International Journal of Multimedia and Ubiquitous Engineering
Vol.11, No.12 (2016), pp.265-284
http://dx.doi.org/10.14257/ijmue.2016.11.12.25

Matching Points Filtering Applied Panorama Image Processing Using the SURF and RANSAC Algorithm

Daewon Kim

Department of Applied Computer Engineering
Dankook University, 152 Jukjeon-Ro, Yongin,
Kyunggi-Do, Republic of Korea, 16890
E-mail: drdwkim@dku.edu

Abstract

Techniques of generating a single panoramic image by using multiple images are being widely studied in a number of areas, such as computer vision and computer graphics. Generating a panoramic image is a good way of overcoming the limitations of the images obtained from one single camera (e.g., those of picture angles, resolutions, information amounts, etc.) and may be applied in a variety of fields such as virtual reality and robot vision that require wide-angle images. A panoramic image has a great significance in that it can provide a greater sensation of immersion compared to a single image. Currently, there are a variety of techniques of producing panoramic images, but most of them commonly use a method of detecting feature points and matching points in each of the panoramic images they generate. In addition, they use the method of converting images after obtaining homography matrix using the RANSAC (Random Sample Consensus) algorithm that uses matching points. The SURF (Speeded Up Robust Features) algorithm used in this study utilizes the black-and-white and local space information of images when detecting their feature points and is widely used because it provides an outstanding performance in detecting the viewpoints and the changes of the image sizes and is faster than SIFT (Scale Invariant Features Transform) algorithm. However, the SURF algorithm also has its weak point of detecting wrong matching points, which may slow down the performance speed of the RANSAC algorithm and thus increases CPU usage occupation rates. The errors in detecting matching points serve as essential elements of lowering the accuracy and resolutions of panoramic images. In order to minimize these errors, this paper went through an intermediate filtering process of removing wrong matching points using the RGB values of 3×3 region around their coordinates and then presented analysis and evaluation results related to improvements in panoramic image construction & processing and CPU usage occupation rates and the decreasing rates and accuracy of the extracted matching points.

Keywords*: panorama, SURF, RANSAC, matching points, feature points, filtering*

1. Introduction

A panoramic image refers to an image that is produced by stitching and synthesizing separate images shot in various directions around the user so that all the surrounding directions from one viewpoint can be seen. The techniques of making panoramic images using these multiple images are being studied in various fields such as computer visions and computer graphics, *etc.* [1-2]. The panoramic imaging techniques are a good way of overcoming the limitations of images that are obtained from one single camera (*e.g.*, those of angle of view, resolution, information amount, *etc.*) and may be applied in a variety of fields such as virtual reality and robot vision that require wide-angle images. In this regard, studies are being performed to extract feature points, identify matching points [3-4], and then synthetically stitch images [5]. A panoramic image has a great advantage of

ISSN: 1975-0080 IJMUE

providing a greater sensation of immersion compared to a single image, [6] and various program settings and algorithms are employed to produce it [7]. A common process in these programs of generating panoramic images is composed of finding out feature points of more than two input images and then by referring to them, identifying the points at which the images match with each other. After that, the images are put together using these matching points [3-7]. However, the common processes of producing such panoramic images are shown to have several limitations. First, the sequence of the image input is a matter of concern. For example, there is a discomfort in previous methods in making a panoramic image using two images: the image that is placed relatively more to the right side or the lower side of the other one should be input first. Second, if there is an error in detecting matching points using the feature points of the images, the calculation of a homography matrix using the RANSAC algorithm will be delayed and the accuracy of stitching the images will be reduced. This paper presents the methods to solve the above-mentioned limitations that arise when panoramic images are generated by using SURF and RANSAC algorithm by means of filtering the matching points and the results of the analysis of the performance improvements. In this study, chapter II introduces panoramic images and SURF and RANSAC algorithm; chapter III gives a description of the solutions suggested by this study; chapter IV discusses the experiment carried out to test the performance of the improved algorithm proposed by this study; and chapter V summarizes the overall results of the experiment by presenting, analyzing, and evaluating the contents of the resolution of the limitations and the improvements in the reduction rate of the matching points, performance speed, CPU usage occupation rate, and accuracy. In the section of the conclusion of this study, the directions of the follow-up studies are discussed.

2. Panoramic Image Processing Techniques

2.1. Panoramic Images

Generating panoramic images is a good way of overcoming the limitations of the images that can be obtained from one single camera and they are applied in various fields that require wide-angle images. Various program settings and algorithms are used to produce panoramic images. Figure 1 shows a process of producing general panoramic images in which various algorithms are used [8].

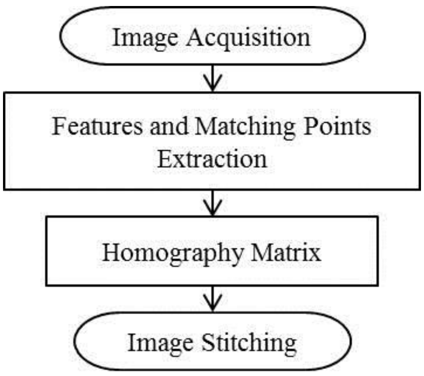


Figure 1. Flowchart of Generation of a Panoramic Image

First of all, more than two images should be obtained to make a panoramic image. After that, the matching points are extracted using the feature points that are extracted from each of the images. In this process, the Harris Corner detector or the Hessian detector or the Fast-Hessian detector is used. However, the above-mentioned algorithms have their weak point of not being robust in the image sizes and the changes of the viewpoints. [9-12]. SIFT (Scale Invariant Features Transform) [13-14] and SURF [15] algorithm are the representative algorithms that are robust in this problem, and the latter was used in this study to extract the feature points and matching points of the images. After that, the extracted matching points were used to convert the image; the RANSAC algorithm was performed to calculate homography matrix [16]; the process of stitching the two images using this matrix was initiated [17-18]. The existing process of producing general panoramic images has several limitations, however. First, the sequence of the image input should be kept suitable for the relevant application programs in the course of obtaining panoramic images. Second, when feature points are used to extract the matching points, some cases arise where several inaccurate ones should be calculated.

2.2. SURF (Speeded Up Robust Features) Algorithm

As mentioned earlier, SURF algorithm is a typical algorithm used to extract feature points [19] of images and largely divided into two categories: expressing and detecting feature points.

$$H_{\approx} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{1}$$

In the process of detecting feature points, the determinant of the approximated Hessian matrix shown in formula (1) is used to determine whether the edge exists. In formula (1), ***D*** which constructs the approximated Hessian matrix means the approximation formula of the second order partial derivative of the Gaussian kernel, and each of the approximation formulas is convoluted in the form of the filter as shown in Figure 2.

