Journal of Southeast University (English Edition) Vol. 24, No. 2, pp. 192 - 196 June 2008 ISSN 1003—7985

Fast global motion estimation and moving object extraction algorithm in image sequences

Zhang Tao Fei Shumin Li Xiaodong Lu Hong

(Key Laboratory of Measurement and Control of Complex Systems of Engineering of Ministry of Education, Southeast University, Nanjing 210096, China)

Abstract: A novel and effective approach to global motion estimation and moving object extraction is proposed. First, the translational motion model is used because of the fact that complex motion can be decomposed as a sum of translational components. Then in this application, the edge gray horizontal and vertical projections are used as the block matching feature for the motion vectors estimation. The proposed algorithm reduces the motion estimation computations by calculating the one-dimensional vectors rather than the two-dimensional ones. Once the global motion is robustly estimated, relatively stationary background can be almost completely eliminated through the inter-frame difference method. To achieve an accurate object extraction result, the higher-order statistics (HOS) algorithm is used to discriminate backgrounds and moving objects. Experimental results validate that the proposed method is an effective way for global motion estimation and object extraction.

Key words: global motion estimation; edge projection; higher-order statistics; moving object extraction

Motion estimation is a key technique in image sequences compression and processing, and in computer vision. A number of very different motion estimation algorithms have been proposed[1-6]. These algorithms are developed for different applications such as image sequences analysis, machine vision, robotics, and image sequences restoration.

Traditionally, motion estimation techniques are categorized into two broad types: pixel recursive and block matching. Both methods are based on the 2-D information extracted from successive picture frames.

The pixel recursive method predicts the displacement of each pixel recursively from its neighboring pixels[7-9]. The block matching method, on the other hand, estimates the displacement vectors by comparing the gray levels of successive frames on a block by block basis[10-13].

At present, the block-matching algorithms attract most of the attention because of their simplicity and effectiveness, and, moreover, the block-matching algorithms are more suitable for a simple hardware realization.

In block-based motion estimation, an image frame is divided into square blocks, typically of size 8×8 or 16×16. For each block, a search space is defined in the previous frame. The conventional approach is to perform a computation intensive exhaustive search to find the best match with respect to the mean absolute difference measure. The implementation always requires massive computational efforts and huge hardware costs to achieve significant speed.

This paper proposes a fast feature-based block matching algorithm to estimate the global motion by using the method of edge gray projection. The horizontal and vertical projections are the sums of the pixel values of the horizontal and vertical lines, respectively. So the horizontal and vertical projections within a block are the low-frequency features of the block. If two blocks are similar, their projection features should be similar. This motivates us to perform motion estimation in the feature domain.

Since the higher-order statistics (HOS) algorithm has the merit of being able to extract non-Gaussian signals from Gaussian signals, the method based on four moments of difference image for extracting moving objects[14] is used in this paper. Experimental results demonstrate that the method proposed in this paper is more effective.

1 Global Motion Modeling and Global Motion Estimation

1.1 Global motion modeling

The choice of the appropriate motion model depends on the target application and a variety of motion models can be taken into account[15].

The success of the translational motion model is primarily due to its simplicity and it is justified by the fact that complex motion can be decomposed as a sum of translational components. With the perspective projection and a rigid plane assumption of an object surface, the mapping from the point (x_1, y_1) on the arbitrarily shaped object in the current frame into the point (x_2, y_2) in the next frame is described by a vector $\{d_x, d_y\}$. And the global motion is modeled by

$$\begin{Bmatrix} x_2 \\ y_2 \end{Bmatrix} = \begin{Bmatrix} x_1 \\ y_1 \end{Bmatrix} + \begin{Bmatrix} d_x \\ d_y \end{Bmatrix} \tag{1}$$

where (x_1, y_1) and (x_2, y_2) are the coordinates of corresponding points in the two consecutive image frames, respectively. The vector $\{d_x, d_y\}$ defines the translation of the camera between frames.

1.2 Horizontal motion d_x estimation

Since moving objects usually occupy the centre of any scene, the margins of the image can be considered as the background. Thus a block which has $h\times N$ pixels is selected at the top of the image with a size of $M\times N$.

The vertical projections of each column contain information of the gray levels of its pixels along vertical directions; thus, we can use them as features of the sub-block considered.

Received 2007-10-26.

Biographies: Zhang Tao(1981—), male, graduate; Fei Shumin(corresponding author), male, doctor, professor, smfei@seu.edu.cn.

