논문 14-39C-10-08 The Journal Of Korea Information And Communications Society '14-10 Vol.39C No.10
Http://Dx.Doi.Org/10.7840/Kics.2014.39C.10.957

Fast Image Stitching For Video Stabilization Using Sift Feature Points

Mostafiz Mehebuba Hossain◆, Hyuk-Jae Lee*, Jaesung Lee°

Abstract

Video Stabilization For Vehicular Applications Is An Important Method Of Removing Unwanted Shaky Motions From Unstable Videos. In This Paper, An Improved Video Stabilization Method With Image Stitching Has Been Proposed. Scale Invariant Feature Transform (Sift) Matching Is Used To Calculate The New Position Of The Points In Next Frame. Image Stitching Is Done In Every Frame To Get Stabilized Frames To Provide Stable Video As Well As A Better Understanding Of The Previous Frame'S Position And Show The Surrounding Objects Together. The Computational Complexity Of Sift (Scale-Invariant Feature Transform) Is Reduced By Reducing The Sift Descriptors Size And Resticting The Number Of Keypints To Be Extracted. Also, A Modified Matching Procedure Is Proposed To Improve The Accuracy Of The Stabilization.

Key Words : Video Stabilization, Sift, Image Stitching, Image Transformation, Ransac

Ⅰ. Introduction

Video stabilization for vehicular applications is one of the most discussed topics for many years. Most of the papers related to video stabilization concentrate only on shaky videos generated from hand-held devices. However this research is mainly focused on videos generated from in-vehicle camera for various event detection during driving so that it can be used in a car app such as Google's Waze. Also, this algorithm can be used to stabilize any kind of unstable video. The videos generated from in-vehicle camera have several motions and they are very uncertain, which can be removed by the proposed algorithm. In addition, as the stabilization is done by stitching frames together, it is easy to know the surroundings of the vehicle. The traditional video stabilization algorithms outputs only the transformed next frame according to the previous frame, but the proposed algorithm will output the stabilized current frame stitched with all other previous frames.

Video Stabilization techniques have been studied in several ways with different pros and cons. In early days, 'block matching' was used to estimate local motion and remove unwanted motions[1-4]. Those algorithms mostly gives good results but sometimes get misled by any moving objects presents in the un-stable video. Because of the moving objects wrong motions are estimated and those wrong estimations results into incorrect wrong stabilized video.

※ This Research Was Supported By The Msip(Ministry Of Science, Ict And Future Planning), Korea, Under The C-Itrc(Convergence Information Technology Research Center) Support Program (NIPA-2014-H0401-14-1007) Supervised By The NIPA(National It Industry Promotion Agency) And Basic Science Research Program Through The National Research Foundation Of Korea(Nrf) Funded By The Ministry Of Education, Science And Technology (No. 2012R1A1A1038515)

◆ First Author : Department Of Electrical Engineering And Computer Science, Seoul National University, Mostafiz@Capp.Snu.Ac.Kr, 학생회원
° Corresponding Author : Korea National University Of Transportation Department Of Electrical Engineering, Jaesung.Lee@Ut.Ac.Kr, 정회원
* Department Of Electrical Engineering And Computer Science, Seoul National University, Hjlee@Capp.Snu.Ac.Kr, 정회원
논문번호 : KICS2014-07-275, Received July 23, 2014; Revised September 15, 2014; Accepted September 15, 2014

Other important techniques are feature based video stabilization. Those approaches mainly have three stages: motion estimation, motion filtering, and image composition. For motion filtering several filters have been used, Kalman filtering[5], Gaussian filtering[6], Motion Vector Integration (MVI) with adaptive damping coefficient[7,8]. However, those algorithms are computationally intensive as the filtering needs to be done in every frame.

The proposed algorithm presents a method which uses image stitching to perform video stabilization.[9] SIFT features are extracted in the current frame and matched with the next frame to know the new position of the same points. A homography matrix with RANSAC algorithm is estimated to determine the translation and rotation of the points more efficiently from current frame to next frame. The SIFT feature points of next frame is transformed according to the homography matrix and then the transformed next frame is stitched with the current frame to get the stabilized frame.

