simulation

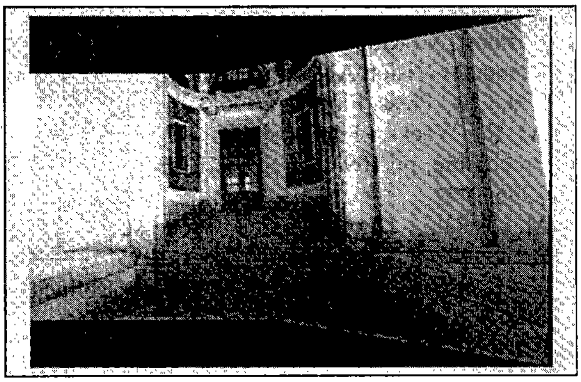
As we can see the table 1, though, when the matching is done by human intervention the error is least, but this can not construct the mosaics automatically. Moreover, since the Levenberg-Marquardt method doesn't reflect the enough geometrical feature, we can often see the visual artifact. For the most of the experimented images, however, the proposed algorithm can construct mosaic automatically and obtain the more precise projective transform than the Levenberg-Marquardt method. The table 2 presents the comparison between the general block matching criteria and the proposed block matching criteria and the table 3 presents the performance analysis whether the BVD is included or not. The experimental results present that the proposed algorithm is better than the others. The figure 9 presents some examples of mosaics that apply the proposed algorithm.

6. Conclusion

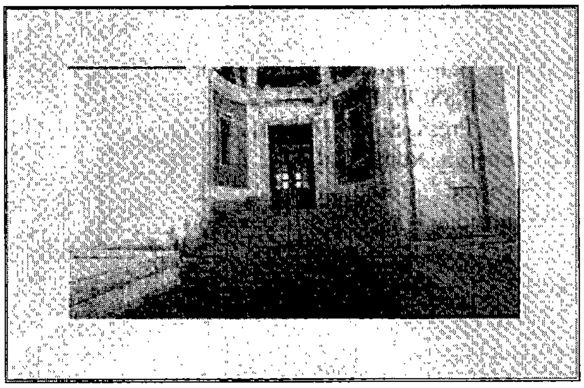
We have proposed an algorithm that extracts the *quasi*-feature points and their correspondence to compute the projective transformation between two consecutive images for image mosaics. The proposed algorithm defines the *quasi*-feature point and selects the four *quasi*-feature points in the reference image. Then their corresponding points are searched in the target image using the block matching algorithm that is considered both the image distortions and the variation of the illumination conditions.

To obtain the optimized transform, in the previous methods, one intervenes or computes the nonlinear equations iteratively. However the proposed algorithm can automatically construct the mosaic image without human intervention and needs not any iteration in order to compute the optimized projective transform. We use only four *quasi*-feature points and their correspondence. Moreover, when searching for the correspondence, we apply the block matching algorithm for only four blocks that include each *quasi*-feature point. Therefore, we can decrease the computational cost. The proposed algorithm is very simple and can be implemented easily, but the experimental results present that the proposed algorithm has better performance than the previous methods for the most of the experimented images.

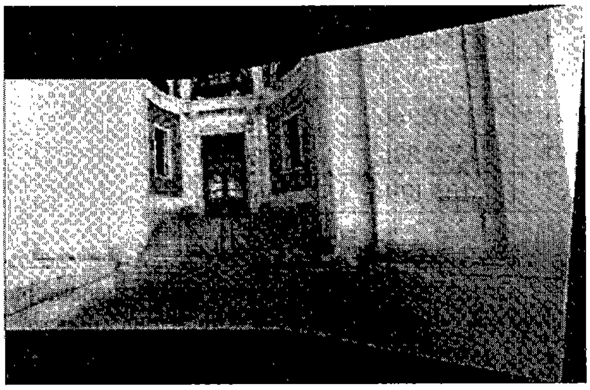
Now we don't consider the seam lines between two images. So, in the future works, we would study the removal of the seam lines and this research will be a fundamental foundation for the novel view scene reconstruction and 3D image reconstruction.