Foundation item: The National Natural Science Foundation of China(No. 60574006).

Citation: Zhang Tao, Fei Shumin, Li Xiaodong, et al. Fast global motion estimation and moving object extraction algorithm in image sequences[J]. Journal of Southeast University (English Edition), 2008, 24(2): 192 - 196.

First, the vertical projections of each column in the block of the current frame k are determined using Eq. (2), and compose a one-dimensional projection vector $\boldsymbol{H}_k$ (see Fig. 1(a)).

$$\boldsymbol{H}_k(j) = \sum_{i=0}^{h} f_k(i,j) \qquad j = 1, 2, \ldots, N \tag{2}$$

$$\boldsymbol{H}_k = \{\boldsymbol{H}_k(1), \boldsymbol{H}_k(2), \ldots, \boldsymbol{H}_k(N)\} \tag{3}$$

where $f_k(i,j)$ denotes the gray level at the (i,j) position of frame k.

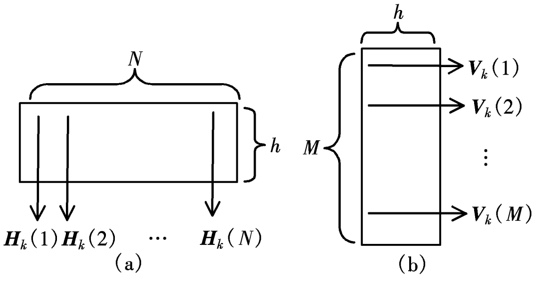


Fig. 1 Projections in a sub-block of the k-th frame. (a) Vertical projections in the $h \times N$ block; (b) Horizontal projections in the $M \times h$ block

To reduce the influence of the slow change of image texture to the background, we subtract the subsequent item of the vector $\boldsymbol{H}_k$ from the previous one, and obtain a $1 \times (N-1)$ vector $\boldsymbol{H}_{D(k)}$.

$$\boldsymbol{H}_{D(k)} = \{\boldsymbol{H}_{D(k)}(1), \boldsymbol{H}_{D(k)}(2), \ldots, \boldsymbol{H}_{D(k)}(N-1)\} \tag{4}$$

$$\boldsymbol{H}_{D(k)}(i) = \boldsymbol{H}_k(i) - \boldsymbol{H}_k(i+1) \qquad i = 1, 2, \ldots, N-1 \tag{5}$$

And for the previous frame, we can obtain vectors $\boldsymbol{H}_{k-1}$ and $\boldsymbol{H}_{D(k-1)}$.

Then we choose a value m and assume that the translational range of the camera is less than m.

The $N-2m+1$ elements are taken from the m-th item in $\boldsymbol{H}_{D(k)}$ and these elements compose a vector $\boldsymbol{H}_{m(k)}$.

$$\boldsymbol{H}_{m(k)} = \{\boldsymbol{H}_{D(k)}(m), \boldsymbol{H}_{D(k)}(m+1), \ldots, \boldsymbol{H}_{D(k)}(N-m)\} \tag{6}$$

And then $N-2m+1$ elements are taken from the i-th ($i = 1, 2, \ldots, 2m-1$) item in $\boldsymbol{H}_{D(k-1)}$. These can compose $2m-1$ vectors $\boldsymbol{H}_{m(k-1)(i)}$ ($i = 1, 2, \ldots, 2m-1$).

$$\boldsymbol{H}_{m(k-1)(i)} = \{\boldsymbol{H}_{D(k-1)}(i), \boldsymbol{H}_{D(k-1)}(i+1), \ldots, \boldsymbol{H}_{D(k-1)}(N-2m+i)\} \qquad i = 1, 2, \ldots, 2m-1 \tag{7}$$

The cost function S_{avgH} is calculated using the values of one-dimensional projection arrays. The cost function is defined by

$$S_{\text{avgH}}(i) = \frac{1}{N-2m+1} \sum_{j=1}^{N-2m+1} (\boldsymbol{H}_{m(k)}(j) - \boldsymbol{H}_{m(k-1)(i)}(j)) \qquad i = 1, 2, \ldots, 2m-1 \tag{8}$$

Determine i which has a correspondence with the minimum of $S_{\text{avgH}}(i)$:

$$i = \arg\min S_{\text{avgH}}(i) \qquad i = 1, 2, \ldots, 2m-1 \tag{9}$$

Let $d_x = i - m$, where d_x is the horizontal global motion between two continuous frames. If d_x is positive, it shows the camera moves to the left d_x pixel. Otherwise, the camera moves to the right $|d_x|$ pixel.