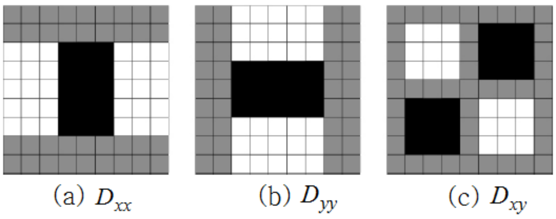


Figure 2. Filters of Approximated Hessian Matrix

As shown in Figure 3, a constant increase of the scales of the filters allows corners on various scales to be extracted. After that, directions of the feature points are calculated using the pixel values around their locations, when the sizes and directions of the slopes, which are outcomes of the convolution with the Haar wavelet filter, are used. The calculated slopes correspond to the colored squares shown in Figure 4 and generate multiple vectors after adding up the sizes within the window with the sliding window in the fan shape with the location of the feature points as its center revolving 360 degrees. The direction of the largest among these vectors is the direction of feature points. The feature points whose directions are calculated in this manner are finally constructed as the descriptor in the form of a vector and come to express themselves [20].

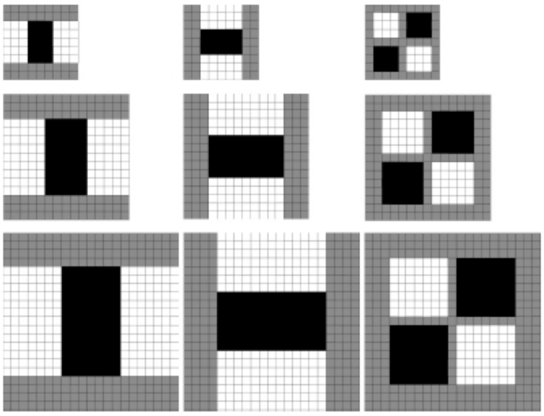


Figure 3. Hessian Matrix Filters with Different Scales (D_{xx}, D_{yy}, D_{xy})

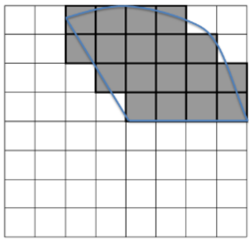


Figure 4. Calculation of the Orientation of Featuring Points Using the Gradient and Sliding Window

This study extracted the feature points of the multiple images based on the SURF algorithm and the extracted appropriate matching points between the two images.

2.3. RANSAC (Random Sample Consensus) Algorithm

The RANSAC algorithm proposed by Fischler [21] is a method to predict an appropriate model parameter out of the original data with heavy measurement noise and generates the homography matrix through these data [22-23]. Based on this theory, the RANSAC algorithm calculates the optimal matching points through the repeated process of randomly making the sampling of the minimum number of the matching points that are essential for determining the homography matrix from all their candidates. The RANSAC algorithm is performed by going through the following steps:

① : N samples of matching points are obtained from all their candidates

② : Homography matrix is predicted assuming that the obtained sample matching points are true values.

③ : Whether the predicted Homography Matrix is correct or not is determined

④ : If the values are not true, the process from ① to ③ is repeated.

N in the step ① corresponds to the number (n=4) of the matching points needed to produce the homography matrix for constructing a panoramic image. In other words, at least 4 matching points are needed to construct a panoramic image, but the problem arises that the frequency of the steps from ① to ④ being repeated will be increased if there are too many wrong matching points, or if the number of the matching points itself is too large, which will be addressed through the filtering of the matching points described in the next chapter.

3. Extracting and Filtering the Matching Points

This study aims to present solutions to the problems related to the limitations that may occur when reconstructing the multiple images obtained from the cameras installed in various directions into a single panoramic image. The process of adding the method proposed by this study to the producing process of a general panoramic image of a total of 6 steps is shown in Figure 5.

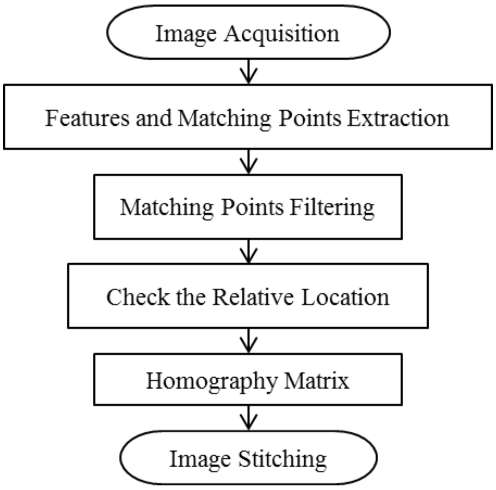


Figure 5. Flowchart of Generation of a Panoramic Image Using the Matching Points Filtering

First, image acquisition from cameras or other image input devices is made and then feature and matching points are identified using the SURF algorithm. Second, the filtering of the extracted matching points is performed to remove wrong ones. The images go through the process of being converted and stitched after the relative locations of the images are identified by using the matching points filtered in this manner and homography matrix is calculated.

3.1. Solutions to the Limitation of Having to Obey the Sequence of the Image Input

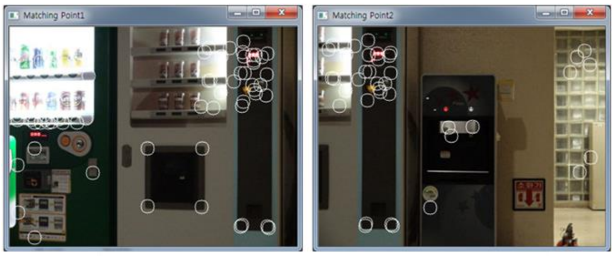
In order to generate a panoramic image, more than two images obtained from the cameras should be input into the relevant application program. In general, the images should be input following the sequence required by the application program in stitching the images. However, as the application programs to construct panoramic images cannot always actively identify the relative locations of the images, the user should differentiate and input the relative locations in person when generating panoramic images by using the images obtained from the cameras. This is why a program is needed so that flexible panoramic images can be generated regardless of whatever sequence of the image input. In this regard, this study supported a function that enables the application programs to identify the relative locations of the images by themselves using the data of the coordinates of the filtered matching points. Overlapping parts are needed for two images to be merged into one, and the locations of the matching points should also be in the overlapping areas. Figure 6 provides a description of the overlapping parts between the two images for two different cases.

