International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol. 8, No. 2 (2015), pp. 361-374
http://dx.doi.org/10.14257/ijsip.2015.8.2.34

Consistent Labeling Approach for a PTZ Camera Based on Template Cache and Least Recently Used Replacement Strategy

Hsien-Chou Liao[1], Chih-Hung Yang[1], Hong-Wei Huang[1] and Jungpil Shin[2]

[1]*Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taiwan (R.O.C.)*
[2]*Department of Computer Software, Graduate School of Computer Science and Engineering, the University of Aizu, Japan*
hcliao@cyut.edu.tw, jpshin@u-aizu.ac.jp

Abstract

Object tracking is an important function of video surveillance system. For the same object in a multi-camera environment, how to assign the same label to this object is so-called consistent labeling problem. Many consistent labeling approaches proposed by previous studies mainly based on the environment with multiple fixed cameras. In this study, a consistent labeling approach is proposed for a PTZ (Pan-Tilt-Zoom) camera. The same object is assigned the same label while the object in the FOV (Field-of-View) of a PTZ camera without influencing by the pan/tilt rotation. In order to achieve the above goal, the proposed approach using several methods, such as temporal differencing, template matching, mean-shift tracking, Kalman filter, and so on. A template cache is also designed for preserving the templates of an object with various angles and a least recently used (LRU) replacement mechanism is used to update the cache. The experimental results show that accuracy of the proposed approach for the consistent labeling of a PTZ camera can reach about 83 percentage.

Keywords: *object tracking, active camera, histogram matching, video surveillance system*

1. Introduction

In a multi-camera environment, a moving object usually passes through the field-of-view (FOV) of different cameras. So, the consistent labeling function is to give the same label to the same object all the time. The consistent labeling enables the tracking of the same object in the multi-camera environment. It is also useful to record and analyze the trajectory of an object precisely. For example, the video of the same object can be retrieved quickly based on the results of the consistent labeling function.

For the previous studies related to consistent labeling, they assumed the environment covered by multiple fixed cameras. The spatial relationships among cameras are either pre-defined or established automatically. Therefore, when a moving object leaves the FOV of one camera, the system can predict the most likely camera that the object may enter. Then, the template or spatial information of the object is matched with the foreground object entering the FOV of another camera to make sure the object the same. Color histogram matching, feature matching, or trajectory matching methods are popularly used to determine whether the same label is given to the object.

In this paper, the consistent labeling function is fulfilled on a pan-tilt-zoom (PTZ) camera. Assume there are many objects are moving in an open area and the area is monitored by using a PTZ camera, *e.g.*, the playground of a kindergarten or the football

ISSN: 2005-4254 IJSIP

players in the field. When a surveillant controls the PTZ camera to monitor a part of the area, the same object should be given the same label when the object appears in the FOV of the PTZ camera. Such a scenario is shown in Figure 1. In the Figure, the same object is still given the same label without influenced by the rotation of the PTZ camera.

Figure 1. The Scenario of Consistent Labeling Function based on a PTZ Camera

2. Related Works

The previous studies related to the consistent labeling issue mainly focus on the environment covered by multiple fixed camera. For example, L. Zhu, J. N. Hwang, and H. Y. Cheng proposed a consistent labeling method on 2009 [1]. They assumed that the FOVs of cameras are either overlapped or separated. In order to achieve the goal of consistent labeling, the FOV line between two cameras is established automatically if the FOVs of two cameras are overlapped. Firstly, the feature points, called landmark points, of the images from two cameras are extracted. SIFT (Scale-Invariant Feature Transform) method is used to obtain the connection between two set of landmark points. Then, RANSAC (Random Sample Consensus) algorithm is used to perform the alignment of two images. The line to align two images is called FOV line. Oppositely, if the FOVs of two cameras are separated, the FOV line is established manually. When all the FOV lines among cameras are obtained, a color histogram matching method is used to balance the brightness of images from two cameras. Then, the location of the object's foot and the similarity of the color histogram are used to make sure the objects in the images from two cameras are the same one. The same object is marked by the same label. An example is shown in Figure 2. The same object can be still marked by the same label under the cameras without overlapping FOVs.

