



Fast Image Registration Using Pyramid Edge Images

Kee-Baek Kim, Jong-Su Kim, Sangkeun Lee, and Jong-Soo Choi*

*The Graduate School of Advanced Imaging Science, Multimedia and Film,
Chung-Ang University, Seoul, Korea(R.O.K)*

*Corresponding author's E-mail address: jschoi@cau.ac.kr

Abstract: Image registration has been widely used in many image processing-related works. However, the existing researches have some problems: first, it is difficult to find the accurate information such as translation, rotation, and scaling factors between images; and it requires high computational cost. To resolve these problems, this work proposes a Fourier-based image registration using pyramid edge images and a simple line fitting. Main advantages of the proposed approach are that it can compute the accurate information at sub-pixel precision and can be carried out fast for image registration. Experimental results show that the proposed scheme is more efficient than other baseline algorithms particularly for the large images such as aerial images. Therefore, the proposed algorithm can be used as a useful tool for image registration that requires high efficiency in many fields including GIS, MRI, CT, image mosaicing, and weather forecasting.

Keywords: Image registration; FFT; Pyramid image; Aerial image; Canny operation; Image mosaicing.

1. Introduction

Image registration is a basic task in image processing to combine two or more images that are partially overlapped. Image registration has been used in various fields such as geographic information system (GIS), aerial, satellite, weather, underwater images, and etc [3-5]. Many image-related works can be separated into three main classes which are the correlation-based, feature-based, and FFT-based methods. However, they cannot find the sub-pixel accuracy information for image registration when the full-sized images need to be registered. In particular, FFT-based approach has been widely used since it is fast and more robust than others in image brightness changes even though it may have some problems when it handles noisy images in the feature detection stage. However, its computation cost increases significantly, and it cannot find the accurate

registration information as image size increases [3]. To solve these problems, this work employs a pyramid-based image decomposition scheme. Specifically, first, we use the Gaussian filter to remove the noise because it causes many undesired problems in image registration works. Next, Canny edge operator is applied on a given input image to reduce the computational cost dealt with. Next, the pyramid-based image decomposition is conducted to obtain the sub-pixel accuracy information and to improve the computation speed by using smaller image size if necessary. Then, we estimate the registration information integrated with the computed values from each pyramid level. It is noted that this work estimates the fundamental geometric changes such as image rotation, scaling, and translation for robust and effective registration.

The experiment shows that the proposed algorithm saves computational cost with maintaining the image quality, and it is particularly efficient in aerial images that contain relatively less geometric

warping problem.

This paper is organized as follows. In Section 2, the existing works are reviewed, and we present a proposed scheme for image registration in Section 3. Section 4 shows the experimental results and compares with other baseline algorithms. In Section 5 concludes this work with some discussions.

2. Background

2.1. Canny Edge Detection

Canny edge detector is defined by three concepts which are error rate, localization, and response [1-2]. Basic idea is to detect the zero-crossings of the second directional derivative of the smoothed image in the direction of the gradient where the gradient magnitude of the smoothed image is greater than some threshold depending on image statistics. Specifically, it first employs a Gaussian function for the image smoothing and uses the first derivative on it, respectively, as

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$\frac{\partial G(x, y)}{\partial x} = \alpha x e^{-\frac{x^2+y^2}{2\sigma^2}}, \quad \frac{\partial G(x, y)}{\partial y} = \alpha y e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

$$\left(\because \alpha = \frac{1}{2\pi\sigma^2} \right)$$

Next, original image and this function are convolved in both the vertical and horizontal directions. This operation is described in the following equation [1-2] as,

$$I'(x, y) = g(k, l) * I(x, y)$$

$$= \sum_{k=-N}^N \sum_{l=-N}^N g(k, l) I(x-k, y-l) \quad (2)$$

Where, I is an original image, and g is a convolution kernel. The non-maximal suppression step finds the local maxima in the direction of gradient, and suppresses all others for minimizing false edges. Then, a hysteresis threshold is applied. The hysteresis threshold uses two level threshold values which are t_h and t_l ($t_h > t_l$) for a high and low levels, respectively [2].

