PERGAMON

Pattern Recognition 33 (2000) 2095–2098

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Rapid and Brief Communication

Region tracking using perspective motion model

Song-Ha Choi, Seong-Whan Lee*

Center for Artificial Vision Research, Department of Computer Science and Engineering, Korea University, Anam-dong, Seongbuk-ku, Seoul 136-701, South Korea

Received 28 January 2000; accepted 3 February 2000

1. Introduction

The video (or motion picture) media has recently grown rapidly and become widely popular such that it is now being used by various desktop PC applications. Among these applications of the video media, there are video conferencing, visual surveillance, agriculture automation, medical imaging, vision-based control, and so on. One of the issues that is becoming continually more important in these applications is region tracking.

There are two traditional region tracking methods: the dense correspondence method and the contour tracking method. The dense correspondence method is computationally too expensive because of its full-search correspondence computation. The contour tracking method also is computationally expensive due to so many iterations and have a danger of running into local minima.

The widely used affine motion model is the simplest and most general motion model for tracking in 2D space. Black and Jepson [1] proposed an eigentracking method with the affine motion model and used multi-scale eigenspace representation of a target region template. However, the affine motion model cannot track the deformation such as the deformation from rectangles to general quadrangles. The perspective (or projective) motion model can be one possible solution. Poelman and Kanade [2] suggested a method of shape and motion recovery. Steinbach et al. [3] proposed a 3D motion estimation from multi-frame image sequence using the perspective motion model. The studies mentioned thus far have utilized the geometric information of images, such as epipolar lines.

In this paper, we suggest a fitting method of the motion vectors in a target region into the perspective motion model and the selection method of feature points. The perspective motion model is basic and flexible to express 3D real motion. Also, since the proposed method considers the motion vectors only of the feature points in the target region, noises of motion information are reduced and corrected throughout the processes. Fig. 1 shows the overall process of the proposed method.

2. Perspective motion tracking

2.1. Feature point extraction

To extract a set of feature points from a target region, we apply difference of Gaussian (DOG) filtering to input images. The DOG function is widely used in biologically motivated vision research because it shows a similar response to the receptive fields of the human vision system [4]. The DOG function is defined as

$$\mathrm{DOG}(x,y)=\frac{\mathrm{e}^{-(x^2+y^2)/2\sigma_1^2}}{2\pi\sigma_1^2}-\frac{\mathrm{e}^{-(x^2+y^2)/2\sigma_2^2}}{2\pi\sigma_2^2}, \tag{1}$$

where σ_1 and σ_2 are variance constants of two Gaussian functions. The width and height of the DOG response can be adjusted with these parameters.

Next, we select local maximum points in the DOG response. The local maximum points are the set of points of current interest. In order to obtain the saccadic movements of the points of current interest to the next position, the self-inhibition function is applied to the points of current interest. The self-inhibition function is defined as

$$M(\mathbf{x}_0,\mathbf{x}') = \left(1+\alpha\exp\left(\frac{|\mathbf{x}_0-\mathbf{x}'|^2}{A^2}\right)\right)\left(\beta\exp\left(\frac{|\mathbf{x}_0-\mathbf{x}'|^2}{B^2}\right)-1\right) \tag{2}$$

where α, β, A, and B are constant values.

* Corresponding author. Tel.: +82-2-3290-3197; fax: +82-2-926-2168.

E-mail addresses: shchoi@image.korea.ac.kr (S.-H. Choi), swlee@image.korea.ac.kr (S.-W. Lee).