(a) Directly matching with human intervention

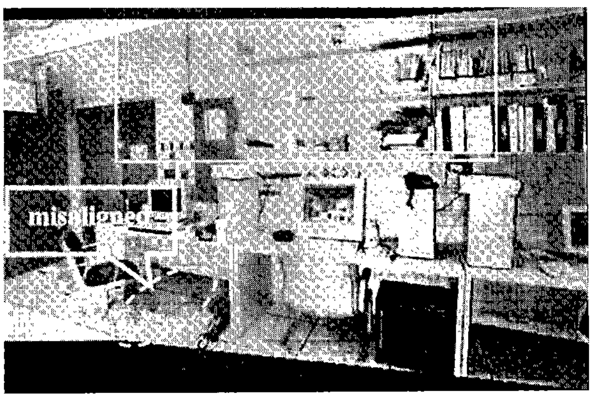


(b) Result image of the Levenberg-Marquardt method

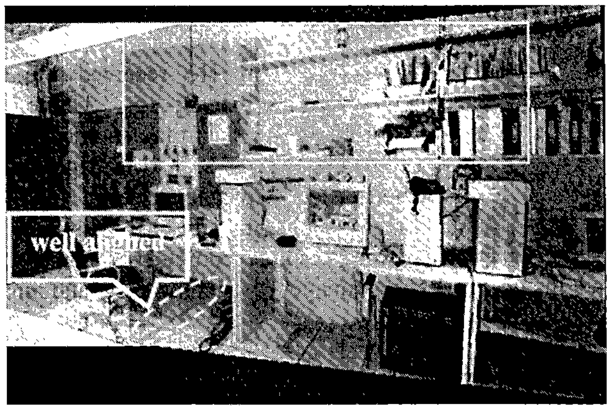


(c) Result image applied the proposed algorithm

Fig. 7 Result images applied the previous methods and the proposed algorithm



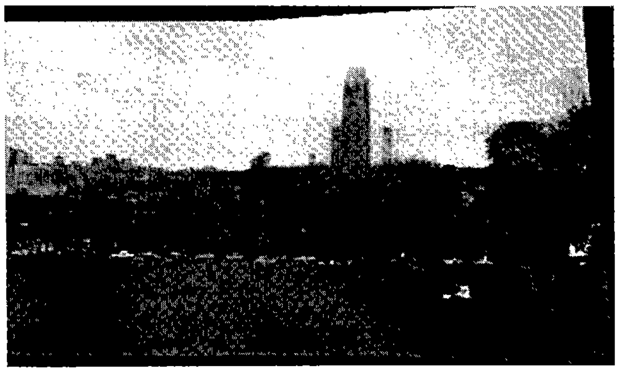
(a) Result image applied the general matching criteria



(b) Result image applied the proposed matching criteria

Fig. 8 Result images applied the general block matching criteria and the proposed block matching criteria

(a) Result image for the 5-(2), (3)

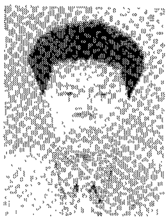


(b) Result image for the 5-(5)

Fig. 9 Examples of the image mosaics that applied the proposed algorithm

7. Biography

<u>Dae-Hyun Kim</u> He received the BS degree from Chung-Ang University, Seoul, Korea, in electronic engineering, in 1999 and he received his MS degree from Chung-Ang University, Seoul, Korea, in image engineering, in 2001. His research interests include computer vision, signal and image processing, image based rendering and image synthesis.



<u>Yong-In Yoon</u> He received his BS and MS degrees in electronics engineering from Chung-Ang University, Seoul, Korea, in 1986 and 1988, respectively. He was with IAE at Daewoo Co., Seoul, Korea, from 1994 to 1999. He is currently working toward his PhD degree in image engineering at Chung-Ang University. His research interests are computer vision and image processing.