(a)

(b)

Figure 6. Areas of Overlapping (Gray-colored) Between the Two Images: (a) Widthwise Case, (B) Lengthwise Case

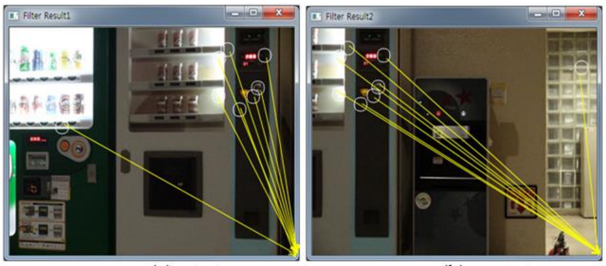
In the Figure 6, (a) and (b) show the examples of the two images constructing a widthwise panoramic image and a lengthwise panoramic image respectively. As shown here, when the two images construct a panoramic image, the two overlapping areas need to exist and the coordinates of the matching points extracted by the filtering process should exist within those areas.



(a) (b)

Figure 7. Matching Points (White circles) without the Filtering Process: (a) Left-side Image, (b) Right-side Image

Figure 7 is the image that shows the location coordinates of the matching points obtained from the extracted feature points using the SURF algorithm to show the reason why the filtering of the matching points is needed. As shown in this figure, there are a considerable number of matching points even outside the overlapping areas. This study, however, randomly used a small number of selected matching points to increase the calculation speed, because it is difficult to make an accurate calculation without filtering the matching points. To address the limit of having to obey the sequence of the image input, coordinates data of the matching points obtained in that manner are used as shown in Figure 8.



(a) (b)

Figure 8. Lines between the Matching Points and Lower-Right Corner: (a) Left-side Image, (b) Right-side Image

First, the location coordinate of each of the matching points are calculated, and then the coordinate of the lower right corner of the images are used to calculate the distance between the two coordinates. Figure 8 (a) is the image that will be placed relatively in the left side when the panoramic image is completed, and (b) is the image that will be placed on the right side. The lines in (a) and (b) in Figure 8 represent the distance from the coordinates of the filtered matching points to the lower right corner of the picture. The relative locations of the images are identified and determined by adding up the distances of these lines of each of the input images. The distances are calculated with the formulas shown from (2) through (6).

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2}$$

$$D_{1n} = \sqrt{(x - x_n)^2 + (y - y_n)^2},\ [Img_1, I_{1n}(x_n, y_n), M_1(x, y)] \tag{3}$$

$$D_{2n} = \sqrt{(x - x_n)^2 + (y - y_n)^2},\ [Img_2, I_{2n}(x_n, y_n), M_2(x, y)] \tag{4}$$

(n = number of matching points)

$$Avg_1 = \frac{D_{11} + D_{12} + \cdots + D_{1(n-1)} + D_{1n}}{n} \tag{5}$$

$$Avg_2 = \frac{D_{21} + D_{22} + \cdots + D_{2(n-1)} + D_{2n}}{n} \tag{6}$$

Equation (2) represents the formula to calculate the distance between the two 2-dimensional points using the Euclidean formula and d refers to the distance value between (x_1,y_1) and (x_2,y_2). Equations (3) and (4) are the formulas to calculate the distance from one of the coordinates of the matching points to the lower right corner of the image. $I_{1n}(x_n,y_n)$ in formula (3) is the coordinate of one of the initially input matching points of the images and $I_{2n}(x_n,y_n)$ in formula (4) is the coordinate of one of the secondly input matching points of the images. n in x_n and y_n represents the number of the matching points and increases from 1 to n during the calculation process. $M_1(x,y)$ represents the coordinate of the lower right corner of the initially input image $M_2(x,y)$ being the coordinate of the right corner of the secondly input image. For instance, if the value of the pixel size of the initially input image is 420×315, $M_1(x,y)$ becomes $M(420,315)$ and this value becomes the eigen value. D_{1n} and D_{2n} in formulas (3) and (4) represent the distance between the random matching points of the initially and secondly input image and the lower right corner of the image. $Avg1$ and $Avg2$ in formulas (5) and (6) mean the averaged distance between the random matching points of the initially input and secondly input image. The sizes of $Avg1$ and $Avg2$ are compared to calculate the relative locations of the images, and the case where $Avg1$ is larger than $Avg2$ leads to the conclusion that the distance between the matching points of the initially input image and the lower right corner of the image is greater than the distance between those and the secondly input image. This means that the overlapping area of the initially input image is concentrated in the left or upper part of the image and that it will be placed relatively to the right or lower part of the panoramic image to be completed. The same holds true for the image to be placed in the lower part of the completed panoramic. On the contrary, if *Avg*1 is smaller than *Avg*2, the first image will be placed in the left or upper part of the panoramic image. Adjusting the locations of the images in their stitching process after identifying them through this method may solve the limit of having to obey the sequence of the image input.

3.2. Matching Points Filtering

Figure 9 gives a description of the necessity of filtering the matching points. In this picture, there are matching points improperly linked to each other, which may constitute a fatal error in constructing a panoramic image.

Figure 9. Lines Connected between the Extracted Matching Points

To address this problem, the feature points extracted by using the SURF algorithm and the data of the coordinates of the matching points initially extracted by using them are used. In addition, all the data of the coordinates of the matching points are separately saved and utilized for each of the images, and whether the matching points are correct is determined by substituting the coordinates of the matching points to the original images and using the RGB values of the surrounding pixel 3×3 mask area. To perform the filtering process, the 3×3 mask is used to calculate the RGB values around the coordinates of the matching point and judge the correlations of the values. For example, if the specific coordinate of one point is (x,y), the 3×3 area can be expressed as the coordinate area in Figure 10.

(x-1, y-1)	(x, y-1)	(x+1, y-1)
(x-1, y)	(x, y)	(x+1, y)
(x-1, y+1)	(x, y+1)	(x+1,y+1)

Figure 10. Coordinates of a Specific Position and its Surrounded Area

In addition, if the values of the RGB pixels are $(x,y)_R$, $(x,y)_G$, and $(x,y)_B$ respectively, the formulas to calculate the values are the expressions in (7), (8), and (9).

$$(x,y)_R = (x-1,y-1)_r + (x,y-1)_r + (x+1,y-1)_r + \cdots + (x,y+1)_r + (x+1,y+1)_r \quad (7)$$

$$(x,y)_G = (x-1,y-1)_g + (x,y-1)_g + (x+1,y-1)_g + \cdots + (x,y+1)_g + (x+1,y+1)_g \quad (8)$$

$$(x,y)_B = (x-1,y-1)_b + (x,y-1)_b + (x+1,y-1)_b + \cdots + (x,y+1)_b + (x+1,y+1)_b \quad (9)$$

The values of the pixels of one particular point within the first image are calculated using the expressions in (7), (8), and (9) and added up into RGB_1 and the result value obtained from the second image is represented as RGB_2. After that, the difference between RGB_1 and RGB_2 is calculated and designated as *Result*, and then a proper error range is designated in consideration of the changes in lighting and difference of the picture angles. This is because the difference in the shooting points and time brings about the difference in lighting and colors, and this difference may cause the problem of removing even the correctly calculated matching points when matching points are filtered with masks.

