An edge-based approach to improve optical flow algorithm

Shih-Kuan Liao[1]
Department of Information Management National Taichung Institute of Technology, Taichung, No.129 Sec.3, San-min Road Taichung, Taiwan
Taichung, Taiwan
skliao@ntit.edu.tw

Baug-Yu Liu[2]
Graduate School of Computer Science and Information Technology National Taichung Institute of Technology, Taichung, No.129 Sec.3 San-min Road Taichung
Taichung, Taiwan
S18973120@ntit.edu.tw

Abstract—Traditional optical flow techniques applied to object tracking generally perform global searching and calculations of brightness and light intensity of the object in the image. In addition, traditional optical flow techniques assume that the light intensity is constant across a series of consecutive images. The goal is to obtain the displacement and moving direction of an object in a series of images. However, most of important information lies in the regions where optical flows vary significantly. Having relatively small optical flow variations usually implies that the information lying in this region is not important. As traditional optical flow techniques employs global searching to obtain optical flow values, the total computations are time consuming and most of time is spent on unimportant regions. If it is acceptable to exclude part of unimportant information then the overall algorithm can omit part of computations and hence shorten the time needed to calculate optical flow field.

To speed up the optical flow calculation, this study proposes an edge-based algorithm for obtaining optical flows. The main ideas are to segments out objects in each of consecutive images and then compare every object's centroid with circumference to identify matching objects of each image. According to the movement data of corresponding objects in each image, optical flow field can be formed and as a result objects can be tracked. Finally, the proposed algorithm in this study has been experimented to effectively decrease computation time while preserving useful information.

Keywords- Optical flow; object feature; Segmentation

I. INTRODUCTION

The main concept of "optical flow method[4]" is to capture serial pictures by using a recording video camera. During each time period, there is a relative motion between the objects in pictures and the recording video camera, and the displacement of illumination would occur, through this we can calculate the light source intensity and scale space[7] in the serial images and find out the corresponding characteristics among the images. Then we can get the statistics by evaluating the light source intensity and scale space, then the displacement of the objects from the serial images can be tracked. But there are other factors would lead image changes rather than the relative motion between the objects in pictures and the recording video camera such as the difference light source in the surroundings. In the case, even if the objects stay still, because of the changes in light source intensity, illumination alteration on the image surface or the object shape would deform would still occur and lead to miscalculation.

The traditional optical flow method uses global calculation which obtains displacement data by comparing and evaluating the serial images through the light source intensity and scale space. This method would cause miscalculation of optical flow value due to the difference of the light source intensity in images and lead to failure in tracking. Moreover, global calculation needs mass time consuming for the complicated process. Many scholars have done researches and experiments about speeding light stream to improve this problem. Also many scholars have mentioned some researchers about optical flow method, such as John S. Zelek[14]、Jeongho Shin[11]、Javier Diaz[2]、Farid Kendoul[5]。

Optical flow method can be applied to tracking[3] or recognizing objects, the application is wide arranged, for instance, the inspecting for gait[9] and innervation zone[8]…etc in the medical field. Also facial recognition[1] and human body displacement trace[10].

Optical flow is a vector quantity; all the optical flow in a scene constructs a optical flow field. For an optical flow field, if there is a longer optical flow in a certain area, it means more scene changes and m ore or better data would be gathered. On the other hand, when the scene is stabilized, only vague optical flow would show because the lack of data. Sometimes only small part of the area would have stronger optical flow, as Fig.1 shows. If we're only concerned about the area with drastic changes but not the tendency of changes in the bigger range, and then consider the data form optical flow, under this circumstance, we think it'd be fine to only choose the stronger part of optical flow instead of the rest in the scene.

978-1-4244-6542-2/$26.00© 2010 IEEE

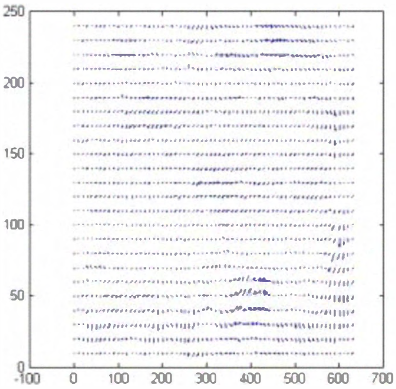

Fig.1 optical flow field

Based on the analyze above, this research addresses a method of using object rim to speed up the optical flow calculation, the main idea is to use the object displacement data from proportional object rim in serial images to calculate optical flow. By the following steps: first, assuming all light source in the area stays still, and use N.Otsu's[6] method to segment the background and object in the images, then use the perimeter of the object and core as the characteristic to find out the matching object in serial images, and this would acquire the matching objects after displacement of position due to the changes of time and environment. By using the method described above, we can lessen calculating unimportant area and lower calculating time.

