

An Electronic Image Stabilization Algorithm Based on Efficient Block Matching on the Bitplane

Luo Fang, Qin Xiaozhen

Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan, China
Email: luofang8816@163.com

Received 2012

ABSTRACT

This paper proposes an electronic image stabilization algorithm based on efficient block matching on the plane. This algorithm uses a hexagonal search algorithm, and uses the bit-planes to estimate and compensate for the translational motion between video sequences at the same time; As for the rotary motion vector generated in the video sequences, in order to highlight the intensity change of the image sequence, the algorithm firstly conducts Laplace transform for the reference frame, then select a number of characteristics at the image edge to make block matching with the current frame, calculate and compensate for the rotational movement that may exist finally. Through theoretical analysis and simulation, we prove that, as for a mixed translational and rotational motion video sequences, the proposed algorithm can reduce required time for block matching computation, while improving the accuracy of the electronic image stabilization.

Keywords: Electronic Image Stabilization Algorithm; Bit Plane; Block Matching; Hexagon Motion Estimation

1. Introduction

At present, video signal acquisition system has been installed in the motive carrier, the strenuous exercise on the carrier can cause jitter of the camera imaging. The electronic image stabilization is a new generation video stabilization technology, which integrates electronics, computers, digital signal processing and other technology. It can be widely used in military and civilian, such as in the automatic camera recognition system of the vehicle license. In the commonly used image stabilization algorithm, the bit-plane method has characteristics such as less computation, high accuracy and so on. It uses flat images to replace the gray-scale images which can be realized quickly binary matching, and then determine the image motion vector. This reduces the computational complexity of image processing, improved stability speed, but the method exists defects when applied to the image with a rotary motion. As for the problem, this paper further integrate the the block matching method to assess the rotary motion that may exist, on the basis of using the bit-plane method to estimate the translational motion of the video sequences, and thus to obtain a more accurate image stabilization effect.

2. The bit-plane method

2.1. The plane decomposition and the Gray code of bit-planes

The main idea of the bit-plane matching algorithm is

using plane images to replace the gray-scale images, in order to achieve fast matching of the image block. According to the literature [1], on the (x, y) position of the t -th frame of image sequences, which has 2^k gray scale, the pixel gray values can be expressed as:

$$G(x, y) = b_k \times 2^k + b_{k-1} \times 2^{k-1} + \dots + b_1 \times 2^1 + b_0 \times 2^0 \quad (1)$$

In which $G(x, y)$ express the pixel gray values on the (x, y) position, b_k as a Boolean value, its value can be 0 or 1. Thus, the Boolean matrix constituted by the k -th bit plane of the image can be said as $b_k(x, y)$, each of these values are 0 or 1, which can simplify the calculation. A level corresponds to every bit plane, the level of the k -th bit plane is 2^k , thus a k -bit grayscale images can be regarded as constituted by k binary bit plane images $b_k(x, y)$. For example, a 256 grayscale image, $k = 8$, after the transformation of the bit-planes, you can get 8 levels of bit plane, in which $b_0(x, y)$ represents the lowest level of image pixels, while $b_8(x, y)$ represents the highest level in the pixels, each plane describes only part of the image information, the larger amount of information is contained in the high level bit plane.

As shown in Figure 1, we can get 8 bitmap by using bit plane decomposition.

But it is noteworthy that, a flaw exists when using the formula 1 to get a bit plane: tiny change of pixel gray value may make an enormous impact on its bit plane decomposition. For example, the gray level is 127, the number of its bit is 01111111, 128 is 10000000, the

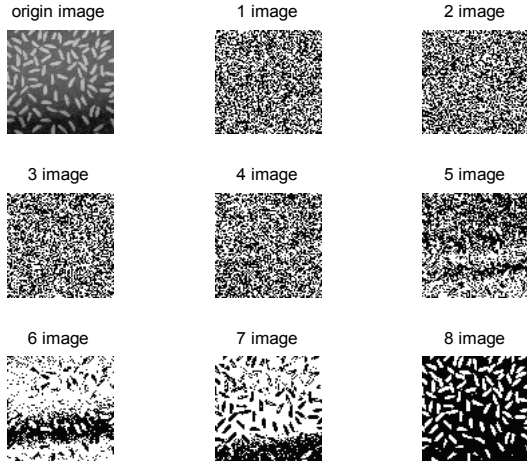


Figure 1 The bit-plane decomposition diagram

grayscale only differs one level but number of bits of the eight bit plane are all different. Therefore, in order to reduce impact when small gray value changes occur in plane, we converse the bit values of the image before matching, improve the highest k bit-planes using the Gray Code [2], the conversion formula is as follows:

$$g_i = \begin{cases} b_i & i = k - 1 \\ b_i \oplus b_{i+1} & 0 \leq i \leq k - 2 \end{cases} \quad (2)$$

