DOI: 10.11591/telkomnika.v12i7.5855

Digital Image Stabilization Based on Improved Scale Invariant Feature Transform

Xiaoran Guo*, Shaohui Cui, Dan Fang
Ordnance Engineering College, Shijiazhuang 050003, China
*Corresponding author, e-mail: vip850522@163.com

Abstract

A novel digital image stabilization approach using Harris and Scale Invariant Feature Transform (SIFT) was presented in this article. Using SIFT in digital image stabilization, too many feature points and matches were extracted, but some of them were not so stable. Using these feature points and matches can not only increase the computational effort, but also enhance the wrong matching probability. We proposed to use SIFT to detect feature points and incorporate the Harris criterion to select the most stable feature points in the video sequence where image motion was happened due to vehicle or platform vibration. With these feature points, we use general feature descriptor and matching algorithm to achieve the image stabilization. Experimental results show that the proposed algorithm can not only bring down the probability of wrong matching and get more accurate matches, but also reduce the computation burden than SIFT effectively.

Keywords*: digital image stabilization, SIFT, Harris*

1. Introduction

Video stabilization techniques have been studied for decades to improve visual qualities of image sequences captured by digital video cameras which are hand held or mounted on unstable platform or vehicle, the captured video generally looks shaky because of undesired camera motions. Unwanted video vibrations would lead to degraded view experience and also greatly affect the performances of applications like video coding, video surveillance, object detection, target tracking, image navigation and guidance. Video stabilization is becoming an indispensable technique in industrial, military and consumer applications fields.

The video stabilization systems can be classified into three major types: electronic image stabilization (EIS), optional image stabilization (OIS), and digital image stabilization (DIS) [1]. The EIS stabilizes the image sequence by employing motion sensors, such as gyroscope and accelerometer, to detect the camera movement for compensation. The OIS adopts a prism assembly which moves opposite the shaking of camera for stabilization [2]. The applications of EIS and OIS are restricted to device, because both are hardware dependent. DIS is the process of removing the undesired motion affects to generate a stable image sequence by using digital image processing techniques without any mechanical devices such as gyroscope, accelerometer, or a fluid prism. DIS system essentially consists of two units: the motion estimation unit and the motion compensation units.

The motion estimation unit estimates the global motion parameters between every two consecutive frames of the input video sequences. With these global motion parameters, the motion compensation unit then generates the compensating motion parameters needed to compensate for the jitter of a frame and creates a more visual stable image sequence. Unlike most motion estimation techniques, in DIS the robustness of the motion estimation is critical associated with the fact that an improper estimate will yield an abrupt jitter in the video sequence.

Various algorithms have been developed to estimate the local motion vectors. Block matching [3] is the conventional way to detect the global motion vector between two consecutive frames. In this method, each image is divided into squares, rectangles or circulars [1] of a certain dimension, and then motion search is executed to find the motion vector for each block in current frame. With these local motion vectors, clustering method [3] is used to find the main motion vector which is recognized as the global motion vector. Because of the tedious

Received February 24, 2013; Revised March 25, 2014; Accepted April 12, 2014

computational cost of block matching with full pixel information, some schemes have been proposed to speed up the process with only partial pixel information, such as representative point matching scheme [4], selected areas matching scheme [5], bit-plane matching scheme [6] and gray-coded bit-plane matching scheme [7]. These schemes can reduce the computation complexity, but also lower the motion estimation accuracy. Besides block matching method with full or partial pixel information, optical flow [8] based method has been proposed for image stabilization. In optical flow algorithm the constraint is the first-order approximation of Taylor series, which means that the motions can not be very large. Otherwise, the higher order terms of Taylor series which are ignored, can lead to significant incorrect estimation. Some feature matching based algorithms, such as phase correlation scheme [9], and edge pattern matching scheme [10] have also been proposed in DIS. These schemes are lack of adaptation to affine and perspective motions.

The more effective methods for affine and perspective motion estimation are local features based ones. Harris [11] corner features are extracted and tracked in order to estimate global motion [12]. Scale invariant feature transform (SIFT) [13] features, considered to be invariant to image rotation and scaling, are being widely used in the latest methods for global motion estimation [14], image matching [15] and image registration [16]. Also many works are proposed to improve the distinctiveness of SIFT descriptor, such as principal component analysis (PCA)-SIFT [17] which was less distinctive than the SIFT descriptor proved by [18], and speeded up robust features (SURF) [19] which is fast in matching and sufficiently distinctive, but weaker than SIFT in the invariant of scaling and rotation. Such above-mentioned approaches mainly focus on the development of descriptors, but in this paper, we care more about the extraction of the most stable feature points, which can not only bring down the probability of wrong matching and get more accurate matches, but also effectively reduce the computational effort. By carefully reviewing the survey on feature point detectors [20], we note that Harris corner could provide stable detection performance with high repeatability and localization accuracy under various distortions and geometric transform. Therefore, we propose to incorporate the Harris criterion in SIFT to select the most stable feature points for image stabilization. With these most stable feature points, we use general feature descriptor and matching algorithm to generate matches, and use those matches to calculate geometrical transform parameters. Finally, we transform one image using the geometrical transformation matrix to align with the other image, and accomplish image compensation.

