

Automatic Detection and Tracking in Multi-fisheye Cameras Surveillance System

Dinchang Tseng, Chienhung Chen*, Chiahui Tseng

The Institute of Computer Science and Information Engineering, National Central University, Jhongli, Taiwan.

* Corresponding author. Tel.: +886-918-220945; email: chienhung66@gmail.com

Manuscript submitted March 23, 2017; accepted May 28, 2017.

doi: 10.17706/ijcee.2017.9.1.370-383

Abstract: The view scope of a single camera is finite and limited by scene structures. Multi-fisheye cameras can monitor a wide area and trace a complete trajectory of a moving object. In this study, an automatic detection and tracking system with two fisheye cameras for environment surveillance is proposed. The proposed system is composed of two major modules: foreground detection and foreground tracking. The background subtraction method is first applied to extract targets. Then use Kalman filtering for pedestrian motion prediction. A transform table is pre-established to associate multi-cameras data in the overlapping areas. When object across disjoint camera views, the data in the lookup table can provide enough information to realize the moving object in camera views actually belonging to the same object, and keep consistent labels on the object. To improve the reliability of the tracking performance, motion and color appearance features are used to match the detected objects in different cameras. It demonstrates that the proposed method can work well under challenging conditions, such as light change, shadow interference, object occlusion.

Key words: Surveillance system, multi-fisheye cameras, Kalman filter, codebook background model.

1. Introduction

Automated security surveillance systems need to develop [1], not only can monitoring and processing abnormality in real time, but also can save money on labor costs. The view scope of a single camera is finite and limited by scene structures. In order to monitor a wide area and trace a complete trajectory of a moving object, multi-cameras video surveillance systems be used. The view angle of a fisheye camera is 180 degree, so it can cover a wider field of view than a normal camera. Thus, in the same surveillance environment, only a few fisheye cameras can replace many traditional cameras to survey the events. In this thesis, we propose an automatic detection and tracking system with two fisheye cameras for environment surveillance. The proposed system is composed of two major modules: foreground detection and foreground tracking. To do surveillance on the environment, we must first be able to capture a moving object in the image sequence.

Temporal difference method [2]-[4], using a continuous image sequences adjacent two images do the subtraction operation, set the threshold to get moving pixels. Background subtraction method [5], [6] using a while to establish the initial background model. Then use the image sequence and the background model to compare new entrants to detect moving pixels. This method can be achieved most complete object pixels, but it is quite sensitive to changes in ambient light, and cannot overcome the problem of camera movement. Optical flow method is using a variation on the estimation of image pixels to infer the position of moving

objects. This method can detect when you move the camera still moving object independently, but the disadvantage is fairly complex calculation, if there is no specific hardware, can't be applied in real-time systems. Background subtraction is quite widely used method of dividing the movement area in the image sequence. The simplest and most common is the use of a background subtraction few seconds of time to build a normal distribution in the background of each pixel, then sent to new imaging sequences and background model for comparison, and set the threshold to distinguish between the foreground and background points many systems take advantage of such a method to detect the movement of the pixel [7]-[12].

Horprasert *et al.* [13] proposed a method to detect moving objects from a static color background images, background model to establish a reference image using a statistical approach. Javed *et al.* [14] proposed a gradient-based background subtraction method, the feature vector from the gradient size and the direction of the composition gradient, based on the establishment of a gradient of the background model. Stauffer and Grimson *et al.* [15] proposed background model of Gaussian mixture model, the background model established is on each pixel to establish several Gaussian distribution, and in accordance with foreground or background Gaussian distribution to classify. This method can avoid detecting the leaves waved with the wind. Kim *et al.* [16] proposed CB algorithm adopts a quantization/clustering technique, used color distortion and brightness distortion classified as background, and established multimode background model. This method can to encode moving backgrounds or multiple changing, and the capability of coping with local and global illumination changes backgrounds.

The main function of multi-cameras tracking model is to aid detection model, when occurring person move into the pillar of environment or two figures cross or overlap each other, object of foreground does not detected by foreground detection system, or identify two objects into one object. The challenge of multi-cameras surveillance is maybe because of angle of view, brightness different, and produce object distortion, color distortion, masking, resolution, posture change and other problem are make it more difficult to match object in tracking technology. Wang *et al.* [17] the multi-cameras tracking technology in accordance with the camera setup relations is divided into two categories: overlapping field-of-view (*FOV*) and non-overlapping field-of-view, both tracking methods in object match technology has a very large difference. Overlapping field-of-view methods [18] some use the camera correction [19]-[22], using different camera observed the feature point conversion to point of three-dimensional world coordinate system. This need take time to do correction, and the disadvantage is the feature point in a different camera, does not always can convert to the same position of world coordinate. Homography [23], [24] used two-dimensional information to create a corresponding point, and matching to a common plane, then use camera *FOV* boundary to match object. Khan *et al.* [25] proposed automatically finding *FOV* method, when the object occur masked or observing object lost, use position of previous time to help calculate the estimated position of the object. Non-overlapping field-of-view methods [26], [27] sometimes because of the position of the camera setup too far away, cause the temporal relationship unreliable, and in the overall monitor area the camera to capture the foreground object was too small, makes the face feature cannot be used to identify. So non-overlapping field-of-view tracking is often used exterior features, as color, point edge, texture, in order to enhance reliability match is typically used in combination more features.