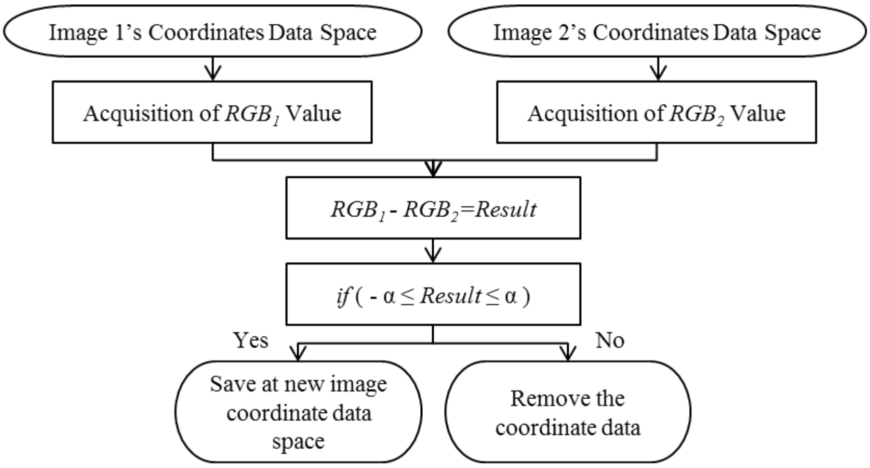


Figure 11. Flowchart for Determining Correct Matching Points through the Additional Filtering Process

Thus, after designating the acceptable error range as ±α, whether the matching points are within the error range of *Result* and thus correct is determined. The flowchart in Figure 11 describes the process of determining whether the matching points are correct or not. Here, if the value of *Result* is within the error range, the matching points are judged to be correct and used to generate the panoramic image; otherwise, they are deleted in the data of the saved coordinates not to be used in the process of performing the RANSAC algorithm. If this process is performed repeatedly on all the matching points, only the correct ones are finally left behind. The RGB calculation by this study with the mask filter of 3×3 size is made to achieve an intuitive calculation, which may have the effect of improving the calculation speed. Various and larger sizes of the filters may rather slow down the overall speed of the calculation. In addition, the problems related to lighting and colors may be minimized by performing the SURF algorithm which is based on black-and-white images and then re-applying the correct coordinates of the matching points to the color images. Then, the correct matching points extracted in this manner are expected to improve the accuracy and speed of constructing panoramic images by minimizing the errors caused by the wrong matching points.

4. Experiment

Two images to construct a panoramic image were obtained from the two SONY HXR-MC50N cameras installed both in parallel and in series and a panoramic image was obtained from the SONY NEX-5n camera to be used in the experiment of measuring MSE (Mean Square Error). OpenCV 2.3.1 was used to create a panoramic image generating program utilizing SURF and RANSAC algorithm. Visual Studio 2010 was used to implement the panoramic image generating program for the experiment and its execution. Through the experiment, data needed for performing the second session of matching points filtering and those of their coordinates were identified.

4.1. Experiment on Solving the Limitation of Having to Obey the Sequence of the Image Input

Different two cases of each of (a) and (b) in Figure 12 show the images to be placed in the right and left side of the panoramic image and the results of the experiment performed in the way introduced in Section 1 of chapter III of this paper.

Figure 12. Results of Two Different Cases for Solving Condition of Image Input Order: (a) Left-side Image, (b) Right-side Image, (c) Output WITH Keeping the Image Input Sequence, (d) Output WITHOUT Keeping the Image Input Sequence

Each of (c) and (d) in Figure 12 show the results of obeying the sequence of the image input and those of not done so. As mentioned earlier, the previous methods had the discomfort of having to identify and input the relative locations in advance when generating panoramic images. From the Figure 12 (c) and (d) we can recognize that the proposed method in this paper works well to generate a panoramic image without keeping the conditions of input image sequence compared to the existing method. In this regard, this study aimed to help solve the problems related to this limitation by introducing a new technique. Generated panoramic images were analyzed by measuring the MSE values and the results are discussed in chapter V.

4.2. Matching Points Filtering

The images in Figure 13 show the two different cases of locations of the matching points obtained through the SURF algorithm without going through the erroneous matching points filtering and the pictures that linked them.

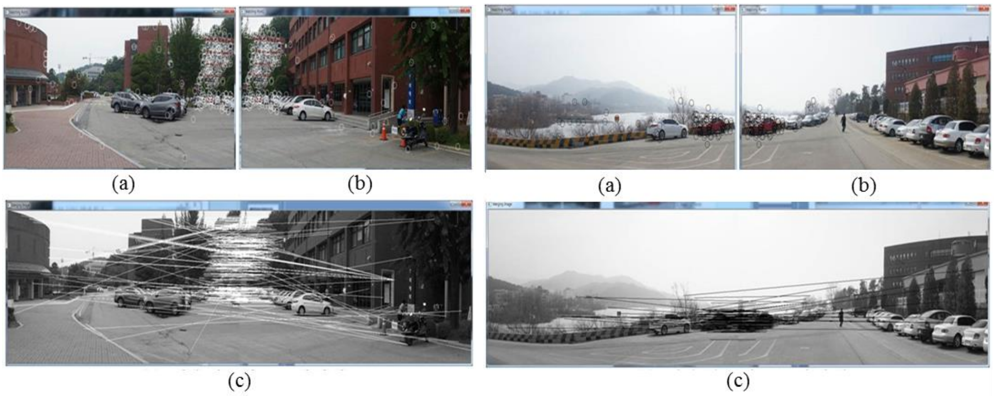


Figure 13. Results of Two Cases for Matching Points and Connected Lines before Error Filtering: (a) Left-side Image, (b) Right-side Image, (c) Lines with Matching Points before Filtering

To produce a panoramic image by minimizing the errors, the so-called overlapping area where the two images on the right and left side are overlapping should appear in the area marked in Figure 14, where most of the matching points exist.