The performance of Canny edge detector depends on the parameters σ , t_h , and t_l . If σ is bigger, we can make a stronger Gaussian filter. Fig. 1 shows the procedures of Canny edge detection.

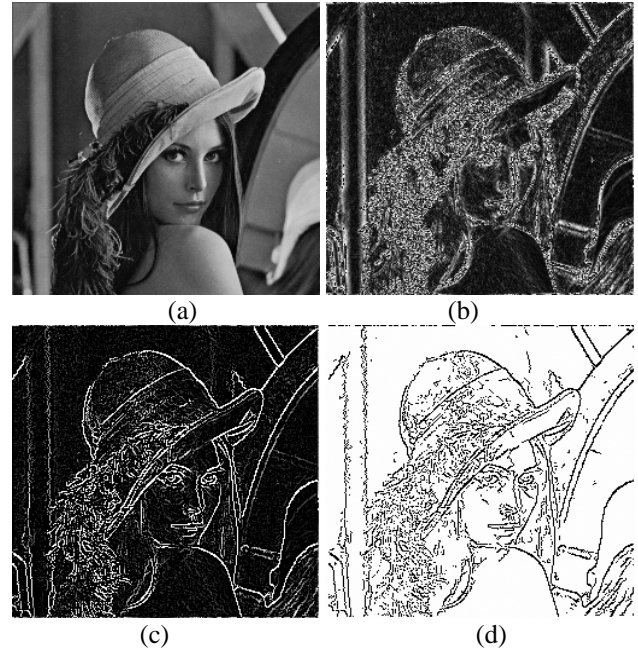


Fig. 1. Filtering results of Canny edge detector : (a) An Origin image, (b) Gaussian filtering with $\sigma=0.5$, (c) Non-maximal suppression, and (d) Hysteresis Thresholding with $t_h=128$ and $t_l=64$.

2.2. Fourier Based Registration

In Fourier-based image registration, the first step is to find the translation information, and then the rotation and scaling Information are computed [4-5].

2.2.1. Translation

A displacement (x_0, y_0) between two consecutive images is formulated by

$$f_2(x, y) = f_1(x - x_0, y - y_0) \quad (3)$$

Using a Fourier shifting property, we get

$$F_2(\omega_1, \omega_2) = F_1(\omega_1, \omega_2) e^{-j(\omega_1 x_0 + \omega_2 y_0)} \quad (4)$$

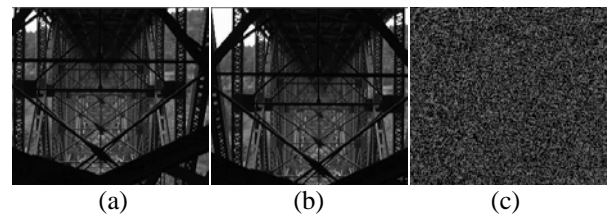


Fig.2. Translation : (a) Input image1, (b) Input image2, and (c) Cross-power spectrum

Next, we can find the cross-power-spectrum between images (see Fig. 2) [4-5] as

$$\frac{F_2(\omega_1, \omega_2) F_1^*(\omega_1, \omega_2)}{|F_2(\omega_1, \omega_2) F_1^*(\omega_1, \omega_2)|} = e^{-j(\omega_1 x_0 + \omega_2 y_0)}$$

$$\stackrel{F^{-1}}{=} \delta(x - x_0, y - y_0) \quad (5)$$

From this, we can compute the translation information easily.