To match objects in different cameras the color information is more sensitive to light. In multi-cameras system have to solve the brightness problem, Javed *et al.* [28] proposed brightness transfer function (*BTF*) to adjust the color difference between the two cameras, and training the luminance conversion function need to collect a lot of images and mark the corresponding position of the camera. Rahimi *et al.* [29] used normalized RGB color, remove the effect caused by brightness. Chen *et al.* [30] used *HSV* color space and determine color similarity only comparison *H-S* channel, removed the brightness *V* channel.

Because fisheye images converted to a normal two-dimensional images methods cannot find a good algorithm to real-time processing images and continuous to monitor. So in this paper, we propose and develop can direct processing fisheye images algorithm, the system is composed of overlooking scene by two fisheye cameras. System can continuously monitor staff moving trajectory, in the overlapping areas of two cameras divided view border, and using pre-established the corresponding table to match foreground object and finding the most appropriate match. Using codebook foreground detection algorithm to adjust brightness change in the moment and make have the noise rejection capability. In accordance with color information as a basis to determine foreground and background with more tolerance. Our multi-cameras system using overlapping field-of-view, therefore, the camera uses two overlapping areas find the corresponding relationship and match foreground object, to implement continuous tracking function.

This paper is structured as follows: The details of the proposed techniques for use in the system are presented in Sections in Section 2. Experimental results are included in Section 3, followed by conclusions in Section 4.

2. Proposed Techniques

The proposed system is composed of two major modules: foreground detection and foreground tracking. These are described in detail below.

2.1. Foreground Detection

In foreground detection is divided into four steps of the foreground detection. First, using background modeling to obtain the foreground, and then to remove shadow and noise, and finally on the foreground image to capture feature.

2.1.1. Background modeling

The codebook algorithm to construct a background model from long observation sequences. It builds a codebook consisting of one or more codewords for each pixel, and each pixel are clustered into the set of codewords based on a color distortion metric together with brightness bounds. It captured structural background motion over a long period of time and captured structural background variation due to periodic-like motion over a long period of time is better than Gaussian mixture model. Experimental of brightness change in the moment and who wears different color clothes at the same time to measure the color tolerate, codebook background model has best result, as shown in in Fig. 1 and Fig. 2.

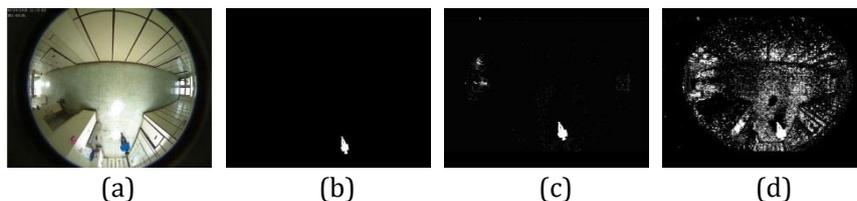


Fig. 1. Detection results of brightness change in the moment. (a) Original image. (b) Ground truth image. (c) Codebook background model. (d) Gaussian mixture model.

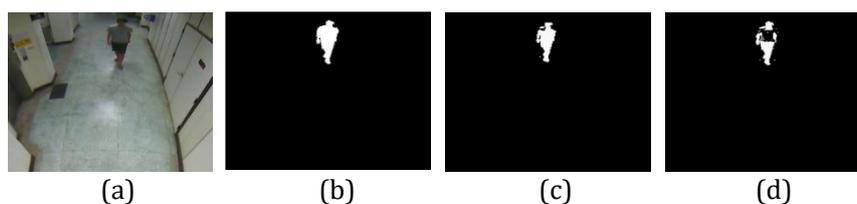


Fig. 2. Wears different color clothes at the same time to measure the color tolerate results. (a) Original image. (b) Ground truth image. (c) Codebook background model. (d) Gaussian mixture model.

Each pixel has a different codebook size based on its sample variation. One codebook consisted with much codewords. Let $C = \{c_1, c_2, \dots, c_L\}$ represent the codebook for the pixel, L is number of codeword, each pixel not owns the same number of codeword. Each codeword $c_i (i = 1, \dots, L)$ consists of an RGB vector $v_i = (\bar{R}_i, \bar{G}_i, \bar{B}_i)$ and a 6-tuple $aux_i = \{\hat{I}_i, \check{I}_i, f_i, \lambda_i, p_i, q_i\}$. Where aux_i contains brightness values, (\check{I}, \hat{I}) is min and max brightness, respectively, of all pixels assigned to this codeword. f is frequency with which the codeword has occurred. λ is maximum negative run-length (MNRL) defined as the longest interval during the training period that the codeword has NOT recurred. p_i is first access times and q_i is last access times.

To determine which codeword will be the best match, using a color distortion measure and brightness bounds. If compute the pixel distance of color and brightness, has the nearest cluster mean, then assigned to background pixel. In Kim *et al.* [16] proposed the color model, as shown in Fig. 3. Consideration in the RGB space, let input pixel of training images $x_t = (r, g, b)$ and a codeword c_i where $v_i = (\bar{r}, \bar{g}, \bar{b})$.