2. Robust Local Feature Points Extraction

Local features, such as corners, blobs, and regions, have been widely used for object detection, recognition, and retrieval purposes in computer vision. The intrinsic advantages of these local features are their invariance under geometric transforms. A comprehensive review of the state-of-the-art local features can be found in [18]. Among various local feature detectors and descriptors, SIFT was shown to be relatively optimal considering the trade-off between robustness, distinctiveness, and efficiency, and Harris corner could provide stable detection performance with high repeatability and localization accuracy under various distortions and geometric transform.

The original SIFT algorithm consists of the following four major steps:

(1) Scale-space extreme detection
(2) Accurate feature points localization
(3) Orientation assignment
(4) Feature points descriptor

Where steps 1 and 2 are the feature detection and steps 3 and 4 are the description and generation of the feature points.

Our method uses the step 1, step 3 and step 4 of the original SIFT algorithm, but in step 2, when the potential feature points have all found, and the points of the edge and the lower contrast points were eliminated. Yet there are still too many feature points, and some of them are not so stable, if adopting all the feature points to execute the feature matching, not only the computational effort will be tremendous, but also the wrong matching probability will be high. To further identify the most stable feature points, we incorporate the Harris criterion to select the most stable SIFT feature points. The underlying reason is that such a criterion helps keep the most stable local patterns with higher gradients distribution but rejects the feature points with lower gradient distribution. Also, the Harris-based criterion is self-adaptive and can yield relatively stable detection performance.

The steps of the proposed algorithm are:

(1) Scale-space extreme detection

The first stage of computation searches over all scales and image locations to find local extrema. It is implemented efficiently by using a series of difference-of-Gaussian (DOG) images in the scale space σ to identify potential feature points that are invariant to scale and orientation.

The image scale space is expressed as a function $L(x,y,\sigma)$, that is generated from the convolution of a series of Gaussian kernel functions $G(x,y,\sigma)$ with consecutively incremental scales, with an input image $I(x,y)$:

$$L(x,y,\sigma)=G(x,y,\sigma)*I(x,y) \tag{1}$$

Where $*$ denote the convolution operator, and the Gaussian function $G(x,y,\sigma)$:

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

The scale space $D(x,y,\sigma)$ is set up from the difference of two nearby scales by a constant multiplicative factor k.

$$\begin{aligned} D(x,y,\sigma) &= L(x,y,k\sigma)-L(x,y,\sigma) \\ &= (G(x,y,k\sigma)-G(x,y,\sigma))*I(x,y) \end{aligned} \tag{3}$$

2) Accurate feature points localization with Harris criterion

When the potential feature points have all found, to further identify the more stable feature points, we will eliminate the unstable points, such as the points of the edge and the lower contrast points. Only the stable feature points remain. According to Taylor expansion, let the derivative of $D(x)$ equal zero after an offset $\hat{x}$ set can be found. This $\hat{x}$ can take a pixel location with true local extreme value, then substitute $\hat{x}$ into the Taylor expansion. If a value of $|\mathrm{D}(\hat{\mathrm{x}})|$ is less than 0.03, we will say the point has low contrast. The $D(x)$ is computed below.

$$D(x)=D+\frac{\partial D^{\mathbf{T}}}{\partial x}x+\frac{1}{2}x^{\mathbf{T}}\frac{\partial^2 D}{\partial x^2}x+\cdots \tag{4}$$

$$\hat{x}=-\frac{\partial^2 D^{-1}}{\partial x^2}\frac{\partial D}{\partial x} \tag{5}$$

$$D(\hat{x})=|D+\frac{1}{2}\frac{\partial D^{\mathbf{T}}}{\partial x}\hat{x}| \tag{6}$$

After eliminating the points with low contrast, we will cancel the points of the edge perhaps. Here we use a 2×2 Hessian matrix $\mathbf{E}$ which enables us to compute the scale and location of the feature point. The matrix is given by:

$$\mathbf{E}=\begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix} \tag{7}$$

Where D_{xx} is the second derivative of the image in the x-direction, and D_{xy} is the first derivative of the image in the x-direction and y-direction respectively, D_{yy} is the second derivative of the image in the y-direction. Then, $\mathbf{E}$ is done in the decomposition of eigenvalues to get the two eigenvalues: α and β.

