공학석사학위논문

Fast Image Stitching for Video Stabilization using SIFT Features

2014년 8월

서울대학교 대학원

전기 컴퓨터 공학부

Mostafiz Mehebuba Hossain

Fast Image Stitching for Video Stabilization using SIFT Features

지도교수 김재하

이 논문을 공학석사 학위논문으로 제출함

2014 년 8 월

서울대학교 대학원

전기 컴퓨터 공학부

Mostafiz Mehebuba Hossain

Mostafiz Mehebuba Hossain 의 공학석사 학위논문을 인준함

2014 년 8 월

위 원 장 : Chae, Soo-Ik ______________ (인)

부위원장 : Kim Jaeha ______________ (인)

위 원 : Lee, Hyuk-Jae ______________ (인)

Abstract

In recent years' uses of hand-held camera, portable camera, firemen's head camera, robot camera has increased a lot. However, the videos captured by these types of cameras are generally unstable, filled with unwanted shaky camera motions. Therefore, a demand for digital video stabilization has increased. There are several researches about digital video stabilization based on block-based motion compensation and feature point based motion compensation for video stabilization. However, those algorithms are computationally expensive and also cannot handle uncertain and big motions such as unstable motions generated by firemen's head cameras. In this paper, an improved video stabilization method with image stitching has been proposed. Also, the computational complexity of SIFT (Scale-Invariant Feature Transform) is reduced and the matching is improved to improve the accuracy of the stabilization.

Image stitching using SIFT (Scale-Invariant Feature Transform) is done to perform improved video stabilization. After SIFT feature points are extracted and matched between two frames of unstable video, the appropriate mathematical model relating pixel coordinated from one frame to pixel coordinates in other frame is done by estimating the homography matrix between them. All the pixels of next frame are transformed according to the current frame with the homography matrix and the pixel values. Transformed frames in every iteration are stitched together to get the stitched stabilized frames.

To improve the SIFT feature point matching and improving 'classification error' the searching area of the keypoints are considered. After initial matching with KNN matcher those matching keypoints are removed from the match list whose displacement from one frame to other is lager then a threshold value.

To reduce the computational complexity of SIFT feature point the descriptor vector's size is reduced to 24 from 128. A restriction also implemented on the number of SIFT feature points to be extracted. A region based SIFT feature points extraction is proposed in the paper to reduce the effect of 'measurement error'. Keypoints are extracted in different regions of the frame and then merged together to the full frames keypoints. This method ensues that the keypoints are well distributed over the image frame.

Keywords: Video Stabilization, Image Stitching, SIFT, Image Processing

Student number: 2012-23951

Table of Content

List of Figures

List of Tables

1. Introduction

1.1. Foreword

Video stabilization is one of the most discussed topics for many years. Video stabilization is done to remove unwanted shaky motions from videos which are common in fireman’s helmet camera, hand-held video cameras, robots cameras etc. Most of the researches concentrate only on shaky videos generated from hand-held devices. This means very little motions are only considered while stabilizing videos. However, this research is mainly focused on videos generated from firemen’s helmet camera. Also, this algorithm can be used to stabilize any kind of unstable video. The videos generated from firemen’s 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 fireman. The traditional video stabilization algorithms out puts 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.

The matching procedure is also improved by checking the keypoints frame wise displacement in this paper to enhance the quality of the video stabilization. Experimental results show the improvements in matching algorithm in the section 6. To reduce the computational complexity of the algorithm SIFT descriptor vectors dimension has been reduced and also the number of keypoints to be extracted in every iteration have been restricted after performing the matching improvements. Section 6 also shows comparison between traditional video stabilization algorithms results and the proposed stabilization approach with image stitching.

1.2. Video Stabilization Problem Definition

Given a video which has several kinds of unwanted shaky motions between any two consecutive frames, we want to remove those uncomfortable video motions to make it stable. Those motions may have rations, translation or scale difference between two consecutive frames. The goal of video stabilization is to get a smooth video without any visible frame-to-frame jitter.

1.3. Research Material

The Research was done under Computer Architecture and Parallel Processing lab. All the source codes, test videos, demo videos are present in Computer Architecture and Parallel Processing lab. A request can be made to capp lab via website (http://capp.snu.ac.kr) and also can directly email the author at mostafiz@capp.snu.ac.kr.

2. Previous Work for Video Stabilization

Video stabilization techniques have been widely studied in several ways with different issues and weak points. Video Stabilization generally follows 3 main steps local and global motion estimation, motion filtering and image composition or image enhancement. Figure 1 shows a general overflow of traditional video stabilization procedures.

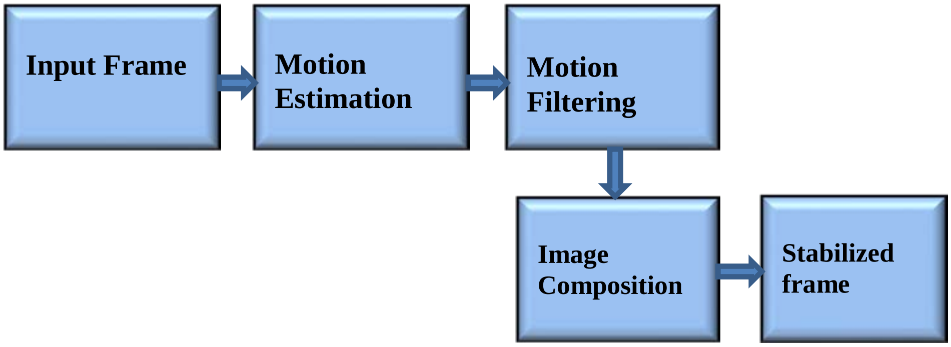


Figure 1: Overflow of traditional video stabilization

In order to understand the traditional procedures of video stabilizations a brief history and discussion is given to understand the whole thing without referring to many other papers.

2.1. Block Matching Based Video Stabilization

In early days, 'block matching' was used to estimate local motion and remove unwanted motions [1], [2], [3]. These algorithms extract global motion information from a sequence of images by using block matching and then process those motion vectors to remove outlier motions to reduce the effect of global motions. They also use different adaptive filters to refine motion estimation from block local vectors. Block matching approaches mostly gives good results but sometimes gets 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 and also, any descriptor is not associated to a block to track the block in consecutive frames. Also Block based stabilization algorithms are sensitive to illumination; noise and motion blur [11].

2.2. Feature Points Based Video Stabilization

Feature points based video stabilization methods are very popular for their good performances. Many different methods have been researched in this field. Some of them uses corner feature points[24] and optical flow to calculated the motion information, where as other researches are done with Scale Invariant Feature Transform(SIFT) feature [26] points and many other uses Speeded Up Robust Features(SURF) feature points [22] to do video stabilization. Most of these feature based stabilization are done by this steps –

- **Feature points are extracted in consecutive frames,**
- **Motions information are calculated by tracking those feature points frame by frame,**
- **Motion filtering is done to distinguish the global and local motions and**

- **Motion compensation to remove the unwanted motion from the unstable videos.**

For motion filtering several filters are used in years such as- Kalman filter [27], particle filter [25], and Gaussian filter [24], Motion Vector Integration (MVI) [23]. Most of these researches give good results, but the conversion from Cartesian to log polar coordinates sometimes brings significant re-sampling error. Also they are computationally very expensive because of the motion filters is done for every frame.

3. SIFT Feature Points Matching and Homography Matrix

For working knowledge in this section previous work related to this researches (SIFT, Homography matrix, RANSAC) is given, so that the reader don’t need to refer several materials/papers for each topic.

3.1. SIFT Feature Points

The Scale Invariant Feature Transform (SIFT) [4] is an algorithm to find and describe scale invariant feature points in image. SIFT transforms image data into a feature vector, each of this vectors are invariant of image rotation, scaling, translation and also robust to local geometric distortion. SIFT feature points detects the dominant gradient orientation at every location and records that according to its orientation to the histogram bin. SIFT point extraction mainly consist of the following stages-

3.1.1. Scale Space Generation

To detect a keypoint the image is examined in different scales with Gaussian blurring to find the strong feature which occurs at multiple sizes of image. To find a scale invariant feature, localization of stable features across multiple scales is done in a space call *scale-space*. Laplacian of Gaussian is found for the image with various **σ** values. LoG detects blobs in various sizes due to change in sigma (**σ)**. SIFT algorithm uses Difference of Gaussians (DoG) which is an approximation of LoG. Difference of Gaussian is obtained as the difference of Gaussian blurring of an image with two different sigma's (**σ).** Which means to create a Gaussian pyramid. The original image is consecutively blur using a Gaussian filter, at the end of the level the original image is scaled down and the process is repeated. The procedure can be represented by the Figure 2.