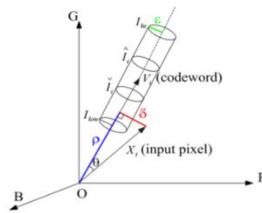


Fig. 3. Kim *et al.* [16] proposed the color model.

Orthogonal distance of x_t to v_i is δ can be calculated by

$$\|x_t\|^2 = r^2 + g^2 + b^2, \tag{1}$$

$$\|v_i\|^2 = \bar{r}_i^2 + \bar{g}_i^2 + \bar{b}_i^2, \tag{2}$$

$$\langle x_t, v_i \rangle = (\bar{r}_i r + \bar{g}_i g + \bar{b}_i b). \tag{3}$$

$$p^2 = \|x_t\|^2 \cos^2 \theta = \frac{\langle x_t, v_i \rangle^2}{\|v_i\|^2}, \tag{4}$$

and

$$colordist(x_t, v_t) = \delta = \sqrt{(\|x_t\|^2 - p^2)} \tag{5}$$

To solve for brightness changes and shadow in detection, store \hat{I} and \check{I} statistics, which are the min and max brightness of all pixels assigned to a codeword. Using \hat{I} and \check{I} of aux and to set a certain range, and allow the brightness change to vary in it. The brightness function is defined as

$$\text{Brightness}(I, (\check{I}, \hat{I})) = \begin{cases} true, & \text{if } I_{low} \leq \|x\| \leq I_{hi}, \\ false, & \text{otherwise.} \end{cases} \tag{6}$$

where

$$I_{low} = \alpha \hat{I}, I_{high} = \min\{\beta \hat{I}, \frac{\hat{I}}{\alpha}\}. \tag{7}$$

2.1.2. Background subtraction method to detect foreground

In equation (6), define background subtraction result $M(x', y', t)$ is pixel of new enter image location (x', y')

as

$$M(x', y', t) = \begin{cases} 0, & \text{if there is no match,} \\ 1, & \text{otherwise.} \end{cases} \quad (8)$$

RGB color and brightness value of x will be subtracted at the same image location on codebook model, if the color distortion is less than the detection threshold ε and brightness lies within the brightness range, then to determine match. Persistent to process image until finishing all pixels, can obtained binarization image of foreground, as shown in Fig. 4.

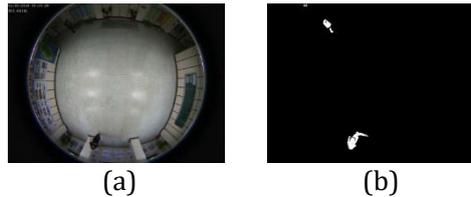


Fig. 4. Detected foreground result. (a) Original image. (b) Binarization image of foreground.

2.1.3. Foreground process

Use morphology erosion and dilation operator obtained best binarization image, as shown in Fig. 5.

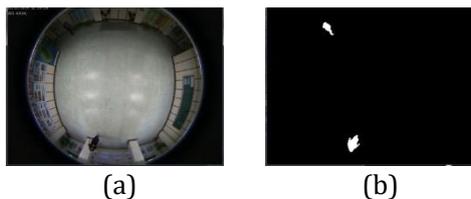


Fig. 5. Foreground process results. (a) Original image. (b) Image after morphology method.

The influence of the environmental light source, these may occur shadow of the foreground. The shading and background have similar chromaticity values, the difference lies in the shadow of the lower luminance than the background. In equation (6) defined, and appropriate to change α and β values can detected belong to the shaded portion and removing it from the foreground area. Results of removed the shadow in different brightness, as shown in Fig. 6. In case of strong light source, because the environment brightness is uniform, so the shade clearly on the ground. In case of weak light source, when person move between light area and dark area will has clearly shadow, and the shadow area was obvious.

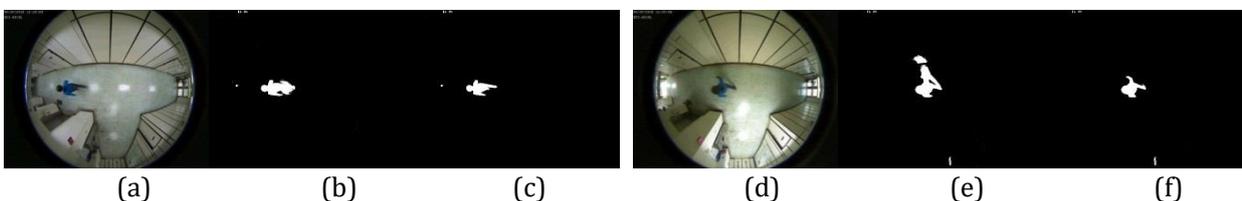


Fig. 6. Results of removed the shadow in different brightness. (a) Original image of strong light source. (b) Before removed the shadow of image (a). (c) After removed the shadow of image (a). (d) Original image of weak light source. (e) Before removed the shadow of image (d). (f) After removed the shadow of image (d).

Image segmentation use neighbor connected method to obtain different foreground regions, then to capture features of foreground. Different foreground regions obtained result, as shown in Fig. 7. Different foreground regions used different color shown.