PII: S0031-3203(00)00058-3

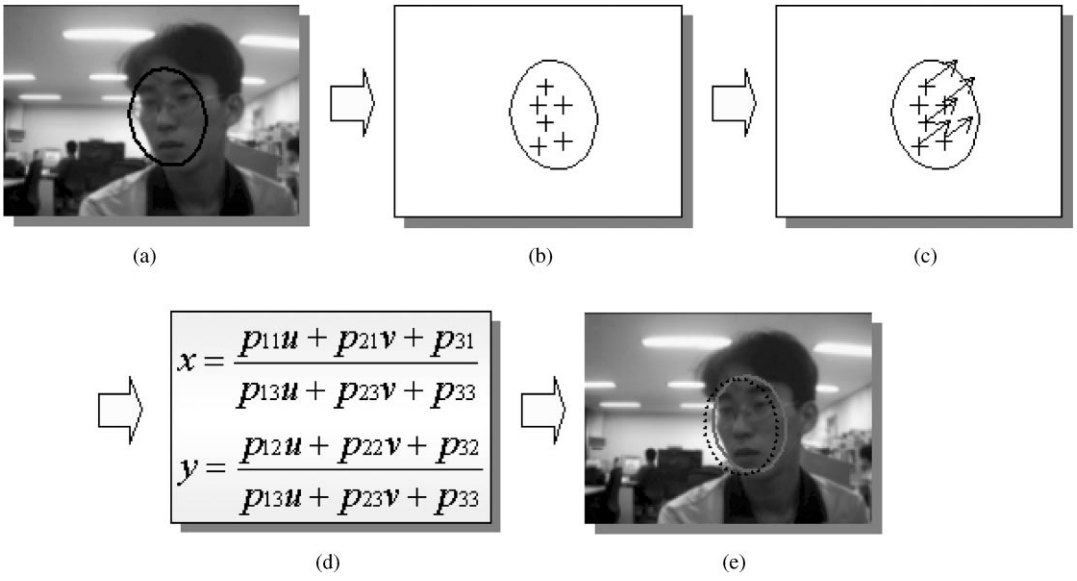

Fig. 1. The process of perspective region tracking: (a) target region selection; (b) feature points extraction; (c) motion estimation; (d) model parameters calculation; (e) target region update.

2.2. Block-based motion estimation in HSV color space

To estimate each motion vector of the feature points obtained in the previous section, a correlation-based motion vector estimation method, known as the block matching algorithm, is used.

The traditional block matching algorithm is performed in the RGB color space and has a great deal of weaknesses. Smith et al. [5] indicated that HSV color space has perceptually color-distinct properties. We adopt Smith's method for motion estimation, and its measure is used for calculation of correspondence. The HSV space difference (D) is defined as follows:

$$D(C_1, C_2) = \frac{1}{\sqrt{5}}\sqrt{\Delta V^2 + \Delta SH_a^2 + \Delta SH_b^2}$$

where $\Delta V = V_1 - V_2, \Delta SH_a = S_1 \cos H_1 - S_2 \cos H_2$, $\Delta SH_b = S_1 \sin H_1 - S_2 \sin H_2$ and subscripts 1 and 2 are each pixel position for comparison.

2.3. The linearizing process of 2D perspective transformation

For fitting 2D apparent motion information into a constrained motion model, we use the least-squares approximation technique (LSA). Here, the optimal solution is approximated to a plane which was fit into the motion model, and the estimated values represent 2D apparent motion vectors, which were obtained through the BMA method. Fig. 2 shows an illustration about solving the minimization problem.

The representation of the perspective transformation can be expressed as:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = P\begin{pmatrix} u \\ v \\ 1 \end{pmatrix}, \quad P = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} \tag{3}$$

where X, Y and Z are the 3D coordinates of the output point, P is the transformation matrix and u, v are the input coordinates. Next, we apply the least-squares algorithm to the linearized perspective model. The optimization of the model parameters is redefined as the minimization problem of F. Equation F is

$$F = \sum_{i=1}^{n} \left\| \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} - \begin{pmatrix} \tilde{X}_i \\ \tilde{Y}_i \\ \tilde{Z}_i \end{pmatrix} \right\|^2 = \sum_{i=1}^{n} \left\| P\begin{pmatrix} u_i \\ v_i \\ 1 \end{pmatrix} - \begin{pmatrix} \tilde{X}_i \\ \tilde{Y}_i \\ \tilde{Z}_i \end{pmatrix} \right\|^2, \tag{4}$$

where $\tilde{X}_i$, $\tilde{Y}_i$ and $\tilde{Z}_i$ are the estimated 3D coordinates obtained from 2D apparent motion.

2.4. Least squares approximation of perspective motion model

Minimizing the sum of errors term F means that the differences between the 2D motion vector and 3D

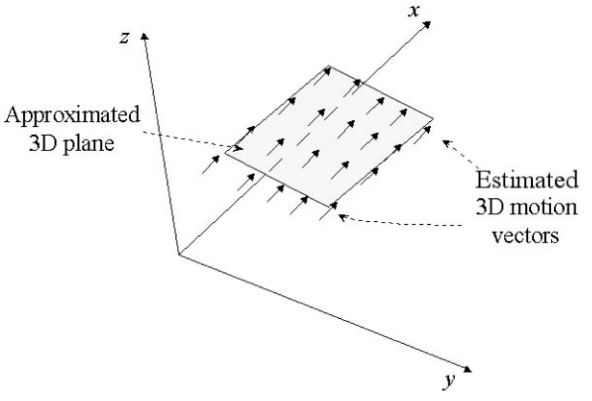

Fig. 2. The perspective motion model fitting with LSA algorithm.

perspective model is reduced. If the sum of errors term become minimized, the difference between the estimated 3D parameters and 3D parameters obtained from 2D parameters is closer.

The value of Z_i can be obtained from the equation of the perspective model function $Z_i = gu_i + hv_i + 1$. Then, we have the following linearized equation:

$$F = \sum_{i=1}^{n} \{(au_i + bv_i + c - \tilde{x}_i(gu_i + hv_i + 1))^2 + (du_i + ev_i + f - \tilde{y}_i(gu_i + hv_i + 1))^2\}, \quad (5)$$

where x, y are the 2D output coordinates.