We use the ratio of the eigenvalues as follows:

$$\frac{Tr(\mathbf{E})}{Det(\mathbf{E})} = \frac{(\alpha+\beta)^2}{\alpha\beta} = \frac{(\gamma+1)^2}{\gamma} \tag{8}$$

$$\alpha = \gamma\beta \tag{9}$$

The following inequality must be satisfied, if not, we say that this point is probably on the edge, and it can be eliminated like this:

$$\frac{Tr(\mathbf{E})}{Det(\mathbf{E})} < \frac{(\gamma+1)^2}{\gamma} \tag{10}$$

So far, we get the a set of SIFT points $P(x_n, y_n, \sigma_n)$, where x_n is the horizontal ordinate, y_n is the vertical ordinate, and σ_n is the scale parameters of the SIFT feature point n. Suppose that there are total N feature points, so $n = 1,2,3,...N$. Then, we add the Harris criterion to select the most stable SIFT feature points. The Harris criterion is based on the autocorrelation matrix, which represents the gradient distribution within a local region of the selected point. We adopt the x_n, y_n, σ_n to calculate the Harris response $R_{\sigma_n}(x_n, y_n)$, where σ_n is the standard deviation of the Gaussian kernel function as the weighted window used to compute the Harris autocorrelation matrix M, which represents the gradient distribution within a local region of the selected point. The Gaussian kernel window in this paper is 3×3. The autocorrelation matrix M of at feature point $P(x_n, y_n, \sigma_n)$ is represented as:

$$M = \sum_{(x,y)\in W(x_n,y_n)} \omega(x,y) \begin{bmatrix} I_x{}^2 & I_x I_y \\ I_x I_y & I_y{}^2 \end{bmatrix} \tag{11}$$

$W(x_n, y_n)$ is the Gaussian kernel window with the (x_n, y_n) as the center to determine the accumulated region, and $\omega(x,y)$ is the Gaussian kernel function of the weighted window, I_x and I_y are the image gradients in x-direction and y-direction. So I_x^2 is the product of the image gradients in x-direction. We use the Gaussian kernel function $\omega(x,y)$ as the weighted window to make the matrix isotropic. If both eigenvalues of matrix M, λ_1 and λ_2 are sufficiently large positive values, this point is a corner point. Harris proposed a formula to evaluate the corner points instead of computing the two eigenvalues as:

$$R = \lambda_1\lambda_2 - \kappa(\lambda_1 + \lambda_2) = \det(M) - \kappa trace^2(M) \tag{12}$$

Where κ is a coefficient with value 0.04-0.06. We set 0.06 as its default value in this paper. $\det(M)$ is the determinant of matrix M, and $trace(M)$ is the trace of matrix M. We calculate the $R_{\sigma_n}(x_n, y_n)$ of each feature point $P(x_n, y_n, \sigma_n)$, and calculate the mean value of them. Finally, we set the mean value as the threshold to select robust SIFT feature points:

$$Threhold = \frac{1}{N}\sum_{n=1}^{N} R_{\sigma_n}(x_n, y_n) \tag{13}$$

If the $R_{\sigma_n}(x_n, y_n)$ of feature point is greater than $Threhold$, we keep this feature point and take it as a most stable feature point, otherwise reject this feature point. With this threshold, we can reserve the most stable feature points with higher gradients distribution, and reject the feature points with lower gradient distribution, and then we could extract stable feature point in images under geometric transforms even under the attack of blurring. Due to this threshold is based on Harris algorithm, it is self-adaptive and can yield relatively stable detection performance.

3) Orientation Assignment

One or more orientations are assigned to each feature point location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.

After finding the feature points, the following step will define the magnitude and orientation for each feature point in order to use image matching. The orientation of each feature point is determined by the peak of the orientation histogram formed by the gradient orientations within its neighborhood. After finding the major orientation for an image feature point, when doing the image matching again, we can rotate to the same orientation in two images, and thus achieve rotation-invariance. Here we are using the distribution of the gradient direction for feature points taken as a consistent orientation to each feature point. Therefore, we will compute the gradient magnitude and the orientation for each image sample $L(x,y)$. The computation of the gradient magnitude and the orientation in pixels are given as below.

$$m(x,y)=\sqrt{(L(x+1,y)-L(x-1,y))^2+(L(x,y+1)-L(x,y-1))^2} \tag{14}$$

$$\theta(x,y)=\tan^{-1}((L(x,y+1)-L(x,y-1))/(L(x+1,y)-L(x-1,y))) \tag{15}$$

After the gradient directions of all pixels within a region around the feature point are computed, we calculate the direction of the most points supporting the primary direction. The weight of each point adjacent to the central pixels is decided using the product of Gaussian distributions and gradient magnitude of the point, where Gaussian distribution is a setting at $\sigma=0.5$.