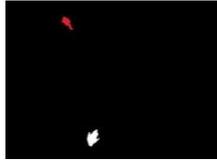


Fig. 3. Different foreground regions obtained result.

2.1.4. Background upgrade

Environment surveillance usually works 24 hours, the environment may be changed a little or vigorously during one day, so designing the background upgrading has adaptability, ensure the foreground object can be capturing correct.

If foreground object appears during the course of training, using background layers model, these is to establish two layers of background, first layer determined pixel belongs to foreground or background. Pixel detected of background to classify in first layer, pixel detected of foreground will be learn and to classify in second layer, some time, pixel detected of foreground will handle to first layer from second layer, to reach of upgrading function of greater difference.

Background changes not only cause by moving objects, the illumination changes is other problem, in this paper the illumination changes problem has divided into (i) gentle illumination change (the naturally gradations of illumination), using background upgrade model to process, it not detected foreground object. (ii) rapid illumination change (turn on or turn off the light), though it happen for time shortest, but will cause by background change enormous. Because the color is changed largely, so will reestablishing the background. In this paper, if the foreground pixel more than sixty percent of the total pixel of the input image, the background will re-establish.

2.1.5. Features capture of foreground

After foreground object obtained then using neighbor connected method to take out foreground regions, and to record data of foreground, this data will become input data of track model, for reference to object match. Then use frame outline of foreground region to select the most appropriate polygon, and using this polygon contents for foreground region to capture outline (the height, width, and color distribution of image) and moving feature (the position of coordinate), as shown in Fig. 8.



Fig. 4. Features captured of foreground process results.

2.2. Foreground Tracking

Tracking model operation methods has divided into (i) single camera tracking method, contains object match, Kalman filter use and the system parameters setup. (ii) multi-cameras tracking method, contains system condition limit and tracking match on different cameras.

2.2.1. Features matching

Multiple object detecting can divided into two steps, first step is using Kalman filter to predict the foreground position of next time, and suppose the person moving with the fixed speed, so choice the move distance for the features. Using center coordinate of Kalman filter prediction $p_1(x_1, y_1)$ and center coordinate of foreground bounding box $p_2(x_2, y_2)$ to calculate distance, the distance of two coordinates used Euclidean distance measurement, as

$$dist(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{9}$$

In this paper, the distance threshold set 100 pixels; this is in accordance with general pedestrian velocity about 3.6 kilometer/hour (1 meter per second). In this velocity, suppose the maximum move distance of neighborhood frame as one tile (90cm * 90cm) in the frame. The feature of fisheye camera is the distance far with frame center has largest distortion, and has least pixel in the same distance of frame border. Therefore, system chose tile length in frame of nearest frame center as the threshold, as shown in Fig. 9. Every tiles edge has marked with blue lines, and calculated the length of between each blue line as shown in Table 1.

Table 1. The Length Variation Results of Different Position in the Frame

| Tile number | 1 | 2 | 3 | 4 | 5 |
|---------------------|-----|-----|----|----|----|
| Frame Length(pixel) | 100 | 100 | 90 | 60 | 40 |

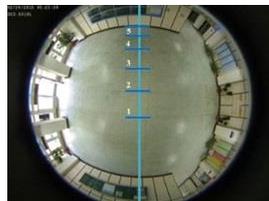


Fig. 5. Every tiles edge marked with blue lines, the distance is reduced.

Second step is to match color of detected multiple objects, processing error foreground object cause nearest distance, then use *RGB* color space transform *HSV* color space to remove the influence of illumination. The image feature described result of *H-S* color distribution map, as shown in Fig. 10. In Fig. 10(b), *x*-axle is *H* of *H-S* color distribution, and the degree 0 to 360 is divided into 30 frames, *y*-axle is *S* of *H-S* color distribution, and the gray-level 0 to 255 is divided into 32 frames. This became a two-dimension color distribution map, illumination to represent amount of distribution, the color that the brightest is corresponding is blue.

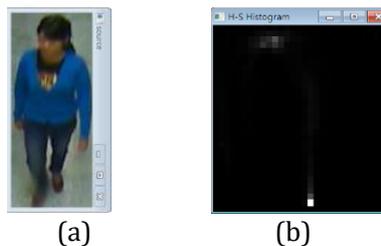


Fig. 6. The image feature described result of *H-S* color distribution map. (a) Blue jacket and dark blue trousers of foreground object. (b) *H-S* color distribution map of foreground object.

In order to match foreground object correct, the center position using predicted position of Kalman filter, using height and width frame of previous moment to choose image contents at record of object database, then to match similarity of color with foreground detected result. The color feature matching is to compare color distribution map of two frames, matching method is use Bhattacharyya distance, as

$$dist(h_1, h_2) = \sqrt{1 - \frac{\sum_I \sqrt{h_1(I)h_2(I)}}{\sqrt{\sum_I h_1(I)\sum_I h_2(I)}}} \tag{10}$$

When the result of two frames has nearest distance of similarity distance, the Bhattacharyya distance as $[0,1]$, through numerous experimented results, the threshold of discriminates similarity of two foreground objects is 0.4. Original image and two test images matched results of color distribution map, as shown in Fig. 11. Original image is blue jacket and dark blue trousers; one test image is blue jacket and dark blue trousers, person the same with original image, the other test image is black jacket and black trousers. Matched results, the same person has nearest distance of similarity distance, mean is similarity.

| | | |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| Original image | | Test image #2 |
|  | Test image #1 |  |
| |  | |
| Bhattacharyya distance | 0.3317 | 0.78 |

Fig. 7. Matched results of color distribution map, has small score mean more similarity.