2.2.2. Rotation

Similarly, to get the rotation information, we first formulate the rotation relation in the spatial domain [4-5] as

$$f_2(x, y) = f_1(x \cos \theta_0 + y \sin \theta_0, -x \sin \theta_0 + y \cos \theta_0) \quad (6)$$

Next, we convert it to the Fourier space as

$$F_2(\omega_1, \omega_2) = F_1(\omega_1 \cos \theta_0 + \omega_2 \sin \theta_0, -\omega_1 \sin \theta_0 + \omega_2 \cos \theta_0) \quad (7)$$

After converting the above equation to the polar coordinates [4-5] as

$$M_2(\rho, \theta - \theta_0), \quad (8)$$

we can get the rotation information θ_0 (see Fig. 3 for the procedure).

2.2.3. Scale

When an image is scaled by a to another image, the relation in Fourier transform is expressed by

$$F_2(\omega_1, \omega_2) = \frac{1}{a^2} F_1(\omega_1 / a, \omega_2 / a). \quad (9)$$

Taking the logarithm, this can be rewritten by [4-5]

$$F_2(\omega_1, \omega_2) = F_1(\log \omega_1 - \log a, \log \omega_2 - \log a). \quad (10)$$

We can get the scale factor a from this. Note that we ignore a constant $1/a^2$ for simplicity.

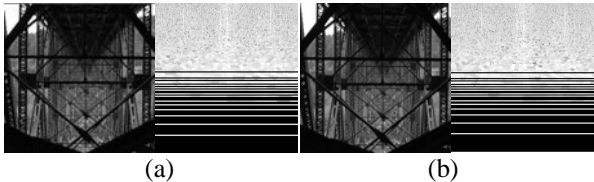


Fig.3. Rotation and scaling : (a) Input image 1 and Log-Polar, (b) Input image 2 and Log-Polar.

3. Proposed Algorithm

There are three main steps in image registration procedures. They are a feature detection, a feature matching, and the panorama image generation. In particular, the feature detection in registration related works is severely affected by brightness images between two given images. To solve this problem and to reduce the computation complexity, we use pyramid edge images in feature detection. Moreover, to accelerate the processing speed, we collect the registration information from FFT-based approach. Fig. 4 shows the whole procedures.

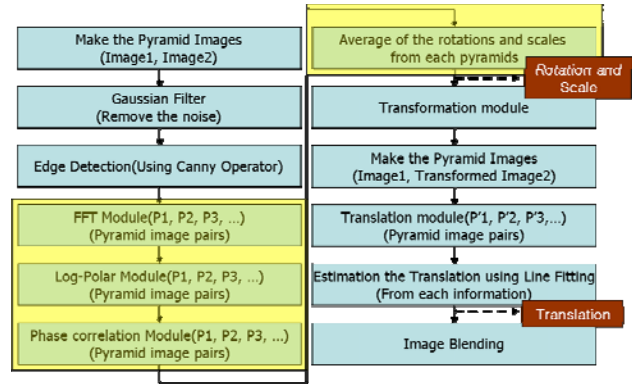


Fig.4. Flow chart of the proposed Algorithm

3.1. Feature detection

In this work, to reduce the computation cost, a pyramid image decomposition scheme is used. Let the corresponding pyramid pairs and input images are g_1^n, g_2^n and I_1, I_2 respectively. Then, the pyramid decomposition [2] is described by

$$g_1^n(x, y) = P^n(I_1(x, y)), \quad g_2^n(x, y) = P^n(I_2(x, y)), \quad n = 1, 2, \dots, n. \quad (11)$$

Where, n and P means pyramid levels and a pyramid function. It is noted that the pyramid levels are determined by the degree of feature matching. In other words, an image is inputted to the next level of pyramid until registration information is not found in feature matching.