Figure 2. The Consistent Labeling Method Proposed by L. Zhu, *et al*.

Besides, S. Calderara, *et al*., also proposed a consistent labeling method with high accuracy [2-3]. The method was called HECOL (Homography and Epipolar-based Consistent Labeling). Assume the FOVs of two cameras are overlapped, the homography is established firstly. Bayesian statistical information is also used to solve the labeling of single or group objects. In the training phase, HECOL method is used to establish the ground-plane homography and epipolar geometry. The results are shown in Figure 3. The experimental results also show that the labeling accuracy of HECOL method is 98.77 percentage. However, the method is only suitable when the FOVs of two cameras are overlapped. Besides, V. Badrinarayanan, *et al.*, proposed a label propagation method for a video [4]. When an object appears in a video for the first time, it is assigned a label. Once the object appears in the video again, the same label is assigned to the object. The method is based on the motion information of the object. The method can also be used for video segmentation or scene analysis.

Figure 3. The Consistent Labeling Method Proposed by S. Calderara, *et al*.

In addition, G. Lian, *et al*., proposed a spatial-temporal probabilistic model to solve the consistent labeling problem when the FOVs of cameras are separate, *i.e.*, non-overlapped [5]. The model is used to predict the next possible camera when an object leaves the FOV of one camera. A method called CMCSHR (Competitive Major Color Spectrum Histogram Representation) is proposed to ensure the object in the FOV of the next possible camera is the same one.

According to the above related works, these methods are mainly designed for the multiple fixed cameras. There is still no method proposed to solve the consistent labeling problem for the PTZ camera. This is the main purpose of this study.

3. Method

When a PTZ camera is used to monitor people moving around a square, a consistent labeling approach is designed to mark the same person with the same label when the object enters the FOV of the PTZ camera. The process of the proposed approach is shown in Figure 4. Firstly, foreground objects are separated from the background image. Then, every object is matched with the templates preserved previously. If an object is matched successfully, it is marked with the same label. Then, a mean-shift method is used to keep tracking the object until it leaves the FOV of the camera. The foreground object detection, consistent labeling, and object tracking is presented in the following subsections.

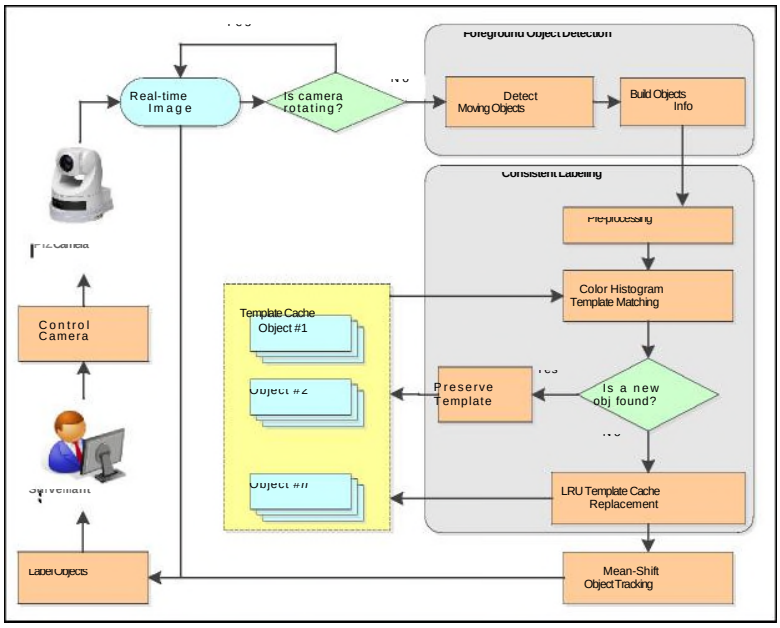


Figure 4. The Process of the Proposed Consistent Labeling Approach

3.1. Foreground Object Detection

Two popular methods are used to detect foreground objects. One is background subtraction, the other is temporal differencing. For the PTZ camera, the rotation function causes the background model is difficult to be established. Therefore, a temporal differencing method is used here. The equation of temporal differencing is shown in formula (1).

$$D(x, y) = i+1(,) - i(,) \quad (1)$$

where P_i and P_{i+1} denotes the gray-scale images of two successive frames. $P_i(x, y)$ is the value of a pixel at the location (x, y). D is denoted as the result of temporal differencing.