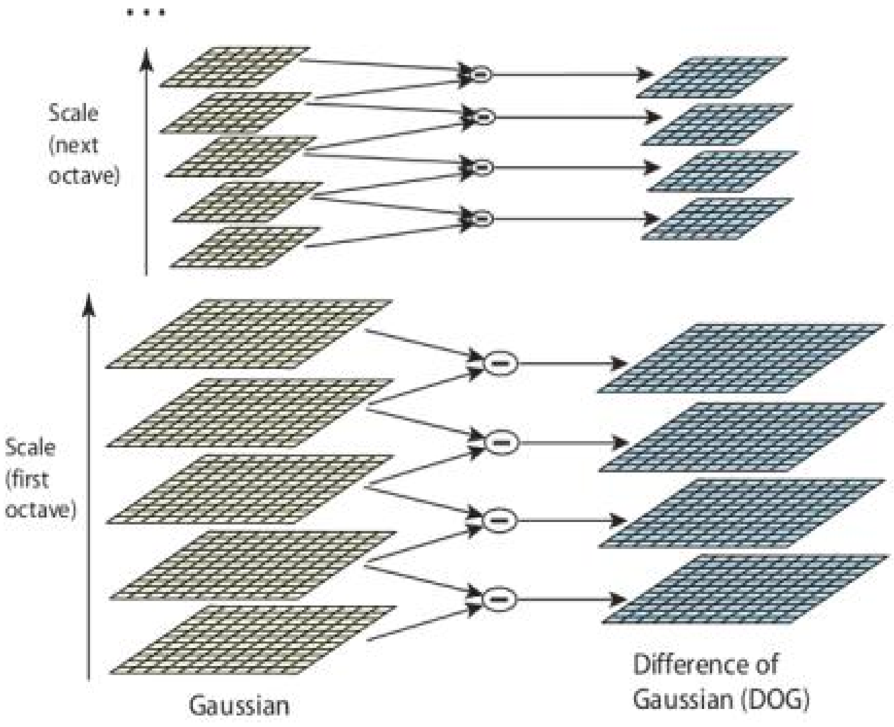


Figure 2: Gaussian pyramid

Once these DoGs are found the images are searched for local extrema over scale and space. If a pixel is extrema that is either minimum or maximum, the process is repeated in the next space scale level. So the points which are not extrema are easily rejected and extrema points are considered as potential keypoint. Because of these local extrema which are found repeatedly in multiple scales SIFT has the scale-invariant property in it.

3.1.2. Keypoint Localization

Once the potential keypoints are found in the previous step, they are refined to get more accurate keypoints in this step. This is done by a Taylor series expansion of scale space to get more accurate location of the extrema. If the intensity of the etrema is less than a threshold value then the extrema is rejected.

For removing the points in the edges two perpendicular gradient is calculated for each point. If the ratio is greater than a threshold value the keypoint is discarded. So, it basically removes those keypoints which are low-contrast or edge keypoints.

3.1.3. Orientation Assignment

This step assigns a consistent orientations to the keypoints survived after previous tests. Use the keypoints scale to select the Gaussian smoothed image L, from above then compute the gradient magnitude, m and θ by using the equation (1) and (2).

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \qquad \text{---(1)}$$

$$\theta(x, y) = \tan^{-1}\left(\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y)} - L(x-1, y)\right) \qquad \text{---(2)}$$

An orientation histogram with 36 bins covering 360 degrees is created. Samples located around the feature are added to the histogram using a Gaussian weighted circular window with σ=1.5 times the scale of the keypoint. The highest peak in the histogram is taken and any peak above 80% of it is also considered to calculate the orientation. It creates keypoints with same location and scale, but different directions. It contributes to the stability of matching.

3.1.4. Keypoints Descriptors

At this stage the keypoint's descriptor is created. To create the descriptor of a keypoint a 16*16 neighborhood is taken and is divided into 16 sub-blocks of 4*4 size. Within each 4×4 window, gradient magnitudes and orientations are calculated. These orientations are put into an 8 bin histogram. So, a 128 dimensional (4*4*8) vector is used to put the 128 bin value for each keypoint descriptor. Finally, the feature vector is further modified to achieve robustness against illumination and rotation changes.

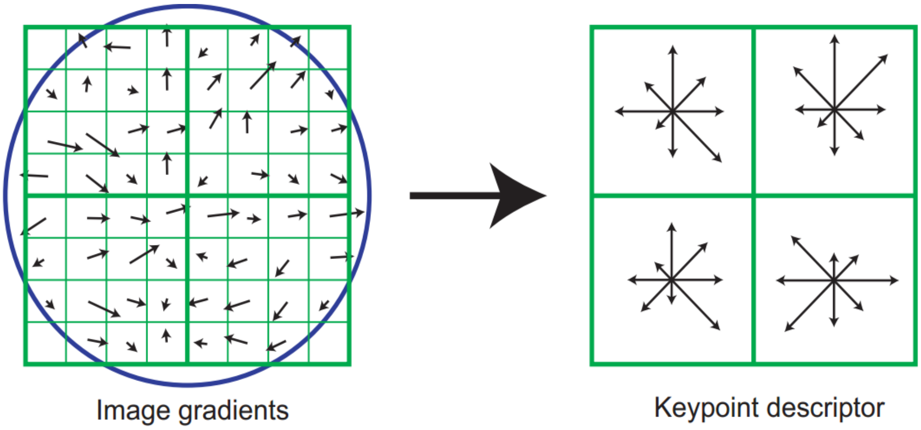


Figure 3: Descriptor vector

Figure 3 shows a 2x2 descriptor array computed from an 8x8 set of samples, whereas the experiments in this paper [4] use 4x4 descriptors computed from a 16x16.

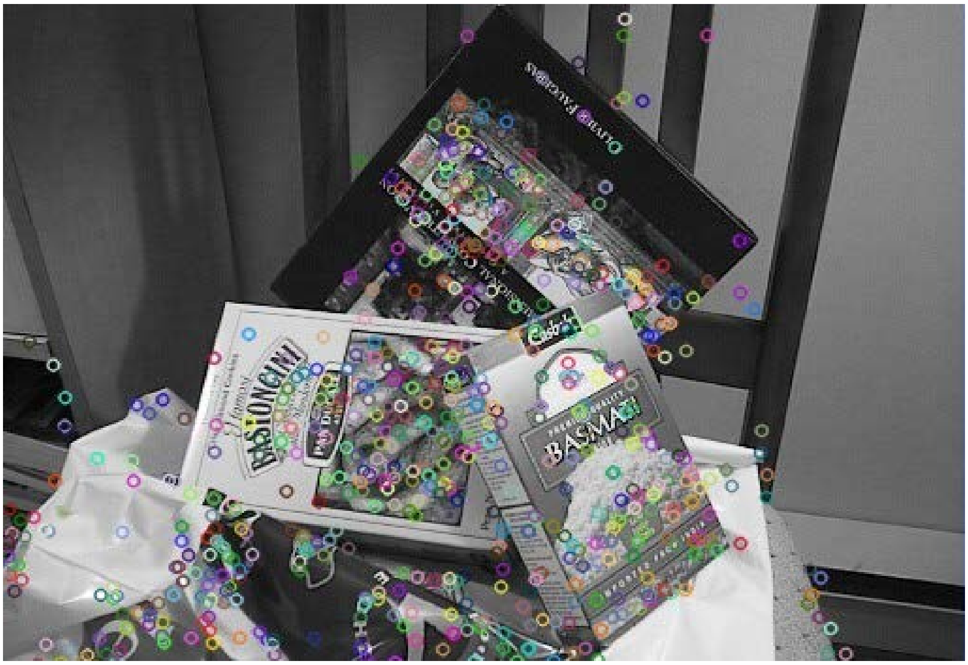


Figure 4: SIFT feature points

Figure 4 shows an example of detected SIFT feature points after performing all the previous steps. The image is taken from the reference images provided by opencv.

3.1.5. Feature Points Matching

Keypoints between two images are matched by identifying their nearest neighbors. 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. Test image scans the feature for best matches and uses a system to decide the correct object. SIFT feature matching can be done by Brute-Force [20] matcher with cross checking. 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 [4]. In this method a Knn (k==1) Matching is done in both the ways i.e KnnMatch(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.

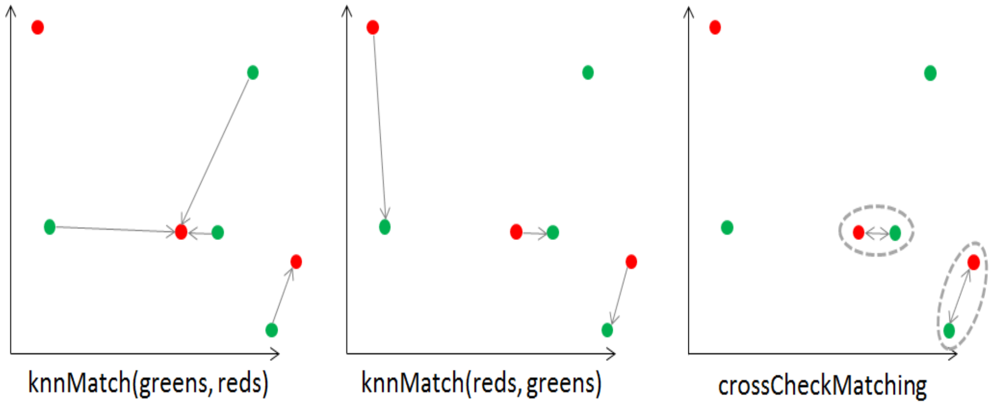


Figure 5: Brute-Force matcher with cross checking

Figure 5 [32], shows an example of Brute Force matcher with cross checking with KNN==1 is considered. Here randomly picked 2D samples data are shown with red and green colors. In the first plot green colored points are the query data set, which means for the green points nearest neighbor red points are searched and each arrow represents a correspondence for that green colored point. In the second plot the query data set and the train data set is changed i.e red colored points are query data and green colored points are train data set. The arrow from re to green color data represents that particular point nearest correspondence. Finally in the third plot a cross checking is done to find bi-directional arrows for good matches.