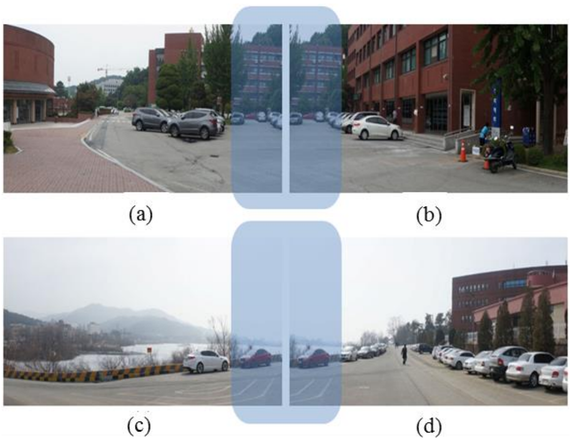


Figure 14. Overlapped Area (Gray-colored) of Two Images: (a) Figure 13 Left-(a), (b) Figure 13 Left-(b), (c) Figure 13 Right-(a), (d) Figure 13 Right-(b)

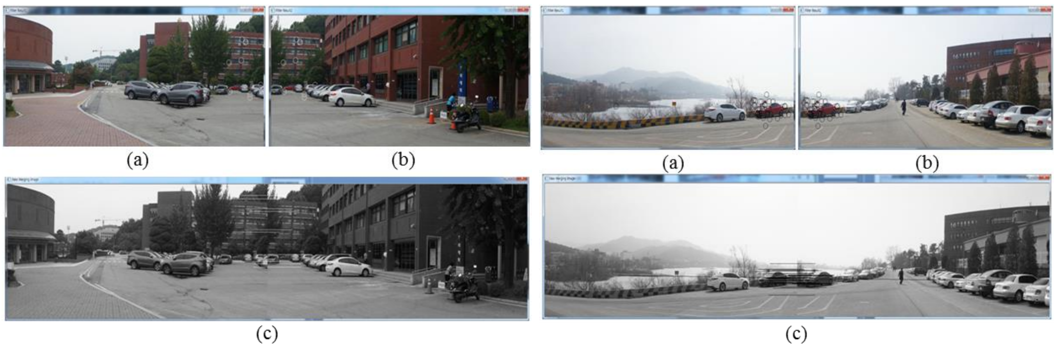


Figure 15. Results of Two Cases after the Matching Points Filtering: (a) Left Image, (b) Right Image, (c) Panorama Image

We can see that the erroneous matching points, as well as the image overlapping area, exist in each of (a) and (b) in Figure 13. Therefore filtering was performed to remove these erroneous matching points, and the results were shown in Figure 15. Erroneous matching points filtering was performed in the way described in chapter III, and these pictures show that most of the matching points exist within the overlapping area and that there are almost no matching points pointing to different portions. The completeness of the panoramic image produced after erroneous matching points are filtered in this manner can be analyzed through the MSE value comparison with the original images. The performance analysis and accuracy judgement for this are discussed in the next chapter.

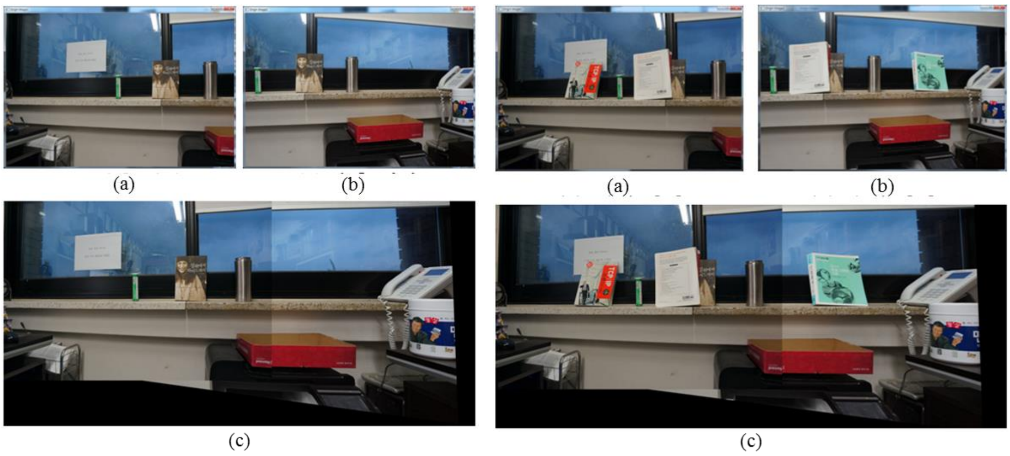


Figure 16. Results of Two cases after the Matching Points Filtering with Respect to the Image's Complexity: (a) Left Image, (b) Right Image, (c) Panorama Image

Figure 16 and 17 show the different forms of images used in the experiment and the results. Figure 16 shows the most simple form of the three images and the moderate complexity and Figure 17 shows the results of the image processing using the most complex images.

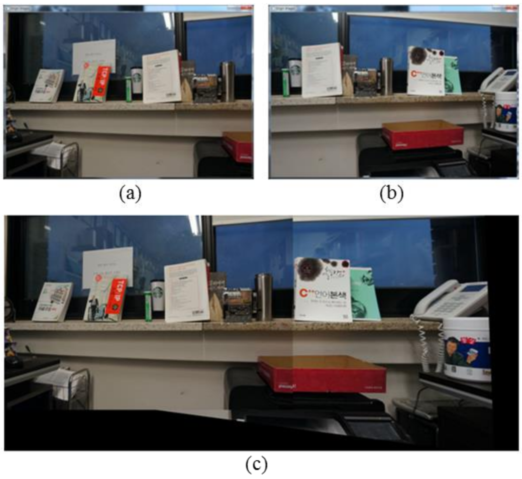
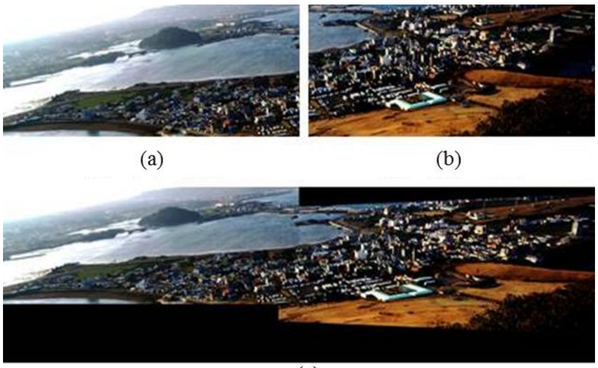


Figure 17. Result after the Matching Points Filtering with the Most Complex Images: (a) Left Image, (b) Right Image, (c) Panorama Image

To test the robustness of the method suggested by this study, the shooting angle of the camera was intentionally distorted when obtaining the images constructing the right and left image in Figure 16 and 17. As shown in each of (a) and (b) in Figure 16 and 17, the horizontal angle of the window shelf where the books or the stainless water bottle were placed is a little twisted, which means some difficulty was expected in the panoramic image processing. However, this was intended to test the functions and effectiveness of the method proposed in this research. The (c) respectively in Figure 16 and 17 shows the results of the panoramic image processing through the shelf of the horizontal window frame and the seam side of the tip lines of the curtain in the upper part of the picture. The parts which were supposed to be connected with straight lines are shown to be twistedly connected. The analysis using the angles is discussed in the sections of result analysis of and discussion in the next chapter. Lastly, Figure 18 shows the result of producing the panoramic image using the images shot with the camera not parallel with the ground.