Where \oplus is the XOR operation, b_i is the i -th Boolean value in the above formula 1, so b_i can be replaced by g_i in formula 1. By using Gray Code, adjacent digital only differ in one bit. For example, when the gray level are 127 and 128, according to formula 2, their grayscale are 01000000 and 11000000 respectively, so 127 and 128 adjacent gray-scale only differ in one bit.

2.2.Bit-plane matching

After the above-mentioned plane extraction and re-encode, the obtained plane can be used for image matching. Assume that at the time $t-1$, the size of matched block is in the reference frame, the search area for the current image is at the t -time, Select the k -th bit-plane to match, as and respectively, and we can get the bit-planes matching measure of the two images by the following formula (3):

$$C_k(m, n) = \frac{1}{M \times N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} b_k^{t-1}(x, y) \oplus b_k^t(x + m, y + n) \quad (3)$$

Which in the video sequence, $C_k(m, n)$ is the average number of bits that do not match that between the search area and matching region of the reference frame in the current frame, the smallest value corresponds to the point

of best matching, (m, n) is the motion vector, its pointing direction is the interface vector direction of movement, and the angle is:

$$J_k^t = \arg \min \{C_k(m, n)\} \quad (4)$$

3.Hexagon search algorithm

Hexagonal matching algorithm is a coarse to fine search method, it is improved by the Diamond Search Algorithm (DS) and can further reduce the complexity of the searching. It first began to match with the seven points (including a search window center point), later select the matching points for searching according to certain principles, the diagram that how to choose the matching points for searching is shown in Figure 2, and the search algorithm described as follows:

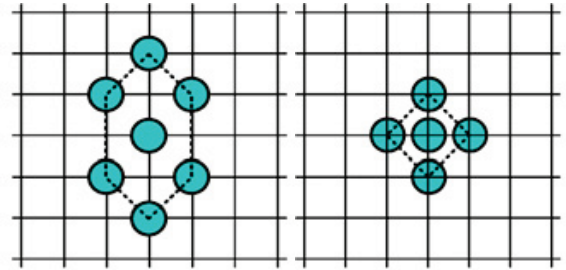


Figure 2 Hexagonal search algorithm

As shown in Figure 2, the algorithm consists of two kinds of search mode: seven search points hexagon search pattern (LHSP) and five search points of the small diamond search pattern (SDSP), in which LHSP search mode is used for coarse positioning and SDSP mode is used for fine positioning. The search strategy is as follows:

- (1) Use the LHSP in the search window center to carry out a search, if the most small pieces of distortion point (MBD) appear in the center point of the LDSP, go to step (3), otherwise go to step (2);
- (2) Use the MBD point in the step (1) as the center to continue the LHSP mode, if the MBD point appears in the center, go to step (3), otherwise repeat this step (2);
- (3) Take the MBD point in the above step as the center to continue SDSP search, further to find the minimum block distortion point in a small diamond-shaped template as the final match point.

The bit plane selection is important when matching. Select only a single plane to match will affect the matching accuracy due to the limited information contained in the image. However, introduction too much of matching

plane, will affect the speed of matching. Therefore, each step in the above search uses only one single plane, different search steps use different bit-planes, not only retains a wealth of image information, but also does not affect the speed of matching.

High-order bit plane contain the most on the vision very important data, so the hexagonal search algorithm uses 4-th plane in step (1), 5-th plane in step (2), 6-th plane in step (3) [3].

The above algorithm is based on an assumption that the motion vector change is not very severe and the vertical jitter is larger in the inter-frame. In response to the scene that the horizontal jitter accounts for the main problems, hexagonal search pattern also provides another search mode, shown in Figure 3. It can do selection depending on the application characteristics, the algorithm is the same with the above.

