H-023

Motion Estimation on Fish-Eye Images Using Modified Motion Model

Arief Suryadi Satyawan[1] Junichi Hara[2,3] Hiroshi Watanabe[1,2]

1. Introduction

Estimating motion of objects directly from 2D fish-eye images can be done by calculating vector fields from spatio-temporal intensity derivative of the image sequence as previously demonstrated by adopting Lucas and Kanade's concept [1]. However, due to massive distortion caused by characteristic of the fish-eye camera which results unique deformation of object's motion during captured by the camera, the performance of motion vector fields yielded seem to be unsatisfying. To improve this, in this research, the general perspective camera motion model is modified by incorporating a parametric motion model of the fish-eye lens before calculating the vector fields.

2. Image Motion Model

2.1 Perspective Image Motion Model Approach

In general perspective image model, a pixel intensity belong to an object within an image captured at time t_1, $I_1(x_1,y_1)$, is usually assumed to be constant although the position within the next image captured at t_2 is located at $I_2(x_2,y_2)$ because of the object movement $(I_1(x_1,y_1,t)=I_2(x_2,y_2,t+1))$, hence the pixel's motion vectors along horizontal and vertical direction can be expressed as $u=x_2-x_1$ and $v=y_2-y_1$ respectively. Additionally, this constant pixel's intensity assumption now can be written as (1).

$$I_1(x_1,y_1,t) = I_2(x_1+u, y_1+v, t+1) \quad \text{……………………}(1)$$

Following this, to be able to calculate u and v, the construction of lower-order Taylor series expansion of (1), in respect to such location of pixel (x_1,y_1) can be done firstly to meet (2), generally named as the optical problem equation. Noted that $I_x=\partial I_2/\partial_x$ and $I_y=\partial I_2/\partial_y$ are the first derivative of the image with reference to horizontal and vertical successively, while $\partial I/\partial_t$ is equal with the difference between the first derivative of I_2 and I_1.

$$\left(\frac{\partial I}{\partial x}\right)u + \left(\frac{\partial I}{\partial y}\right)v = -\left(\frac{\partial I}{\partial t}\right) \quad \text{……………………}(2)$$

Secondly, optimizing (2) in accord with u and v successively by incorporating n x n neighboring pixels according to Lucas and Kanade's idea as expressed by (3), can be done, so that the pixel's motion vectors (U^T) can be obtained from the 2x2 matrix (4) [5].

$$\vec{V}(u,v) = \arg\min_{(u,v)} \textstyle\sum_{x,y\in R} n^2 \left[\left(\frac{\partial I}{\partial x}\right)u + \left(\frac{\partial I}{\partial y}\right)v + \left(\frac{\partial I}{\partial t}\right)\right]^2 \quad \text{…….}(3)$$

$$\underbrace{\begin{pmatrix} \sum n^2 I_x^2 & \sum n^2 I_x I_y \\ \sum n^2 I_x I_y & \sum n^2 I_y^2 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} u \\ v \end{pmatrix}}_{U} = -\underbrace{\begin{pmatrix} \sum n^2 I_x I_t \\ \sum n^2 I_y I_t \end{pmatrix}}_{B} \quad \text{…………………}(4)$$

2.2 Non-Perspective Image Motion Model Approach

To be able to adapt with unique deformation of object shown on the fish-eye images, in this experiment we tried to adopt two parametric image motion model. The first model is dioptric motion model [2], while the second is affine motion model [3].