Matching results directly effects on the homography matrix calculation, so it very important to calculate the correct corresponding matches to improve the accuracy of the stabilization. In this paper, matching procedure is also improved by checking the keypoints frame wise displacement in this paper to enhance the quality of the video stabilization. Several methods have been used previously to improve the matching results [13,18].

The computational complexity of the algorithm is reduced by reducing the dimension of the SIFT descriptor and by restricting the numbers of keypoint to be extracted and to be matched. Several research also have been done to reduce the SIFT complexity. In Paper[17], the complexity is reduced by removing the rotation invariance of SIFT assuming the point for image is relatively stable. In [16], another approach of reducing SIFT computation has given, but the result shows , the proposed algorithm has much better result than Zhu Qidan and Li Ke's approach.

The paper is organized in following sections- Sec. II.1: Details of SIFT feature points II.2: Feature points matching, Sec. II.3: Homography matrix calculations, Sec. III: Image Transformations and Stitching for Stabilization, Sec. IV: Matching Improvement Sec. V.1: Modified SIFT descriptor , Sec. V.2: Region-based Keypoints extraction, Sec. VI: Experimental results and Sec. VII: Conclusions from the study. [fig. 1] shows an overflow of the proposed algorithm.

Ⅱ. Previous Works

In this part a brief descriptions of SIFT feature points, feature points matching and homography matrix is given and how they are used in the proposed algorithm, to better understand the whole flow of the algorithm.

2.1 SIFT Feature Points

The Scale Invariant Feature Transform (SIFT)[9] is a method of transforming image data into scale invariant feature points. These keypoints are robust to scale changes, rotation, illumination variations and viewpoint changes. SIFT feature points detects the dominant gradient orientation at every location and records that according to its orientation to the histogram bin. Since SIFT features are invariant of rotation and scale the proposed algorithm can also make stabilized frames with orientation variations [see fig. 8 & fig. 10]

The SIFT keypoint descriptors, which gives unique information about the surroundings is very useful for accurate feature matching. The descriptor is a vector is combination of orientation histograms. A 8 bin histogram are used and a patch around the feature is split into separate 4x4 region each has its own orientation histogram, so the descriptor is a 128 dimensional vector(8x4x4).

The proposed algorithm uses a smaller vector for SIFT descriptor, which consist of 2x2 window around every pixel and a 6 bin histogram to represent the local gradient for each pixel, so the descriptors dimension becomes 24(2x2x6), which makes the algorithm much faster and gives similar matching results with the original 128 dimensional descriptor.

A restriction is applied in the number of the

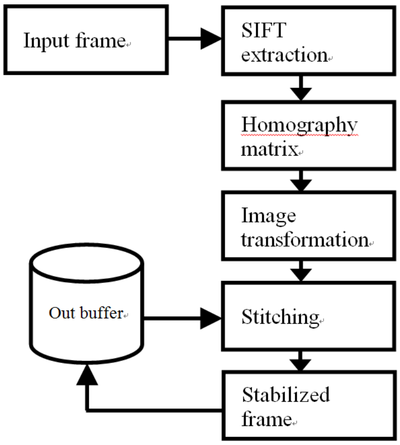

Fig. 1. Algorithm overflow

keypoints to be extract in every frame by 200 to reduce the computational complexity. Experiments have been done with different numbers of keypointsin various types of videos; all of them gave satisfying results with 200 keypoints

2.2 Feature Points matching

Feature points matching needs to be done in order to know the corresponding keypoints in next frame and also to know the new positions of the feature points from current frame to next frame. In the proposed algorithm SIFT feature point's descriptors matching is done by Brute-Force Matcher. The aim is to make the matching results as good as possible. This method gives better result than the ration test proposed by D.Lowe[10]. In this method a Knn(k==1) Matching is done in both the waysi.eKnnMatch (img1, img2) and KnnMatch (img2, img1) and it returns only those pairs (i,j) such that for i-th query descriptor, the j-th descriptor is in the matcher's collection is the nearest for img1 to img2 and vice versa [fig. 2].