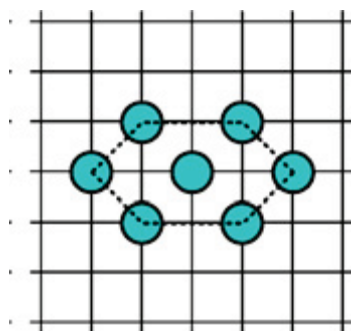


Figure 3 Hexagonal search patterns which emphasis on horizontal jitter

4. The description of Image Stabilization Algorithm

4.1. The motion model of image

The affine motion model is a two-dimensional motion model which describes the translation and the rotational motion of the object by linear transformation. In the two-dimensional coordinate plane, points (x_i, y_i) rotates the angle θ around the center, then gets the point (x_j, y_j) by the following affine transformation formula:

$$\begin{pmatrix} x_j \\ y_j \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} \quad (5)$$

If there are translational motions in the center, you can use the following transformation formula to describe, in where m and n are the displacement of the point in the horizontal and vertical direction.

$$\begin{pmatrix} x_j \\ y_j \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} m \\ n \end{pmatrix} \quad (6)$$

Where, when θ is a positive number, it represents the counterclockwise rotation, and it represents a clockwise rotation when θ is negative.

Based on the above affine transformation model, the actually captured global motion between consecutive two images can be approximate manifested to the translational motion and rotational motion around the optical axis: in reference frame, coordinate points (x_i, y_i) becomes (x_j, y_j) after the angle θ of rotation, and then after a translational motion in the current frame, it becomes (x_k, y_k) . According to this model, we will compensate for the translational motion and rotational motion in the following.

4.2. Calculation of translation vector

Combine bit-plane matching with hexagon searching strategy, select four 64×64 sub-images in the four corners of the image, match each sub-image to obtain local motion vector, and finally derive global motion vector of the image after median filtering. Algorithm can be realized only by using binary logical operations, which significantly reduces the computation; the use of median filtering, can also overcomes the unreliable local motion vector which may bring impact to the global motion of the image. θ is a very small rotation angle between two consecutive frames in video sequences, and the image gray value changes it brings is, Thus, bit-plane algorithm can accurately calculate the image translational motion vector D_x and D_y . Compensate the translational component for the current image frame, there may be only rotation vector difference between the compensated image and the the reference frame, and the rotation center is in the center of the image.

4.3. Calculation of rotation angle

The center of rotation is in the center of the image, so the changes that brought by the rotary motion is the most obvious at the edge of the image. Select a number of small reference blocks at the edge of the reference frame image, match them with the image block that has compensated for translational motion so we can obtain translational motion vectors of the reference block due to the rotation, and then combine with the center coordinates of the reference block and the center coordinates of the entire frame to determine the rotation angle. The specific process is as follows.

In order to include regions that has rich-changes in gray level information, we firstly conduct Laplace transform to the reference frame when selecting block, and then select four points having a maximum of the gray value at the edge of the transformed image, choose a small piece of 8×8 taking these four points as the center. Figure 4 shows the rotation model. (x_i, y_i) is the center coordinates of the reference frame matching block i , due to the rota-

tion effects in the current frame, the center coordinates of the matching block i becomes (x_i', y_i') . Translational motion vectors δx and δy can be obtained by above block matching, and $x_i' = x_i + \delta x$, $y_i' = y_i + \delta y$. The following inverse trigonometric functions can be used to calculate the corresponding angle of (x_i, y_i) :

$$q_i = \arctan\left(\frac{y_i - N/2}{x_i - M/2}\right) \quad (7)$$

the corresponding angle of (x_i', y_i') :

$$q_i' = \arctan\left(\frac{y_i' - N/2}{x_i' - M/2}\right) \quad (8)$$

Where, $(M/2, N/2)$ is the center coordinates of the image. And the rotation angle of the matching block i is $\delta\theta_i = q_i - q_i'$.

Perform median filtering to all rotation angle of the matching block, we can estimate the rotation angle of the entire image $\delta\theta$. Rotate the current image frame, that has been previously compensated for the translational motion, reversely for further compensation, to make the final image frame the translation and rotation corrected.