3.2. Homography Matrix

Homography is a mathematical term for mapping points from one surface to other. In Computer Vision, homography almost always refers to mapping points between two image planes that correspond to the same location on a planar object in the real world [6]. 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-

$$\mathbf{[F1]=H}\begin{bmatrix} \boldsymbol{H11} & \mathbf{H12} & \boldsymbol{H13} \\ \boldsymbol{H21} & \mathbf{H22} & \mathbf{H23} \\ \boldsymbol{H31} & \mathbf{H32} & \boldsymbol{H33} \end{bmatrix}\mathbf{*[F2]} \qquad \text{--------} \ \mathbf{(3)}$$

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. So, to estimate a robust homography matrix RANdom SAmple Consensus (RANSAC) method had been implemented in the proposed algorithm, which is described in section 3.2.1.

3.2.1. RANSAC Algorithm

The RANSAC [6] 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 points requires estimating the mathematical entry [10].

In [6], author shows, how even in a simple two dimensional case, the extreme outlier ("poisoned point") fools a highest disagreement elimination iterative least square method. RANSAC was created to be robust in presence of outliers. Rather than attempting to operate on as much of data as possible to find an initial solution, RANSAC begins with the smallest solution set possible for task and keep on modifying it until the end.

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 who fulfils the conditions forms the support set ('inliers') of the homogrpgy matrix. The larger the support set is the probability of getting the homography matrix accurately 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 in Figure 6.

```
1. Select four matches (randomly),
2. Compute homography H,
3. Keep largest set of inliers ,
4. Go to step 1 ,
5. Re-compute H estimate with all the inliers.
```

Figure 6: RANSAC algorithm for homography matrix [14]

3.3. Image Stitching

Image stitching is a process to combine or integrate two or more images, which have some over lapping areas. In general, images are matched with some feature point information and then images are transformed according to same view points and stitched together in a bigger image. Image stitching is widely used in several fields over the years' such as- Panorama image generation [7] and [17] for creating wide angle and high resolution images medical image generation [15], for image mosaics [18] and [19]. For image stitching feature based approaches are mainly used for example- corner point based image stitching [21], Edge based stitching[19], SIFT points based[7], SURF points based[16] etc. Corner features and edge feature based image stitching are sensible to noise, whereas SIFT based image stitching is more accurate as SIFT points are rotation, translation and scale invariant. Though SIFT based stitching are computationally high but the quality of stitching is much better with it. In this research paper SIFT based image stitching used for doing video stabilization.

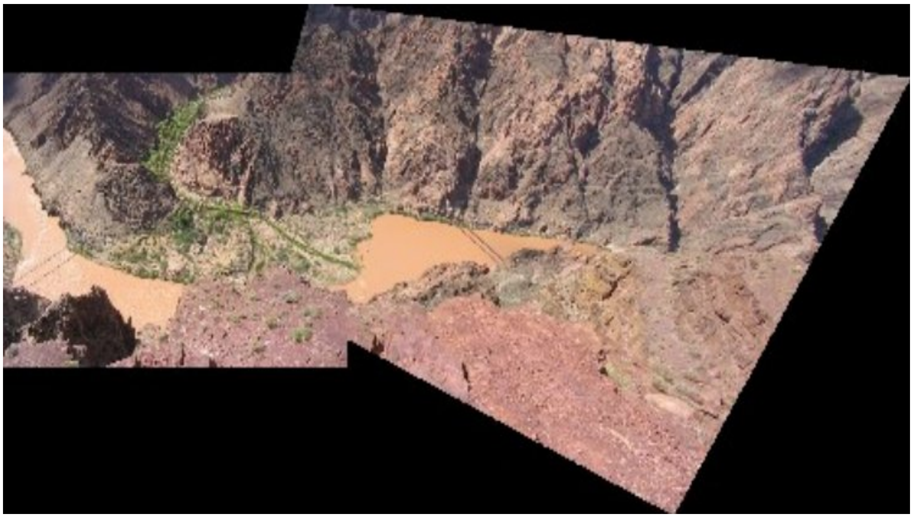


Figure 7: Example of image stitching

Figure 7 shows an example of image stitching; here two different images were taken from different point of view. Second image is transforming according to first image’s view point and then in a larger blank image.

4. Improved Matching and Fast SIFT Extraction

To improve the accuracy of the stabilization an improved matching procedure is proposed to get more correct matches. To reduce the complexity two things is done, first the keypoints descriptors dimension is reduced, 2nd the number of keypoints to be extracted is restricted to a certain number. A region based SIFT feature point extracting is also proposed to ensure that feature points are well distributed over the frame.

4.1. Modified Matching Approach

Classification errors generally occur when a feature detector incorrectly identifies a portion of an image as an occurrence of a feature. (e.g. wrong match). To improve the stabilization accuracy the matching procedure is need to be improved. Most of the classification errors are removed with the two way Brute-Force matching procedure. To further remove the matches which are the reasons for the classification error is removed by reducing the searching area for keypoints in the next frame detailed description in section 4.1.1.

4.1.1. Searching Area Restriction

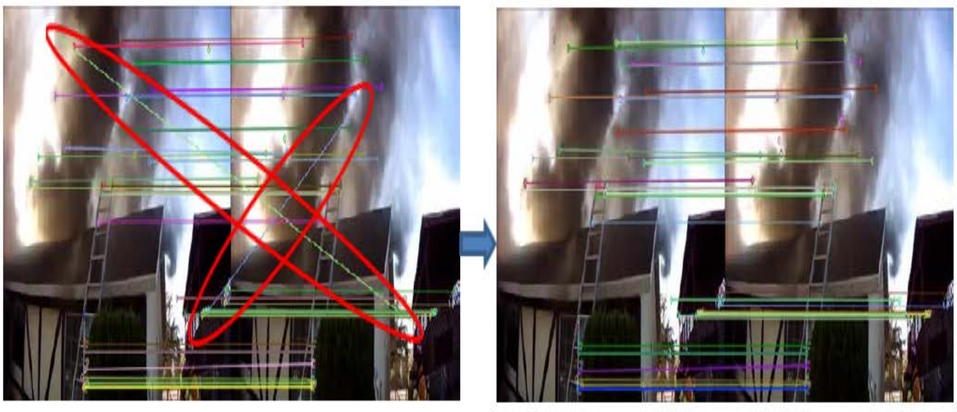
For perfect image stitching it is very important to have an accurate homography matrix. And, the estimation of homography matrix as described in section 3.2 is directly related to the matched keypoint. So, it is very important that the matching set have correct matches as much as possible. The initial matched keypoints are found with Brute-Force Matcher as described in section 3.5.1. To further remove the incorrect matches the area to search the correspondences in the next frames is restricted with in some specified area. Different researches also have been done to improve the matching procedure [8], [9], [12].

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. Which means the displacement of the matched keypoint in the next frame is checked to get some good matched keypoints. 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.a) & (4.b). 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)). Experimental results in section 6.2 shows that [table 2] in any video the frame to frame ratio of displacement for a pixel with the minimum displacement cannot be more than 50 pixel even if there exist any moving objects in the frame.

(|x1-x2|)+(|y1-y2|)<(initial_match_size*(min_dist_x+min_dist_y))

------ (4.a)

(|x1-x2|)+(|y1-y2|)/ initial_match_size <50 ------ (4.b)



(a)Before (b) After

Figure 8: Removing incorrect matching, (a) shows matching results before improvement, (b) matching results after improvements

Figure 8 shows matching how wrong matches are removes by the proposed approach. Figure 8(a) shows initial matches result before restricting the keypoints searching areas in the next frame and Figure 8(b) shows after applying the proposed approach and removing wrong matches from the same image.

4.2. Modified Keypoint Descritors

Further, to reduce the complexity of the SIFT keypoints a modified SIFT describtors vector is proposed. In general, the SIFT [4] keypoint descriptors, which gives unique information about the surroundings is very useful for accurate feature point matching. The descriptor vector of traditional SIFT 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) as described in section 3.1.4. Rather than using 4*4 window around the pixel a 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 dimention becomes 24(2*2*6).

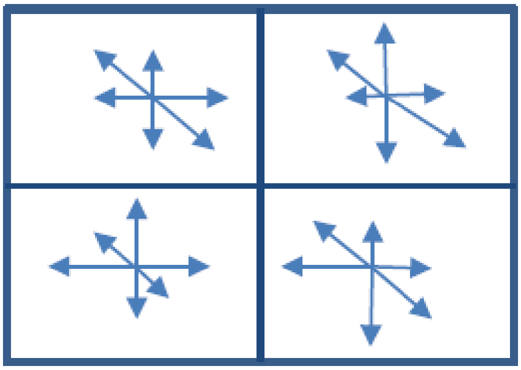


Figure 9: Modified keypoint descriptor

Figure 9 shows a 2*2 window around the pixel with 6 bins histogram. For stabilization with stitching, several test have been done with different kind of videos to check the accuracy of stabilization with the modified SIFT descriptor vector. Almost every video gives 100% accuracy with the modified SIFT descriptor vector.