2.2.2. System parameters setup of KALMAN filter

Use Kalman filter to track image, it using constant prediction and revision update to estimate correct in the next frame of motion states, therefore can to reduce the search region of foreground, and the tracking speed, accuracy, stability, and multiple object tracking had to promote. Match tracker and detection, and to compute similarities of feature then store in cost matrix, if has greatest similarities, the cost is smallest, therefore used hungarian algorithm to match tracker and detection. The hungarian algorithm process as follows:

- 1) The detection have not been match is a new foreground object, will in accordance with detection information initialing tracker, and add the object data to the database.
- 2) The tracker have not been match as failed link with foreground object, cause of foreground object leaving screen or to hide, at this time will using prediction position to take as foreground object position, then prediction next time states, in this paper, when continuous occurred with six times, the prediction reliable will decreased, system will determined the tracker already lost the foreground object, and to kill this tracker.
- 3) If the foreground object and tracker have match, used foreground object position to prediction next time states, and update the contents of database.

2.2.3. Multi-cameras tracking

We used multi-cameras to extend surveillance region, when foreground object moving position at present camera to another camera, in order to make another camera control continuously, need to exchange information of two camera. In this paper, used overlapping field-of-view features and field-of-view line (*FOV* line) of two cameras, to find out the corresponding relation position point near *FOV* line, and can through the corresponding point of two cameras, to match foreground object of cross over the cameras. When foreground object moving position at present camera to another camera, the foreground object can match directly in overlap region, and newly-increased tracker will be offered the original number, continues to use the information of that number in the object database. Two scene overlap area will be split up into independent area by *FOV* line, and the scene content of the sideline extremity is controlled by different cameras, as shown in Fig. 12.

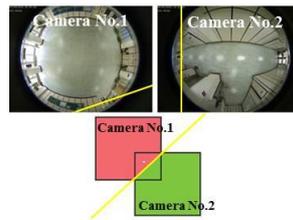


Fig. 8. Field-of-view diagram of two cameras.

Corresponding point to establish steps:

Step 1: in offline stage, capture foreground objects moving video in overlapping screen.

Step 2: detecting foreground object, selected bounding box of foreground object, and record center coordinate position of the bounding box.

Step 3: use two cameras obtaining the center value, and establish coordinates corresponding table of two cameras screen.

Partial of the contents of the corresponding table, as shown in Table 2.

Table 2. The Length Variation Results of Different Position in the Frame

| Camera No.1 | Camera No.2 |
|-------------|-------------|
| (770, 800) | (180, 470) |
| (780, 803) | (179, 460) |
| (800, 792) | (185, 450) |
| (820, 785) | (183, 420) |
| (840, 785) | (179, 380) |

When the foreground object across border of field-of-view and to communicate information with near camera, converting the coordinate position to near camera used the corresponding table, near camera to match the coordinates position of the coordinate detected of the foreground object center and coordinate was the original camera after conversion, the number value was given to the foreground object.

3. Experimental Results

In this section, we will show experimental results of multi-camera video surveillance systems, including foreground detection and foreground tracking results. System uses capture images device is D-link DCS-6010L fisheye cameras. Experimental videos size is 1200×900, and using two fisheye cameras to capture images. The two cameras were installed in the lobby and the passage leading to the laboratory, and cameras can cover the wide area. Camera in the lobby, can monitor area are 11 meters length × 9 meters width, camera in the passage, can monitor area are 16 meters length × 11 meters width, configuration diagram of experimental environment, as shown in Fig. 13 actual scene of experimental environment, as shown in Fig. 14.

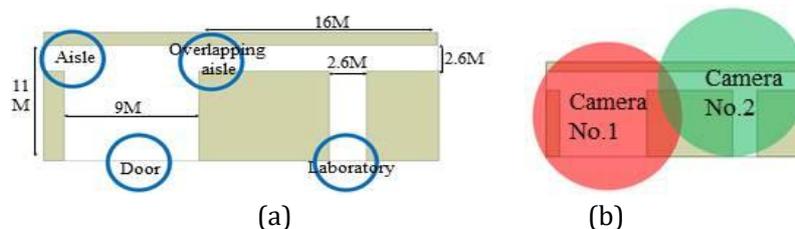


Fig. 9. Experimental environment diagram. (a) Configuration diagram, blue circle is the entry and exit space. (b) Monitor range of two fisheye cameras.



Fig. 10. Actual scene of experimental environment. (a) Door. (b) Aisle. (c) Overlapping aisle. (d) Laboratory.

3.1. Foreground Detection Experimental Results

Using the first fifteen input image do background training, non-background objects appear frequency in background training has decided the background mode is good or bad. If want to reduce the effect by foreground, training time can be extended or using background update to get better results. Two fisheye cameras detected results, as shown in Fig. 15 and Fig. 16.