(a) (b) (c)

Figure 18. Result after the Matching Points Filtering of an Image that is Out of the Horizontal with the Ground: (a) Left Image, (b) Right Image, (c) Panorama Image

The photographs of landscapes shot in a high place were used, and since they were not parallel with the ground and the angles of the right and left side were twisted, the black empty spaces were found to be generated at the top and bottom of the image produced by using them. Through this process, the panoramic image generating algorithm using the techniques following the matching points filtering was found to properly work even for the uneven input images. Table 1 shows the change amount of the pre- and post-filtering matching points extracted from the images presented in the results in Figure 15 to 18.

Table 1. Changing Rate of the Image's Matching Points presented in Figure 15 through Figure 18 (a: Left Case of Figure 15, b: Right Case of Figure 15, c: Left Case of Figure 16, d: Right Case of Figure 16)

Changing Rate of the Number of Matching Points				
Image	Size	Before Filtering	After Filtering	Reduction Rate (%)
A	1100×453	304	10	96.71
B	1100×453	106	41	61.32
C	1000×664	372	17	95.43
D	1000×664	326	111	65.95
Figure 17	1000×664	398	25	93.71
Figure 18	1100×450	41	12	70.70

Left case of Figure 15 (A) shows a 96.7% reduction rate as the number of the matching points was reduced from 304 to 10 after their filtering; and the reduction rate in C, D, Figure 17, and 18 was 95.4%, 65.9%, 93.7%, and 70.7% respectively with the averaged reduction rate of 80.6%. The number of matching points in the process of generating a panoramic image greatly affects the performance speed of the algorithm, CPU occupation rate, and the accuracy of the generated image, *etc.* In the next chapter, analysis and discussion on the experiment results are given.

5. Discussion and Analysis of the Results

In this section, the solutions are presented to solve the limitation of having to obey the sequence of the image input order in constructing the panoramic images. In addition, the measurement is made of reduction rate of the matching points of various sized images, performance duration, CPU occupation rate, and the accuracy using the angles of the partial images included in the resultant images.

5.1. Results of the Experiment on Solving the Limitation of Having to Obey the Sequence of the Image Input

The experiment has been performed to determine the right and left locations using the two images in each of (a) and (b) in the left-case of Figure 12. According to the results, the initial input image would be placed on the left side since averaged distance between each of the matching points of the initial input image after filtering and the right lower corner of the image was 330 while the averaged distance in the case of the second input image was 1,037. Therefore, it is possible to have accuracy in aligning the images either horizontally or vertically when constructing a panoramic image using these results.

5.2. Results of Matching Points Extraction and Filtering

5.2.1. Evaluation and Analysis of the Reduction Rate of Matching Points

Table 2 shows the filtering results of the matching points of all the images presented in Figure 15 to 18; A and B show the filtering results of the images with the pixel sizes at 100, 2750, and 5500 respectively, with C, D and Figure 17 at 1000, 2500, and 4000 respectively and Figure 18 at 1100, 2750, and 4400 respectively.

Table 2. Changing Rate of the Matching Points (a: Left Case of Figure 15, b: Right Case of Figure 15, c: Left Case of Figure 16, d: Right Case of Figure 16)

Changing Rate of Matching Points							
Image			Number of Feature Points		Number of Matching Points		Reduction Rate (%)
			Left	Right	Before Filtering	After Filtering	
A	Size	1100×453	1284	1468	304	10	96.7
		2750×1133	6521	6450	1382	30	97.8
		5500×1813	14631	13538	3471	3197	7.8
B	Size	1100×453	835	1254	106	41	61.3
		2750×1133	4177	5636	397	11	97.2
		5500×1813	11077	13741	4704	170	96.3
C	Size	1000×664	1050	1165	372	17	95.4
		2500×1661	3656	4045	1124	6	99.4

		4000×2658	6728	7642	1994	51	99.9
D	Size	1000×664	1161	1256	326	111	65.9
		2500×1661	4203	4918	1039	328	68.4
		4000×2658	8781	9960	1956	28	98.5
Figure 17	Size	1000×664	1449	1518	398	25	93.7
		2500×1661	5198	5949	1232	37	96.9
		4000×2658	10524	11924	2378	6	99.7
Figure 18	Size	1100×450	643	1006	41	12	70.7
		2750×1126	2931	4850	167	16	90.4
		4400×1800	5592	9446	334	22	93.4

Feature points were extracted by using the SURF algorithm from each of the images and their number was expressed at that time. The larger number of the feature points means that the images have more features, which is related to their complexity. In other words, the larger number of the feature points means that the images are more complex. The images in A and B showed an averaged 76.1% of the reduction rate of the matching points, with those in C, D and Figure 17 an averaged 90.8%, and those in Figure 18 an averaged 84.8%, which indicated the results of the erroneous matching points filtering were very satisfactory.

5.2.2. Evaluation and Analysis of the Performance Speed of the RANSAC Algorithm

Table 3 shows the results of measuring the performance speed of the RANSAC algorithm by comparing the case where the erroneous matching points were filtered with the case where they were not.

Table 3. Comparison of RANSAC Performance Speed (a: Left Case of Figure 15, b: Right Case of Figure 15, c: Left Case of Figure 16, d: Right Case of Figure 16)

Comparison of Performance Speed for Filtering and RANSAC Algorithm (10^{-2} *sec.*)						
Image	Size	Filtering	RANSAC	Total	RANSAC Only	Reduction Rate (%)
A	1100×453	0.006	0.139	0.146	0.264	44.7
	2750×1133	0.034	0.137	0.172	0.661	73.9
	5500×1813	0.161	0.840	1.002	1.228	18.3
B	1100×453	0.002	0.143	0.145	0.280	48.1
	2750×1133	0.009	0.136	0.146	0.434	66.2
	5500×1813	0.176	0.192	0.368	1.334	72.3
C	1000×664	0.008	0.270	0.278	0.714	60.1
	2500×1661	0.028	0.117	0.145	1.722	91.5
	4000×2658	0.051	0.785	0.83	3.207	73.9