4.2.1. 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. Figure 10 shows an example of extracted traditionally extracted SIFT keypoints when the number is restricted to 60. The keypoints are mainly concentrated in one area.

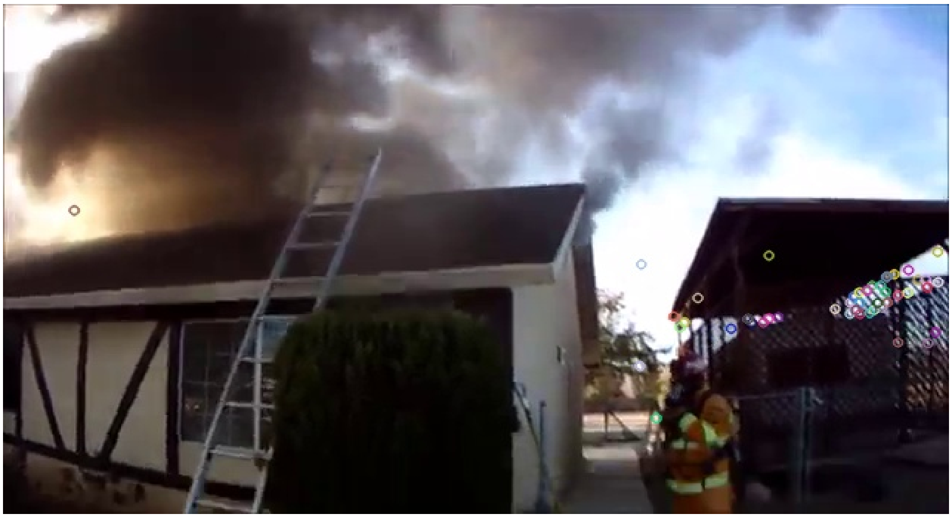


Figure 10: traditionally extracted SIFT feature points

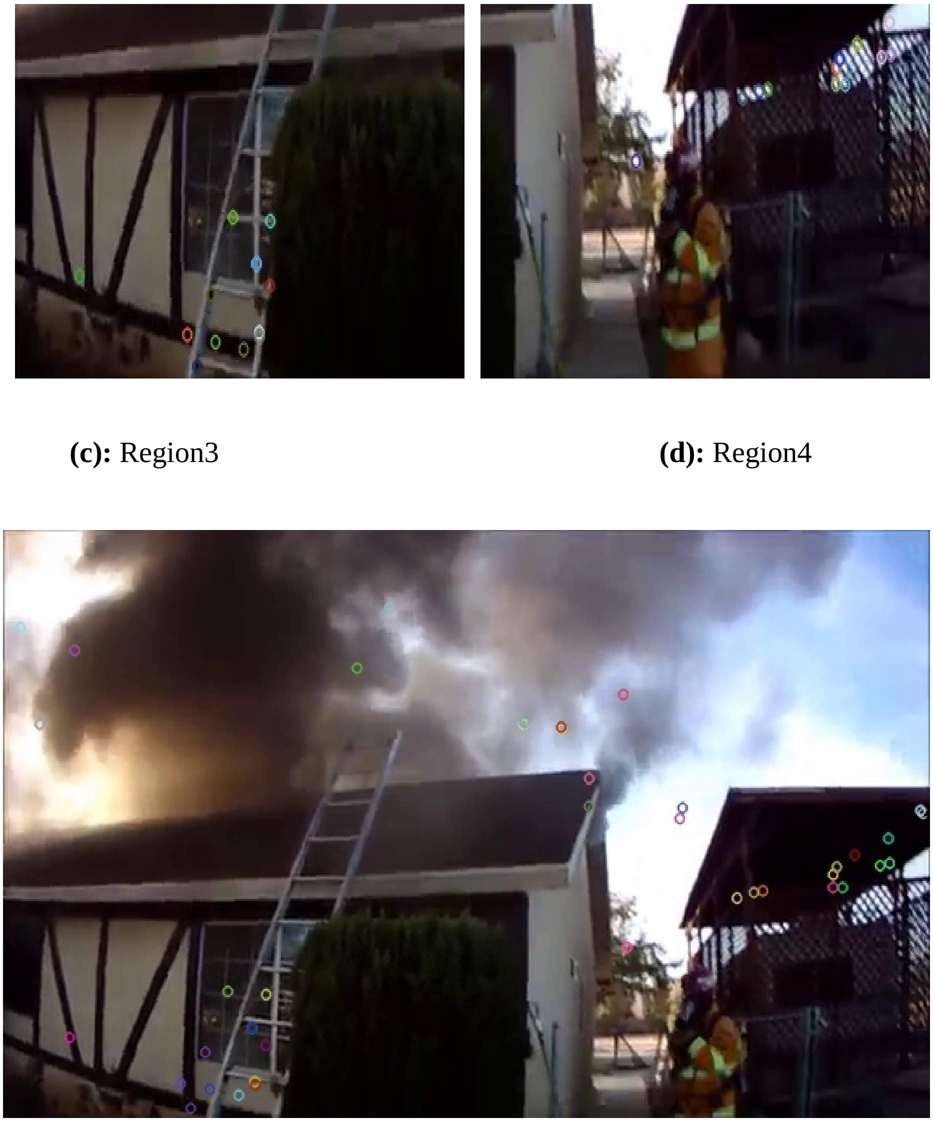
When keypoints are concentrated in one area there are chances of **Measurement error** to occur while transforming the image frame with homography matrix. **Measurement errors** generally occur when the feature detector correctly identifies the feature while matching, but slightly miscalculates one of its parameters (e.g., its image location). Measurement error can have a huge effect in image transformation. Let's consider an example with 1D data. Say we have two points that are supposed to place at the exact location 0 and 30.8, but instead they are found at 0 and 30. This is around 2.6% error. If those points are used to transform a 1D image of size 500, some pixels can be off by 13 pixels (2.6% of 500). The measurement error generally follows a normal distribution, and therefore, smoothing assumptions is applicable to them.

Now the problem is we need less number of keypoints to be extracted for reducing the time complexity and also need to guarantee that the extracted keypoints are well distributed to reduce the effect of Measurement error.

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.

(a): Region1

(b): Region2



(c): Region3

(d): Region4

(e): All regions Keypoints are merged together

Figure 11: (a), (b), (c), (d) shows extracted keypoints in specific regions. Fig (e) shows keypoints after merging them together.

Figure 11 shows an example of proposed region wise extracted SIFT keypoints. In every region 15 SIFT keypoints are extracted [Figure 11(a), 11(b), 11(c), and 11(d)] and they are merged in 11(e) to get well distributed keypoints. Same image and same number of keypoints are extracted in Figure 10 and Figure 11 to compare the traditional and proposed approach of SIFT keypoints extraction.

5. Frames Transformation and Stitching for Video Stabilization

The proposed algorithm presents a method which uses image stitching to perform video stabilization. SIFT features are extracted in the current and next frame then they are 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.

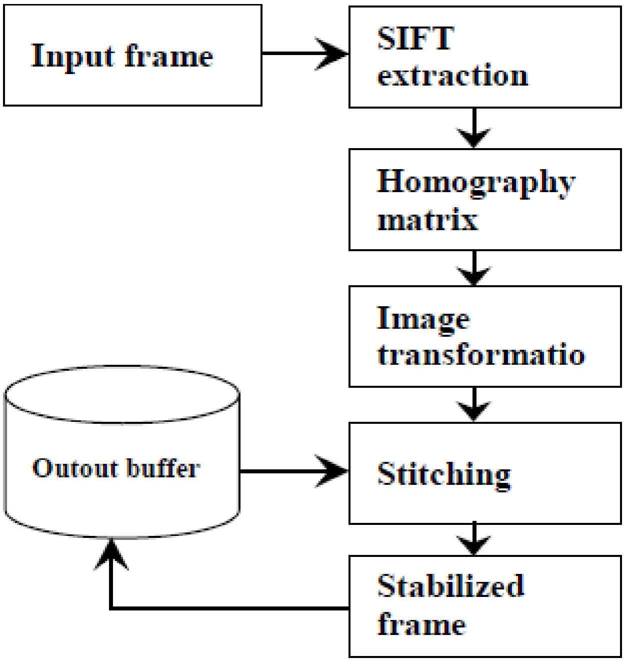


Figure 12: Stabilization with stitching algorithm’s over flow

The computational complexity of the algorithm is reduced by reducing the dimension of the SIFT as described in Section 4. 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 color value of that part of the image. The whole procedure of stitching for stabilization with some examples are explained step by step from the next paragraph-

Step1: SIFT Keypoints in current_frame (Kpt1) and next_frame (Kpt2) are extracted. Keypoints matching is done as described in section 3.1.5 by Brute-force matches Figure 13. Say, points in current_frame and next_frame after matching are (Xc,Yc) and (Xn,Yn). A homography matrix H (3x3) is calculated between (Xc,Yc) and (Xn,Yn) using equation (3).