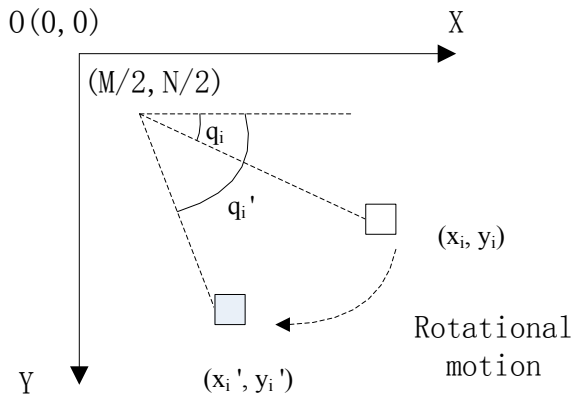


Figure 4 Offset angle calculation of image

5. Demonstration

To verify the validity of the algorithm, this experiment adopt a continuous image of 320×256 pixels, select different frames of image sequences for four experiments. Peak signal to noise ratio (PSNR) reflects the maximum signal to noise ratio between the reference image frame and the current image frames, it is an objective evaluation of stability in an image sequence. Once the PSNR value is larger, the restored image is more close to the original image. The expression of PSNR algorithm is as follows:

$$PSNR = 10 \lg \frac{255 \times 255}{MSE} \quad (9)$$

in which:

$$MSE = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [F(x, y) - R(x, y)]^2}{M \times N} \quad (10)$$

Where $F(x, y)$, $R(x, y)$ indicates gray value of (x, y) in the current image and reference image respectively, $M \times N$ indicates pixels of the image. The experimental results of calculated set are shown in Table 1.

Table 1 Comparison of estimated motion vector and PSNR values

The experimental group name	The translation vector this algorithm calculated	This algorithm calculated the angle	PSNR of two images before image stabilization	PSNR after compensating for translation vector using bit-plane	PSNR after compensating for motion vector using the proposed algorithm
Translation group 1	(1, -3)	0	21.111	33.426	33.812
Translation group 2	(0, 3)	0.005	21.285	32.231	32.996
Translation rotation group 1	(0, -2)	0.223	23.696	25.013	28.915
Translation rotation group 2	(1, 2)	-0.028	23.257	23.648	26.836

In the above four experiments, Comparing the PSNR before and after using bit plane algorithm for image stabilization, the results show that it can accurately handle the translational motion, but the image processing results of the rotating movement are poor; but comparing the PSNR values before and after using the improved algorithm, the results show that the algorithm can not only deal with translational motion, but also can effectively make up the defects of rotational motion for the original algorithm and improve the accuracy of image stabilization.

6. Conclusion

Using block-matching algorithm to optimize the bit-plane algorithm, This paper proposes an improved electronic image stabilization algorithm. In all algorithms of electronic image stabilization based on block-matching, full search block matching algorithm can come up with the most accurate translation motion vector, but also consume the large amount of calculation and time. Diamond search algorithm (DS) is a widely used block matching algorithm that can make balance between accuracy and efficiency in the current; In this

paper, compared with the diamond search algorithm, the block matching algorithm based on the hexagon, reduces the number of searching points (reduced from nine to seven) under the premise of guarantee the accuracy, results in improved motion estimation speed, and further reduces the time of block matching and image stabilization compensating. The experimental results show that the improved image stabilization algorithm is a better solution to the existence of rotational motion in the video images, well meets the image stabilization demands while ensuring the real-time performance, and also has a good prospect of application.

REFERENCES

- [1] Ko S J, Lee S H, and Lee K H. Digital image stabilizing algorithms based on bit-plane matching [J], IEEE Transactions on Consumer Electronics, Aug 1998,44 (3):617-622.
- [2] Ko S J, Lee S H, Jeon S W, et al. Fast digital image stabilizer based on gray-coded bit-plane matching[J], IEEE Transactions on Consumer Electronics, Aug 1999,45(3):598-603.
- [3] Yankun Wang and Liu Fang. Global motion estimation using the bit-planes and diamond searching [J], Ordnance Industry Automation, (200711):80-85.