4) Feature Points Descriptor

The local image gradients are measured at the selected scale in the region around each feature point. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination. Here we adopte an approach which is basically based on the orientation histogram. The computation and description of the image gradient is done by using the feature points and adjacent pixels, we employ a 16×16 local neighborhood block around the feature points, then divide it up into 4×4 sub-blocks again, and define the eight directions, therefore, we treat the $4\times4\times8=128$ dimensional vector as the feature points descriptor. The feature vector is then normalized to decrease the illumination alteration effects.

3. Feature Matching and Image Compensate

Once the most stable feature points have been detected in the images, the next step is to match them. In this paper, feature points are matched by constructing KD-tree with best-bin-first search, the advantage of KD-tree is reducing the number of feature that needs to be searched. The feature descriptors of the reference frame are clustered by building KD-tree. A feature is then randomly selected to be the partition node. Each feature extracted from the current frame traverses the built KD-tree to find its corresponding point with best-bin-first search. The pair of matching features in two different images is called a correspondence. When all correspondences have been found, the transformation matrix can be computed using random sample and consensus (RANSAC). In this study, the focus will be on stabilization of aerial images where the distance of the camera to the scene is much larger than changes in scene depth, or if depth variation is large, the camera motion is small enough so that the frames being registered have negligible local geometric difference. We will use projective transformation to register the image frames. Assume that the number of mismatches can be reduced by using our improved approach and KD-tree with best-bin-first search, small amount of mismatches still occur, and this may lead to unreliable prediction of motion estimation. Thus, a further check of matching errors is a significant part of the algorithm. We use the well known RANSAC algorithm for the purpose of both elimination the mismatches and estimation of the parameters of the projective transform model. Based on the correspondences, RANSAC can iteratively compute the parameters of the motion model by detecting the inliers and outliers among these matching pairs. Traditionally, the geometric transformation between two images can be described by a homography which is a 3D model.

The procedure to calculate the homography can be expressed as follows:

(1) We begin with a simple liner algorithm for determining homography $\mathbf{H}$ by giving a set of four point correspondences, $\mathbf{P}_i \leftrightarrow \mathbf{P}_i'$, which are not three points collinear. The transformation is given by equation $\mathbf{P}_i' = \mathbf{H}\mathbf{P}_i$.

(2) Compute a similarity transformation $\mathbf{T}$ for the points in the reference image, consisting of a translation and scaling, that takes points $\mathbf{P}_i$ to a new set of points $\tilde{\mathbf{P}}_i$, the points are translated and scaled so that the centroid of the points $\tilde{\mathbf{P}}_i$ is at the coordinate origin $(0,0)^{\mathbf{T}}$, and their average distance from the origin is equal to $\sqrt{2}$. So coordinates $\mathbf{P}_i$ in reference image are replaced by $\tilde{\mathbf{P}}_i = \mathbf{T}\mathbf{P}_i$. Compute a similar transformation $\mathbf{T}'$ for the points in the current image, transforming points $\mathbf{P}_i'$ to $\tilde{\mathbf{P}}_i'$, that is coordinates $\mathbf{P}_i'$ in current image are replaced by $\tilde{\mathbf{P}}_i' = \mathbf{T}'\mathbf{P}_i'$. $\mathbf{T}$ and $\mathbf{T}'$ are 3×3 matrixes. Substituting in the equation $\mathbf{P}_i' = \mathbf{H}\mathbf{P}_i$, we derive the equation $\tilde{\mathbf{P}}_i' = \mathbf{T}'\mathbf{H}\mathbf{T}^{-1}\tilde{\mathbf{P}}_i$. This relation implies that $\tilde{\mathbf{H}} = \mathbf{T}'\mathbf{H}\mathbf{T}^{-1}$ is the transformation matrix for the point correspondences $\tilde{\mathbf{P}}_i \leftrightarrow \tilde{\mathbf{P}}_i'$. Note that the equation $\tilde{\mathbf{P}}_i' = \tilde{\mathbf{H}}\tilde{\mathbf{P}}_i$ involves homogeneous vectors, thus the 3-vectors $\tilde{\mathbf{P}}_i'$ and $\tilde{\mathbf{H}}\tilde{\mathbf{P}}_i$ are not equal, they have the same direction but may differ in magnitude by a nonzero scale factor. The equation may be expressed in terms of the vector cross product as $\tilde{\mathbf{P}}_i' \times \tilde{\mathbf{H}}\tilde{\mathbf{P}}_i = 0$. This form will enable a simple linear solution for $\tilde{\mathbf{H}}$ to be derived. If the j-th row of the matrix $\tilde{\mathbf{H}}$ is denoted by $\tilde{\mathbf{H}}_j^{\mathrm{T}}$, then we may write:

$$\tilde{\mathbf{H}}\tilde{\mathbf{P}}_i = \begin{pmatrix} \tilde{\mathbf{H}}_1^{\mathrm{T}}\tilde{\mathbf{P}}_i \\ \tilde{\mathbf{H}}_2^{\mathrm{T}}\tilde{\mathbf{P}}_i \\ \tilde{\mathbf{H}}_3^{\mathrm{T}}\tilde{\mathbf{P}}_i \end{pmatrix} \tag{16}$$

Writing $\tilde{\mathbf{P}}_i' = (\tilde{x}_i', \tilde{y}_i', \tilde{\omega}_i')^{\mathrm{T}}$, the cross product may then be given explicitly as:

$$\tilde{\mathbf{P}}_i' \times \tilde{\mathbf{H}}\tilde{\mathbf{P}}_i = \begin{bmatrix} \tilde{y}_i'\tilde{\mathbf{H}}_3^{\mathrm{T}}\tilde{\mathbf{P}}_i - \tilde{\omega}_i'\tilde{\mathbf{H}}_2^{\mathrm{T}}\tilde{\mathbf{P}}_i \\ \tilde{\omega}_i'\tilde{\mathbf{H}}_1^{\mathrm{T}}\tilde{\mathbf{P}}_i - \tilde{x}_i'\tilde{\mathbf{H}}_3^{\mathrm{T}}\tilde{\mathbf{P}}_i \\ \tilde{x}_i'\tilde{\mathbf{H}}_2^{\mathrm{T}}\tilde{\mathbf{P}}_i - \tilde{y}_i'\tilde{\mathbf{H}}_1^{\mathrm{T}}\tilde{\mathbf{P}}_i \end{bmatrix} \tag{17}$$

Since $\tilde{\mathbf{H}}_j^{\mathrm{T}}\tilde{\mathbf{P}}_i = \tilde{\mathbf{P}}_i^{\mathrm{T}}\tilde{\mathbf{H}}_j$, for $j = 1,2,3$, this gives a set of three equations in the entries of $\tilde{\mathbf{H}}$, which may be written in the form:

$$\begin{bmatrix} \mathbf{0}^T & -\tilde{\omega}_i'\tilde{\mathbf{P}}_i^{\mathrm{T}} & \tilde{y}_i'\tilde{\mathbf{P}}_i^{\mathrm{T}} \\ \tilde{\omega}_i'\tilde{\mathbf{P}}_i^{\mathrm{T}} & \mathbf{0}^T & -\tilde{x}_i'\tilde{\mathbf{P}}_i^{\mathrm{T}} \\ -\tilde{y}_i'\tilde{\mathbf{P}}_i^{\mathrm{T}} & \tilde{x}_i'\tilde{\mathbf{P}}_i^{\mathrm{T}} & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{H}}_1 \\ \tilde{\mathbf{H}}_2 \\ \tilde{\mathbf{H}}_3 \end{bmatrix} = \mathbf{0} \tag{18}$$

That is,

$$\begin{bmatrix} 0 & 0 & 0 & -\tilde{\omega}'_i\tilde{x}_i & -\tilde{\omega}'_i\tilde{y}_i & -\tilde{\omega}'_i\tilde{\omega}_i & \tilde{y}'_i\tilde{x}_i & \tilde{y}'_i\tilde{y}_i & \tilde{y}'_i\tilde{\omega}_i \\ \tilde{\omega}'_i\tilde{x}_i & \tilde{\omega}'_i\tilde{y}_i & \tilde{\omega}'_i\tilde{\omega}_i & 0 & 0 & 0 & -\tilde{x}'_i\tilde{x}_i & -\tilde{x}'_i\tilde{y}_i & -\tilde{x}'_i\tilde{\omega}_i \\ -\tilde{y}'_i\tilde{x}_i & -\tilde{y}'_i\tilde{y}_i & -\tilde{y}'_i\tilde{\omega}_i & \tilde{x}'_i\tilde{x}_i & \tilde{x}'_i\tilde{y}_i & \tilde{x}'_i\tilde{\omega}_i & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{h}_0 \\ \tilde{h}_1 \\ \tilde{h}_2 \\ \tilde{h}_3 \\ \tilde{h}_4 \\ \tilde{h}_5 \\ \tilde{h}_6 \\ \tilde{h}_7 \\ \tilde{h}_8 \end{bmatrix} = \mathbf{0} \qquad (19)$$

These equations have the form $\mathbf{A}_i\tilde{\mathbf{h}} = \mathbf{0}$, where $\mathbf{A}_i$ is a 3×9 matrix, and $\tilde{\mathbf{h}}$ is a 9-vector made up of the entries of the matrix $\tilde{\mathbf{H}}$, with $\tilde{h}_i$ the i-th element of $\tilde{\mathbf{h}}$.