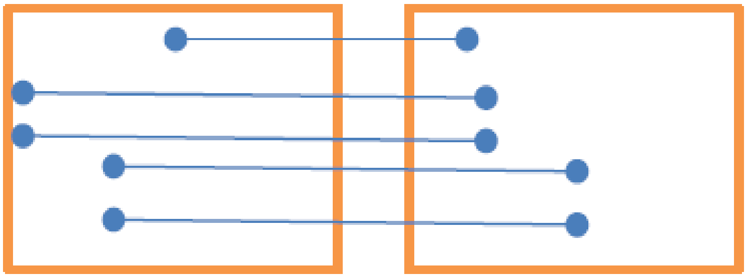


Figure 13: SIFT matching between current_frame and next_frame

Step2: A new blank output image of size S is created as in Figure 14. At first the current_frame is copied in the (0, 0) location of the black image to get the initial output image.

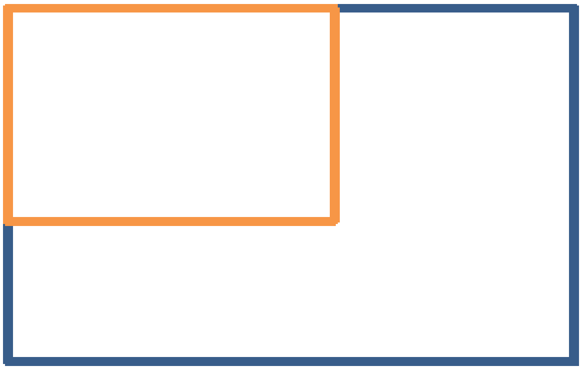


Figure 14: Created initial output image for stabilization

Step3: Next_frame is transformed 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 (5), where H11, H12, H13 , …, H33 are the 9 element of the H matrix.

$$\mathbf{Dest(Xi,Yi)=Source(}\frac{\mathbf{H11*Xi+H12*Yi+H13}}{\boldsymbol{H31*Xi+H32*Yi+1}},\ \frac{\boldsymbol{H21*Xi+H22*Yi+H23}}{\boldsymbol{H31*Xi+H32*Yi+1}}\ \mathbf{)} \qquad \text{---- (5)}$$

For example the pixel value at (1, 2) point in transformed image can be calculated with the **H**$\begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{2} \\ \mathbf{2} & \mathbf{1} & \mathbf{3} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}$. Pixel value at (1, 2) = pixel value at next_frame (5, 6). Figure 15 shows an example of transformed image.

Step4: P1’(X1, Y1), P2’(X2, Y2), P3’(X3,Y3), P4’(X4,Y4) points in the transformed image[Figure 15] can be similarly calculated by the following equation(6)-

$$\mathbf{(X', Y')} = \begin{bmatrix} \boldsymbol{H}\mathbf{11} & \mathbf{H12} & \boldsymbol{H}\mathbf{13} \\ \mathbf{H21} & \mathbf{H22} & \boldsymbol{H}\mathbf{23} \\ \boldsymbol{H}\mathbf{31} & \boldsymbol{H}\mathbf{32} & \boldsymbol{H}\mathbf{33} \end{bmatrix} \mathbf{*(X, Y)} \qquad \text{---- } \mathbf{(6)}$$

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 the homography matrix is $\mathbf{H}\begin{bmatrix} \mathbf{1} & \mathbf{1} & \mathbf{2} \\ \mathbf{2} & \mathbf{1} & \mathbf{3} \\ \mathbf{0} & \mathbf{0} & \mathbf{1} \end{bmatrix}$, then the corner points in the transformed image are as shown in equation (7).

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

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

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

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

----- (7)

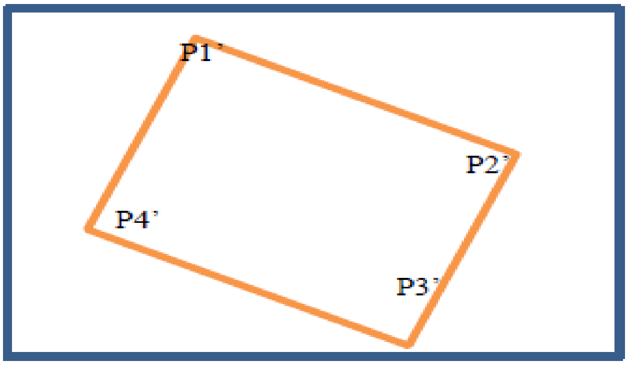


Figure 15: Transformed next_frame according to homography matrix

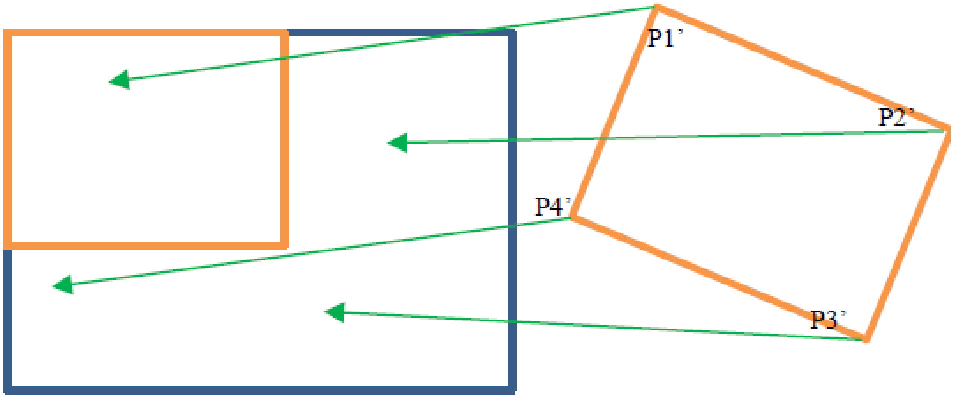


Figure 16(a): Previous output image, the arrow line shows where the bounding part of P1', P2', P3', P4' will go in the output image

Figure 16(b): the bounding part to be copied from transformed image to previous output from Figure 10.

Step5: Now to stitch the transformed next_frame [Figure 15] with the initial output image[Figure 14], the bounding part of transformed corner points, P1'(2,3), P2'(642,1283), P3'(1002,1643) and P4'(362,363) , calculated in step4 is copied from the transformed image to the output image at the same position as shown in the Figure 16(a),(b) and (c).

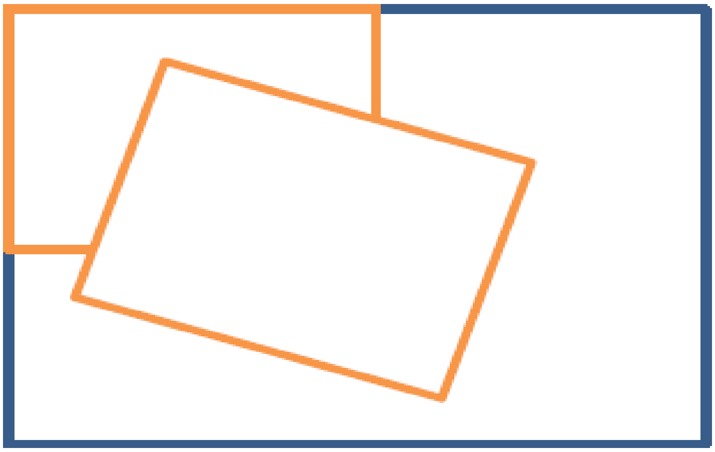
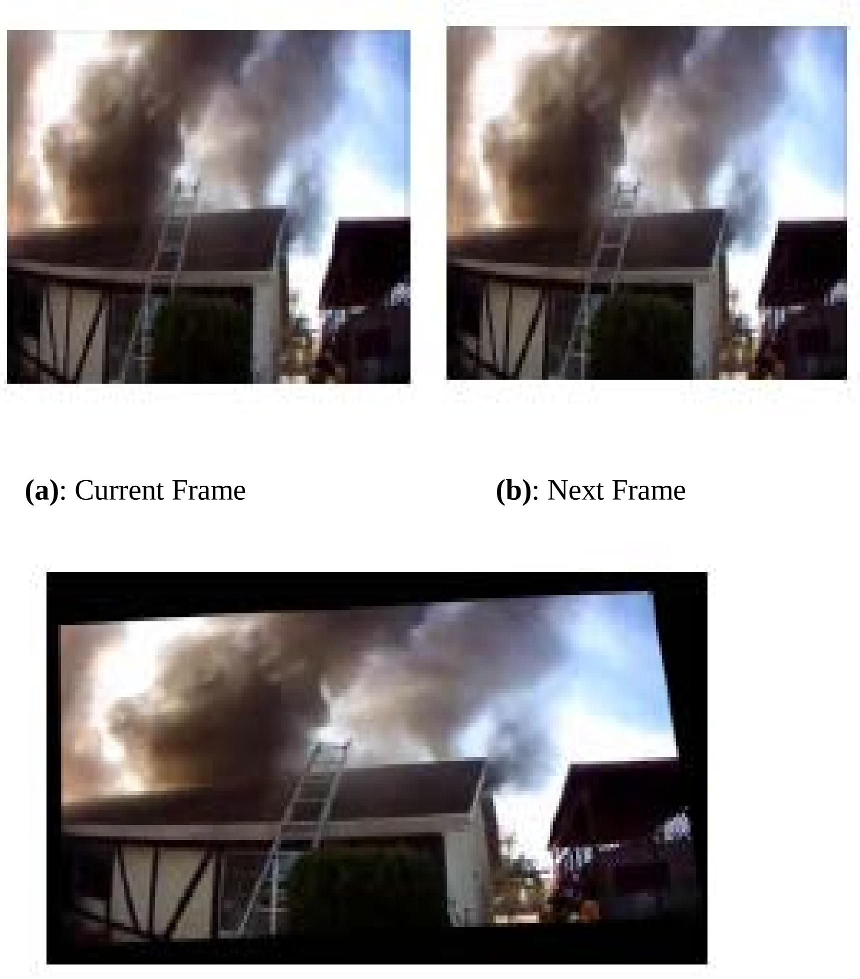