<u>Jong-Soo Choi</u> He received his BS degree from Inha University, Inchon, Korea, his MS degree from Seoul National University, Seoul, Korea, and his PhD degree from Keio University, Yokohama, Japan, all in electrical engineering, in 1975, 1977, and 1981, respectively. He joined the faculty at Chung-Ang University in 1981, where he is now a dean of the Graduate School of Advanced Imaging Science, Multimedia, and Film. His current research interests are in computer vision, image coding, and electro-optical systems.

Acknowledgments

This research was supported by the Ministry of Education, Seoul, Korea, under the BK21 project, and by the Ministry of Science and Technology, Seoul, Korea, under the NRL project.

(Manuscript received Oct.2, 2000, revised Dec.22, 2000)

References

[1] R. Szeliski, "Image Mosaicing for Tele-Reality applications," IEEE Workshop on Applications of Computer Vision (WACV'94), pp. 44-53, Florida, December 1994.

[2] Z. Imad, O. Faugeras, R. Deriche, "Using geometric ccorners to build a 2D mosaic from a set of images," Proc. 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp.420-425, Puerto Rico, June, 1997.

[3] R. Szeliski, S. B. Kang, "Direct Methods for Visual Scene Reconstruction," IEEE Workshop on Representations of Visual Scenes, pp. 26-33, Massachusetts, June 1995.

[4] L. G. Brown, "A Survey of Image Registration Techniques," ACM Computing Surveys, Vol.24, No.4, pp.325~376, Dec. 1992.

[5] B. S. Reddy, B. N. Chatterji, "An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration," IEEE Trans. On Image Processing, Vol.5 No.8 pp.1266-1271, Aug. 1996.

[6] Gerhard X. Ritter, Joseph N. Wilson, Handbook of Computer Vision Algorithm in Image Algebra, pp.225-239, CRC Press, Florida, 1996.

[7] A. R. Weeks, Fundamentals of Electronic Image Processing, pp.109-120, SPIE/IEEE Press, New York, 1996.

[8] M. Tekalp, Digital Video Processing, pp.95-116, Prentice Hall, Upper Saddle River, 1995.

[9] Y. Huang, X. Zhuang, "An Adaptively Refined Block Mathching Algorithm for Motion Compensated Video Coding," IEEE Trans. On Circuits/Systems for Video Tech. Vol.5, No.1 pp.56-59, Feb. 1995.

[10] H. M. Jong, L.G. Chen, T.D. Chiueh, "Accuracy Improvement and Cost Reduction of 3-Step Search Block Matching Algorithm for Video Coding," IEEE Trans. On Circuits/Systems for Video Tech. Vol.4, No.1 pp.88-90, Feb. 1994.

[11] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery, Numerical Recipes in C : The Art of Scientific Computing 2nd Edition, pp.656~706, Cambridge Univ. Press, New York, 1992.

Table 1. Performance estimation between the previous method and the proposed algorithm

Image	E		
	Manual Matching	Maquardt Method	Proposed Method
(1)	334.685	635.088	398.219
(2)	2374.994	2887.741	2338.182
(3)	2310.340	2602.408	1654.783
(4)	1507.370	1617.515	1659.938
(5)	105.179	166.756	145.686

Table 2. Performance estimation between the general block matching criteria and the proposed block matching criteria

Image	E	
	General BMA	Proposed Method
(1)	391.594	398.219
(2)	2459.840	2338.182
(3)	2012.009	1654.783
(4)	2486.434	1659.938
(5)	140.042	145.686

Table 3. Performance estimation whether the BVD term include or not

Image	E	
	without BVD	with BVD
(1)	381.166	398.219
(2)	2397.136	2338.182
(3)	1785.568	1654.783
(4)	1689.331	1659.938
(5)	150.314	145.686