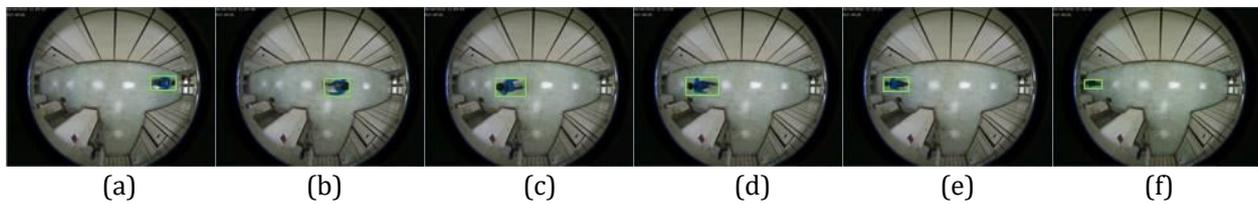


Fig. 11. Continuous images of camera No.1 detected results.

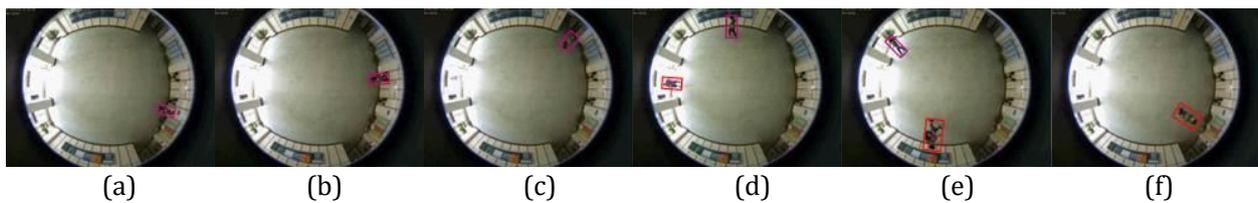


Fig. 12. Continuous images of camera No.2 detected results.

3.2. Foreground Tracking Experimental Results

When the image occur masking, use tracker predicted position to determine position of the foreground object. Every camera has its own tracker to trace multiple target trajectories even if the moving objects are partial and complete occluded. As in Fig. 17, the screen bottom, the person walks to the back of pillar (water blue trajectory), person disappeared for some time then appeared again, this is complete occluded, according tracker trust parameters we set, when the tracker was not matched to the foreground, it can still own predicted position of foreground object. As in Fig. 18, the screen top, two persons walking towards each other and interlaced, distance from the screen center close person (No. 2) covered the distance from the screen center far person (No. 0), when two persons approached system only detected one foreground object (red frame in screen), by tracker predicted position, when persons apart, use our features matching method can maintain the number of two persons.

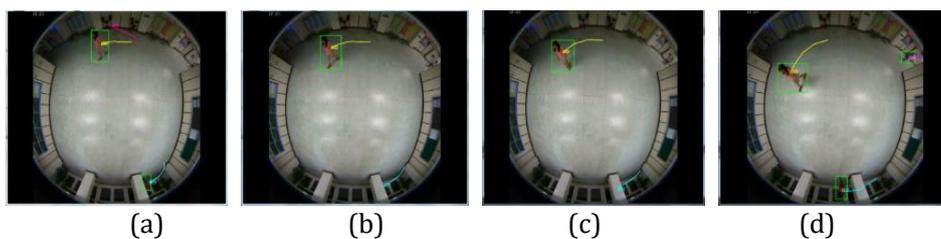


Fig. 13. Tracking results of foreground object complete occluded.

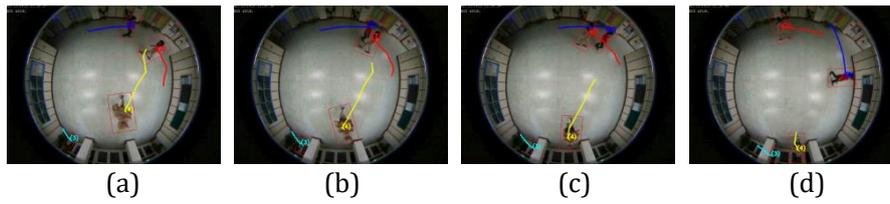


Fig. 14. Tracking results of foreground object partial occluded.

Foreground object (No. 0) moving results of different cameras, after information communication of multi-cameras can keep track and maintaining the same number, as shown in Fig. 19.

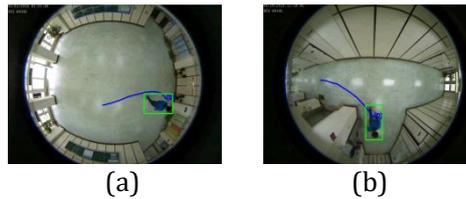


Fig. 15. Joint tracking results of multi-cameras. (a) Camera No.1. (b) Camera No.2.

The performance of detection method can be addressed by estimating the following parameters: (i) True Positive (TP) rate, (ii) False Negative (FN) rate, (iii) False Positive (FP) rate, and (iv) True Negative (TN) rate. Then can define the accuracy, sensitivity, and specificity. The experimental results of system assessment four video, each video length from 140 frames to 250 frames, an average of two persons appear in the screen. Effectiveness evaluation results of foreground detection, as shown in Table 3, the average sensitivity is 96.7 percent and the average false positive rate is 0.45 percent.