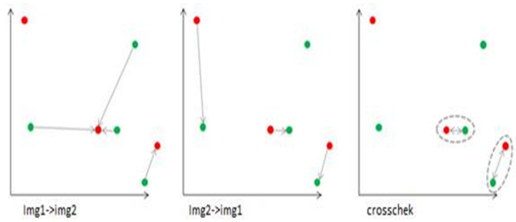

Fig. 2. Brute-Force Matcher with cross checking

2.3 Homography Matrix with RANSAC

Homography is a mathematicalterm for mapping points from one surface to other surface. In computer vision, homography almostalways refers to mapping points between two image planes that correspond to the same location on a planar object in the real world [14]. Here, homography is a 3 by 3 matrix, which can give information's about image translations and rotation. To find such information between current frame and next frame a homography matrix is estimated. If F1 and F2 are points from frame1 and frame2 correspondingly then the homography matrix (H) between them can be represented by-

$$[F1]=H*[F2] \tag{1}$$

If two images are related by a homography, it is possible to transform image points from one image to other, once the matrix is estimated. In order to get correct and stable transformed image, it has to be ensured that the homography matrix is estimated with good matched points only. Otherwise the estimated homography matrix will be wrong, which will affect the transformed image. However, in practical, it is difficult to guarantee that the SIFT point matching with descriptor information will return perfect results. If the frames are not stitched properly, the output stabilized frame will be incorrect. So, to estimate a robust homography matrix RANdomSAmple Consensus (RANSAC) method had been implemented in the proposed algorithm.

The RANSAC algorithm aims at estimating a given mathematical entity from a data set that may contain a number of outliers. The idea is to randomly select some data points from the whole data set and perform the estimation only with the randomly selected data points. The number of randomly selected points should be the minimum number of point's requires estimating the mathematical entry[12]. In case of homography matrix we need minimum 4 pairs of points i.e four SIFT feature points from current frame and 4 feature from the next frame whose descriptors matches. Initially, homography matrix is estimated with those randomly selected 4 matches. All the other matches in the match set are tested against the epipolar constraint that derives from the matrix. The matches which fulfils the constrains forms the support set ('inliers') of the homogrpgy matrix. The larger the support set is the probability of getting the homography matrix exactly increases. If one or more randomly selected matches are wrong, the support set size will decrease and the estimates homography matrix will also be wrong. The aim is to get a large set of inliers. RASAC algorithm is summarized at [fig. 3][15].

a. Select four matches(randomly), b. Compute homography H, c. Keep largest set of inliers, d. Go to step a, e. Re-compute H estimate with all the inliers.

Fig. 3. RANSAC algorithm

Ⅲ. Frames Transformations and Stitching for Stabilization

As mentioned earlier once the homography matrix is estimated the image points from the next_frame can be transformed according to the current_frame's view point. Also, it is possible to transform all the pixels of the next_frame according to the current_frame, even for those pixels which falls outside the current_frame's boundaries. Suppose the next_frame shows a portion of the scene that is not visible in the current_frame, then it is possible to transform those parts of the next_frame according to current_frames view point using the information of homography matrix and the pixels colour value of that part of the image.The whole procedure with some examples are explained step by step from the next paragraph-

Step1 : Keypoints in current_frame(Kpt1) and next_frame(Kpt2) are extracted. Keypoints matching is done [fig. 4]. Say, points in current_frame and next_frame after matching are (Xc,Yc) and (Xn,Yn). A homography matrix H(3x3) is calculated between

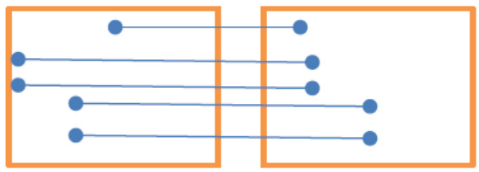

Fig. 4. SIFT matching between current_frame and next_frame

(Xc,Yc) and (Xn,Yn) using equation (1)

Step2 : A new blank output image of size S is created [fig. 5]. At first thecurrent_frame is copied in the (0, 0) location.