An example of foreground object detection is shown in Figure 5. Figure 5(a) and 5(b) are the current and previous frames, respectively. The corresponding gray-scale images are shown in Figure 5(c) and 5(d). Figure 5(e) is the result of temporal differencing. The edge of the foreground object is obvious unconnected. In order to solve such problem, the Otsu algorithm with an adaptive threshold [6] is used on the current and previous frames. The results are shown in Figure 5(f) and 5(g), respectively. Then, the above images are performed the same differencing operation and the result is shown in Figure

5(h). By observing the results shown in Figure 5(e) and 5(h), the edges of the foreground objects in the latter result are better than the edges of the previous one. Then, two results shown in Figure 5(e) and 5(h) are merged by using formula (2), where $I_1(x, y)$ and $I_2(x, y)$ are the pixel values of an image on the coordinates (x, y). The merged result is shown in Figure 5(i).

$$M(,) = Max(_1(,), _2(,)) \quad (2)$$

Then, some fragments of the above result are connected by using dilation and erosion operations as shown in Figure 5(j). Finally, a hole filling operation [7] is performed to

generate the expected foreground objects as shown in Figure 5(k). Therefore, the foreground objects in a frame are detected according to the above process.

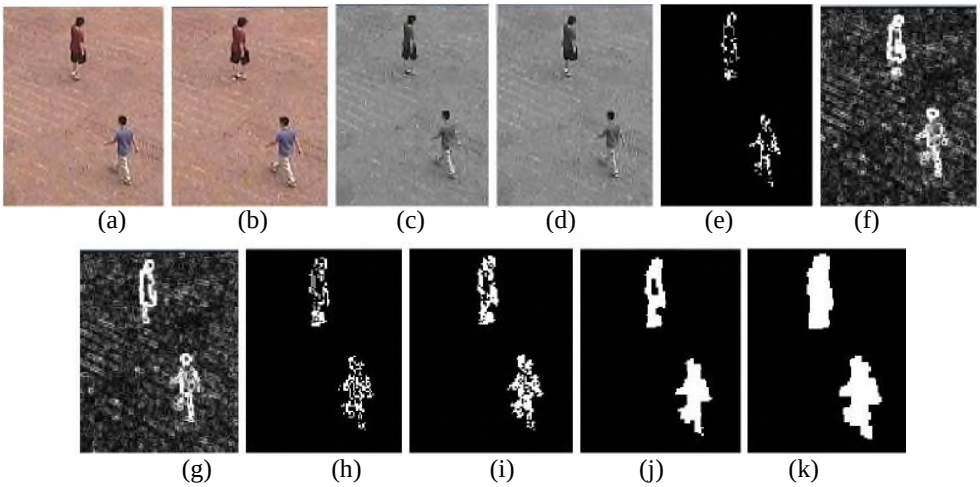


Figure 5. An Example of Foreground Object Detection (a) Current Frame, (b) Previous Frame, (c) and (d) the Gray Image of (a) and (b), (e) Temporal Differencing of (c) and (d), (f) and (g) the Binarization of (c) and (d), (h) Temporal Differencing of (f) and (g), (i) the Addition of (e) and (h), (j) the Dilation, (k) Hole Filling

3.2. Consistent Labeling

When the foreground objects are detected in the previous process, the key consistent labeling process is performed. The following factors are considered on designing the consistent labeling approach for a PTZ camera:

- A PTZ camera can be rotated to view any position in the monitoring area. During the period of rotation, the real-time image is changed dramatically. So, the foreground object detection is paused temporarily during such period.
- After the rotation of a PTZ camera is done, a foreground object is then re-marked either its original label or a new label.

According to the above factors, the following steps are designed to fulfill the consistent labeling of a PTZ camera:

(1) Intersection Checking of Objects in Two Successive Frames: Foreground objects detected in the previous frame are performed intersection checking with those objects in the next frame. The checking is based on the rectangle area of an object. There are three possible conditions: one-to-one (1-1), one-to-many (1-M), and many-to-one (M-1). If 1-1 is found, two objects in two successive frames are deemed as the same one. Oppositely, 1-M and M-1 mean that the split and join conditions occur, respectively. Aspect ratio checking and template matching presented later are used to handle such conditions.