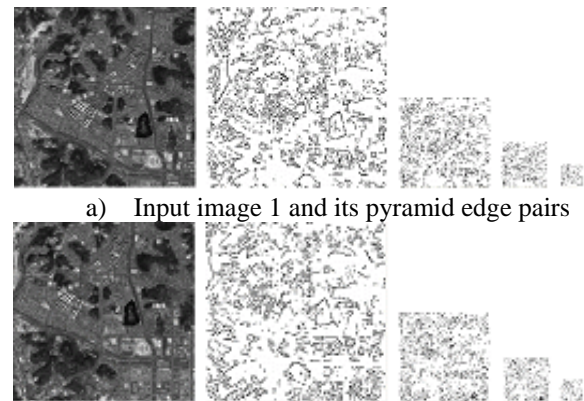


Fig.5. Pyramid edge pairs

To reduce more computation cost, the edges in pyramid images are used since pixel-by-pixel processing is generally more complicated. Moreover, when edge images are used in image registration, the noise from brightness differences is reduced [3-5]. For this purpose, we use a Canny operator. Canny operator has many merits. For example, this can reduce the noise by a Gaussian function and it gives the ideal edges. Let pyramid edge images be c_1^n, c_2^n and the first derivative of a Gaussian function be

$h(k,l)$. Then, the corresponding pyramid edge images [2] can be defined by

$$\begin{aligned} c_1^n(x,y) &= h(k,l) * g_1^n(x,y) \\ c_2^n(x,y) &= h(k,l) * g_2^n(x,y) \end{aligned}, \quad n=1,2,\dots,n. \quad (12)$$

Fig. 5 shows the pyramid edges pairs at each level.

3.2. Feature matching

Once feature detection is done, a feature matching step is conducted. In the feature matching step, the rotation and scale information are computed before the translation estimations [4-5].

3.2.1. Estimation of Rotation and scale

To obtain the rotation information, first, each pyramid layers determined by the previous feature detection step is transformed into the spectrum domain. Next, the transformed image is expressed again in polar coordinate system as shown below. Suppose that C_k^n is a Fourier transformed image, and it is converted again into the polar coordinate M_k^n for $k=1, 2$ at the n th level. Then, we can easily compute M_k^n from Eq. (8) with taking scale information into account [4-5] as

$$M_1^n(\rho, \theta) = M_2^n(\rho/a, \theta - \theta_0). \quad (13)$$

Finally, the logarithm is taken for simple computation [4-5] as

$$M_1^n(\log \rho, \theta) = M_2^n(\log \rho - \log a, \theta - \theta_0). \quad (14)$$

In this step, we can get the scale and rotation angles from each pyramid layer. The final scale and rotation information in the original size layer is defined by averaging. Let the scales and rotations be S_n and R_n , respectively, at n th level. Then, the original registration information is defined by

$$S = \frac{\sum_{n=1}^N S_n}{N}, R = \frac{\sum_{n=1}^N R_n}{N} \quad (n \text{ is pyramid degrees}) \quad (15)$$

(S is the original scale, and R is the original rotation)

Translation is estimated after an input image 2 is transformed using a transformation module and values gained by this step.

3.2.2. Estimation of Translation

Let C_2^n be a Fourier image for an edge image c_2^n from a pyramid layer g_2^n which is adjusted with the computed scale and rotation information. Then, we can estimate the translation information between the given input image 1 C_1^n and an adjusted input

image 2 C_2^n by Fourier shift property [4-5] as

$$\frac{C_2^n(\omega_1, \omega_2) C_1^n(\omega_1, \omega_2)^*}{|C_2^n(\omega_1, \omega_2) C_1^n(\omega_1, \omega_2)|} = e^{-j(\omega_1 x_0 + \omega_2 y_0)} \quad (16)$$

Suppose that the obtained values in the n th layer are T_n . Then, to estimate registration information with sub-pixel accuracy for an original image, we use the values T_n for computing a mapping function based on χ^2 function because this approach is more robust than the others. The line fitting is performed to meet the following conditions [6].