Certain restriction would occur when using traditional optical flow method for calculating optical flow, this might lead to optical flow value miscalculation, and also lower calculation efficiency. Therefore, by using the method suggested in this research, we can lower calculating time and enhance object tracing effect.

II. RELATED WORK

A. *Optical Flow*

The optical flow technique begins by taking consecutive pictures of the targeting object. In every time interval, there exists a vector indicating the movement of the object relative to camera. Object's movement will cause shifts in brightness intensity of the object. This phenomenon is used to find matching feature and location among images. With the assumption that the light intensity stays constant over whole series of images, the movement direction and displacement can be tracked. As shown in Fig. 2, a point v is located at coordinate (x, y) in the initial image. As time elapses, the light intensity changes and a new image with different brightness distribution has formed. As shown in the figure, the second image is captured after ΔT and point v has moved to a point u, with new coordinate values $(\bar{x}, \bar{y})$. Connecting the old coordinate to the new coordinate forms a pair of pints (u, v) and this pair of points describes the degree of change in light source intensity. Connecting old coordinates to new coordinates yield a set of vectors and these vectors represent the optical flow distribution.

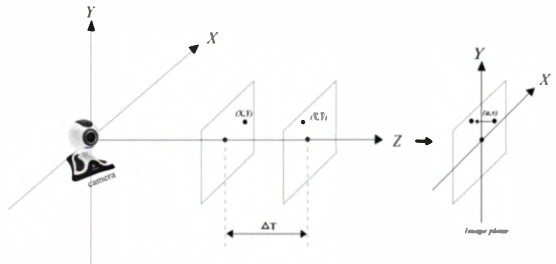

Fig.2 The formation of optical flow

N.Otsu defines optical flow as an approximation gradient based method for dynamic vector evaluation . Since the formation of optical flow field requires the original coordinate on the first image and its corresponding new coordinate on the second consecutive image, a single image is not sufficient for calculating optical flow. As described in equation (1), the coordinate $I(x, y, t)$ shifts to a new coordinate as time elapses:

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \tag{1}$$

Image flow[13] is the flow of pixels on the projected image. Target object's 3D coordinate is transformed to 2D coordinate on the image (projection) plane. Image flow processes and tracks target object's shape and does not need to care about changes in light intensity. As shown in Fig.3, the point P with world coordinate (X,Y,Z) moves towards the direction pointed by vector S. On the image plane, the projected point is P(x,y) and velocity vector is $\bar{S}$.

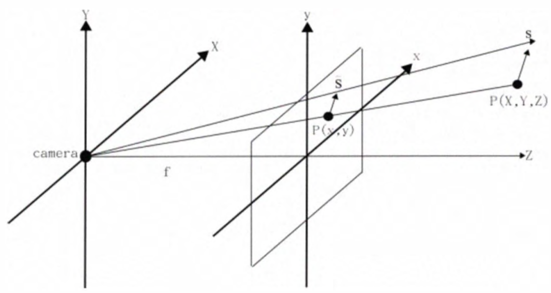

Fig.3 The formation of image flow

When calculating brightness intensity of optical flow and searching for matching features, there exist many interfering factors in the environment and difficult problems. Hence, many researchers limit the application of image flow to the problem of solving constraints in optical flow calculations. By doing this, constraints of shadows and interfering factors in the environments can be resolved. Deploying the concept of image flow in the optical flow algorithm does not affect other steps of the algorithm.