1.3 Vertical motion d_y estimation

Being similar to the horizontal motion estimation, a region which has $M \times h$ pixels is selected on the left of the image. Fig. 1(b) shows the step of the horizontal projections of each row in the block.

$$\boldsymbol{V}_k(i) = \sum_{j=1}^{h} f_k(i,j) \qquad i = 1, 2, \ldots, M \tag{10}$$

$$\boldsymbol{V}_k = \{\boldsymbol{V}_k(1), \boldsymbol{V}_k(2), \ldots, \boldsymbol{V}_k(M)\} \tag{11}$$

where $f_k(i,j)$ denotes a gray level at the (i,j) position of frame k.

We subtract the subsequent item of the vector $\boldsymbol{V}_k$ from the previous one, and obtain a $1 \times (M-1)$ vector $\boldsymbol{V}_{D(k)}$.

$$\boldsymbol{V}_{D(k)} = \{\boldsymbol{V}_{D(k)}(1), \boldsymbol{V}_{D(k)}(2), \ldots, \boldsymbol{V}_{D(k)}(M-1)\} \tag{12}$$

$$\boldsymbol{V}_{D(k)}(i) = \boldsymbol{V}_k(i) - \boldsymbol{V}_k(i+1) \qquad i = 1, 2, \ldots, M-1 \tag{13}$$

And for the previous frame, we can obtain vectors $\boldsymbol{V}_{k-1}$ and $\boldsymbol{V}_{D(k-1)}$.

Then we choose a value m and assume that the translational range of the camera is less than m.

The $N-2m+1$ elements are cut from the m-th item in $\boldsymbol{V}_{D(k)}$ and these elements compose a vector $\boldsymbol{V}_{m(k)}$.

$$\boldsymbol{V}_{m(k)} = \{\boldsymbol{V}_{D(k)}(m), \boldsymbol{V}_{D(k)}(m+1), \ldots, \boldsymbol{V}_{D(k)}(M-m)\} \tag{14}$$

Then we obtain $2m-1$ vectors $\boldsymbol{V}_{m(k-1)(i)}$ by taking the $\boldsymbol{V}_{D(k-1)}$ from the i-th ($i = 1, 2, \ldots, 2m-1$) element.

$$\boldsymbol{V}_{m(k-1)(i)} = \{\boldsymbol{V}_{D(k-1)}(i), \boldsymbol{V}_{D(k-1)}(i+1), \ldots, \boldsymbol{V}_{D(k-1)}(M-2m+i)\} \qquad i = 1, 2, \ldots, 2m-1 \tag{15}$$

The cost function is defined by

$$S_{\text{avgV}}(i) = \frac{1}{M-2m+1} \sum_{j=1}^{M-2m+1} (\boldsymbol{V}_{m(k)}(j) - \boldsymbol{V}_{m(k-1)(i)}(j)) \qquad i = 1, 2, \ldots, 2m-1 \tag{16}$$

Find i which is corresponding to the minimum of $S_{\text{avgV}}(i)$:

$$i = \arg\min S_{\text{avgV}}(i) \qquad i = 1, 2, \ldots, 2m-1 \tag{17}$$

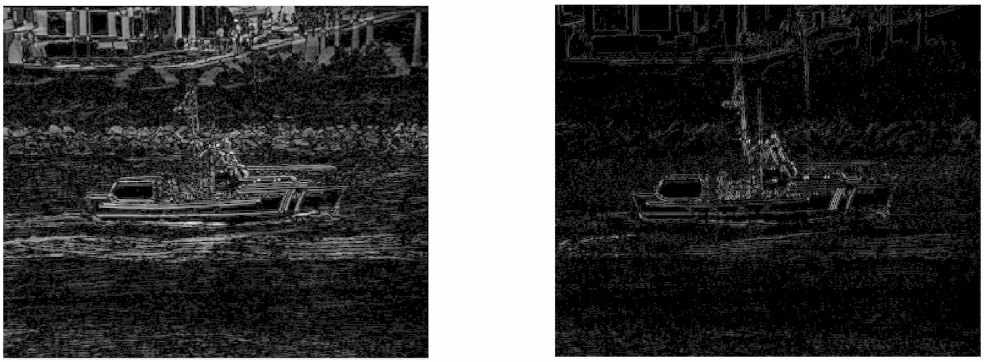
Let $d_y = i - m$, where d_y is the vertical global motion estimation between two continuous frames. If d_y is positive, it shows the camera moves up d_y pixel. Inversely, the camera moves down $|d_y|$ pixel.