(2) Aspect Ratio Checking of an Object: The join or split of foreground objects frequently occurs. Such conditions can be found by checking the aspect ratio of an object in the successive frames. That is, the aspect ratio of an object increase or decrease significantly means the join or split condition occurs, respectively. Here, an object is performed the following consistent labeling steps only when its aspect ratio is stable, i.e., the ratio is less than a pre-defined threshold . Assume the width and height of an object in the previous

frame and current frame is denoted as W_{i-1}, H_{i-1}, W_i and H_i, respectively. The change of the aspect ratio in width and height is computed by using formulas (3) and (4):

$$\Delta = \frac{|-1-|}{\max(-1,)} \quad (3)$$

$$\Delta = \frac{|-1-|}{\max(-1,)} \quad (4)$$

So, The following steps are performed when ΔW or Δ is smaller than the threshold , which means that the join or split conditions don't occur.

(3) Template Collection: When a PTZ camera is used to monitor an object moving in an open area, the same object may appear in the image with various angles. This is quite different with the fixed camera. Therefore, a template cache (TC) is designed in this step to preserve the possible templates of an object. Assume the size of TC is *N*, there are *N* templates stored in the TC for every labeled object. An example of TC is shown in Figure 6.

Figure 6. An Example of the Template Cache with size *N*=8 (Left Column are Labeled Objects, Right Column are Corresponding Templates in TC)

An object can be performed the tracking and consisting labeling function only when its size of templates in TC is equal or larger than □ (□□*N*). It is because the labeling function is easily failed if the size of templates is not enough. Besides, when the cache is full and a new template is found, a replacement strategy must be designed. Here, the basic least recently used (LRU) replacement strategy is used. The strategy will be represented in the later step.

(4) Color Histogram Matching: When an object is tracking, the template matching is mainly based on the appearance model, *i.e.*, the clothes of an object. Therefore, a color histogram matching method is used here. For two templates to be matched, the color

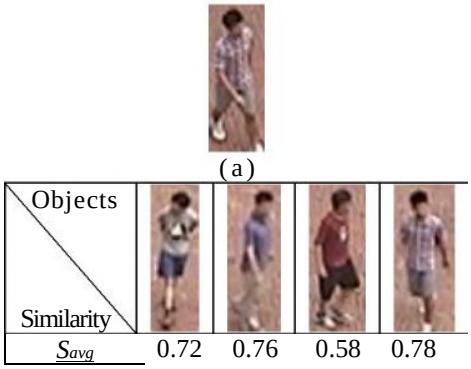
histograms of R, G, and B channels are accumulated separately for 16 bins. Then, the corresponding bins of two images are computed to estimate the similarity using formulas (5) and (6) as shown below.

$$= / \quad (5)$$

$$= \sum_{=1}^{16} \min(\quad ,) \times \quad (6)$$

where k represents R, G, or B channels. represents the number of pixels in the i-th bin of the channel k. $_{Proral}$ is the total number of pixels in an image. So, is the ratio of i-th bin of the channel k in an image. and represents the pixels count of the template or object in the i-th bin of the channel k. S^k denotes the overall similarity of the channel k.

Then, the final similarity denoted $_{Savg}$ equals the average of S^R, S^G, and S^B. Its value is within 0 and 1. Four examples of the above color histogram matching are shown in Figure 7. The template is shown in Figure 7(a). The matching results of four objects are marked on the line to indicate the similarity between the template and the object.



(a)

Figure 7. Four Examples of the Color Histogram Matching (a) Template (b) Results

By observing the above examples in Figure 7, the similarities by matching the whole template and object image causes the values are too close. The real object is the left-most image. However, its similarity is close to the first and the second objects on the left-hand side. In order to solve such problem, the obvious difference in appearance can be found on the T-shirt/pants. Therefore, two rectangles for the upper/lower parts of the image are defined separately and the color histogram matching is also performed separately. The settings of two rectangles are mainly from the empirical study as listed in Table 1. Four examples of two rectangles in an object image are shown in Figure 8.