$$T(X, Y) \leq T_n(x_n, y_n)$$

This is expressed by

$$Y(X) = y_n(x_n; a, b) = a + bX, \quad n=1,2,3,\dots,n. \quad (17)$$

To find the offset b and slope a the above Eq. (17) can be computed [6] as,

$$\begin{aligned} \Delta &\equiv SS_{xx} - (S_x)^2, \\ a &= \frac{S_{xx}S_y - S_xS_{xy}}{\Delta}, b = \frac{SS_{xy} - S_xS_y}{\Delta}. \end{aligned} \quad (18)$$

$$\left(\begin{aligned} S &\equiv \sum_{i=1}^N \frac{1}{\sigma_i^2}, S_x \equiv \sum_{i=1}^N \frac{x_i}{\sigma_i^2}, S_y \equiv \sum_{i=1}^N \frac{y_i}{\sigma_i^2}, \\ S_{xx} &\equiv \sum_{i=1}^N \frac{x_i^2}{\sigma_i^2}, S_{xy} \equiv \sum_{i=1}^N \frac{x_i y_i}{\sigma_i^2} \end{aligned} \right)$$

Once the a and b are solved, we can estimate the original information X and Y as follows:

$$\begin{aligned} Y &= a + bx_1 \times 2 \\ X &= \frac{2 \times y_1 - a}{b} \quad (\because X \cong 2 \times x_1, Y \cong 2 \times y_1). \end{aligned} \quad (19)$$

Then, we can get the information for the original image using the first pyramid layer, $T_1(x_1, y_1)$, because $T_1(x_1, y_1)$ gives about the half of original positions.

4. Experiment and Result

To evaluate the performance of the proposed approach, we use two types of images: stereo and aerial images. Additionally, we compare the results of the proposed scheme with those of the baseline algorithms for the performance comparison.

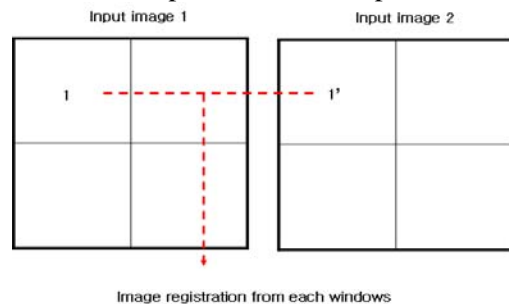


Fig.6. Image registration using fixed windows



Fig. 7. the general stereo image sets (Up : a ship, Middle : a doll, Bottom : a piano).

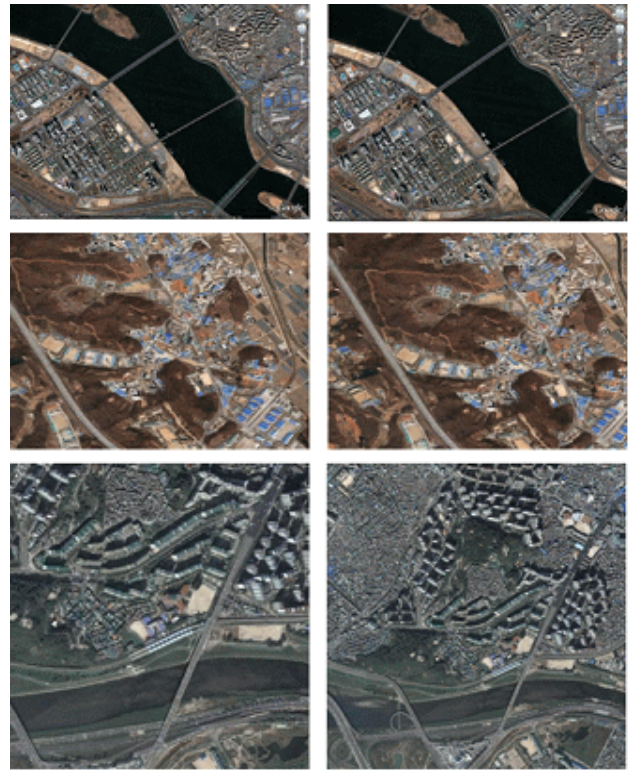


Fig. 8. Aerial image sets (Up : Aerial images 1, middle : Aerial images 2, bottom : Aerial images3).