Fig. 5. Created initial output image for stabilization

Step3 : Next_frame is transformed [fig. 6] according to the H matrix. Size of the transformed image is same as the output image, S. Pixel value of every pixel in the transformed image can be calculated by the equation (2), where H11, H12, H13, ⋯, H33 are the 9 element of the H matrix.

$$Dest(\Xi, Yi) = Source\left(\frac{H11^{*}Xi + H12^{*}Yi + H13}{H31^{*}Xi + H32^{*}Yi + 1}, \frac{H21^{*}Xi + H22^{*}Yi + H23}{H31^{*}Xi + H32^{*}Yi + 1}\right) \quad (2)$$

For example, the pixel value at (1, 2) point in transformed image can be calculated with the H $\begin{bmatrix} 1 & 1 & 2 \\ 2 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}$. Pixel value at (1,2)= pixel value atnext_frame($\frac{1^{*}1+1^{*}2+2}{0^{*}1+0^{*}2+1}, \frac{2^{*}1+1^{*}2+3}{0^{*}1+0^{*}2+1}$)= pixel value at next_frame(5,6).

Step4 : P1'(X1,Y1), P2'(X2,Y2), P3'(X3,Y3), P4'(X4,Y4) points in the transformed image can be similarly calculated by the following equation(3)-

$$(X', Y') = \begin{bmatrix} H11 & H12 & H13 \\ H21 & H22 & H23 \\ H31 & H32 & H33 \end{bmatrix} * (X, Y) \qquad (3)$$

Where, (X',Y') are the points in transformed image and (X,Y) are the points in next_frame. For example, if the next _frame is of size 640*360 before transformation then the corner points of next_frame P1,P2,P3,P4 are (0,0),(640,0),(640,360),(0,360) accordingly before transformation. Now, if thehomography matrix is H $\begin{bmatrix} 1 & 1 & 2 \\ 2 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}$, then the corner points in the transformed image are-

$$P1'\left(\frac{1*0+1*0+2}{0*0+0*0+1}, \frac{2*0+1*0+3}{0*0+0*0+1}\right) = P1'(2,3),$$

$$P2'\left(\frac{1*640+1*0+2}{0*640+0*0+1}, \frac{2*640+1*0+3}{0*640+0*0+1}\right) = P2'(642,1283),$$

$$P3'\left(\frac{1*640+1*360+2}{0*640+0*360+1}, \frac{2*640+1*360+3}{0*640+0*360+1}\right) = P3'(1002,1643),$$

and

$$P4'\left(\frac{1*0+1*360+2}{0*0+0*360+1}, \frac{2*0+1*360+3}{0*0+0*360+1}\right) = P4'(362,363).$$

Step5 : Now to stitch the transformed next_frame [fig. 6] with the output image [fig. 5], the bounding part of transformed corner points,P1'(2,3), P2'(642,1283), P3'(1002,1643) and P4'(362,363), calculated in step4is copied from the transformed image to the output image at the same position as shown in the [fig. 7(a), fig. 7(b) and fig. 7(c)].