Table 1. The Settings of Rectangles for Body and Foot

	Left	Top	Width	Height
Body	0.3	0.2	0.5	0.35
Foot	0.3	0.6	0.5	0.3

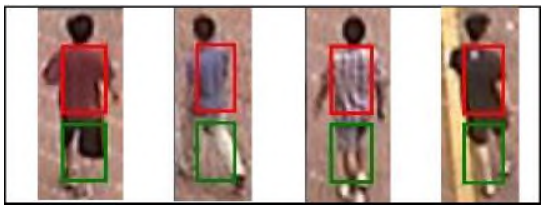
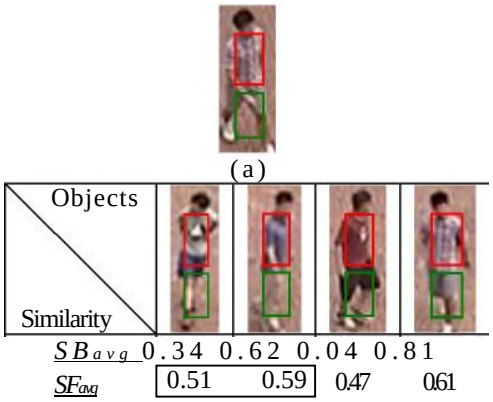


Figure 8. Four Examples of the Rectangles for Body and Foot

Then, the body and foot parts are performed the same color histogram matching separately. Assume the similarity of the body and foot parts are denoted as SB_{avg} and SF_{avg}. For the same examples shown in Figure 7, the results of SB_{avg} and SF_{avg} are shown in Figure 9. By observing the results, the similarities of the left-most object, 0.81 and 0.61, are quite different from that of the first and second objects on the left-hand side.



(a)

Figure 9. Four Examples of the Color Histogram Matching for Body and Foot (a) Template (b) Results

(5) Template Cache Replacement: After the similarities of the body/foot parts are estimated, two thresholds, T_b and T_f, are defined for the SB_{avg} and SF_{avg} separately. A new detected foreground object is deemed as an existing labeled object when both of two SB_{avg} and SF_{avg} are larger than T_b and T_f, respectively. When the current number of templates in TC doesn't reach the maximum number N, the matched object image is added into the TC. When the TC is full, a LRU replacement strategy is used here. A time stamp is attached to every template in TC. The least recently used template in TC will be replaced by the matched object image.

On the other hand, when an object image cannot be matched with all the templates of all the labeled objects' TC, and the maximum similarity with all templates is smaller than another threshold □. It means that the object is quite different with all the templates of all the labeled objects. The object is deemed as a new one and is assigned a new label. The consistent labeling and tracking function is performed on this new object later.

For the join and split conditions detected by the aspect ratio checking step mentioned in the first intersection checking step, the template matching will be still performed. It the join object or the multiple split objects cannot be matched with the templates of all the current labeled objects, the object will be deemed as a new one and assigned a new label just like in the previous paragraph.

3.3. Labeled Object Tracking

When a foreground object is either matched with a labeled object or assigned a new label, the object tracking is then performed until the object leaves the FOV of the camera. The tracking method is based on the mean-shift algorithm. It is used to estimate the most similar location of a template in a region [8]. Assume *S* denotes {}=1,2..., that are a sample of independent random variables from some distribution with an unknown density *f*(*x*). The size of a template image is M × N pixels. The kernel density can be estimated by using formula (7).

$$() = 1 \times \sum \quad h \underline{(|-} \quad 2$$
$$\_\_\_\_\_ \times \quad h \,|\,) \qquad (7)$$
$$=1$$

where *h* is the bandwidth, and *h*□0. *h* $\underline{h}\,|\,)$ is the weighted mean on *x* coordinate.

Then, Bhattacharyya coefficient is applied to calculate the similarity between each candidate and the template. The Bhattacharyya coefficient is defined as follows:

$$() \equiv \sqrt{1}, \bar{} \; () \subseteq \sum \quad \sqrt{()} \qquad (8)$$
$$=1 \qquad (9)$$

where *q* is the template model, () is the candidate model, and *B* is the number of bins used to calculate the model.