1.4 Global motion compensation result

In this section, we use the typical image sequence Coastguard to illustrate the motion estimation algorithm mentioned above. Fig. 2(a) and Fig. 2(b) indicate the original frames 120 and 121 of Coastguard sequences, respectively. We use the algorithm mentioned above to estimate the global motion, where $m = 20$, and $h = 60$. The calculation shows the background of frame 120 shifts 3 pixels to the left and moves down 1 pixel relative to the background of frame 121 (i.e., $d_x = -3$ and $d_y = -1$). Fig. 3(a) shows the inter-frame difference before motion compensation. It can be noticed that the trees, houses and the coast are declared as the moving objects due to the motion of the camera. After motion compensation, the result is shown in Fig. 3(b). We can see that the boat is detected but not accurately because of the background noises. Similarly, another two continue frames of the coastguard sequences are shown in Fig. 4. And the same robust detection results are shown in Fig. 5.

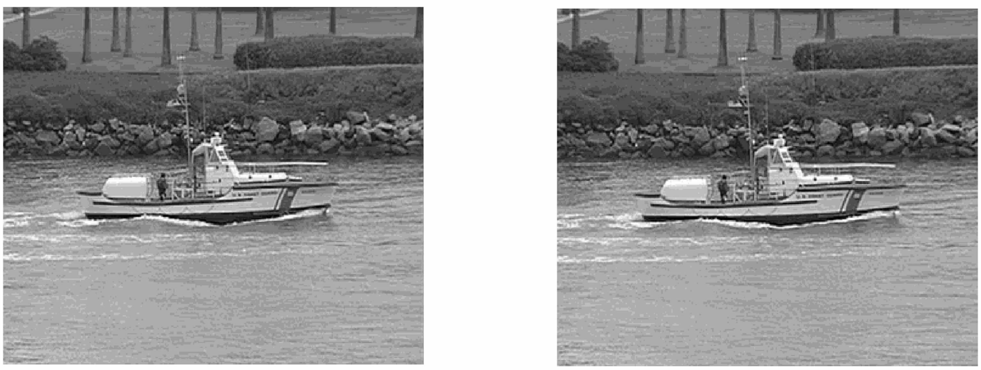
(a) (b)

Fig. 2 Frames of Coastguard sequences. (a) Frame No. 120; (b) Frame No. 121



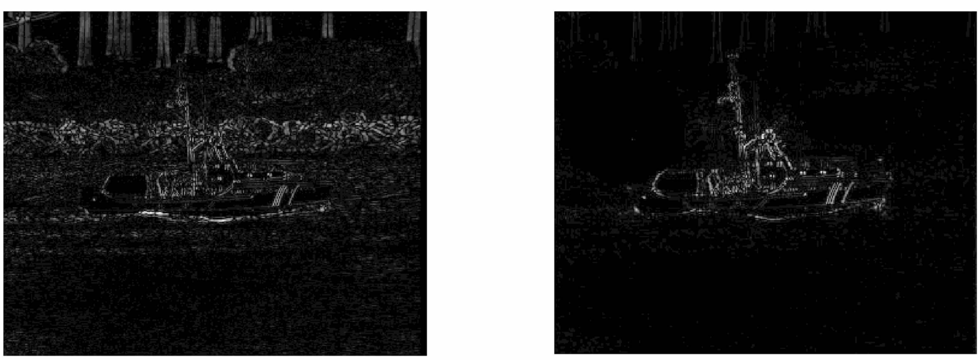
(a) (b)

Fig. 3 Result of the inter-frame difference algorithm using the motion compensation and non-using motion compensation. (a) Non-use of motion compensation; (b) Use of motion compensation



(a) (b)

Fig. 4 Frames of Coastguard sequences. (a) Frame No. 270; (b) Frame No. 271



(a) (b)

Fig. 5 Result of the inter-frame difference algorithm using the motion compensation and non-using motion compensation. (a) Non-use of motion compensation; (b) Use of motion compensation

Additionally, in order to obtain the object exactly, we use the HOS algorithm to reduce the influence of noises.