Figure 16(c): Created new stabilized output image

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

For example, Figure 17(a) and Figure 17(b) shows current frame and next frame before performing matching and transforming feature points from the test unstable video. Figure 17(c) shows transformed next frame according to the steps described in section 5.



(a): Current Frame

(b): Next Frame

(c): Transformed next frame

Figure 17: (a) shows current frame, (b) next frame and (c) next frame after transformation

Now this transformed image is stitched with the initial output as described in described in **step3** to get the stabilized frame from the same view point. For the test sequences, 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 as the first frame is the reference frame. And from the next iteration next frames are only grabbed as the previous next from becomes current frame for the next iteration. In next every iteration transformed frame is estimated. From the transformed image as shows in Figure 17(C) only the color part removing the black part is copied to the output image and the output image is saved for next iteration in the image buffer. The algorithm continues until there is no more next_frames available in the video. So, stitching is done in every iteration to get the stitched stabilized frames. Result of the stabilization algorithm is shown in section 6. Figure 20 and Figure 21 shows experimental results and comparison between proposed and traditional method of stabilization.

6. 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. Different size of videos was used to measure the speed and efficiency. The results of proposed stabilization with stitching are compared with the traditional stabilization algorithms to show the difference between the traditional and proposed algorithm. Experimental results also shows how proposed SIFT is computationally faster than the D. Lowe’s Traditional SIFT [6]. How performance of the matching algorithm is improved by is also shown in this section.

6.1. Improved Matching

The threshold value to remove some of the wrong matches described in section 4.2.2 is set to 50 pixels. Several experiments are done with different videos to justify the threshold value. The experimental results shows that, if the threshold value is lesser than 50, the percentage of correct matches and the number of matches decreases which means some of the correct matches are removed reducing the threshold value. On the other hand, making the threshold value larger than 50 reduces correct matches’ percentage whereas the average number of matches increases, which means increasing the threshold value further more includes some of the wrong matches in the matches set.

Table 1 shows, how the correct matches and the number of matches are changed with varying the threshold value. All the calculations are done with first

60 frames of the videos. Threshold value is selected to 50 because, if the threshold value is reduce the number of total matches increases and the percentage of correct matches reduces, which means some of the correct matches are removed in the matches. On the other hand if the threshold value is increased the number of total matches increases whereas the percentage of correct matches recues, which means some of the wrong matches are included in the matches

	Threshold	Avg. Correct matches (%)	Avg. Number of matches (%)
Video1	40	99.51	39.99
	45	99.52	40.05
	50	99.54	40.18
	55	99.35	40.36
	60	99.27	40.45
Video2	40	99.19	42.25
	45	99.21	42.68
	50	99.24	42.87
	55	99.13	43.13
	60	99.01	43.2
Video3	40	98.63	27.38
	45	98.63	27.46
	50	98.14	27.55
	55	98.11	27.73
	60	98.11	27.73

Table 1: Experimental results with different threshold values

The correctness of the matching procedure is improved a lot with the proposed approach described in section 4.2.2. Experiments have been done with different unstable videos and the results shows the proposed approach give more than 98% correct matches [Table2]. 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 [8] & [9] but the results cannot be directly compared as, the previous works deals with still images and the proposed algorithm mainly considers video sequences. In [11] and [13], they uses video frames for improving matches, their results gives 90% of matching results which is lower than the proposed approach.

	Video1	Video2	Video3	Video4
Avg. Correct Match before (%)	88.89	91.05	88.02	81.23
Avg. Correct Match (%)	99.54	99.21	98.64	98.42

Table 2: Comparison of correct Matches before and after

6.2. Fast SIFT

The computational complexity is reduced by reducing the number of keypoints and size of descriptor as described in section 4.2 . The Figure 18 shows that by reducing the descriptors dimension how the time complexity reduces of the algorithm. It also shows that by restricting the number of SIFT keypoints to be extracted by 60 the time complexity can be dramatically reduced. Figure 19 shows how the computational complexity of the algorithm is affected by after performing the matching improvements. It shows, even after matching improvements are done, with 24 dimensional descriptor vector and redirecting the keypoints to 60, takes lesser time than with 128 dimensional original SIFT without performing any matching improvements.

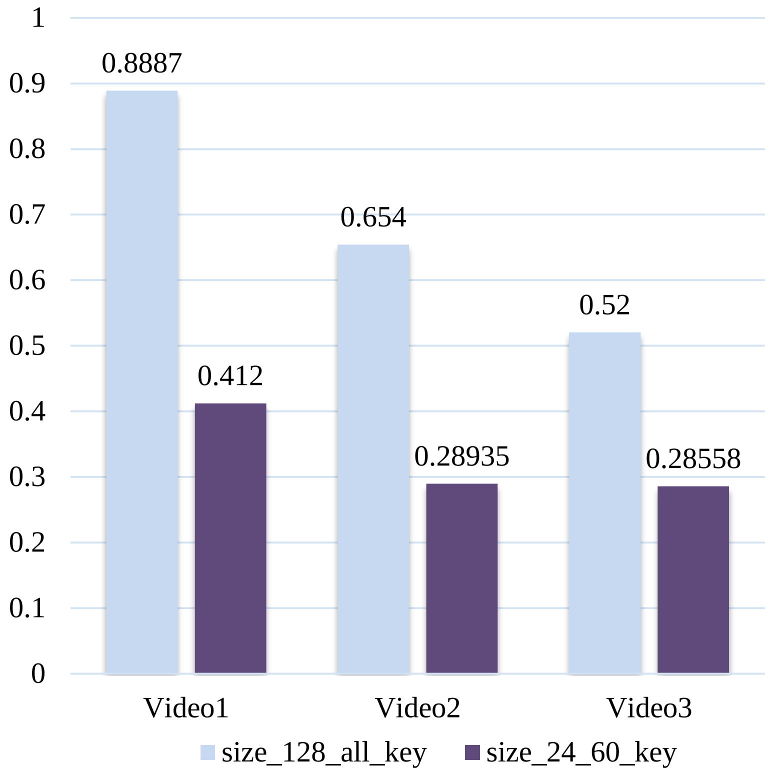


Figure 18: Time complexity comparison after reducing the descriptor size and restricting the number of keypoints to be extracted

In Zhu Qidan and Li ke proposed algorithm [7] which performs similar approach, the SIFT 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,

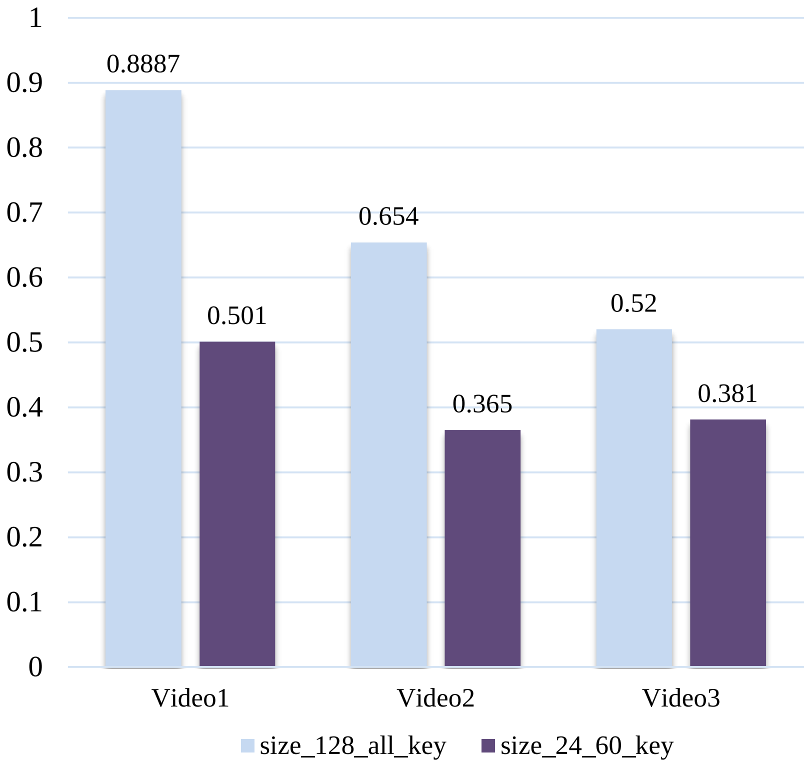


Figure 19: comparison of time complexity after matching improvements

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 3 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.