Step6 : New next _frame is grabbed. Keypoints are only extracted in new next_frame, as previous

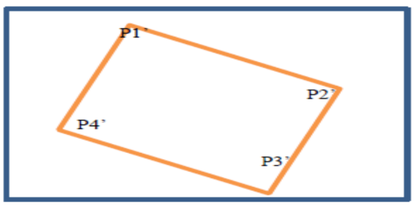

Fig. 6. Transformed next_frame according to homography matrix

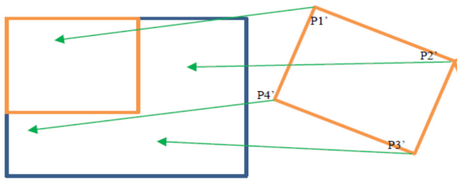

Fig. 7. (a) Previous output image, the arrow line shows where the bounding part of P1', P2', P3', P4' will go in the output image, (b) the bounding part to be copied from transformed image to previous output

Fig. 7. (c) Created new output image

Fig. 7. (a), (b), (c): Stitching transformed frames together for stabilization

Next_frame becomes current_frame and Kpt1=Kpt2. Keypoints matching is done, (Xc,Yc) and (Xn,Yn) are the matched point. As mentioned earlier in equation (2), new position of (Xc,Yc) in the output image are calculated in similar way. Say, the new position of (Xc,Yc) are (Xt,Yt). Now the homography matrix will be calculated between (Xt,Yt) and (Xn,Yn), in order to know the next_frames position according to the output image.

Step7 : If current_frame is not last_frame go to step3.

[fig. 8] shows transformation results of next_frame according to current_frame using homograpy matrix. After the transformed next frame is calculated it is stitched with the curre_frame to get a stabilized frame from the same view point. In the proposed algorithm, a blank frame of size 2*frame_size was created to show the stabilized output frame. Initially first frame is copied to the blank output image as described in step2. And In next every iteration transformed frame is estimated. From the transformed image [fig. 8, (C)] only the colour part removing the black part is copied to the output image. The algorithm continues until there is no more next_frames available in the video. Result

(a) Current_frame

(b) next_frame

(c) Transformed image

Fig. 8. Transforming next frame according to current_frame

of the stabilization algorithm is shown in [fig. 9].

Ⅳ. Matching Improvement

For perfect image stitching it is very important to have an accurate homography matrix. And, the estimation of homography matrix is directly related to the matched keypoint. So it is very important that the matching set have correct matches as much as possible. To further remove the incorrect matches the area to search the correspondences in the next frames is restricted with in some specified area. This is possible as the algorithm deals with video frames and frame to frame displacement is very less in video. Different researches also have been done to improve the matching procedure[13,18,19].

Suppose the pixel value of a keypoints in the current_frame is (x1,y1) and correspondence keypoint for that point after initial matching is (x2,y2), then this matching will be considered as a correct match only if the keypoint in next frame(x2, y2) is present around the area where it was in the current frame. To do so, at first minimum displacement according to x-axis and y-axis (x_min_dis, y_min_dist) from current frame to next frame is calculated at every iteration. Now for every initial matched keypoints the displacement in x axis (|x1-x2|) and y axis (|y1-y2|) are calculated and checked with the equation (4). Assuming all the keypoints in initial match have minimum displacement then total average minimum displacement for all initial match would be (initial_match_size* (min_dist_x+min_dist_y)).

$$(|x1-x2|)+(|y1-y2|) < (\text{initial\_match\_size}*(\text{min\_dist\_x}+\text{min\_dist\_y})) \quad (4)$$

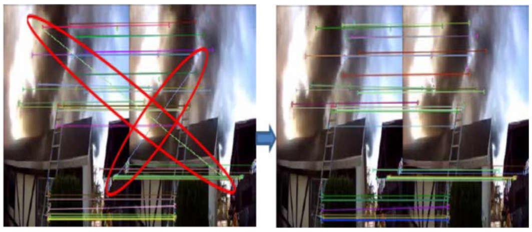

Fig. 9. Removing incorrect matches by proposed approach

Ⅴ. Complexity Reduction

To reduce the computational complexity of the algorithm, atfist SIFT descriptor vectors dimention is reduced and then the number of keypoints to be extracted in every frame is restricted.