When the location of a labeled object in a frame is estimated using the mean-shift tracking method, the locations in continuous frames may be unstable caused by the noise or brightness. Therefore, a Kalman filter is used to smooth the locations. A Kalman filter is defined by the following equations [9]:

$$\hat{} - = \; -1 + -1 -1 + - \qquad (10)$$
$$- = (- +)^{-1} \hat{} - + (- \qquad (11)$$
$$\hat{} = \; -) (-)- \qquad (12)$$
$$= \qquad (13)$$
$$= \qquad (14)$$

where $\hat{} -$ is a priori state estimate at step *k* given knowledge of the process prior to step *k*, $\hat{}$ is a posteriori state estimate at step *k* given measurement. z_k The matrix *A* relates the state at the previous time step *k*-1 to the state at the current step *k*, and matrix *B* relates the optional control input u_k to the state *x*. P_k is estimate error covariance, K_k is Kalman gain, *Q* is process noise covariance, and *R* is the measurement noise covariance. Formulas (9) and (10) are time update equations. Formulas (12)-(14) are measurement update equations. An example by filtering *x* coordinates with/without Kalman filter is shown in Figure 10. It is obvious that the curve of the *x* coordinates processed by the Kalman filter is smoother than the curve of the original *x* coordinates.

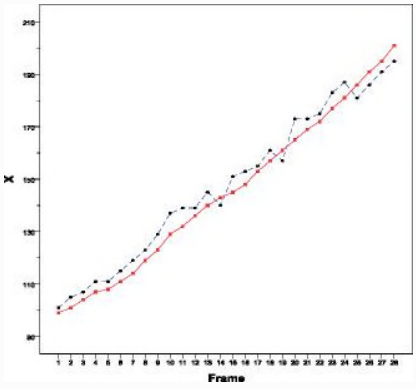


Figure 10. An Example of Kalman Filter (Q=0.001, R=0.0012)

4. Experimental Study

In order to evaluate the performance of the proposed consistent labeling approach of a PTZ camera, a set of persons walks randomly in the experimental site. However, such a way is impractical since it is time-consuming. If an error is found, it is also difficult to be removed because those persons cannot walk in the same way repeatedly. Therefore, a video is recorded for the experiment. A prototype was also implemented to perform the function of consistent labeling on the video. A PTZ camera is simulated by viewing only one-fourth of the video frame. The viewing windows can be adjusted to simulate the operation of a PTZ camera. Then, an experiment is designed and can be repeated to analyze the accuracy of the proposed consistent labeling approach.

4.1. Prototype

A prototype was implemented by using Visual C# 2010 and two popular libraries, AForge.NET [10] and EmguCV [11]. Its screen shot is shown in Figure 11.