				7		
D	1000×664	0.007	1.131	1.139	1.619	29.6
	2500×1661	0.027	1.354	1.381	2.019	31.5
	4000×2658	0.084	1.364	1.448	3.104	53.3
Figure 17	1000×664	0.009	0.329	0.338	1.012	66.5
	2500×1661	0.031	0.960	0.991	2.076	52.2
	4000×2658	0.129	0.263	0.393	4.04	90.2
Figure 18	1100×450	0.098	0.368	0.369	6.488	94.3
	2750×1126	0.478	0.387	0.392	1.519	74.1
	4400×1800	1.088	0.357	0.368	1.082	65.9

The performance speed of each of the images used in this experiment showed that it took a shorter time to perform the RANSAC algorithm when the filtered matching points were used. The time it took to perform the erroneous matching points filtering proposed by this study was measured as part of the whole time it took to perform the RANSAC algorithm. A and B show an averaged 53.9% reduction of the performance speed of the RANSAC algorithm, with C, D and Figure 17 an averaged 60.9% and Figure 18 an averaged 78.1% respectively. It can be said that the results were encouraging given the fact that the reduction of the speed of the entire performance was made with the erroneous matching points filtering.

5.2.3. Evaluation and Analysis of CPU Occupation Rate Reduction

Table 4 shows the performance occupation rate of CPU, one of the computer central processing units in the two cases: one in which the RANSAC algorithm including the erroneous matching points filtering was performed and the other in which only the RANSAC algorithm was performed.

Table 4. CPU Occupation Rate of RANSAC Algorithm (a: Left Case of Figure 15, b: Right Case of Figure 15, c: Left Case of Figure 16, d: Right Case of Figure 16)

CPU Occupation Rate (%)					
Image			RANSAC W/ Filtering	RANSAC Only	Reduction Rate (%)
A	Size	1100×453	0.086	0.126	31.7
		2750×1133	0.014	0.032	56.2
		5500×1813	0.004	0.01	60.0
B	Size	1100×453	0.1	0.116	13.7
		2750×1133	0.02	0.026	23.0
		5500×1813	0.004	0.018	77.7
C	Size	1000×664	0.162	0.198	18.1
		2500×1661	0.01	0.076	86.8
		4000×2658	0.022	0.074	70.2

D	Size	1000×664	0.128	0.226	43.3
		2500×1661	0.044	0.072	38.8
		4000×2658	0.016	0.038	57.8
Figure 17	Size	1000×664	0.08	0.176	54.5
		2500×1661	0.034	0.068	50
		4000×2658	0.012	0.062	80.6
Figure 18	Size	1100×450	0.07	0.17	58.8
		2750×1126	0.044	0.075	41
		4400×1800	0.01	0.06	83.3

These results show that the averaged CPU occupation rate fell by a significant amount (52.5%) when the RANSAC algorithm including the erroneous matching points filtering was performed. It can be seen that the data processing load of CPU was also reduced when the panoramic image processing was performed by using the techniques presented in this research.

5.2.4. Evaluation and Analysis of Accuracy

Table 5 shows the results of measuring the accuracy by using the MSE (Mean Square Error) value to determine the difference between the panoramic images with existing methods and the resulting image that was reconstructed with the method suggested by this study.

Table 5. Comparison of MSE for the Two Cases in Figure 15 (a: Left Case of Figure 15, b: Right Case of Figure 15)

MSE				
Image			Filtering Included	RANSAC Only
A	Size	1100×453	9.256841	11.31281
		2750×1133	0.635182	0.843366
		5500×1813	0.562384	0.827647
B	Size	1100×453	4.256398	6.459141
		2750×1133	0.452156	0.559321
		5500×1813	0.468523	0.537658

In order to measure the MSE value, one image was first generated by performing the panoramic image processing including the erroneous matching points filtering while the other one was produced by using the existing method with no filtering, and then the difference was calculated between each of the generated images and the original panoramic image. The comparative analysis of the MSE values in the two cases found that the proposed method in this paper produced moderately superior results than that of existing methods from the view point of error measurements.

(a) (b)

Figure 19. Measurement and Comparison of Accuracy: (a) WITH the Filtering, (b) WITHOUT the Filtering

Figure 19 shows a part of the panoramic image finally generated after the RANSAC algorithm was performed by using the two images on the left and right side. Homography matrix value was calculated to measure the accuracy of the experiment in the case of Figure 16, 17 that had no original panoramic image. With the lines in the images on the left and right side constructing the panoramic image, the two angles of the borderlines were supposed to have little difference with almost 0º, but the result was otherwise, which is attributed to the fact that the calculated matching points and the homography matrix were both incorrect. In other words, it means that if the upper linking part of the white curtain in Figure 19 looks straight connected, its angle is likely become 0º, which in turn increases the accuracy of the constructed panoramic image. Table 6 shows the results of measuring, comparing, and analyzing the accuracy of the panoramic images generated by applying the lines shown in Figure 19 to those in Figure 16 and 17.

Table 6. Measurement and Comparison of Accuracy for Figure 16 and Figure 17 (c: Left Case of Figure 16, d: Right Case of Figure 16)

Image			Accuracy (º: degree)		Concordance Rate (%)	
			Filtering Included	RANSAC Only	Filtering Included	RANSAC Only
C	Size	1000×664	3	4	94.6	92.8
		2500×1661	3	4	94.6	92.8
		4000×2658	2.5	4	95.5	92.8
D	Size	1000×664	3	4	94.6	92.8
		2500×1661	2.5	4	95.5	92.8
		4000×2658	3	3.5	94.6	93.7
Figure 17	Size	1000×664	2.5	3	95.5	94.6
		2500×1661	2	3	96.4	94.6
		4000×2658	3	4	94.6	92.8

In Table 6, the two results were presented and compared: one for the case where the experiment included the erroneous matching points filtering, and the other for the case where it did not. In the experiment with the filtering included, the averaged value of the accuracy was 2.70º for the angle difference and 95.1% for the concordance rate, while in the experiment with the filtering excluded, the averaged value of the accuracy was 3.70º for the angular difference and 93.3% for the concordance rate. The results of the experiment showed that there was a higher accuracy when the panoramic images were generated by using the filtered matching points than they were not.

