Object Tracking and Local Appearance Capturing in a Remote Scene Video Surveillance System with Two Cameras

Wenming Yang[1,2], Fei Zhou[1,2], and Qingmin Liao[1,2]

[1] Visual Information Processing Lab., Graduate School at Shenzhen,
Tsinghua University, Shenzhen, China
[2] Tsinghua-PolyU Biometric Joint Lab., Shenzhen, China
yangelwm@163.com, flying.zhou@163.com, liaoqm@sz.tsinghua.edu.cn

Abstract. Local appearance of object is of importance to content analysis, object recognition and forensic authentication. However, existing video surveillance systems are almost incapable of capturing local appearance of object in a remote scene. We present a video surveillance system in dealing with object tracking and local appearance capturing in a remote scene, which consists of one pan&tilt and two cameras with different focuses. One camera has short focus lens for object tracking while the other has long ones for local appearance capturing. Video object can be located via just one manual selection or motion detection, which is switched into a modified kernel-based tracking algorithm absorbing both color value and gradient distribution. Meanwhile, local appearance of object such as face is captured via long focus camera. Both simulated and real-time experiments of the proposed system have achieved promising results.

Keywords: Remote scene, video surveillance, object tracking, local appearance capturing.

1 Introduction

Recently, intelligent video surveillance system has attracted more and more attention from different researchers[1] [2] [3], because it possesses great potential in both civilian and military security applications. However, most of existing video surveillance systems either work on close-up view with short surveillance distance or work on remote scene surveillance with incapability of providing local appearance of object. However, local appearance of object often plays a crucial role in content analysis, object recognition and forensic authentication. For example, it is an urgent need for video surveillance system to provide distinguishable local appearance or features such as human face when intruder appears in surveillance scene. In addition, many of real-time video surveillance systems are only motion detection, not object tracking indeed. So these systems just can deal with surveillance by static camera, and fail once camera becomes moving. Especially, tracking in a remote scene and capturing local appearance simultaneously become a challenging task.

S. Boll et al. (Eds.): MMM 2010, LNCS 5916, pp. 489–499, 2010.

Various tracking algorithms have been proposed to overcome the difficulties that arise from noise, occlusion and changes in the foreground objects or in the surrounding such as Kalman Filter(KF), Particle Filter(PF), Active Shape Model(ASM), meanshift, and their variants[4], [5], [6], [7]. Among these algorithms, meanshift has achieved considerable success in object tracking due to its simplicity and robustness[8], [9]. However, it takes feature spatial distribution into little consideration, so that object and background with partly similar color or texture can not be distinguished.

Yang[10] proposed an object tracking method in joint feature-spatial spaces, feature spatial distribution such as pixel position has been taken into consideration, and experiments on several video sequences achieved satisfactory results. However, in Yang's method, time-consuming iteration always is a problem since it perhaps leads to local optimum. In addition, the use of feature spatial distribution is not adequate in Yang's method, more than one kind of feature need to be introduced simultaneously.

Collins[11] proposed an online tracking features selection method from linear combination of RGB values. This method can adaptively select the top N features, and meanshift algorithm was utilized. The median of N locations produced by meanshift is selected as the final object location. Liang[4] pointed the scale adaptation problem of literature[11] out, giving a corresponding resolution. And an evaluating principle is suggested to perform feature selection. Unfortunately, in their work[4],[11], feature selection was focused on while feature spatial distribution was ignored.

In this paper, we design a remote scene video surveillance system for object tracking and local appearance capturing with two cameras, and present a probabilistic tracking framework integrating feature value combination and feature spatial distribution. In the proposed system, we introduce: 1) a video surveillance system construction consisting of two cameras and a pan&tilt; 2) a 1-dimension feature combining hue, saturation and value; 3) a new formula to compute weight image, avoiding the interference from cluttered background; 4) a modified kernel-based tracking approach absorbing both color value and gradient of sample pixels, which is performed on weight image, not the whole image.

The structure of this paper is as follows: Section 2 presents hardware structure of the proposed system and the computing method of relationship of two views from two cameras. Section 3 shows feature value combining method, new computing method of weight image and a modified kernel-based tracking approach. Both simulated and real-time experimental results are shown in Section 4. Finally, Section 5 concludes the paper with a discussion.

2 The Proposed Surveillance System Structure

2.1 The Hardware Structure

To track object in remote scene and capture local appearance of this object, two kinds of focus lens are necessary in video surveillance system: one is short focus for tracking while the other one is long focus for capturing local appearance. Also two image outputs are necessary for performing different processing algorithms. Meanwhile, to deal with the condition of moving camera, a pan&tilt at least is need to rotate the camera. The ideal case of imaging device is to find a camera with two different focus lens of the same optic axis and two according video outputs, and a pan&tilt used to rotate the camera to tracking object. Unfortunately, no such camera can be found. Since the distance of surveillance is between 50m and 150m, so we select two cameras with different focus lens to simulate ideal camera with different focus lens. For convenience, we name the short focus camera as panorama camera and name the long focus camera as close-up camera in the following context.