Table. 1 The PSNR and Time (CLK) for stereo sets

	Ship		Doll		Piano	
	PSNR	Time(CLK)	PSNR	Time(CLK)	PSNR	Time(CLK)
FFT	9.487530	20282	8.857208	38782	9.821853	24390
Edge-FFT	9.442783	20906	8.624562	38735	9.807317	31640
Windows	8.638269	22484	8.172996	41531	9.692507	30250
Feature-point	12.794282	61125	13.3499	86223	17.878	1929
Block	10.806005	6688	9.7693	11875	22.753	4062
Proposed	11.910829	14297	12.206663	24953	12.296949	16422

Table. 2 The PSNR and Time (CLK) for aerial sets

	Aerial images 1		Aerial images 2		Aerial images 3	
	PSNR	Time(CLK)	PSNR	Time(CLK)	PSNR	Time(CLK)
FFT	9.752090	92675	10.019081	48406	17.907309	312359
Edge-FFT	9.524965	93828	12.344001	51984	22.221701	452172
Windows	9.774270	95985	11.083047	83422	17.515963	302266
Feature-point	11.8713	60922	NF	NF	NF	NF
Block	11.95228	24735	TL	TL	TL	TL
Proposed	13.201385	63953	11.918487	58454	20.693722	187579

The first baseline algorithm is a basic FFT method. This method requires too much computation and is too difficult to find the image features [4-5].

For the second approach, we extract the edge images from original images before a basic FFT method. After we get edge images, we process the whole images using a basic FFT method [10].

For the third one, we split an image into four non-overlapped sub-images. In the feature matching step, we obtain the required registration information from each sub-image. Note that it is performed in the frequency domain. Main benefit of this approach is low cost and fast [9].

The fourth one is based on feature-points using RANSAC [11]. To find features and the correspondences, we use a fast corner detection and zero-normalized cross correlation (ZNCC). This method can detect more robust features in images. However, this method requires too much computation, and is easily stuck in local minima. Nevertheless, this method usually detects the most robust features from the given images [12-13].

The last approach is block-based matching. This method relatively requires high cost even though it gives good feature detection results, and its performance is more robust than the others [14-15].

In order to evaluate the performance of the algorithms with respect to peak signal-to-noise ratio (PSNR) and speed, we use two sets of images: stereo and aerial images. The first set includes the stereo pairs (“ship”, “Doll”, and “piano”) which are commonly used in stereo-related works. The main reason selecting these pairs is to test the accuracy for image translation. The second one includes aerial images for evaluating the accuracy of each algorithm with respect to the rotation and scaling. Fig. 7 and 8 show the first and second sets.

Table 1 and Table 2 show the results of the compared algorithms in terms of PSNR and Time (CLK) for the stereo sets and aerial sets, respectively. It is noted that the used pyramid level is the 4th degrees in the case of aerial sets, while the 3rd level in the case of stereo sets. The reason of the different degree pyramids is because the proposed algorithm does not go further when it cannot find the registration information from each pyramid level. Additionally, canny operator parameter σ is 0.6.

In Table 2, ‘TL’ and ‘NF’ denote ‘too long computation time’ and ‘no feature found’, respectively. When the input images have the blurring and large scaling as in aerial image 2 and 3,

and their disparity vectors are large, the feature point-based method fails to find the exact correspondent points by ZNCC. Therefore, it cannot compute the homography in the images. This case is defined as ‘NF’ in this paper. Similar situation is happened in the block-based approach. It takes too long time to find the features even though we manually adjust the block and search window sizes, because this approach does not take into account the rotation and scaling. We call this case as ‘TL’.

It is interesting that the performance of proposed algorithm was more efficient in aerial image registrations. However, it was difficult for the proposed approach to apply for the general stereo pairs containing low feature.

Fig. 9 shows a graphical result for the estimated the original translation of aerial images 3 using the values gained from each pyramid level.