According to [2], motion model of dioptric camera can be expressed as $u=a(x-x_0)^2+a(y-y_0)^2+c$ and $v=b(x-x_0)^2+b(y-y_0)^2+d$. Therefore, calculation of the pixel's motion vectors along horizontal and vertical space can be started by modifying (2) in section 2.1 becomes (5).

$$\left(\frac{\partial I}{\partial x}\right)(a(x-x_0)^2 + \mathrm{a}(y-y)^2 + \mathrm{c}) + \left(\frac{\partial I}{\partial y}\right)(b(x-x_0)^2 + \mathrm{b}(y-y)^2 + \mathrm{d}) = -\left(\frac{\partial I}{\partial t}\right) \quad \text{……………}(5)$$

Next step will be also the same with second procedure, except now the optimization firstly with regard to a, b, c and d as mentioned by (6), before calculating u and v. As a result, Lucas and Kanade's matrix expression can be modified to perform (7) (noted that k is equal with $((x-x_0)^2+(y-y_0)^2)$.

$$\vec{V}(u,v) = \arg\min_{(a,b,c,d)} \textstyle\sum_{x,y\in R} n^2 \left[\left(\frac{\partial I}{\partial x}\right)(a(x-x_0)^2 + \mathrm{a}(y-y)^2 + \mathrm{c}) + \left(\frac{\partial I}{\partial y}\right)(b(x-x_0)^2 + b(y-y)^2 + d) + \left(\frac{\partial I}{\partial t}\right)\right]^2 \quad \text{…………}(6)$$

$$\underbrace{\begin{pmatrix} \sum n^2 I_x^2 k^2 & \sum n^2 I_x I_y k^2 & \sum n^2 I_x k & \sum n^2 I_x I_y k \\ \sum n^2 I_x I_y^2 k^2 & \sum n^2 I_y^2 k^2 & \sum n^2 I_x I_y k & \sum n^2 I_y^2 k \\ \sum n^2 I_x k & \sum n^2 I_x I_y k & \sum n^2 I_x^2 & \sum n^2 I_x I_y \\ \sum n^2 I_x I_y k & \sum n^2 I_x k & \sum n^2 I_x I_y & \sum n^2 I_y^2 \end{pmatrix}}_{D} \underbrace{\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}}_{E} = -\underbrace{\begin{pmatrix} \sum n^2 I_x I_t k \\ \sum n^2 I_y I_t k \\ \sum n^2 I_x I_t \\ \sum n^2 I_y I_t \end{pmatrix}}_{F} \quad \text{…………}(7)$$

On the other hand, affine tend to compose motion model as $u=a_1x+a_2y+a_3$ *and* $v=a_4x+a_5y+a_6$ [3]. Hence this may change the (2), we can get (8).

$$\left(\frac{\partial I}{\partial x}\right)(a_1\mathrm{x} + a_2 y + a_3) + \left(\frac{\partial I}{\partial y}\right)(a_4\mathrm{x} + a_5 y + a_6) = -\left(\frac{\partial I}{\partial t}\right) \quad \text{…}(8)$$

The optimization of (8) is now applied to obtain a_1, a_2, a_3, a_4, a_5, *and* a_6 as performed by (9), so that u and v can be calculated from modified version of Lucas and Kanade's approach as described by the matrix (10).

$$\vec{V}(u,v) = \arg\min_{(a,b,c,d)} \textstyle\sum_{x,y\in R} n^2 \left[\left(\frac{\partial I}{\partial x}\right)(a_1 x + \mathrm{a}_2 y + a_3) + \left(\frac{\partial I}{\partial y}\right)(a_4 x + a_5 y + a_6) + \left(\frac{\partial I}{\partial t}\right)\right]^2 \quad \text{…………}(9)$$