Unstable Videos	Accuracy(%) of stabilization with different descriptors 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

Table3: Accuracy test with different descriptors size

6.3. Stabilization Results

Proposed algorithm is applied to different unstable videos having different kinds of shaky motion to test the quality of stabilization of the algorithm. The results are also compared with the traditional stabilization methods to show that the results are better with the proposed algorithm. Figure 19a shows frame1, frame30 and frame50 from an unstable video. Figure 19b shows stabilized frames with traditional methods done by motion estimation and distinguishing between local and global motions as described in section 2.2. Figure 19c shows stabilized frames with proposed algorithm by stitching images to get stabilized frames.

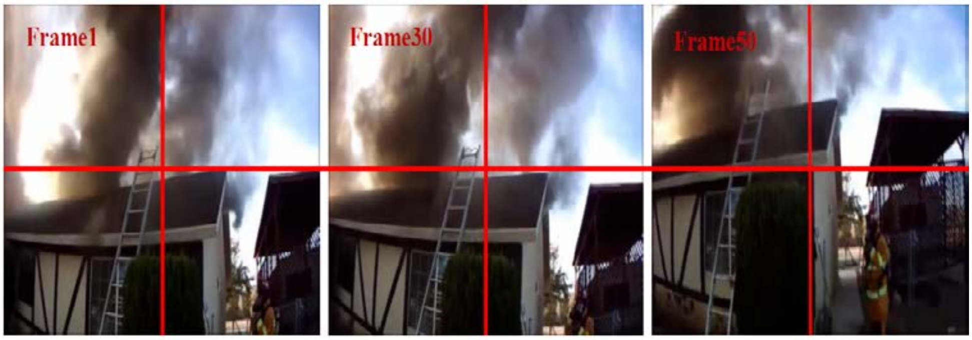


Figure 20(a): Unstable video frames

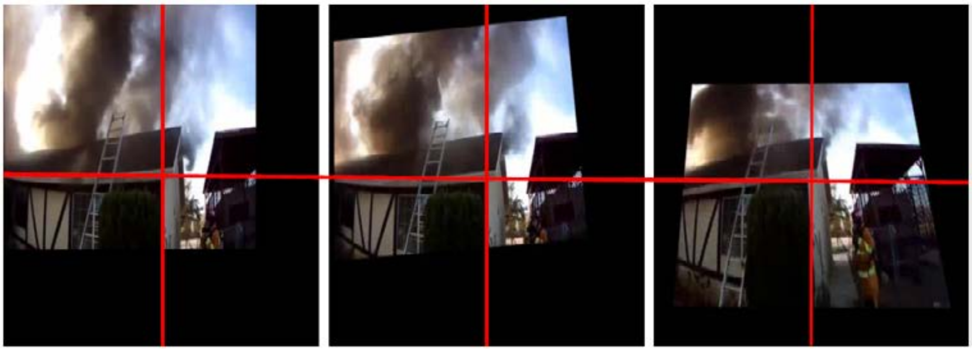


Figure 20(b): Stabilized frames with traditional approach

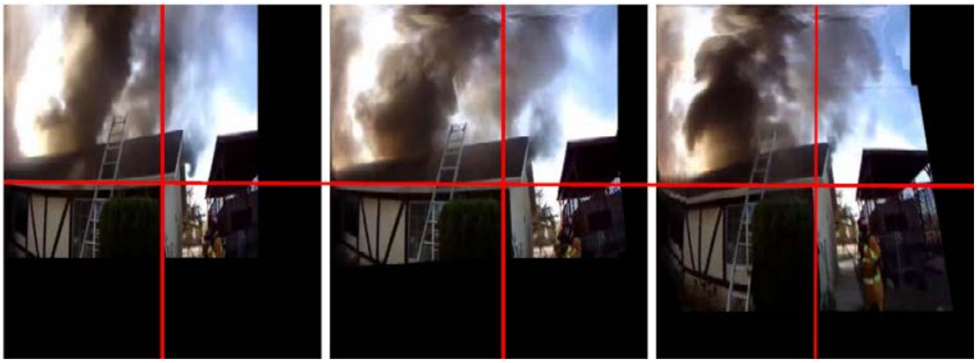


Figure 20(c): Stabilized frames with proposed approach

As in the proposed algorithm SIFT keypoints are used, the stabilization algorithm is also rotation invariant. This means even though the unstable videos have rotational motion, it can be compensated and the video sequence can be stabilized with the proposed algorithm. Figure 20 shows an example of stabilization where the unstable video had rotational camera motion. 20(a) shows unstable, Frames – frame1, frame60 and frame 160 from an unstable video.

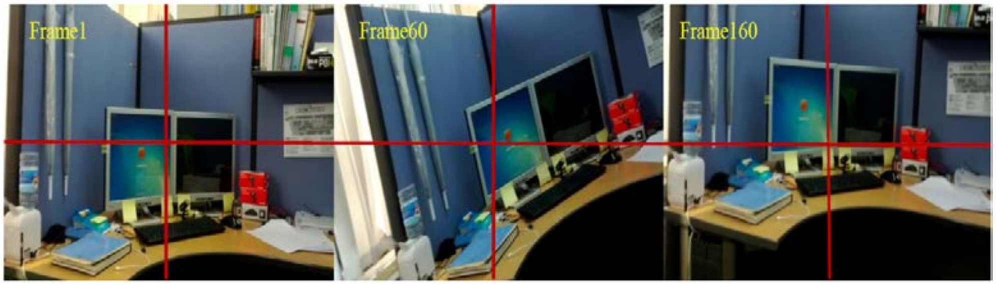


Figure 21(a): Unstable frames with rotational motion

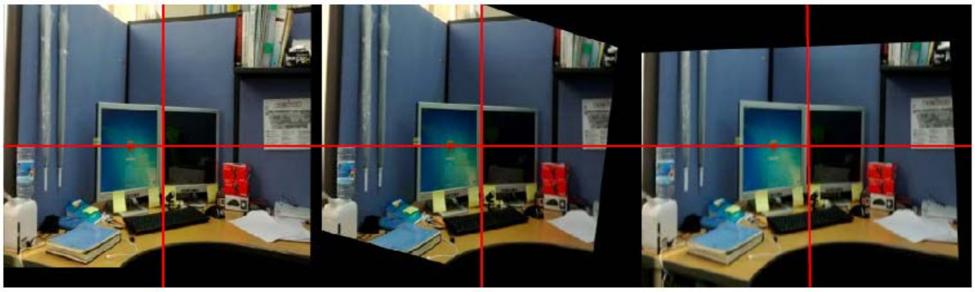


Figure 21(b): Stabilized frames with traditional approaches

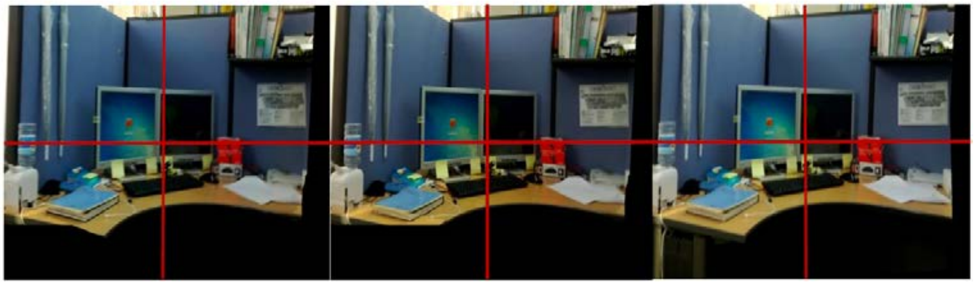


Figure 21(c): Stabilized frames with proposed approach

The stabilized frames with the proposed approach in Figure 20(c) show that it performs much better than the traditional stabilization methods in Figure 20(b). It not only removes the camera motions but also retains the previous frames data as stabilization is done by stitching transformed frames together in every iteration.

7. Conclusions

An improved stabilization method with image stitching is proposed in this research paper. By following this method the hassle of estimating global, local motions and distinguishing between them or filtering global motion 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 without any prior information of camera parameters as SIFT (Scale Invariant Feature Transform) feature points are used for image stitching. Also, as image stitching is done for stabilization, the stabilized frames shows previous frames data as well and most of the black parts is removed compared to the traditional approach Experimental results shows that the proposed algorithm perform have much better results than the traditional motion based stabilizations.

Restricting the search area for matching keypoints in the corresponding frames by calculating the frame wise minimum displacement helps improving the matching results which directly effects on classification error and helps reducing it. Also, Matching results directly effects on the homography matrix calculation, so it very important to calculate the correctness corresponding matches to improve the accuracy of the stabilization. Experimental results shows percentages of correct matches are more than 98% in all the test videos.