B. *Lucas-Kanade Algorithm*

Lucas–Kanade algorithm[12] is one of widely used methods in fields that are related to object motion. This algorithm uses the concept of optical flow to find variation of object's movement. mentioned that using Gaussian weighted function to effectively limit the strengths of neighboring pixels is one way to determine the coordinate of the optical flow. This is represented by the following two equations:

$$\begin{aligned}\frac{\partial E(u,v)}{\partial u} &= \sum_{\vec{x}} g(\vec{x})[uI_x^2 + vI_xI_y + I_xI_y + I_xI_t] = 0 \\ \frac{\partial E(u,v)}{\partial v} &= \sum_{\vec{x}} g(\vec{x})[vI_x^2 + uI_xI_y + I_xI_y + I_xI_t] = 0\end{aligned} \tag{2}$$

The function E in the above equations is the least square estimate used to lower the deviations, as shown in equation 3. In equation 3, the least square estimate is adopted to decrease square error. The $g(\vec{x})$ in the equation is the Gaussian weighted function used to estimate strength.

$$E(u,v) = \sum_{\vec{x}} g(\vec{x})\left[I_x u + I_y v + I_t\right]^2 \tag{3}$$

Then the partial derivatives are plugged to equation 4 to calculate the x, y coordinate and direction t of the optical flow:

$$\begin{gathered}\vec{u} = (A^TA)^{-1}A^Tb \\ \underbrace{\begin{bmatrix}\sum gI_x^2 \sum gI_xI_y \\ \sum gI_xI_y \sum gI_x^2\end{bmatrix}}_{A^Ta}\begin{bmatrix}u \\ v\end{bmatrix} = -\underbrace{\begin{bmatrix}\sum gI_xI_t \\ \sum gI_yI_t\end{bmatrix}}_{A^Tb}\end{gathered} \tag{4}$$

Through equation 4, the equations to estimate optical flow can be obtained:

$$\begin{cases}I_x = m_x \times (I_1 + I_2) \\ I_y = m_y \times (I_1 + I_2) \\ I_t = m_t \times (I_2 - I_1)\end{cases}, \quad \begin{cases}m_x = \frac{1}{4}\begin{pmatrix}-1 & 1 \\ 1 & 1\end{pmatrix} \\ m_y = \frac{1}{4}\begin{pmatrix}-1 & -1 \\ 1 & 1\end{pmatrix} \\ m_t = \frac{1}{4}\begin{pmatrix}1 & 1 \\ 1 & 1\end{pmatrix}\end{cases} \tag{5}$$

As for Lucas-Kanade Algorithm, Horn and Schunck[4] have brought in the rule of fixed brightness and the constraint of smoothness. This method is consistent for estimating all light sources (I_x, I_y, I_t) .Through the constraint of smoothness, changes, magnitudes and directions of neighboring motion vectors will be continuous and smooth. In short, they should refer to the same point at the same time. For estimating the intensity of optical flow, a bright point is treated as the center and intensity calculations are performed on the neighborhoods.

III. THE METHOD

The algorithm proposed in this study applies optical flow techniques to edges of the target object. In this manner, the time for optical flow calculation can be decreased and the goal of tracking object is still met. Specifically, the processing targets are edges that are prominent and easy to distinguish. Bring this concept to optical flow technique for feature extraction has waived calculations of unwanted and redundant features. Furthermore, more effective tracking may result if only the most prominent edge features are considered. Another advantage is that the computation time is greatly reduced. Fig. 4 depicts the flow of the proposed algorithm. The first part of the algorithm is the pre-processing. The first two steps in this part transform input images to gray scale images. The next step is to segment foreground out from background by using method devised by Otsu. The next part utilizes masks to process pixels that lie between black and white regions. In this part, whole edges of the target object are obtained. The final part is to obtain optical flow distribution. The first two steps in this part are to find matching features in two consecutive images based on object's centroid and edges. The third step is to connect all edges that correspond to the same object. The final result is a picture of optical flow distribution. The next section will explain how to find edges and how to apply them to the optical flow algorithm.

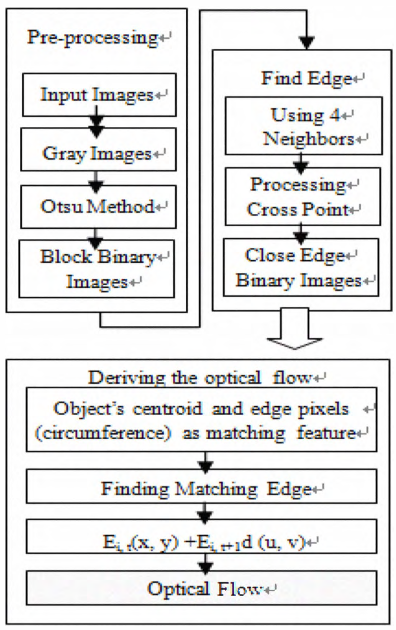

Fig.4 Algorithm flow

A. *Pre-processing*

In the proposed algorithm, the objective of pre-processing is to separate the target object and background. As mentioned in the previous section, sampled frames are first transformed to grayscale images to facilitate latter processing steps. Then we use Otsu's method to obtain a threshold value. Otsu's method primarily uses statistical methods to perform calculations on pixel distribution of the image. The calculation result is a threshold value that can be used to binarized the image. The binarized image will be a black and white image where the white part represents the background and the black part indicates the foreground region. The foreground region is the target to be processed by the algorithm.