5.1 Modified SIFT Descriptor

The descriptor vector of traditional SIFT[10] is a combination of orientation histograms. An 8 binhistogram is used and a patch around the feature point is split into separate 4x4 regions. Each has its own orientation histogram, so the descriptor is a 128 dimensional vector (8x4x4)[10]. As the algorithm deals with video frames and corresponding frames have similar information, so it is sufficient to use a smaller descriptor vector to keep information. Here, rather than using 4*4 window around the pixel a

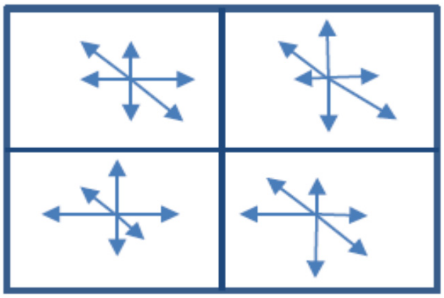

Fig. 10. modified descriptor vector

2*2 window around the pixel is considered in the modified SIFT. And around every pixel a 6 bin histogram is used to represent the local gradient for each pixel. So, the modified SIFT descriptors dimension becomes 24(2*2*6). Experimental results in [Table 2] show almost every video gives 100% accuracy with the modified SIFT descriptor vector.

5.2 Region-Based Keypoints extraction

The computational complexity of the algorithm is further reduced by reducing the number of SIFT keypoints to be extracted in every frames in each iteration. As the computational complexity of SIFT keypoints extraction is very high, reducing the number of keypoints to be extracted will reduce the time complexity effectively. But when SIFT keypoints are extracted traditionally; they are assigned a value and arranged in ascending order with their score. Restricting the number of keypoints results only top most ranked keypoints, but it doesn't guarantee that the keypoints will be well distributed in all over the image. If the keypoints are not well distributed over the frame while transforming frames error may occur. So, the problem is less number of well distributed keypoints are needed.

To ensure the SIFT keypoints are well distributed over the image frame, keypoints extraction is done region wise. The image frame is initially divided into 4 regions or blocks. 4 blocks are chosen because to estimate homography matrix minimum 4 pairs of keypoints are needed. So, even if every region has at least one keypoints it is possible to estimate the homography matrix. After keypoints extraction in every block they are merged together to get well distributed keypoints all over the frame.

Ⅵ. Experimental Results

The performance of the proposed algorithm is evaluated with several unstable video sequences covering different types of senses to observe the efficiency and quality of the stabilized frames. The results of proposed stabilization with stitching are compared with the traditional stabilization algorithms to show the difference between the traditional and proposed algorithm. Fig 11 shows stabilization results of different frames and a comparison between traditional and proposed method.

Experiments have been done with different unstable videos and the results shows that the proposed approach give more than 98% correct matches [Table 1]. The table shows percentage of correct matches before and after doing matching improvements. The proposed matching results have more than 10% improvements. Related approaches are described in [13] & [18] but the results cannot be directly compared as, the previous works deals with still images and the proposed algorithm mainly considers video sequences. In [20] and [21], they uses video frames for improving matches, their results gives 90% of matching results which is lower than the proposed approach.

Fig 12 show, how the computational complexity is reduced by reducing the SIFT descriptor vector's dimension and restricting the number of keypoints to 60 in every frame.

In Zhu Qidan and Li ke proposed algorithm[13] which performs similar approach, the SIFT

Table 1. Comparison of correct Matches before and after

	Avg. correct match before (%)	Avg. correct match after (%)
Video1	88.89	99.54
Video2	91.05	99.21
Video3	88.02	98.64
Video4	81.23	98.42