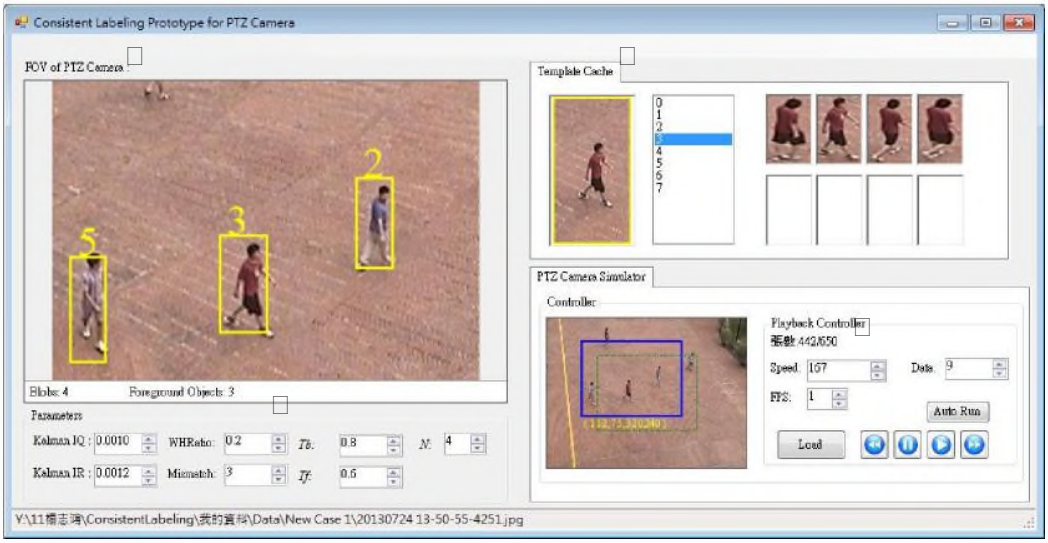


Figure 11. The Screen Shot of the Prototype

The top-left corner marked □ is the FOV of a simulated PTZ camera. All the labeled objects can be viewed in the top-right area marked □, including its templates in TC. The input video for the experiment is shown in the area marked □. The blue rectangle is the FOV of a simulated PTZ camera. A user can click on the image to change the center of the rectangle. The buttons and parameters for controlling the playback of the video are in the panel marked □. The related parameters of consistent labeling approach are listed in the area marked □, including Q and R for Kalman filter, a, □, T_b, T_1, and the cache size N.

The testing video is the random walk of five persons in a square. There are totally 650 frames with resolution 640□480 pixels as shown in Figure 12. During the walking of five persons, three unknown persons are passing through the five persons. The same video is performed 10 different control sequences to simulate the different operations of a PTZ camera. Most of the sequences are normal operations. Some operation sequences are quick and some are normal. Every control sequence is repeated eight times for the cache size N from one to eight. So, there are 80 tests totally.

A suitable setting of parameters was determined from the experiments, including a=0.2, □=3, T_b=0.8, and T_1=0.6.

Figure 12. Four Frames of the Testing Video

4.2. Results

There is still no related study of the consistent labeling approach using a PTZ camera. Therefore, the analysis of the experimental results focuses on the influence of the cache size N on the labeling accuracy. As mentioned previously, there are 10 different control sequences operated on the testing video. And, every sequence is operated for eight times for the cache size N from one to eight. In order to ensure one control sequence is the same for the different cache size. A sequence is recorded and then controlled automatically by the prototype in the experiment. Therefore, the influence of the operator can be reduced to minimum. The screen shots of the experiment are shown in Figure 13:

Figure 13. Screen Shots of the Experiment

The 80 tests are recorded and then checked visually. Two results are calculated for a test: one the total number of foreground objects, denoted as *TObj*, the other is the number of objects with an error label, denoted as *LErr*. An object is labeled consistently when it meets the following criteria:

(1) After the FOV is adjusted, the object can still be marked with the same label.
(2) If the object is joined with other objects, the object can be marked with the same label after splitting has occurred.

On the other hand, an object is deemed as labeled error either if the label is wrong or no label is assigned to it. The accuracy of consisting labeling is computed using formula (15):

$$\text{Accuracy} = (1 - \text{rr}) \times 100\% \quad (15)$$

The average accuracy of the ten control sequences for *N* from one to eight is listed in Table 2. By observing the results in the table, the accuracy is the lowest when *N* equals to one. It is the same as our expectation since the insufficient templates did decrease the accuracy. Oppositely, the highest accuracy appears when *N* equals to four. Then, the accuracy is decreased as the increasing of the cache size *N*.

For the cache size *N* equals to four, the accuracy of ten tests is listed in

Table **3**. Among them, the accuracy of the fifth and tenth control sequences is low since the PTZ camera is controlled to be operated very frequently. Some objects are going to leave the FOV and the time is not enough to mark its label. On the other hand, the highest accuracy appears on the ninth testing sequence. All the accuracy of the ninth testing sequence for *N* is from one to eight is also listed in Table **4**. The highest accuracy appears on *N* equals to 2. In fact, the accuracy of *N*=3 or 4 is still good and over 90 percentage.

In addition to the analysis of the labeling accuracy, the processing time is also analyzed. For the processing of a single frame, the foreground object detection took 16.7 ms, the consistent labeling function took 21.2 ms, and the mean-shift tracking took 6.1 ms. The overall processing time is about 44 ms. It means that the proposed approach is able to achieve 22 FPS (frames-per-second).