To find the solution of the above equation, we find the value of the partial differentiation of the equation above for each parameter. If each equation equals zero, the value of the target function becomes minimized. Therefore, our new target function is defined as

$$\left(\frac{\partial F}{\partial a}, \frac{\partial F}{\partial b}, \ldots, \frac{\partial F}{\partial h}\right) = 0. \quad (6)$$

2.5. Extracting 3D transformation parameters

To estimate the 3D pose of target region, we must construe 3D parameters from 2D perspective parameters. The 3D transformation parameters are defined as

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = P\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = R\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} + T.$$

From above equations, the rotation and translation parameters are acquired as (in case of counterclockwise)

$$\theta_x = \tan^{-1}\left(\frac{ad + be}{ag + bh}\right), \quad \theta_y = \sin^{-1}\left(\frac{ag + bh}{bd - ae}\right),$$

$$\theta_z = \tan^{-1}\left(\frac{b}{a}\right),$$

$$T_x = c - \sin\theta_y, \quad T_y = f + \sin\theta_x \cos\theta_y,$$

$$T_z = 1 - \cos\theta_x \cos\theta_y$$

3. Experimental results and analysis

Our experiment of the proposed method was performed on an IBM compatible 500MHz Pentium PC under the Windows 98 operating system. The data set used for the experiment consisted of the image sequences from *Claire* and *Susie*.

3.1. Experimental results

We compared the performance of the affine and perspective tracking methods based on the SSD (sum of squared differences). The results of the measurement test are shown in Fig. 3. The x coordinate represents the frame number, and the y coordinate represents the SSD differences between the affine and perspective tracking. The graph shows a positive value when the performance of perspective tracking is better than that of affine tracking.

The time comparison between affine tracking and perspective tracking is not principal, because its difference is very trivial. Table 1 shows the time measurements example.

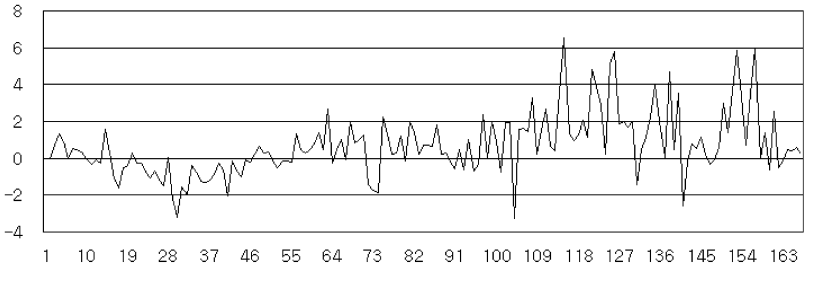

Fig. 3. The graph of difference between perspective SSD and affine SSD for 'Claire' sequences.

Table 1
Time measurements from our test sequences

	Speed (ms/f)	Ratio (%)
Image loading	45.98	11.82
Motion estimation	334.69	86.09
LSA computation	0.22	0.05
Displaying	7.83	2.02
Total	388.72	100

3.2. Comparison of time complexity between affine tracking and perspective tracking

We can calculate the time complexity of affine tracking and perspective tracking. The amount of computations on perspective parameters and affine parameters is $18n$ and $61n$, respectively. Therefore, the time complexity of the two methods are the same as $O(n)$. This is due to the linear property of the 3D perspective motion model.

4. Conclusions and further researches

In this research, we have shown a solution to a perspective tracking problem. The proposed method is easy to understand and to implement. It has shown in our experiment result that the proposed method takes almost similar time and is more accurate than the affine tracking method.

The proposed method has some shortcomings in that the non-planar target region cannot be tracked. In that case, the method can still be applied if the non-planar target is divided into several sub-planes. The case of the target region being occluded or excessively distorted is not considered the proposed method. Therefore, further study is required on the proper division of region, which have complex joints or those which are non-rigid.

Acknowledgements

This research was supported by Creative Research Initiatives of the Korean Ministry of Science and Technology.

References

[1] M.J. Black, A.D. Jepson, Eigen tracking: Robust matching and tracking of articulated objects using view-based representation, Int. J. Comput. Vision 26 (1) (1998) 63–84.
[2] C.J. Poelman, T. Kanade, A paraperspective factorization method for shape and motion recovery, IEEE Trans. on Pattern Anal and Mach Intell 19 (3) (1997) 206–218.
[3] E. Steinbach, S. Chaudhuri, B. Girod, Data-driven multiframe 3D motion estimation, Proceedings of the International Conference on Image Processing, Santa Barbara, 1997, pp. 464–467.
[4] D. Marr, Vision, Freeman, New York, 1982.
[5] J.R. Smith, Integrated spatial and feature image systems: retrieval, compression and analysis, Ph.D. Thesis, Graduate School of Arts and Sciences, Columbia University, February 1997.