The general algorithm of Otsu's method is as follows. Assume that there exists a threshold value *TH.* Pixels can be classified into two groups where one group contains pixels with values between 0 to *TH* and another group contains pixels with values between *TH*+1 ~ 255. Equation (11) can be applied to obtain the variance within each group. Equation (6) can be applied to obtain the difference value between the two groups. An optimal *TH* value would minimize the variance within each group and maximize the difference value between two groups.

$$\sigma_{\omega}^{2} = \omega_0 \sigma_0^2 + \omega_1 \sigma_1^2 \tag{6}$$

In equation 6, ω_0 is the probability of having 0~*TH* and ω_1 is the probability of having *TH*+1~255. σ_0^2 and σ_1^2 are the respective variances of the two pixel value ranges. With some calculations, variance within group σ_{ω}^2 can be obtained. Equation 7 is to calculate difference value σ_b^2 between two groups. In equation 7, μ_0 and μ_1 are the respective mean pixel values of the two pixel value ranges.

$$\sigma_b^2 = \sigma^2 - \sigma_{\omega}^2 = \omega_0 \omega_1 [\mu_0 - \mu_1]^2 \tag{7}$$

Fig. 5 shows a set of images and corresponding binarized images, which are binarized with the optimal *TH* values obtained through Otsu's method. In this figure, it is seen that backgrounds are nicely segmented and the foreground (the target object) is kept.

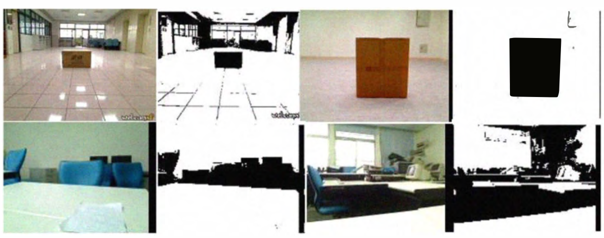

Fig.5 Results by using Otsu's method

B. *Finding the edges of the target object*

After pre-processing, the foreground and the background are segmented out from the image. Then we use a 4-neighbor mask to process prominent edges. Referring to Fig. 6, if the pixel in the center does not have any non-zero neighboring pixel, then this pixel is considered to be an independent pixel. However, if there exist one or more non-zero neighboring pixels, then the central pixel is considered to be non-independent pixel. In the case of non-independent central pixel, the algorithm searches its neighboring pixels clockwise for more non-independent pixels and then connects all continuous non-independent neighboring pixels.

	P(x,y-1)	
P(x-1,y)	P(x,y)	P(x+1,y)
	P(x,y+1)	

Fig.6. 4-neighbor processing mask

As shown in equation (8), independent pixels are removed by the above method and connected continuous pixels form closed edges of the target object. That is, connecting non-independent pixels can obtain the profile of the target object. Independent pixels are discarded.

$$\begin{cases} P(x,y) = 0, delete \\ P(x,y) = 1 \,\&\&\, Independent, delete \\ P(x,y) = 1 \,\&\&\, Dependent, Link \end{cases} \tag{8}$$

As depicted in Fig. 7, edges of the object can be segmented out by using the 4-neighbor mask method. Image a, c, e, and g are obtained by using Otsu's method while image b, d, f and g are obtained by using equation 8.