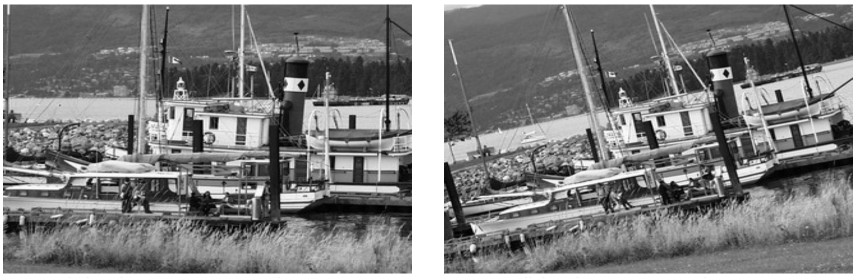
$$\tilde{\mathbf{h}} = \begin{bmatrix} \tilde{\mathbf{H}}_1 & \tilde{\mathbf{H}}_2 & \tilde{\mathbf{H}}_3 \end{bmatrix}^{\mathrm{T}}, \tilde{\mathbf{H}} = \begin{pmatrix} \tilde{h}_0 & \tilde{h}_1 & \tilde{h}_2 \\ \tilde{h}_3 & \tilde{h}_4 & \tilde{h}_5 \\ \tilde{h}_6 & \tilde{h}_7 & \tilde{h}_8 \end{pmatrix} \qquad (20)$$

Although there are three equations in (18), only two of them are linearly independent (since the third row is obtained, up to scale, from the sum of $\tilde{x}'_i$ times the first row and $\tilde{y}'_i$ times the second). Each point correspondence gives rise to two independent equations in the entries of $\tilde{\mathbf{H}}$. Given a set of four such point correspondences, we obtain a set of equations $\mathbf{A}\tilde{\mathbf{h}} = \mathbf{0}$, where $\mathbf{A}$ is the matrix of equation coefficients built from the matrix rows $\mathbf{A}_i$ contributed from each correspondence, and $\tilde{\mathbf{h}}$ is the vector of unknown entries of $\tilde{\mathbf{H}}$. Executing the SVD (Singular Value Decomposition) of $\mathbf{A}$, the unit singular vector corresponding to the smallest singular value is the solution $\tilde{\mathbf{h}}$, and the matrix $\tilde{\mathbf{H}}$ is determined from $\tilde{\mathbf{h}}$. Then we can calculate homography $\mathbf{H}$ by $\tilde{\mathbf{H}} = \mathbf{T}'\mathbf{H}\mathbf{T}^{-1}$.

(3) We calculate the error of a correspondence from homography $\mathbf{H}$ using the symmetric transfer error, that is $d^2_{transfer} = d(\mathbf{P}_i,\ \mathbf{H}^{-1}\mathbf{P}'_i)^2 + d(\mathbf{P}'_i,\ \mathbf{H}\mathbf{P}_i)^2$. If the $d^2_{transfer}$ is less than the given threshold value, then confirming the correspondence inliers, otherwise outliers. Repeat for a number of samples, and record the largest number of inliers. Finally, re-estimate $\mathbf{H}_R$ from all correspondences classified as inliers, and using the final homography $\mathbf{H}_R$ to implement image compensate.

4. Results and Analysis

To test the stabilization ability of the proposed algorithm and compare it with traditional SIFT algorithm, some image pairs with different changes are used.