Further, the modified SIFT descriptor vector with smaller size compared to the traditional SIFT increases the performance of the algorithm. Restricting the number of SIFT keypoints and by extracted region wise SIFT keypoints helps to reduce the measurement error. This helps improving the quality of the stabilized frames as well as improves the performance of the whole algorithm. Experimental results show how computational complexity is reduced by reducing the descriptor size of the SIFT keypoints and restating the number of keypoints to be extracted in every frame.

The complete algorithm gives a video stabilization method with image stitching, which is simply but effective, have better performance and takes lesser time than other existing motion based video stabilization algorithms.

Bibliography

[1] Stephane Auberger and Carolina Miro. Digital video stabilization architecture for low cost devices”. In. *Proceedings of the 4th International Symposium on Image and signal Processing and Analysis*. In. 2005

[2] Seok-Woo Jang, Marc Pomplun, Gye-Young Kim and Hyung-II Choi . “Adaptive robust estimation of affine parameters from block motion vectors”. In. *Image and Vision Computing 23*. In. 2005.

[3] Filippo Vella, Alfio Castorina, Massimo Mancuso and Giuseppe Messina. “Digital image stabilization by adaptive block motion vectors filtering”. In. *IEEE Transactions on Consumer Electronics, Vol. 48, No. 3*. In 2002.

[4] David G. Lowe. “Distinctive image features from scale-invariant keypoints”. In. *International Journal of Computer Vision, Vol. 60, No.2.* In 2004.

[5] Gary Bradski and Adrian Kaebler’s Book “Learning OpenCV”. In. 2008.

[6] Martin A. Fishler and Robert C. Bolles “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography” . In. *Communication ACM 24.6.* In.1981.

[7] Zhu Qidan and Li Ke. “Image stitching Using Simplified SIFT”, In. *Proceedings of the IEEE International conference on Information and Automation.* In. 2010

[8] Z. Yi , C. Zhiguo and X. Yang “Multi-Spectral remote image registration based on SIFT”. In. *IEEE electronics letter Vol. 44, No. 2*. In. 2008.

[9] Pengfei Du, Ya Zhou, Qiaona Xing and Xiaoming Hu. “Improved SIFT matching Algorithm for 3D reconstruction from Endoscopic Images”. In.

VRCAI: Virtual Reality Continuum and its Applications in Industry. In 2011.

[10] Robert Laganiere’s book. “OpenCV 2 computer Vision Application Programming Cookbook”. In. 2011.

[11] Ting Chen. “Video Stabilization algorithm Using a Block-Based Parametric Motion Model” In. *Stanford University, EE392J Project Report winter.* In. 2000.

[12] Li Lingjun and Long- Xiang. “LSIFT- an Improved SIFT algorithm with a New Matching Methode”. In. *International Conference on Computer Application and System Modelling.* In. 2010.

[13] Zheng Ying and Li-Da-Hui. “Video Image Tracing Based on Improved SIFT Feature Matching Algorithm”. In. Journal of Multimedia, Vol.9, No.1. In 2014.

[14] Cansın Yıldız." An Implementation on Recognizing Panoramas ". In. *Department of Computer Engineering Bilkent University Ankara, Turkey*. In. 2005.

[15] Zhao Xiuying, Wang Hongyu. “Medical Image Seamlessly Stitching by SIFT and GIST”. In. *E-Product E-Service and E-Entertainment International Conference.* In. 2010.

[16] Luo Juan and Oubong Gwun. “SURF applied in Panoram Image Stitching”. In. *Image Processing Theory, Tools and Applications IEEE*. In. 2010.

[17] Matthew Brown and David G. Lowe. "Automatic panoramic image stitching using invariant features". In. *International Journal of Computer Vision, Vol.74, No. 1*. In. 2007.

[18] Hongyan Wen and Jianzhong Zhou1. “An Improved Algorithm for Image Mosaic”. In. *International Symposium on Information Science and Engineering*. In.2008.

[19] Paul Bao and Dan Xu. “Complex wavelet-based image mosaics using edge preserving visual perception modeling*”. In. Computer and graphics Vol. 23, No. 3*. In 1999.

[20] Steve Haynal and Heidi Hayanal. “Brute-Force search of Fast convolution algorithms”. In. *Acoustics, Speech and Signal Processing, IEEE International Conference*. In. 2013.

[21] Zoghlami I, Faugeras O and Deriche R. “ Using geometric corners to build a 2D mosaic from set of images”. In. *Proceeding of IEEE conference on Computer Vision and Pattern Recognition*. In 1997.

[22] Minqi Zhou and Vijayan K Asari. “A Fast Video Stabilization System Based on Speed-Up Roboust Features”. In. *International Symposium on Visual Computing, part II*. In. 2011. pp 428-435.

[23] Sebastiano Battiato , Giovanni Gallo, Giovanni Puglisi and Salvatore Scellato. “SIFT Features Tracking for Video Stabilization”. In. *International Conference on Image Analysis and Processing, IEEE*. In.2007.

[24] Labeeb Mohsin Abdullah, Nooritawati Md Tahir and Mustaffa Samad. “Video Stabilization based on Point Feature Matching Technique”. In. *IEEE Control and System Research Colloquium*. In. 2012.

[25] Junlan Yang, Dan Schonfeld, Chong Chen and Magdi Mohamed. “Online Video Stabilization based on Particle filters”. In. *Proceedings of the IEEE International Conference on Image Processing*. In 2006.

[26] K. Madhavi, B Sreekant Reddy and Ch. Ganapathy Reddy. “Weighted Feature Points Extraction based Video Stabilization”. In. *International Journal of Science and Modern Engineering, Vol.1, No.10*. In. 2013.

[27] S. Erturk. “Image Stabilization based on Kalman Filtering of Frame positions”. In. *IEEE Electronics Letters, Vol.37, No.20*. In 2001.

[28] Steve Haynal and Heidi Hayanal. “Brute-Force search of Fast convolution algorithms”. In. *Acoustics, Speech and Signal Processing, IEEE International Conference*. In. 2013.

[29] Luo Juan and Oubong Gwun. “A Comparison of SIFT, PCA-SIFT and SURF”. In. *International Journal of Image Processing, Vol.3, No.4*. In. 2009.

[30] Cheng-Yuan Tang, Yi-Leh Wu, Maw-Kae Hor and Wen-Hung Wang. “Modified SIFT Descriptor for Image Matching under Interference”. In. *Proceedings of the Seventh International Conference on Machine Learning and Cybernetics*. In. 2008.

[31] Li-Lingjun and Long-Xiang. “LSIFT-An Improved SIFT Algorithm with A New Matching Method”. In. *International Conference on Computer Application and System Modeling*. In. 2010.

ABSTRACT

최근 몇 년 동안 휴대용 카메라, 소방관의 머리 카메라, 로봇 카메라의 사용이 증가하고 있다. 그러나 이러한 카메라에 의해 촬영 된 동영상은 일반적으로 불안정 하며 원치 않는 카메라 움직임으로 가득하다. 따라서 디지털 비디오 안정화를 위한 수요는 증가하고 있다.

비디오 안정화를 위한 연구 중에는 이 블록 기반 움직임 보상과 포인트(feature point) 기반 움직임 보상을 통한 비디오 안정화 에 대한 것이 있다. 그러나 이러한 알고리즘은 많은 계산량을 요구하고 또한 소방관 의 머리의 카메라 에 의해 생성된 영상과 같이 큰 움직임을 가지는 영상을 처리할 수 없다. 본 논문에서는 Image stitching을 이용한 영상 안정화 방법 이 제안한다. 제안 알고리즘은 SIFT (Scale-Invariant Feature Transform)의 계산량은 감소시켜서 사용하지만, 매칭 정확도는 향상되었다.

비디오 안정화를 위해 SIFT (Scale-Invariant Feature Transform)와 image stitching을 사용하였다. SIFT 특징점이 불안정한 두 개의 비디오 영상에서 적절한 수학적 모델과 픽셀값을 통해 추출된 후 서로 다른 영상 사이에서 호모 그래피 행렬을 이용하여 매칭이 수행된다. 다음 영상의 모든 픽셀은 현재의 영상으로 호모 그래피 행렬 과 화소값에 따라 변환된다.

변환된 영상은 반복적으로 stitching되며 이를 통해 안정화된 영상을 얻는다.

SIFT의 매칭을 개선하기 위해 특징점의 검출영역을 이용한다. KNN 매처와 초기에 매칭한 후 매칭된 특징점은 특정 조건보다 큰 경우에 매칭 가능 리스트에서 지워진다.

SIFT의 계산량을 줄이기 위해 descriptor 벡터의 크기를 128에서 24로 줄였다. 또한 추출할 SIFT 특징점의 수 역시 조건을 이용하여 제한하였다. 지역기반의 SIFT 특징점 추출 방법을 본 논문에서 제안한다. 각각의 특징점들은 영상을 지역 기반으로 분할 된 뒤 추출되며 추출된 지역 기반 특징점들을 병합하여 영상 전체의 특징점으로 사용한다. 이런 방식을 통해 특징점들이 전체 영상에 대해 잘 분포될 수 있다.

Keywords: Video Stabilization, Image Stitching, SIFT, Image Processing

Student number: 2012-23951