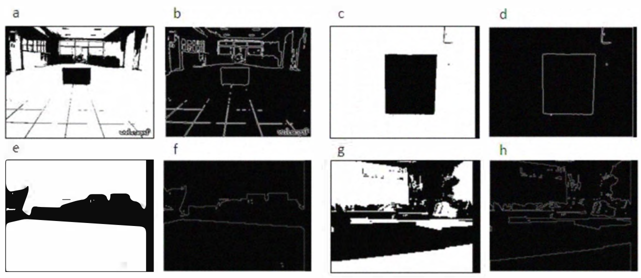

Fig.7 Binarized images before and after processing

C. *Obtaining the optical flow*

Through the procedure as described in section 3.2,a number of binarized images containing closed-edge objects are generated. This study proposes a method that uses centroid and edge pixel (circumference) as object's matching feature. For each closed object, the method finds that object's matching objects, which have moved, in the rest of consecutive images. Lastly, the method uses the information provided by a pair of matching objects to form a optical flow.

Centroid is referred to as center of area or center of mass of the shape. It mainly describes the center of a line, an area or a geometric shape. This study adopts a method that captures the cross section of an object to find its centroid. The cross section is treated as a symmetric cardboard and the centroid of the cross section is the cardboard's center of mass. If the cross section has a symmetry axis, then the centroid will be on the symmetry axis. If there exist two symmetry axes, then the centroid is the cross point of two symmetry axes. Through obtaining the coordinate values of the centroid of composite cross section which consists of n cross sections, the most suitable centroid can be obtained. As shown in equation (9), $Area_i$ is the area of the i th cross section. Variables x_i and y_i respectively represent x and y coordinates of the i th cross section. Variables x' and y' respectively represent the x and y coordinates of the most suitable centroid point.

$$x' = \frac{\sum_{i=1}^{n} Area_i x_i}{\sum_{i=1}^{n} Area_i}$$

$$y' = \frac{\sum_{i=1}^{n} Area_i y_i}{\sum_{i=1}^{n} Area_i} \tag{9}$$

Through the calculation of equation (9), centroid of every object can be obtained. The algorithm proceeds to calculate the total amount of edge pixels. Sum of edge pixels is also referred to as the circumference which represents the distance travelled around the closed-curve line once. The equation of calculating object's circumference is as described in equation (10):

$$L_i = \sum_{i=1}^{n} e_{i,t}(x,y)$$

$$L_j' = \sum_{j=1}^{n} e_{j,t+1}(x,y) \tag{10}$$

In equation (10), circumference L_i is composed by n pixels in the original image. Summing all n pixels yields the sum of the object's total edge pixels. The variable e represents the boundary of the i th object. The term $t+1$ represents the image captured after moving. L_j' is the circumference of each object in the image captured after moving.

After obtaining the centroid and circumference features through equations (9) and (10), an object's matching positions can be found by using these feature values. As shown in the following procedure, the first step is to find the centroid $(x',y')_t$ before movement and centroid $(x',y')_{t+1}$ after movement. The second step calculates circumferences L_i 、 L_j' which correspond to the circumference of the object before movement and after movement respectively. The third step checks whether the moving distances (u,v) between $(x',y')_t$ and $(x',y')_{t+1}$ and the moving distance between L_i and L_j' are all smaller than 5 or not. If all moving distances are indeed smaller than five, then it implies that the two objects are matching objects.

```
For i = 1 to n
  according to (x',y')_t and L of i th object of t
  For j = 1 to n
    search for n and L' in j for every t+1
        if {u ,v of | (x',y')_t - (x',y')_{t+1} | < 5
&& | L - L' | < 15}
    then i and j are matching features
  End
End
```

By using above procedure, the displacement of object's x coordinate value and y coordinate value $d(u,v)$ can be obtained. By adding information about moving direction and moving displacement to edges of every matching object, an optical field distribution can be formed. This process is illustrated by equation (11):

$$\text{Optical Flow Field} = e_{i,t}(x,y) + e_{i,t+1}d(u,v)$$

$$d(u,v) = e_{i,t+1}(x,y) - e_{i,t}(x,y) \tag{11}$$

In the above equation, the term $e_{i,t}$ represents a pixel that belongs to an edge of the i th closed object in the original image t . The number pair (x,y) tells the coordinate values of that pixel. The term $e_{i,t+1}$ represents a pixel that belongs to an edge of the i th closed object in the image $t+1$ after moving.

To effectively mitigate the constraints in traditional optical flow techniques, this study proposed three steps to improve the technique. The first step is to transform all consecutive images into grayscale images and then segment out all objects in these images by using Otsu's method . The next step is to extract all objects' edges which lie between the background and the objects. This step help obtain binarized images containing all closed-edge objects. The third step calculates centroid as well as circumference of each closed object and then makes them the matching feature. The algorithm repeats all these three steps to process each one of other consecutive images and obtains matching .Finally, through finding the matching feature and calculating moving displacement and moving direction, the object that has moved can be tracked.