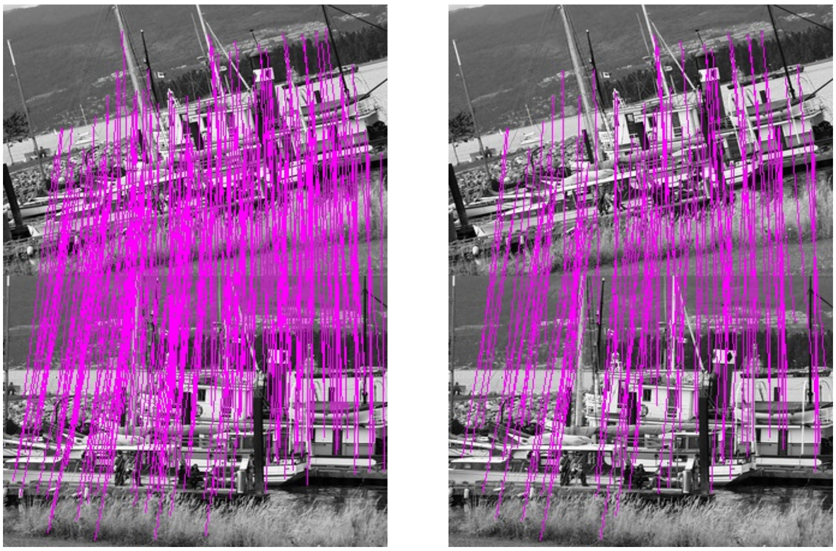


(a) The reference image (b) The current image

Figure 1. Input Images (a) Reference Image and (b) Current Image

Figure 1 is the input images, with scaling and rotational variations. Figure 1(a) is the reference image, and Figure 1(b) is the current image. As we can see, Figure 1(b) takes great scaling and rotational change compared to Figure 1(a).

Figure 2 shows the matching points from SIFT algorithm and our algorithm. As we can see, SIFT generates too many feature points and matches. At the same time, our algorithm generates less feature points and matches but with high correctness, which makes the subsequent calculation of transform very promising. This allows us to gain better accuracy than SIFT.

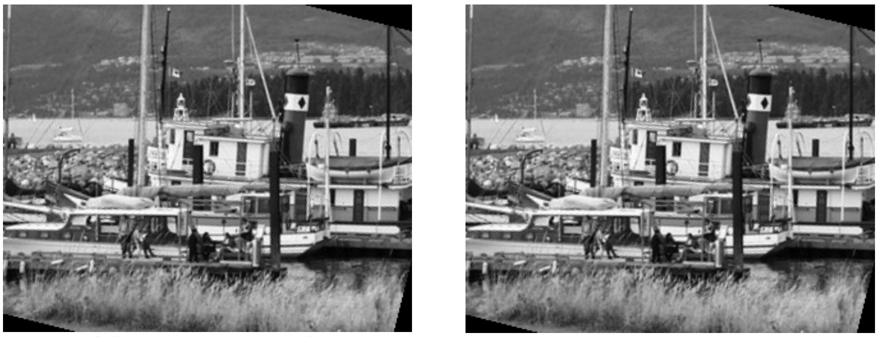


(a) SIFT approach

(b) Proposed approach

Figure 2. Matching Results from (a) SIFT and (b) the Proposed Approach

Figure 3(a) shows the compensated image using SIFT, and Figure 3(b) shows the compensated image using our algorithm.



(a) SIFT approach

(b) Proposed approach

Figure 3. Stabilization Images from (a) SIFT and (b) the Proposed Approach

We also test the stabilization ability of the two algorithms using the images with illumination changes, viewpoint changes, and blur.

Numerical evaluation of the quality of the image stabilization is fulfilled using peak signal-to-noise ratio (PSNR) as error measure. PSNR between frame t and frame $t+1$ is defined as:

$$MSE\,(t) = \frac{1}{MN}\sum_{y=1}^{M}\sum_{x=1}^{N}[I_t(x,y) - I_{t+1}(x,y)]^2 \tag{21}$$

$$PSNR\,(t) = 10\log_{10}\frac{I_{MAX}^{2}}{MSE\,(t)} \tag{22}$$

Where $MSE(t)$ is the mean-square error between frames, I_{MAX} is the maximum intensity value of a pixel, and M and N are the frame dimensions. $PSNR$ measures the similarity between two images, hence, it is useful to evaluate how much a image is stabilized by the algorithm by simply evaluating the similarity of the two images.

In order to prove the stabilization effect of the algorithm, we calculated the $PSNR$ between each frame of the successive 50 frames unstable video sequences, the experimental results are shown in Figure 4. The curve at the bottom represent the $PSNR$ of the original non-compensated video sequences, the middle one is the $PSNR$ of compensated video sequences using SIFT method, and the above one is the $PSNR$ of compensated video sequences using the proposed approach.

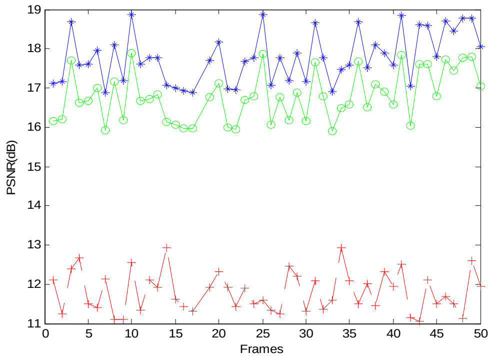


Figure 4. Experimental Results of PSNR

At the same time, we use the execution time to evaluate the efficiency of the algorithms. In Table 1, execution time comparisons among various variations between SIFT and our approach are shown. As can be seen from Figure 4 and Table 1, the $PSNR$ of our approach is a little bigger than SIFT under the various variations, and the execution time of our approach is much less than SIFT, so our approach outperforms the SIFT approach.