Table 2. The Average Accuracy of Experiments for N from One to Eight

N	*Average Accuracy* (%)
1	68.6
2	77.8
3	81.0
4	83.2
5	82.4
6	83.0
7	82.4
8	80.6

Table 3. The Accuracy of Ten Testing Sequence for *N*=4

i	*TObj*	*LErr*	*Accuracy* (%)
1	25	4	84.0
2	26	3	88.4
3	26	5	80.7
4	23	3	86.9
5	31	7	77.4
6	27	4	85.1
7	29	2	93.1
8	24	2	93.5
9	31	2	93.5
10	26	9	79.0
Average	27.1	4.6	83.2

Table 4. The Accuracy of the Ninth Control Sequence for *N* from One to Eight

N	*TObj*	*LErr*	*Accuracy* (%)
1	31	6	80.6
2	31	2	93.5
3	31	3	90.3
4	31	3	90.3
5	31	5	83.8
6	31	4	87.0
7	31	4	87.0
8	31	5	83.8

5. Conclusion

In this paper, a consistent labeling approach is proposed for a PTZ camera. Unlike the previous approaches based on multiple fixed cameras, an object may appear in the image with various angles. Therefore, a template cache is designed and LRU (Least Recently Used) replacement strategy is also used to maintain the set of templates for every object. The experimental results show that the average accuracy is 83 percentages. The results of different cache size *N* also show that the cache size can influence the labeling accuracy. A large cache size results in a long matching time. A suitable cache size is from two to four. Two factors influencing the labeling accuracy are listed below:

(1) The tracking is sometimes failed since the template matching process cannot be finished before the switching of the FOV.
(2) The distance between the camera and the object is quite long. The object image is small and may influence the similarity of the template matching that is based on the apparent model.

In the future, the distance between the camera and the subjects will be shortened and the facial biometrics can be used to identify the subject to assign the correct label. In fact, the zoom capability of a PTZ camera can achieve such goal. It may also influence

the design of the consistent labeling approach. The above issue will be studied further to improve the labeling accuracy.

Acknowledgements

This work is partially supported by National Science Foundation under grant NSC 101-2221-E-324-040.

References

[1] J. P. Lewis, "Fast template matching", Vision Interface, (**1995**) May 15-19, pp. 120-123, Quebec City, Canada.
[2] S. Calderara, R. Cucchiara and A. Prati, "Bayesian-competitive consistent labeling for people surveillance", IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 30, no. 2, (**2008**), pp. 354-360.
[3] S. Calderara, R. Cucchiara and A. Prati, "Group detection at camera handoff for collecting people appearance in multi-camera systems", Proceedings of the IEEE International Conference on Video and Signal Based Surveillance, (**2006**) November 22-26, pp. 36-41, Sydney, Australia.
[4] V. Badrinarayanan, F. Galasso and R. Cipolla, "Label propagation in video sequences", Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition, (**2010**) June 13-18, pp. 3265-3272, San Francisco, USA.
[5] G. Lian, J. H. Lai and W. S. Zheng, "Spatial–temporal consistent labeling of tracked pedestrians across non-overlapping camera views", Pattern Recognition, vol. 44, issue 5, (**2011**), pp. 1121-1136.
[6] N. Otsu, "A threshold selection method from gray-level histograms", IEEE Transactions on Systems, Man and Cybernetics, vol. 9, no. 1, (**1979**), pp. 62-66.
[7] C. C. Hsieh and S. S. Hsu, "A simple and fast surveillance system for human tracking and behavior analysis", Third International IEEE Conference on Signal-Image Technologies and Internet-Based System, (**2007**) December 16-19, pp. 16-18, Shanghai, China.
[8] Y. Kang, W. Xie and B. Hu, "A scale adaptive mean-shift tracking algorithm for robot vision", Advances in Mechanical Engineering, vol. 2013 (**2013**), pp. 12.
[9] G. Welch and G. Bishop, "An introduction to the Kalman filter", Univ. North Carolina, Chapel Hill, Tech. Rep. TR95-041, (**2004**), pp. 16.
[10] AForge.NET Framework: http://code.google.com/p/aforge/.
[11] EmguCV: http://sourceforge.net/projects/emgucv/.