IV. Experimental Results

Some real-world scenes were used to test against the algorithm proposed in this study. Experiment results have proved that the proposed algorithm does indeed generate optical flow field faster than others while the optical flow field still preserves important information. In the first experiment, the camera moved to the right. Fig 8(a) depicts the original image. Fig 8(b) is the image after movement had occurred. Fig 8(c) shows the optical flow field generated by the Lucas-Kanade Algorithm while Fig.8(d) shows the optical flow field generated by the algorithm proposed in this study.

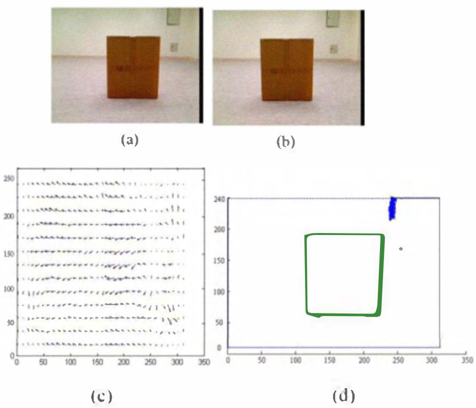

Fig.8 Optical flow fields generated by Lucas-Kanade Algorithm and the algorithm proposed in this study

For the optical flow field generated by Lucas-Kanade Algorithm, optical flows are distributed evenly. There is no prominent information available in the scene to help determine the movements of objects. Under this circumstance, the algorithm proposed in this study is able to outline each object clearly and identify objects having prominent features. Some information of the scene is still lost. However, most important information has been extracted.

Fig. 9 illustrates an image where the camera also moved towards the right but the scene was more complex. Fig.9(a) is the original image. Fig.9(b) is the image after camera had moved. Fig 9(c) displays the optical flow field generated by the Lucas-Kanade Algorithm while Fig.9(d) displays the optical flow field generated by the algorithm proposed in this study.

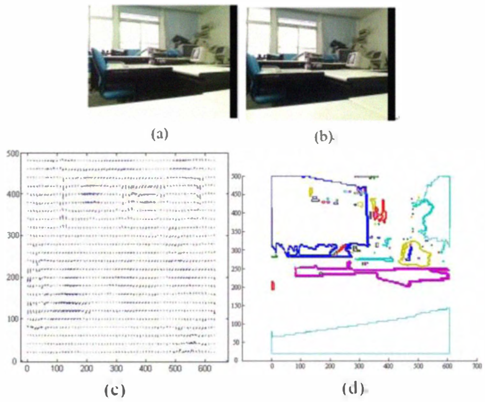

Fig.9 Optical flow fields of images capturing more complex scene generated by Lucas-Kanade Algorithm and the algorithm proposed in this study

When the scene is more complex, Lucas-Kanade Algorithm still generates evenly distributed optical flow field. However, the algorithm proposed in this study is able to outline each object clearly and identify objects having prominent features. Although partial information of the scene has been lost, important information was captured.

The algorithm proposed in this study can decrease computation time for calculating optical flows. The following pictures shows the scene and compare two execution time which correspond to Lucas-Kanade Algorithm and our proposed algorithm. The resolution of the images are: 320x240 for scene A, 620x480 for scene B, 620x480 for scene C and 320x240 for scene D

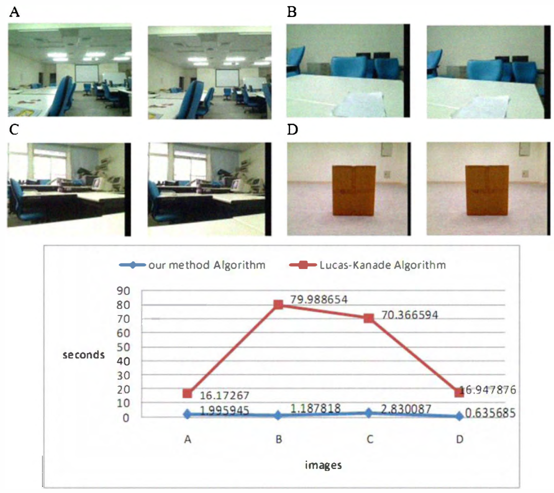

Fig.10 Computation time of the algorithm in this study and Lucas-Kanade Algorithm

As can be seem in Fig. 10, regardless of the complexity of the scene, the computation time required by the algorithm proposed in this study is significantly less than traditional optical flow techniques. Since typical optical flow algorithms consider all pixels when calculating optical flows, a higher resolution would make performance worse. The algorithm proposed in this study filters out unimportant information during preprocessing stage and uses objects' edges to generate optical flow field. The edges are found by using centroid and circumference as matching feature which does not require high computation resources. As the result, the overall computation time is decreased.