2 Moving Object Extraction Algorithm

2.1 HOS algorithm

The moving object in the inter-frame difference will occur as a high value, while background will represent a lower value respectively. In theory, the object and background induce different statistical signal distributions. The background signal is the Gaussian distribution, while the moving object of the inter-frame difference is a strong deviating from Gaussianity and can be considered as a non-Gaussian signal. Change based on motion detection is a traditional and very efficient technique for distinguishing the signal statistic difference between moving objects and backgrounds. The background signal can be detected by a second-order statistic detector and the moving object can be detected by the fourth-order moment measure[16-18].

For each pixel site (x, y), the fourth-order moment $m_{\mathrm{d}}^{(4)}(x, y)$ evaluated on a moving window $\eta(x, y)$ of $N_\eta = 9$ (N_η is the number of pixels in $\eta(x, y)$) of the inter-frame difference is

$$m_{\mathrm{d}}^{(4)}(x, y) = \frac{1}{N_\eta} \sum_{(s, t) \in \eta(x, y)} (\mathrm{diff}(s, t) - \overline{\mathrm{diff}}(x, y))^4 \quad (18)$$

where diff (s, t) is the inter-frame difference,

$$\mathrm{diff}(s, t) = f_t(s, t) - f_{t-1}(s, t) \quad (19)$$

And $\overline{\mathrm{diff}}(x, y)$ is the sample mean in the window $\eta(x, y)$,

$$\overline{\mathrm{diff}}(x, y) = \frac{1}{N_\eta} \sum_{(s, t) \in \eta(x, y)} \mathrm{diff}(s, t) \quad (20)$$

Such a moment $m_{\mathrm{d}}^{(4)}(x, y)$ is compared block by block with a threshold, which is proportional to the square of the noise variance σ_{od}^2. The moving block is detected by

$$\text{block} = \begin{cases} \text{object} & \text{if } m_{\mathrm{d}}^{(4)}(x, y) > c\sigma_{\mathrm{od}}^2 \\ \text{background} & \text{otherwise} \end{cases} \quad (21)$$

where the constant c is approximately independent of σ_{od}^2, and its value has been optimized on the basis of a statistical analysis performed during the experimental activity. In our practice, the c equaling 81 has the best performance. If the block moment is above threshold $c\sigma_{\mathrm{od}}^2$, the block is marked as a moving block.

Here, the noise variance σ_{od}^2 is estimated on a subset S of the static background:

$$\sigma_{\mathrm{od}}^2 = \frac{1}{N_S} \sum_{(s, t) \in S} (d(s, t) - m_{\mathrm{d}})^2 \quad (22)$$

Since moving objects usually occupy the centre of a scene, the four corners are the static background. Here the region S is selected on the four corners. The size of each corner is 12×12 pixels. First, we compute the noise variance σ_i^2 on each corner according to Eq. (22), and then make the noise variance σ_{od}^2 equaling the maximal σ_i^2.

2.2 Extraction results and post-processing

In this section, the HOS algorithm performance is illustrated. Fig. 6(a) shows the object extraction results obtained by the HOS algorithm regarding Fig. 3(b). It can be noted that some regions of the boat are declared as background due to the fluctuation motion of the boat on the river. This is also due to the homogeneous texture in the interior region of the moving object. In order to obtain the whole motion mask, a post-processing step is necessary to obtain the results by using the HOS. Post-processing consists of morphological opening and closing operations, which are used to eliminate noise. Fig. 6(b) shows the result obtained after the opening and closing operations on the HOS results.

(a)

(b)

Fig. 6 Extracted object sample from the Coastguard sequence. (a) Using the HOS algorithm; (b) Using the morphological operation

3 Conclusion

This paper presents a fast motion estimation method by using edge gray projections of the image edge as the features to be matched. This method can estimate the global motion quickly because we only need to calculate the one-dimensional vectors rather than the two-dimensional ones. To achieve an accurate object extraction result, we use the higher-order statistics (HOS) algorithm to discriminate the background and the moving object. From the experimental results, it can be concluded that the proposed method works well. The motion parameters can be efficiently estimated and the moving object can be extracted successfully.

References

[1] Namuduri K R, Aiyuan J. Computation and performance trade-offs in motion estimation algorithms [C]//*Proc of Intl Conf on Information Technology*: *Coding and Computing*. Las Vegas, USA, 2001: 263 - 267.

[2] Chimienti A, Ferraris C, Pau D. A complexity-bounded motion estimation algorithm [J]. *IEEE Trans on Image Processing*, 2002, **11**(4): 387 - 392.