Fig. 11. (a): Unstable frames

Fig. 11 (b): Stabilized with traditional approach

Fig. 11. (c): Stabilized with proposed approach

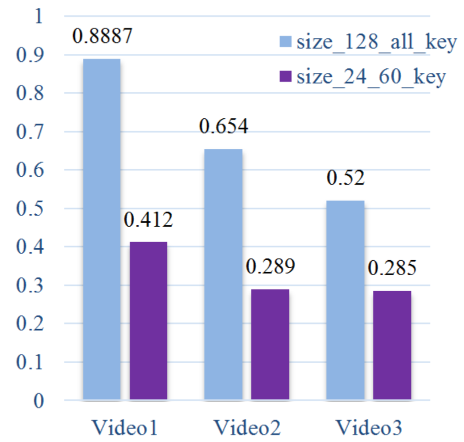

Fig. 12. comparison of time complexity after matching improvements

computational complexity is reduced by simplifying the descriptors, but there result shows, still the computational time/frame is around 3.8 sec to 3.9 sec as they do not restrict the number of keypoints to be extracted. But, in the proposed algorithm the computational complexity is further reduced by restricting the number of keypoints and also reducing the descriptors size. The average time/frame in current approach is 0.389 sec, which means the performance is improved by almost more than 90%.

Table 2 shows that how the descriptors size affects the accuracy of the stabilization algorithm. Currently the algorithm uses descriptor size 24 because further reduction of the descriptor reduces the time complexity of the algorithm but it also affects the accuracy of the stabilization algorithm. Further reducing the descriptor size gives inaccurate results.

Table 2. Accuracy test with different descriptors size

Unstable videos	Accuracy(%) of stabilization with different descriptors size						
Descriptor size	128	96	64	32	24	16	8
Video1	100	100	100	100	100	70.74	17.91
Video2	100	100	100	100	100	87.45	19.23
Video3	100	100	100	100	100	87.45	60.36

Ⅶ. Conclusions

This paper presents an improved video stabilization algorithm with image stitching for vehicular applications. By following these methods the hassle of estimating global, local motions and distinguishing between them to do motion compensation for video stabilization can be easily avoided. The proposed algorithm not only removes unwanted shaky motions but also removes rotation and translation which may occur because of camera movements. The accuracy of the algorithm is improved by improving the matching procedure. Due to the modified descriptor size and restriction on number of SIFT keypoints the efficiency of the algorithm remains perfect and it becomes much faster than the traditional SIFT based algorithms. Further work includes improvements of the algorithm and hardware implementation of the algorithm.

References

[1] S. Auberger and C. Miro, "Digital video stabilization architecture for low cost devices," *Proc. 4th Int. Symp. Image and signal Process. Anal.*, pp. 474-479, Zagreb, Croatia, Sept. 2005.

[2] H. C. Lee and D. M. Lee, "A study on the pattern matching algorithm of 3D coordinates for quality control in ship blocks," *J. KICS*, vol. 37, no. 10, pp. 933-939, Oct. 2012.

[3] S.-W. Jang, M. Pomplun, G.-Y. Kim, and H.-I. Choi, "Adaptive robust estimation of affine parameters from block motion vectors," *Image and Vision Computing*, vol. 23, no. 14, pp. 1250-1263, Dec. 2005.

[4] F. Vella, A. Castorina, M. Mancuso, and G. Messina, "Digital image stabilization by adaptive block motion vectors filtering," *IEEE Trans. Consumer Electron.*, vol. 48, no. 3, pp. 796-801, Aug. 2002.

[5] S. Eriturk, "Image sequence stabilization based on kalman filtering of frame positions," *IEEE Electron. Lett.*, vol. 37, no. 20, pp. 1217-1219,

Sept. 2001.

[6] L. M. Abdullah, N. Md Tahir, and M. Samad, "Video stabilization based on point feature matching technique," *IEEE Control and System Graduate Research Colloquium*, pp. 303-307, Shah Alam. Selangor, Malaysia, Jul. 2012.

[7] M. Zhou and V. K. Asari, "A fast video stabilization system based on speed-up robust features," *Advances in Visual Computing*, pp. 428-435, Berlin, Germany, 2011.

[8] G. Gankhuyag, E. G. Hong, G. Kim, Y. Kim and, Y. Choe, "A real-time video stitching algorithm in H.264/AVC compressed domain." *J. KICS,* vol. 39, no. 6, pp. 503-511, Jun. 2014.