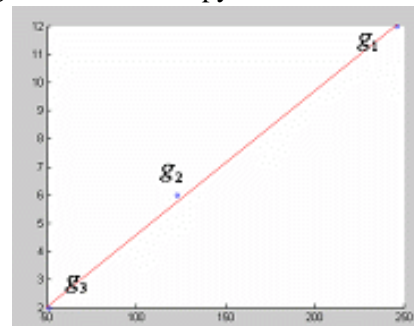


Fig.9. Translation estimation for an aerial image3.

For the comparative results, the registered images from each baseline algorithm with one of the stereo and two of aerial sets are shown in Fig. 10, Fig. 11, and Fig. 12, respectively.

In this experiment, we verify if an input image is a plane, our algorithm is the most efficient than the others. Moreover, it is more efficient for the larger image size.

5. Conclusion

This work presented a FFT-based image registration by using a pyramid edge representation and a line fitting. Experiment results with several sets of data show that the proposed algorithm is more efficient than the other baseline ones with respect to PSNR and executive speed.

However, it is noted that the proposed algorithm is not that efficient when a given input image size is small since we use the pyramid images. In the other words, if the number of pyramid levels is small, then it is difficult to estimate the accurate registration information because the line fitting function uses all

the information from each pyramid levels.

In contrast, the proposed algorithm is efficient and fast for the large-sized images like aerial images. Therefore, our future work will focus on the fitting methods to estimate more accurate information for the small sized images. We believe that the proposed algorithm can be used as an efficient tool for the applications that requires fast processing with the large sized images including aerial images.

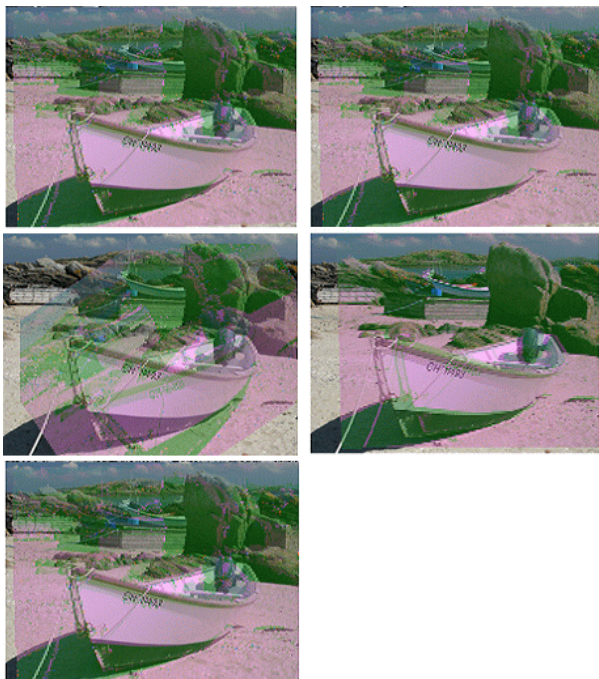


Fig. 10. Overlapped images in stereo sets (Ship).(Left – Up: FFT, Middle : Window, Bottom : proposed Right – Up : Edge-FFT, Middle : Feature-point).



Fig. 11. Overlapped images in Aerial sets(Aerial 1)(Left – Up: FFT, Middle : Window, Bottom : Block Right – Up : Edge-FFT, Middle : Feature-point, Bottom : proposed).