Table 1. Execution Time Comparisons

variation	execution time of SIFT	execution time of our approach
rotations and scale	6203.7ms	3394.9ms
illumination	2385.6ms	1711.0ms
viewpoint	4153.5ms	2895.6ms
blur	3716.0ms	2868.1ms

5. Conclusion

In conclusion, we propose a novel robust local feature point extraction approach in image stabilization. First, we use SIFT algorithm detect the feature point of the image, then adopt Harris criterion to select the most robust feature points. The Harris $Threhold$ is self-adaptive and can yield relatively stable detection performance. Experimental results show the proposed scheme yields better performances than SIFT, it has a little better stabilization accuracy, and saves a lot of computation burden.

References

[1] Xu LD, Lin XG. Digital image stabilization Based on Circular Block Matching. *IEEE Transactions on Consumer Electronics*. 2006; 52: 566-574.

[2] Sato K. Control techniques for optical image stabilizing system. *IEEE Transactions on Consumer Electronics.* 1993; 39: 461-466.

[3] Vella F, Castorina A, Mancuso M. Digital image stabilization by adaptive block motion vectors filtering. *IEEE Transactions on Consumer Electronic*. 2002; 48: 796-800.

[4] Uomori K. Automatic image stabilizing system by full-digital signal processing. *IEEE Transactions on Consumer Electronics*. 1990; 36: 510-519.

[5] Engelsberg A, Schmidt G. A comparative review of digital image stabilising algorithms for mobile video communications. *IEEE Transactions on Consumer Electronics.* 1999; 45: 592-597.

[6] Ko SJ, Lee SH, Lee KH. Digital image stabilizing algorithms based on bit-plane matching. *IEEE Transactions on Consumer Electronics.* 1998; 44: 617-622.

[7] Ko SJ, Lee SH, Jeon SW. Fast digital image stabilizer based on gray-coded bit-plane matching. *IEEE Transactions on Consumer Electronics.* 1999; 45: 598-603.

[8] Chang JY, Hu WF, Cheng MH. Digital image translation and rotation motion stabilization using optical flow technique*. IEEE Transactions on Consumer Electronics.* 2002; 48: 108-115.

[9] Erturk S. Digital image stabilization with sub-image phase correlation based global motion estimation, *IEEE Transactions on Consumer Electronics.* 2003; 49: 1320-1325.

[10] Paik JK, Park YC, Kim DW. An adaptive motion decision system for digital image stabilizer based on edge pattern matching. *IEEE Transactions on Consumer Electronics*. 1992; 38: 607-616.

[11] Harris C, Stephens M. *A combined corner and edge detection.* Proc. 4th Alvey Vision Conference. 1988; 189-192.

[12] Zhu JJ, Guo BL. Electronic image stabilization system based on global feature tracking. *IEEE Transactions on Consumer Electronics*. 2008; 19: 228-233.

[13] Lowe DG. Distinctive Image Features from Scale-Invariant Key Points. *International Journal of Computer Vision.* 2004; 60: 91-110.

[14] Yang JL, Schonfeld D, Mohamed M. Robust Video Stabilization Based on Particle Filter Tracking of Projected Camera Motion*. IEEE Transactions on Circuits and Systems for Video Technology*. 2009; 19: 945-954.

[15] Can Sun, Jin-ge Wang, Zaixin Liu, Junmin Li. Weighted Multi-Scale Image Matching Based on Harris-SIFT Descriptor. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2013; 11(10): 5965-5972.

[16] Quan Sun, Jianxun Zhang. Parallel Research and Implementation of SAR Image Registration Based on Optimized SIFT. *TELKOMNIKA Indonesian Journal of Electrical Engineering*. 2014; 12(2): 1125-1131.

[17] Yan K, Sukthankar R. *Pca-sift: A more distinctive representation for local image descriptors*. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. 2004; 506-513.

[18] Mikolajczyk K. A performance evaluation of local descriptors*. Journal of the Chinese Institute of Engineers*. 2005; 27: 1615-1630.

[19] Bay H. *Surf: Speeded up robust features.* Computer Vision and Image Understanding. 2008; 110: 1615-1630.

[20] Tuytelaars T. Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision.* 2008; 3: 177-208.