[9] W.-R. Lee, K.-H. Kang, and J. Yoo, "Object-based conversion of 2D image to 3D," *J. KICS,* vol. 36, no. 9, pp. 555-563, Sept. 2011.

[10] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Computer Vision*, vol. 60, no. 2, pp. 91-110, Nov. 2004.

[11] V. F. da CamaraNeto, "An improved methodology for image feature matching," *XXII Brazilian Symp. Computer Graphics and Image Processing*, pp. 307-314, Rio de Janeiro, Brazil, Oct 2009.

[12] R. Laganiere, *OpenCV 2 Computer Vision Application Programming Cookbook*, Packt Publishing, 2011.

[13] Z. Yi, C. Zhiguo, and X. Yang "Multi-spectral remote image registration based on SIFT," *IEEE Electron. Lett.*, vol. 44, no. 2. pp. 107-108, Jan. 2008.

[14] G. Bradski and A. Kaebler, *Learning Opencv: Computer Vision with the Opencv Library*, NY: O'Reilly, 2008.

[15] M. A. Fishler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381-395, Jun. 1981.

[16] Z. Qidan and L. Ke. "Image stitching using simplified SIFT," in *Proc. IEEE Int. Conf. Inf. Automation*, pp. 1134-1137, Harbin, China, Jan. 2010.

[17] L. Ledwich and S. Williams, "Reduced SIFT features for image retrieval and indoor localisation," *Australian Conf. Robotics and Automation*, Canberra, Australia, Dec. 2004.

[18] P. Du, Y. Zhou, Q. Xing, and X. Hu, "Improved SIFT matching algorithm for 3D reconstruction from endoscopic images," *VRCAI: Virtual Reality Continuum and its Applications in Industry*, pp. 561-564, New York, United States of America, 2011.

[19] L. Lingjun and L.- Xiang, "LSIFT- an improved SIFT algorithm with a new matching methode," *Int. Conf. Comput. Appl. Syst. Modelling*, pp. V4-643-V4-646, Taiyuan, China, Oct. 2010.

[20] T. Chen, *Video stabilization algorithm using a block-based parametric motion model*, EE392J Project Report, Inf. Syst. Laboratory, Dept. of Electrical Engineering, Stanford University, United States America, 2000.

[21] Z. Ying and Li-Da-Hui. "Video image tracing based on improved sift feature matching algorithm," *J. Multimedia*, vol. 9, no. 1, pp. 130-137, Jan. 2014.

모스타피즈 메헤부바 후세인
(Mostafiz Mehebuba Hossain)

2012년 2월 : 인도 B.P.Poddar 경영기술대학교 컴퓨터공학과 졸업
2013년 3월~현재 : 서울대학교 전기컴퓨터공학부 석사과정
<관심분야> Image Processing, Video Processing

이 혁 재 (Hyuk-Jae Lee)

1987년 2월 : 서울대학교 전자공학과 졸업

1989년 2월 : 서울대학교 전자공학과 석사

1996년 12월 : 퍼듀대학교 전기컴퓨터공학부 박사

1996년 8월~1998년 8월 : 루지에나 공과대학 조교수

1998년 11월~2001년 2월 : Intel Corporation, 선임연구원.

2001년 3월~현재 : 서울대학교 전기컴퓨터공학부 교수

<관심분야> Multimedia Processor, Computer Architecture, System-on-Chip(SoC), VLSI Design

이 재 성 (Jaesung Lee)

2001년 2월~2011년 8월 : 한국전자통신연구원 선임연구원

2003년 3월~2008년 8월 : 서울대학교 전기컴퓨터공학부 박사 졸업

2011년 9월~2013년 9월 : 한국교통대학교 전자공학과 조교수

2013년 10월~현재 : 한국교통대학교 전자공학과 부교수

<관심분야> VLSI, System-on-Chip, Image Processing, Computer Vision