For the proposed video surveillance system, some points need to be addressed:

(i) Though the panorama camera has a relatively broad view, no camera can cover overall angles of view in a remote scene without rotation.

(ii) The tracked object, which is usually in a remote scene, needs to capture local appearance in the close-up camera during the surveillance process.

(iii) It is troublesome to align the views of two cameras at the same center exactly. The common case is that the close-up view corresponds to some part of the panoramic view, rather than the central part.

Considering above issues, we fix the two cameras on the same pan&tilt, i.e., the position relationship of two cameras' view is unchangeable in the running. And a controller and computer are employed to adjust the pan&tilt according to the tracking results in panoramic view. Our system is illustrated in Fig. 1.

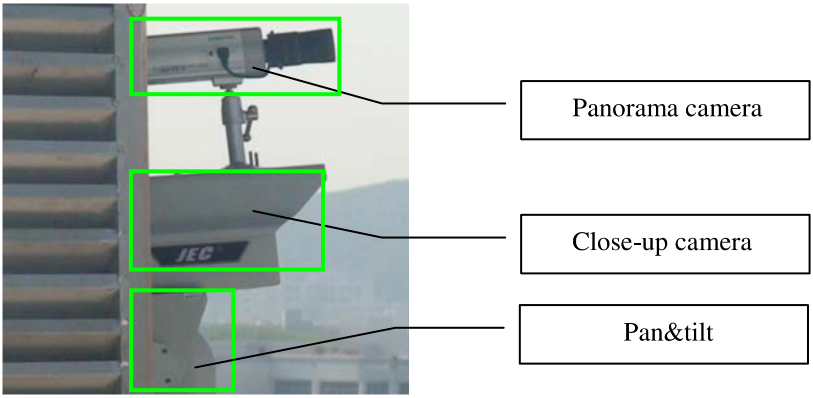


Fig. 1. Hardware structure of the proposed system

2.2 Computing the Relationship of Views from Two Cameras

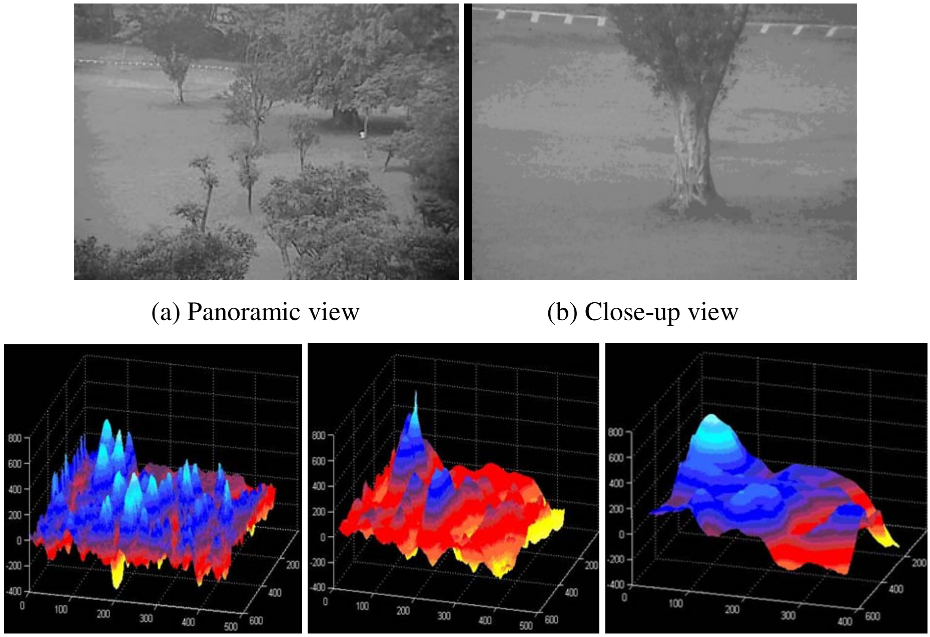
When an object is found to appear at a certain position in panoramic view, the controller should be informed the direction in which the pan&tilt is rotated for the close-up view, since these two views are not aligned to the same center. This means, though the position relationship of the views of two cameras is unchangeable, the relationship still remains unknown. However, this relationship is of utmost importance to make object remain in close-up view.