Table 3. Effectiveness Evaluation Results of Foreground Detection

| Video No. | TP+FN | TN+FP | TP | FP | False positive rate | Sensitivity |
|-----------|-------|-------|-----|----|---------------------|-------------|
| Video 1 | 130 | 100 | 130 | 1 | 0.1% | 100% |
| Video 2 | 138 | 30 | 120 | 1 | 1% | 86.9% |
| Video 3 | 145 | 87 | 145 | 0 | 0% | 100% |
| Video 4 | 155 | 43 | 155 | 3 | 0.7% | 100% |
| Average | | | | | 0.45% | 96.7% |

Effectiveness evaluation results of foreground detection with Kalman filtering for pedestrian motion prediction, as shown in Table 4, the average sensitivity rises to 98.55 percent, and obtained the steady position and smooth trajectories of foreground object.

Table 4. Effectiveness Evaluation Results of Foreground Detection with Kalman Filter

| Video No. | TP+FN | TN+FP | TP | FP | Sensitivity |
|-----------|-------|-------|-----|----|-------------|
| Video 1 | 130 | 100 | 130 | 0 | 100% |
| Video 2 | 138 | 30 | 130 | 0 | 94.2% |
| Video 3 | 145 | 87 | 145 | 0 | 100% |
| Video 4 | 155 | 43 | 155 | 1 | 100% |
| Average | | | | | 98.55% |

Effectiveness evaluation results of proposed algorithm, as shown in Table 5, average cost time of foreground detection is 98 ms, the foreground tracking cost time will be different with object quantity of the foreground, the longest cost time is 6 ms. The execute rate of more than 9 fps in videos size 1200 × 900.

Table 5. Effectiveness Evaluation Results of Proposed Algorithm Execute Rate

| Video No. | Total time | Foreground detection time | Foreground tracking time |
|-----------|------------|---------------------------|--------------------------|
| Video 1 | 100.6 ms | 98 ms | 2.6 ms |
| Video 2 | 102 ms | 98 ms | 4 ms |
| Video 3 | 102 ms | 98 ms | 4 ms |
| Video 4 | 104 ms | 98 ms | 6 ms |

4. Conclusion

In this paper, we have proposed foreground detection and tracking system to adapt to brightness changes, the background subtraction method used in foreground detection can effectively detect foreground objects in the screen, and the light processing approach is simple and intuitive, overall system does not require to do camera transform processing, and used block size of foreground representative person to remove noise and not interested part. Using KALMAN filter to track and recognition persons, use geometric and shape features, can provide assistance in poor effect of foreground detection, from the experimental data validate our method can ensure the following benefits: (i) when the screen appears two person at the same time has average sensitivity rises to 96.7 percent, (ii) to trace multiple target trajectories even if the moving objects are partial and complete occluded, and (iii) the moving object in camera views actually belonging to the same object, and keep consistent labels on the object. System only used two fisheye cameras will be able to implement wide area detection and tracking, we proposed system process can be applied a large number of cameras to do surveillance, as analyzing multiple target trajectories, and can be added PTZ cameras to increase the detection effectiveness.

References

- [1] Mubashir, M., Shao, L., & Seed, L. (2013). A survey on fall detection: Principles and approaches. *Neurocomputing*, 100, 144-152.
- [2] Hu, W., Tan, T., Wang, L., & Maybank, S. (2004). A survey on visual surveillance of object motion and behaviors. *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 34(3), 334-352.
- [3] Masoud, O., & Papanikolopoulos, P. (2001). A novel method for tracking and counting pedestrians in real-time using a single camera. *IEEE Trans. on Vehicular Technology*, 50(5), 1267-1278.
- [4] Paragios, N., & Tziritas, C. (1996). Detection and location of moving objects using deterministic relaxation algorithms. *Proceedings of the 13th Int. Conference on Pattern Recognition* (pp. 201-205).
- [5] Fleet, D. J., & Jepson, A. D. (1990). Computation of component image velocity from local phase information. *Int. Journal of Computer Vision*, 5(1), 77-104.
- [6] Wang, M. L., Huang, C. C., & Lin, H. Y. (2006). An intelligent surveillance system based on an omnidirectional vision sensor. *Proceedings of IEEE Conf. on Cybernetics and Intelligent Systems* (pp. 1-6). Thailand: Bangkok.
- [7] Lee, D. S., (2005). Effective gaussian mixture learning for video background subtraction. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(5), 827-832.
- [8] Oliver, N. M., Rosario, B., & Pentland, A. P. (2000). A bayesian computer vision system for modeling human interactions. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8), 831-843.
- [9] Boulton, T. E., Micheals, R., Gao, X., Lewis, P., Power, C., Yin, W., & Erkan, A. (1999). Frame-rate omnidirectional surveillance and tracking of camouflaged and occluded targets. *Proceedings of the 2nd IEEE Workshop on Visual Surveillance*, Fort Collins, Colorado (pp. 48-55).