$$\underbrace{\begin{pmatrix} \sum n^2 I_x^2 x^2 & \sum n^2 I_x^2 xy & \sum n^2 I_x^2 x & \sum n^2 I_x I_y x^2 & \sum n^2 I_x I_y xy & \sum n^2 I_x I_y x \\ \sum n^2 I_x^2 xy & \sum n^2 I_x^2 \mathrm{y}^2 & \sum n^2 I_x^2 \mathrm{y} & \sum n^2 I_x I_y xy & \sum n^2 I_x I_y y^2 & \sum n^2 I_x I_y y \\ \sum n^2 I_x^2 x & \sum n^2 I_x^2 \mathrm{y} & \sum n^2 I_x^2 & \sum n^2 I_x I_y x & \sum n^2 I_x I_y y & \sum n^2 I_x I_y \\ \sum n^2 I_x I_y x^2 & \sum n^2 I_x I_y xy & \sum n^2 I_x I_y x & \sum n^2 I_y^2 x^2 & \sum n^2 I_y^2 xy & \sum n^2 I_y^2 x \\ \sum n^2 I_x I_y xy & \sum n^2 I_x I_y y^2 & \sum n^2 I_x I_y y & \sum n^2 I_x I_y xy & \sum n^2 I_x I_y y^2 & \sum n^2 I_x I_y y \\ \sum n^2 I_x I_y x & \sum n^2 I_x I_y y & \sum n^2 I_x I_y & \sum n^2 I_y^2 x & \sum n^2 I_y^2 y & \sum n^2 I_y^2 \end{pmatrix}}_{G} \underbrace{\begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \end{pmatrix}}_{H} = -\underbrace{\begin{pmatrix} \sum n^2 I_x I_t x \\ \sum n^2 I_x I_t y \\ \sum n^2 I_x I_t \\ \sum n^2 I_y I_t x \\ \sum n^2 I_y I_t y \\ \sum n^2 I_y I_t \end{pmatrix}}_{I} \quad \text{…………}(10)$$

3. Experimental

3.1 Experimental Setting

There are two image models used in this experiment. While the first image model taken from Ricoh Theta fish-eye camera describing slow movement of object (a hand) from the left to the right position, the second one is synthetic fish-eye images showing motion of flowers [4]. Two pairs of each model are shown in Fig.

1 ― Graduate School of Fundamental Science and Engineering, Waseda University
2 ― Global Information and Telecommunication Institute, Waseda University
3 ― RICOH Company, LTD

1. The procedure of estimating fish-eye images from taking pairs of image to evaluate quantitatively the performance of the estimation can be described in the Table 1.

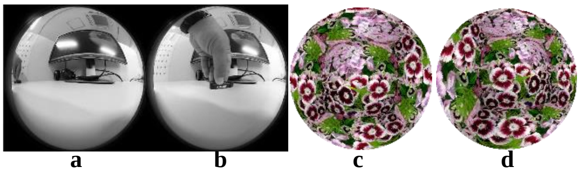
Table 1. Experimental Design Algorithm

Algorithm for calculating modified Lucas and Kanade's optical flow method and the evaluation of its performance

Input: 126 one side of fish-eye images
Output: reconstructed image, vector flows (u & v), motion compensated, & PSNR

for** each two successive images, **do
  $im1_{st}$ & $im2_{nd}$ ← converting the two images to grayscale
  $im1_{st}$ & $im2_{nd}$ ← smoothing the two grayscale images by using Gaussian smooth filter
  I_x, I_y & I_t ← obtaining spatial and temporal gradient of the images
  u & v ← calculating vector flows by using each matrix in section 2.
  $im1_R$ ← $im1_{st}$ moved by u & v (reconstructing image)
  im_inter-difference ← subtract $im2_{nd}$ with im1R
  PSNR ← Calculating maximum error between $im2_{nd}$ and $im1_R$
end for
Return

3.2 Experimental Results



a b c d

Fig 1. Fish-Eye Images (**a** and **b** are real image, whilst **c** and **d** are synthetics images).