A reasonable assumption is made that there is no rotational diversity between two coordinate systems from different views, and it can be guaranteed readily while cameras are fixed. Therefore, translation and scale are the parameters that we take into consideration. We adopt convolution between panoramic view image and close-up view image with varying resolutions to estimate position $\mathbf{x}$ in the panoramic view image to which the close-up view image corresponds:

$$\mathbf{x} = \arg\max_{\mathbf{x}} \left(\frac{\mathbf{P}(\mathbf{x}) * \left(\mathbf{G}_{\sigma} - Avg\left(\mathbf{G}_{\sigma}\right)\right)}{Area\left(\mathbf{G}_{\sigma}\right)} \right) \tag{1}$$

where $\mathbf{G}_{\sigma}$ is a scaled version of close-up view image with variable scale parameter σ, and $\mathbf{P}(\mathbf{x})$ which has the same width and height as $\mathbf{G}_{\sigma}$ is the image blob located at the position $\mathbf{x}$ of panoramic view image. $Avg\left(\mathbf{G}_{\sigma}\right)$ is the average of pixel values in $\mathbf{G}_{\sigma}$, and $Area\left(\mathbf{G}_{\sigma}\right)$ is the area of $\mathbf{G}_{\sigma}$. The symbol * indicates spatial convolution. And convolution is performed on gray images.

$Area\left(\mathbf{G}_{\sigma}\right)$ is used as a normalization term in order to remove the influence of $\mathbf{G}_{\sigma}$'s size. This convolution can be treated as run a mask (i.e. $\mathbf{G}_{\sigma} - Avg\left(\mathbf{G}_{\sigma}\right)$) through the panoramic view image. Like some other digital masks, e.g. Sobel and LoG, the average of the above mask is zero. Considering that $\mathbf{x}$ only needs to be



(a) Panoramic view (b) Close-up view

(c) σ =0.1 (d) σ =0.2 (e) σ = 0.4

Fig. 2. Two gray images from the proposed system and the comparison of their convolution results at different parameters

computed once after cameras are fixed, we search exhaustively in the spatial-scale space to guarantee global optimum of $\mathbf{x}$, without considering the computational efficiency.

Fig. 2(a) and Fig. 2(b) show two gray images from panoramic view and close-up view, respectively. The size of the two images is 640×480 pixels. Fig. 2 (c), Fig. 2(d) and Fig. 3(e) are the convolution results with σ =0.1, 0.2 and 0.4. In our experiment, peak value of convolution results gets maximum, when σ =0.2. Therefore, the location of peak value in Fig.2 (d) is the estimated as position $\mathbf{x}$.

3 Tracking Approach

3.1 Color Components and Combination

HSV color model had been used in many image retrieval algorithms for having identical color discrimination with human vision systems. Hue is used as the only feature of object in traditional camshift, in which 1-dimension hue histogram is computed. Lacking of saturation and value, the tracker appears far from robustness. To overcome this disadvantage, many literatures [12], [13] prefer to employ higher-dimension feature spaces.

However, high-dimension space is not suitable to our applications, since our tracked object in the remote scene is relative small, containing few samples pixels in image. Using few samples to cover high-dimension feature space will lead to “curse of dimensionality” without doubt. Therefore, we adopt a 1-dimension feature combining hue, saturation and value with different weights. Hue is non-uniform quantized into 32 portions, while saturation and value are non-uniform quantized into 6 portions respectively:

$$L = w_h Q_h + w_s Q_s + w_v Q_v \tag{2}$$

$$Q_h = 32 \text{ , } Q_s = Q_v = 6 \tag{3}$$

$$w_h : w_s : w_v = 16 : 2 : 1 \tag{4}$$

where Q_h , Q_s and Q_v are the quantitative level of hue, saturation and value respectively, i.e.,. w_h , w_s and w_v are the weight of hue, saturation and value, respectively. The range of L is from 0 to 511.

3.2 A New Formula of Computing Weight Image

Weight image is an effective tool for estimating the position and the deformation of object in simple scene. It is generated in traditional camshift by using histogram back projection to replace each pixel with the probability associated with that color value in the target model.

$$I_o(\mathbf{x}) = \eta \cdot b(\mathbf{x}) \tag{5}$$

where $I_o(\mathbf{x})$ is the value of $\mathbf{x}$ in weight image, and $b(\bullet)$ is a function to map pixels into its color feature space has. η is a constant to normalize the value from 0 to 511.

Camshift algorithm works well in simple scenes, but fails in cluttered scenes due to the contamination of weight image. A new formula of computing weight image is proposed to make the weight image more reliable. This formula is established under the observation that the size and the location of tracked object changes smoothly in successive frames.