6. Conclusions

A panoramic image is a good way of overcoming the limitation of images obtained from a single camera and applied to various areas that require wide-angle images. Various program settings and algorithms are used in producing panoramic images in a limited environment, and among them the SURF algorithm to extract the feature points and the RANSAC algorithm to calculate the homography matrix for an image conversion are routinely used. In this paper, the method of overcoming some limitations imposed by the matching points left behind after the feature points were extracted through filtering the erroneous matching points when panoramic images were generated by using the SURF and RANSAC algorithm, and the matching points unnecessary for constructing panoramic images were removed. The followings are summary of the improvement of the performance and the solution to the limitations: First, the matching of the erroneous matching points was prevented through their filtering. Second, the reduced number of the matching points through their filtering improved the overall performance speed of the panoramic image processing and remarkably dropped the CPU occupation rate. Finally, the limitation of having to obey the sequence of inputting images at top and bottom or on the right and left side was resolved, and the angular errors in the matching parts were compared and analyzed using the straight lines within the images to measure the accuracy of the generated images. In addition, the difference with the original image was determined by calculating the MSE, and the completeness of the generated image was measured. The techniques for solving the limitation of having to obey the sequence of the image input and the method of filtering the erroneous matching points described in this study may contribute to the performance improvement in the fields that need to generate real-time high definition images and applied to various computer visions and graphics. We plan to perform further studies on generating massive panoramic images that judge the right and left and top and bottom side by themselves using a good quantity of images.

Acknowledgments

This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the CPRC(Communication Policy Research Center) support program (IITP-2016-H8201-16-1005) supervised by the IITP(Institute for Information & communications Technology Promotion)

References

[1] D. H. Kim and J. S. Choi, “View Interpolation Algorithm for Continuously Changing Viewpoints in the Multi-panorama Based Navigation”, IEIE Journal (SP), vol. 40, no. 6, **(2003)**, pp. 141-148.

[2] S. H. Kim, K. Y. Kim and W. T. Woo, “Multiple Camera Calibration for Panoramic 3D Virtual Environment”, IEIE Journal (CI), vol. 41, no. 2, **(2004)**, pp. 137-148.

[3] S. H. Rhee, J. H. Lee, S. G. Lee and M. G. Kang, “Panoramic Image Reconstruction Based on a Camera Motion Model”, KOSBE Journal, **(1999)**, pp. 41-46.

[4] J. S. Joe, B. J. Lee and J. S. Park, “Creating Full View Panorama Image from Multiple Images”, KIISE Journal, vol. 34, no. 2, **(2007)**, pp. 162-166.

[5] Z. Xiong, M. Zhang, Y. Wang, T. Li and K. Li, “Fast Panorama Unrolling of Catadioptric Omni-Directional Images for Cooperative Robot Vision”, Proceedings of the 11th International Conference on Computer Supported Cooperative Work in Design, DOI 10.1109/SCCWD.4281594, **(2007)**, pp. 1100-1104.

[6] S. Kim and W. Woo, “Panoramic Image-Based Virtual Environment Using the 3D Generation”, IPIU, vol. 13, **(2003)**, pp. 111-116.

[7] Y. Seungbeom, J. HyeokJun and H. Chijeion, “Features-Based Panoramic Image Combinations”, KICS Journal, vol. 35, no. 8, **(2010)**, pp. 179-184.

[8] S. Richard, "Image Alignment and Stitching: A Tutorial", Journal of Foundations and Trends in Computer Graphics and Vision, vol. 2, no. 1, **(2006)**.
[9] J. M. Morel and G. Yu, "ASIFT: A New Framework for Fully Affine Invariant Comparison", SIAM Journal on Imaging Sciences, vol. 2, no. 2, **(2009)**.
[10] K. Mikolajczyk and C. Schmid, "Scale and Affine Invariant Interest Point Detectors", International Journal of Computer Vision, vol. 60, no. 1, **(2004)**, pp. 63-86.
[11] J. Matas, O. Chum, M. Urban and T. Pajdla, "Robust Wide Baseline Stereo From Maximally Stable Extremal Regions", BMVC, **(2002)**, pp. 384-393.
[12] T. Tuytelaars and K. Mikolajczyk, "Local Invariant Feature Detectors", Journal of Foundations and Trends in Computer Graphics and Vision, vol. 3, no. 3, **(2008)**, pp. 177-280.
[13] M. Brown and D. Lowe, "Automatic Panoramic Image Stitching using Invariant Features", International Journal of Computer Vision, vol. 74, no. 1, **(2007)**, pp. 59-73.
[14] C. Park, S. Lee and Y. Jeong, "A Hardware Design of Feature Detector for Real-time Processing of SIFT (Scale Invariant Feature Transform) Algorithm in Embedded Systems", IEIE Journal (SD), vol. 46, no. 3, **(2009)**, pp. 86-95.
[15] H. Bay, T. Tuytelaars and L. V. Gool, "SURF: Speeded Up Robust Features", Computer Vision of Lecture Notes in Computer Science, vol. 3951, **(2006)**, pp. 404-417.
[16] S. Seo, S. Jeong, Y. Han, J. Choi and S. Lee, "Efficient Homography Estimation for Panoramic Image Generation", vol. 50, no. 8, **(2013)**, pp. 215-224.
[17] L. Juan, S. Shin, H. Park and O. B. Gwun, "Stitching for Panorama Based on SURF and Multi-Band Blending", KMMS Journal, vol. 14, no. 2, **(2011)**, pp. 201-209.
[18] T. Kim, Y. Choi, B. Nam and H. Hong, "Tele-Presence System using Homography-based Camera Tracking Method", IEIE Journal (CI), vol. 49, no. 3, **(2012)**, pp. 27-33.
[19] J. Im and J. Paik, "Feature-Based Panoramic Background Generation for Object Tracking in Dynamic Video", IEIE Journal (SP), vol. 45, no. 6, **(2008)**, pp. 108-116.
[20] K. MinGoo, C. Wongook and M. Seounbin, "SURF Feature Extraction Algorithm for Face Recognition Research", IEIE Journal (CI), vol. 48, no. 3, **(2011)**, pp. 46-53.
[21] M. Fischler and R. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting Applications to Image Analysis and Automated Cartography", Proceedings of the Image Understanding Workshop, **(1980)**, pp. 71-88.
[22] J. Son, M. Kang and K. Sohn, "View Invariant Image Matching using SURF", KOSBE Conference, **(2011)**, pp. 222-225.
[23] O. Chum, T. Pajdla and P. Sturm, "The Geometric Error for Homographies", Computer Vision and Image Understanding, vol. 97, no. 1, **(2005)**, pp. 86-102.

Author

Daewon Kim, received a M.S. (1996) from the University of Southern California, LA, California, USA, and a Ph. D. (2002) in Electrical and Computer Engineering from Iowa State University, Ames, Iowa, USA. He worked as a senior researcher at Samsung Electronics Co. Ltd., Suwon, South Korea (2002-2004). He is currently a professor in Applied Computer Engineering department at Dankook University, South Korea. His research interests include signal processing, mobile applications, and nondestructive evaluation.