[3] Chang Meng-Chou. Chien Jung-Shan. An adaptive search algorithm based on block classification for fast block motion estimation [C]//*Proc of IEEE Intl Symp on Circuits and Systems*. Kos, Greece, 2006: 3982 − 3985.
[4] Kim S E, Han J K, Kim J G. An efficient scheme for motion estimation using multireference frames in H.264/AVC [J]. *IEEE Trans on Multimedia*, 2006, **8**(3): 457 − 466.
[5] Zan J, Ahmad M O, Swamy M N S. Comparison of wavelets for multiresolution motion estimation [J]. *IEEE Trans on Circuits and Systems for Video Technology*, 2006, **16**(3): 439 − 446.
[6] Demir B, Erturk S. Block motion estimation using adaptive modified two-bit transform [J]. *IET Image Processing*, 2007, **1**(2): 215 − 222.
[7] Braun M, Hahn M, Ohm J R, et al. Motion-compensating real-time format converter for video on multimedia displays [C]//*Proc of Intl Conf on Image Processing*. Santa Barbara, CA, 1997: 125 − 128.
[8] Namazi N M, Penafiel P, Fan C M. Nonuniform image motion estimation using Kalman filtering [J]. *IEEE Trans on Image Processing*, 1994, **3**(5): 678 − 683.
[9] Schreer O, Brandenburg N, Kauff P. Real-time disparity analysis for applications in immersive teleconference scenarios-a comparative study [C]//*Proc of the 11th Intl Conf on Image Analysis and Processing*. Palermo, 2001: 346 − 351.
[10] Moschetti F, Kunt M, Debes E. A statistical adaptive block-matching motion estimation [J]. *IEEE Trans on Circuits and Systems for Video Technology*, 2003, **13**(5): 417 − 431.
[11] Yang M, Cui H, Tang K. Efficient tree structured motion estimation using successive elimination [J]. *IEE Proceedings Vision, Image and Signal Processing*, 2004, **151**(5): 369 − 377.
[12] Patras I, Hendriks E A, Lagendijk R L. Probabilistic confidence measures for block matching motion estimation [J]. *IEEE Trans on Circuits and Systems for Video Technology*, 2007, **17**(8): 988 − 995.
[13] Huang Yu-Wen, Chien Shao-Yi, Hsieh Bing-Yu, et al. Global elimination algorithm and architecture design for fast block matching motion estimation [J]. *IEEE Trans on Circuits and Systems for Video Technology*, 2004, **14**(6): 898 − 907.
[14] Zhong Xingrong, Huang Xianwu, Wang Jiajun, et al. Video object segmentation based on HOS and multi-resolution watershed [C]//*Proc of the* 2004 *Intl Symp on Intelligent Multimedia, Video and Speech Processing*. Hong Kong, China, 2004: 274 − 277.
[15] Stiller C, Konrad J. Estimating motion in image sequences [J]. *IEEE Signal Processing Magazine*, 1999, **16**(4): 70 − 91.
[16] Blanco D, Mulgrew B, Ruiz D P, et al. Independent component analysis in signals with multiplicative noise using fourth-order statistics [J]. *Signal Processing*, 2007, **87**(8): 1917 − 1932.
[17] Weiss M, van de Beld D. A generalized shaping filter method for higher order statistics [J]. *Probabilistic Engineering Mechanics*, to appear.
[18] Neri A, Colonnese S, Russo G, et al. Automatic moving object and background separation [J]. *Signal Processing*, 1998, **66**(2): 219 − 232.

图像序列中快速全局运动估计和运动目标提取算法

张　涛　　费树岷　　李晓东　　路　红

(东南大学复杂工程系统测量与控制教育部重点实验室,南京 210096)

摘要:提出了一种基于边界灰度投影匹配的全局运动估计和运动目标提取算法. 算法将边界灰度水平投影和垂直投影值作为匹配特征,较好地估计了全局运动参数. 由于只需计算一维特征向量所以降低了全局运动估计的计算量. 经过全局运动补偿后,可以运用传统的帧间差法得到运动目标. 为了减少噪声的影响,准确地提取到目标,采用了高阶统计量的方法(HOS)来区分背景和运动目标. 试验结果证明所提出的方法在估计全局运动参数和提取运动目标方面有较好的鲁棒性.

关键词:全局运动估计;边界投影;高阶统计量;运动目标提取

中图分类号:TP391.41