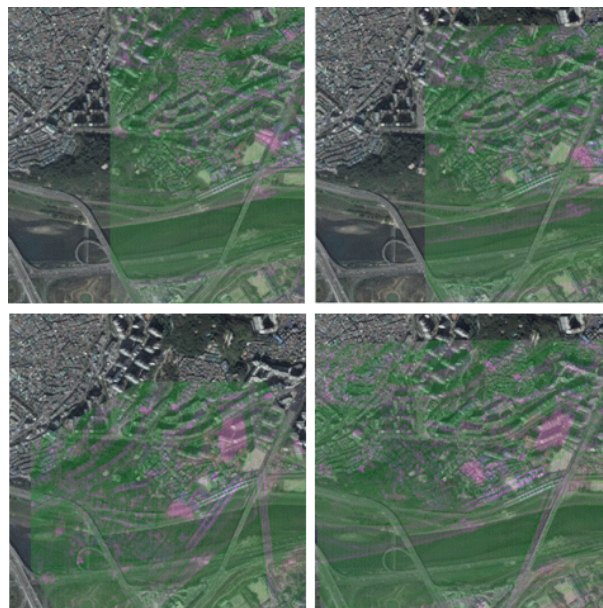


Fig. 12. Overlapped images in Aerial sets (Aerial 3).(Left – Up: FFT, Bottom : Window Right – Up : Edge-FFT, Bottom : proposed).

Acknowledgments

This work was supported by Korean Research Foundation under BK21 project, Seoul R&BD Program(TR080601), Seoul Future Contents Convergence (SFCC) Cluster established by Seoul R&BD Program(10570).

References

- [1] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 679-714, Nov. 1986.
- [2] Hong Shan Neoh and Asher Hazanchuk, "Adaptive Edge Detection for Real-Time Video Processing using FPGAs," *GSPx 2004 Conference*, Altera, NO. CF-EDG031505-1.0, May 2005.
- [3] Barbara Zitova and Jan Flusser, "Image Registration methods: Survey," *Image and Vision Computing*, vol. 21, Issue 11, pp. 977-1000, Oct. 2003.
- [4] Ryan Eustice, Oscar Pizarro, Hanumant Singh, and Jonathan Howland, "UWIT: Underwater Image Toolbox for Optical Image Processing and Mosaicking in MATLAB," *Underwater Technology, 2002. Proceedings of the 2002 International Symposium on*, pp. 141-145, Aug. 2002.
- [5] B.S. Reddy and B.N. Chatterji, "An fft-based technique for translation, rotation, scale-invariant image registration," *IEEE Transactions on Image Processing*, vol. 5, no. 8, pp. 1266-1271, Aug. 1996.
- [6] William H.Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery, *Numerical Recipes Electronic Edition Second Edition*, Cambridge, Chapter 15, 1992.

- [7] Leon J. Gleser and David S. Moore, "The Effect of Dependence on Chi-Squared and Empiric distribution Tests of Fit," *The Institute of Mathematical Statistics*, vol. 11, no. 4, pp. 1100-1108, Dec. 1983.
- [8] Andreas Krutz, Michael Frater, and Thomas Sikora, "Window-Based Image registration using variable window sizes," *IEEE International Conference on Image Processing*, pp. 369-372, Sep. 2007.
- [9] Lisa Gottesfeld Brown, "A survey of image registration techniques," *Columbia University*, Jan. 1992
- [10]Tomas C. Henderson, E. Triendl, and R. Winter, "Edge-Based Image Registration," *Proc 2nd Scandinavian Conference on Image Analysis*, pp. 106-111, Jun. 1981.
- [11]Richard Hartley and Andrew Zisserman, *Multiple View Geometry in computer vision 2nd edition*, Cambridge, Chapter 4, 2003.
- [12]Edward Rosten and Tom Drummond, "Machine learning for high-speed corner detection," *European Conference on Computer Vision*, vol. 1, pp. 430-443, May 2006.
- [13]Edward Rosten and Tom Drummond, "Fusing points and lines for high performance tracking," *IEEE International Conference on Computer Vision*, vol. 2, pp. 1508-1511, Oct. 2005.
- [14]C. K. Cheung and L. M. Po, "Normalized partial distortion search algorithm for block motion estimation," *IEEE Transactions on Circuits System for Video Technology*, vol. 10, no. 3, pp. 417-422, Apr. 2000.
- [15] J. N. Kim, S. C. Byun, Y. H. Kim, and B. H. Ahn "Fast Full Search Motion Estimation Algorithm Using Early Detection of Impossible Candidate Vectors," *IEEE Transactions on Signal Processing*, vol. 50, no. 9, pp. 2355-2365, Sep. 2002.