V. CONCLUSION

This paper proposed an improved object tracking algorithm that applies the technique of optical flow. The proposed algorithm utilizes centroid and circumference as the matching features to obtain edges of objects in question. This method has been experimented to effectively acquire regions having important information and discard regions without important information. Although traditional optical flow techniques have been widely used, they consume too much computation time and do not change regions having relatively important information.

This study adopts Otsu's method to segment objects in an image out of background. The algorithm discards any unnecessary feature and uses centroid and circumference as the matching features. Based on the two matching features the algorithm obtains the edges of objects and then processes these edges as described in section 3.1 and 3.2. The circumference and centroid as matching features help find matching objects and hence indentify edges' movement information. As indicated by experiment results, the proposed algorithm saves a lot of unimportant computations and achieves a relative fast performance.

REFERENCES

[1] B. Axel, S. Tamara, L. Sara, D. Carmen, A. Johnson, J. Budi, E. David, D. Gamini, "Optical flow based analyses to detect emotion from human facial image data," Expert Systems with Applications, 2010.

[2] D. Javier, R. Eduardo, A. Rodrigo, B. Jose Luis, "Superpipelined high-performance optical-flow computation architecture," Computer Vision and Image Understanding, vol.112, pp.262-273, 2008.

[3] D. Simon, C. Vinod, S. Sridha, "An adaptive optical flow technique for person tracking systems," Pattern Recognition Letters, vol.28, pp.1232-1239, 2007.

[4] H. Berthold K.P., S. Brian G., "Determining Optical Flow," Artificial Intelligence, vol.9, pp.185-203, 1981. in press

[5] K.Farid, F. Isabelle, N. Kenzo, "Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles," Robotics and Autonomous Systems, vol. 57, pp.591-602, 2009.

[6] N. Otsu, "A threshold selection method from gray-scale histogram," IEEE Tran. On System, Man and Cybernetics, vol.8, pp.62-66, 1979

[7] N. Fatemeh Karimi, G. Ben G.H., H. Serge P., "Optical flow based vehicle tracking strengthened by statistical decisions," ISPRS Journal of Photogrammetry & Remote Sensing, vol.61, pp.159-169, 2006. in press

[8] O. Nils, G. Bjorn, J. Stefan Karlsson, "Location of innervation zone determined with multichannel surface electromyography using an optical flow technique," Journal of Electromyography and Kinesiology vol.17, pp.549-555, 2007.

[9] P. Jean, F. Anne Marie, F. Michelangelo, B. Bernard, "Unintentional modulations of human gait by optical flow," vol.38, pp.275-281, 1990. in press

[10] P. janez, S. Vildana, K. Matej, P. Matej, P. Klemen, K. Stanislav, "Histograms of optical flow for efficient representation of body motion," Pattern Recognition Letters, vol.33, pp.1369-1376, 2010.

[11] S. Jeongho, K. Sangjin, K. Sangkyn, L. Seong-Won, P. Joonki, A. Besma, A. Mongi, "Optical flow-based real-time object tracking using non-prior training active feature model," Real-Time Imaging, vol.11, pp.204-218, 2005.

[12] S. Lee Yee, M. Siti Mokri, H. Aini, I. Norazlin, M. Mohd-Marzuki, "Motion Detection Using Lucas Kanade Algorithm and Application Enhancement," 2009 International Conference on Electrical Engineering and Informatics, Selangor, Malaysia, 5-7 August 2009.

[13] W. Kwangyoen, W. Allen M., "The analytic structure of image flows: Deformation and segmentation," Computer Vision, Graphics, and Image Processing, vol.49, pp.127-151, 1990. in press

[14] Z. John.S, "Towards Bayesian real-time optical flow," Image and Vision Computing, vol.22, pp.1051-1069, 2004.