- [10] Kim, Z., (2008). Real time object tracking based on dynamic feature grouping with background subtraction. *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition* (pp. 1-8).
- [11] Unno, H., Ojima, K., Hayashibe, K., & Saji, H. (June 13-15, 2007). Vehicle motion tracking using symmetry of vehicle and background subtraction. *Proceedings of IEEE Workshop on Vehicles* (pp. 1127-1131). Istanbul, Turkey.
- [12] Yao, J., & Odobez, J. M. (June 17-22, 2007). Multi-layer background subtraction based on color and texture. *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition* (pp. 1-8.). Minneapolis, Minnesota.
- [13] Horprasert, T., Harwood, D., & Davis, L. (January 8-11, 2000). A robust background subtraction and shadow detection. *Proceedings of Asian Conf. on Computer Vision* (pp. 983-9880). Taipei, Taiwan.
- [14] Javed, O., Shafique, K., & Shah, M. (December 5-6, 2003). A hierarchical approach to robust background subtraction using color and gradient information. *Proceedings of Workshop on Motion and Video Computing* (pp. 22-27). Orlando, Florida.
- [15] Stauffer, C., & Grimson, W. E. L. (Jun. 23-25, 1999). Adaptive background mixture models for real-time tracking. *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition* (pp. 246-252). Fort Collins, CO.
- [16] Kim, K., Chalidabhongse, T. H., Harwood, D., & Davis, L. (2005). Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11(3), 172-185.
- [17] Wang, X. (2013). Intelligent multi-camera video surveillance: A review. *Pattern Recognition Letters*, 34(1), 3-19.
- [18] Liem, M., & Gavrilu, D. M. (September 7-10, 2009). Multi-person tracking with overlapping cameras in complex, dynamic environments. *Proceedings of British Machine Vision Conf.* London, UK.
- [19] Carroll, R., Agrawal, M., & Agarwala, A. (2009). Optimizing content-preserving projections for wide-angle images. *ACM Trans. on Graphics-TOG*, 28(3), 1-9.
- [20] Gennery, D. B., (2006). Generalized camera calibration including fish-eye lenses. *Int. Journal of Computer Vision*, 68(3), 239-266.
- [21] Hughes, C., Glavin, M., & Jones, E. (2011). Simple fish-eye calibration method with accuracy evaluation. *Electronic Letters on Computer Vision and Image Analysis*, 10(1), 54-62.
- [22] Mundhenk, T. N., Rivett, M. J., Liao, X., & Hall, E. L. (Nov.7-8, 2000). Techniques for fisheye lens calibration using a minimal number of measurements. *Proceedings of Conf. on Intelligent Robots and Computer Vision XXI: Algorithms, Techniques, and Active Vision* (pp. 181-190). Boston, MA.
- [23] Eshel, R., & Moses, Y. (June 23-28, 2008). Homography based multiple camera detection and tracking of people in a dense crowd. *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition* (pp. 1-8). Anchorage, AK, ,
- [24] Shao, J., Dong, N., Liu, F., & Li, Z. (2011). A close-loop tracking approach for multi-view pedestrian tracking. *Journal of Computational Information Systems*, 7(2), 539-547.
- [25] Khan, S., & Shah, M. (2003). Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(10), 1355-1360.
- [26] Javed, O., Rasheed, Z., Shafique, K., & Shah, M. (October 13-16, 2003). Tracking across multiple cameras with disjoint views. *Proceedings of IEEE Int. Conference on Computer Vision* (pp. 952-957). France: Nice.
- [27] Song, B., & Roy-Chowdhury, A. K. (2008). Robust tracking in a camera network: A multi-objective optimization framework. *IEEE Journal on Select. Top. in Signal Process.*, 2(4), 582-596.
- [28] Javed, O., Shafique, Rasheed, K., Z., & Shah, M. (2008). Modeling inter-camera space-time and appearance relationships for tracking across non-overlapping views. *Computer Vision and Image Understanding*, 109(2), 146-162.

- [29] Rahimi, A., Dunagan, B., & Darrell, T. (June 27-July 2, 2004). Simultaneous calibration and tracking with a network of non-overlapping sensors. *Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition* (pp. 1187-1194). Washington DC.
- [30] Chen, X., An, L., & Bhanu, B. (2015). Multitarget tracking in nonoverlapping cameras using a reference set. *IEEE Sensors Journal*, 15(5), 2692-2704.



Dinchang Tseng received his Ph.D. degree in computer science and information engineering from National Chiao-Tung University, Hsinchu, Taiwan, in June 1988. He has been a professor in the Department of Computer Science and Information Engineering at National Central University, Jhongli, Taiwan since 1996. He is a member of the IEEE. His current research interests include computer vision, image processing, and virtual reality; especially in the topics of computer vision techniques for advanced driver assistance systems and human computer interaction.



Chienhung Chen received his M.S. degree in electrical engineering from National Chung Cheng Institute of Technology, Taoyuan, Taiwan, in 2003. He is currently pursuing the Ph.D. degree in the Department of Computer Science and Information Engineering at National Central University, Jhongli, Taiwan. His research interests include computer vision, image processing, and real-time data processing.



Chiahui Tseng received his B.S. degree in the Department of Computer Science and Information Engineering from Yuan Ze University, Jhonhli, Taiwan, in 2013, and M.S. degree in the Institute of Computer Science and Information Engineering from National Central University, Jhongli, Taiwan, in 2015.