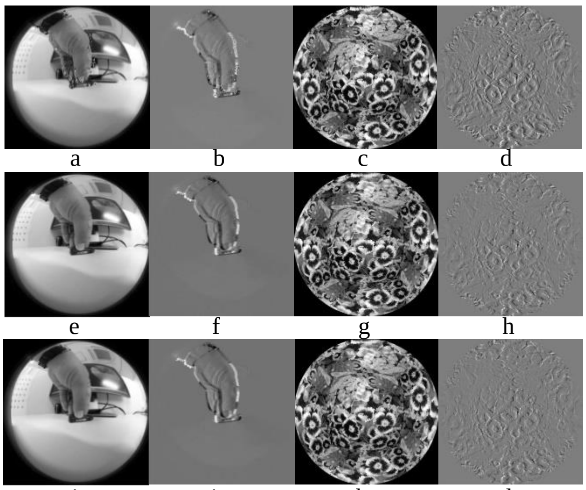


Fig 2. Reconstructed Images for Perspective Image Motion Model (a and c), Fish-Eye/Dioptric Image Model (e and g), and Affine Motion Model (i and k). Inter-Different Images for Perspective Image Motion Model (b and d), Fish-Eye/Dioptric Image Model (f and h), and Affine Motion Model (j and l).

It can be seen from Fig. 2, that there is improvement of motion estimation calculation denoted visually by reconstructed image when adopting dioptric or affine motion model in comparison with perspective motion model for two types of image source. Moreover, the results tend to be followed by the quantitative measurement (PSNR). The dioptric motion model achieves the highest performance (about 24-48 dB for real image and about 16-18 dB for synthetic image), while the affine follows in second with the range of about 23-45 dB for real image and about 15-17 dB. They lead by about 1-3 dB to the perspective model as the lowest.

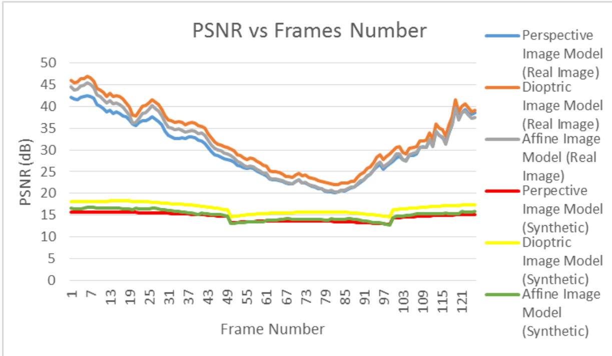


Fig 3. PSNR for Every Pair Image

The trend of PSNR data describes that object movement on each image source affects to the results in difference way. In real image, the lowest performance for each motion model occurs when the single object captured moves around center area of the fish-eye image. This condition seems to be caused by sudden movement of the object during passing central area and in the same time there is a part of the object moves around the top of circle area. On the other hand, the synthetic image which includes small movement on the whole fish-eye image tends to produce constant performance although in the lower level. This because all area of image suffers from motion, therefore the contribution of error massively occurs.

4. Conclusion

Both dioptric and affine motion model can be adopted to increase the performance of estimating motion directly from fish-eye images by using Lucas and Kanade's optical flow concept until 1-3 dB gain in compare with perspective model. Images with detail movement throughout the fish-eye area will have biggest opportunity to produce massive error.

5. References

[1] A.S. Satyawan, J. Hara, and H. Watanabe, "Motion estimation on fish-eye images," IEICE General Conference, BS-1-9, Mar. 2017.
[2] A. Radgui, C. Demonceaux, E. Mouaddib, D. Aboutajdine, and M. Rziza, "An adapted Lucas-Kanade's method for optical flow estimation in catadioptric images," The 8th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras - OMNIVIS, Oct 2008, Marseille, France. 2008, <inria-00325390>.
[3] Y. Altunbasak, R.M. Mersereau, and A.J. Patti, "A fast parametric motion estimation algorithm with illumination and lens distortion correction," IEEE Transactions on Image Processing, vol. 12, no. 4, pp.395-408, April 2003.
[4] Fish-eye image data set: www.lms.lnt.de/fisheyedataset
[5] S. Baker and I. Matthews, "Lucas-Kanade 20 Years On: A unifying framework," International Journal of Computer Vision, vol.56, no.3, pp.221-255, Feb.-Mar. 2004.