Before the computation of weight image, we first analyze the connected regions of original weight image $I_o(\mathbf{x})$ using the method in [14], and then a map of size of connected regions is generated by

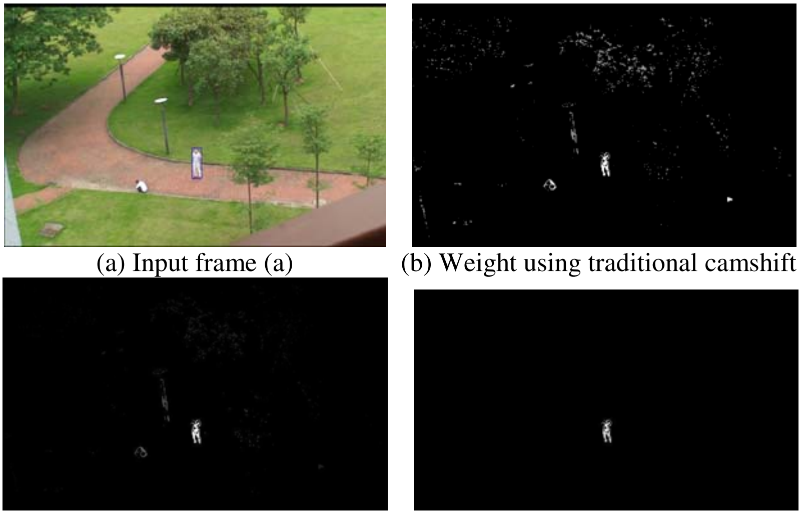
$$J(\mathbf{x}) = sizeof\left(R_i \mid \mathbf{x} \in R_i\right) \tag{6}$$

where R_i is the i-th connected region, and $sizeof(\bullet)$ is a function to map pixels into the size of the connected region which it belongs to. The new formula can be written as:

$$I_n(\mathbf{x}) = C_n \cdot b(\mathbf{x}) \cdot \exp\left(-\alpha \left\| J(\mathbf{x}) - J_t \right\| - \beta \left\| \mathbf{x} - \mathbf{C} \right\|\right) \tag{7}$$

where J_t is the number of pixels in the target model region, and $\mathbf{C}$ is the centroid of tracking result in last frame. α and β are the adjustable parameters. This formula means to give different weights to $b(\mathbf{x})$. The pixels in object tend to get boosted weights while those in background tend to get suppressed weights.

Fig. 3 illustrates the comparison of weight images from different methods and parameters. Fig. 3(a) is the input frame. Fig. 3(b) is the weight image generated by traditional camshift. Fig. 3(c) and Fig. 3(d) are the weight images generated by proposed formula with different parameters.


(a) Input frame (a)

(b) Weight using traditional camshift

(c) $\alpha=0$ and β=0.01

(d) $\alpha=0.1$ and β=0.01

Fig. 3. Comparisons of different weight images

3.3 A Modified Kernel-Based Tracking Approach Absorbing Color Value and Gradient Distribution

Although Collins' method[11] and Liang's method[4] selected and updated the features set, the tracking results mainly relied on "shift" of mass center of target image region, which is numerically unstable. Yang[10] proposed an efficient tracking algorithm via meanshift and new similarity measure, one of the advantages of this algorithm is to embed the Euclid distance between feature pixels and region center in meanshift. The new similarity measure is as follows:

$$J(I_x,I_y)=\frac{1}{MK}\sum_{i=1}^{K}\sum_{j=1}^{M}w(\|\frac{(x_*-x_i)}{\tau}\|^2)w(\|\frac{y-y_j}{\tau}\|^2)\kappa(\|\frac{u_i-v_j}{h}\|^2) \tag{8}$$

where I_x, I_y are sample points in model image and target image respectively, K is width of object region and M is height, x and y are the 2D coordinates of model image and target image, u and v are the selected feature vector for model image and target image, x_* and y are the center of sample points in the model image and the current center of the target points, respectively. $\kappa(x)$ is the Radial-Basis Function(RBF) kernel, h is bandwidth, $w(x)$ is another RBF kernel in the spatial domain, and τ is bandwidth. Yang's method[10] actually integrated feature value with the Euclid distance between feature pixels and region center. It provides a fine kernel-based tracking framework. However, there are three points that need to be strengthened in Yang's method.

Firstly, it's not enough that only the Euclid distance between feature pixels and region center is absorbed into kernel-based tracking algorithm. The more precise information of feature spatial distribution needs to be introduced. Since gradient is one of most discriminative spatial features, so we can define gradient similarity and embed it into similarity measure in literature[10]. Let p and q are the corresponding gradients of model image and target image, respectively. Then new similarity measure is defined as follows:

$$J(I_x,I_y)=\frac{1}{MK}\sum_{i=1}^{K}\sum_{j=1}^{M}w(\|\frac{(x_*-x_i)}{\tau}\|^2)w(\|\frac{y-y_j}{\tau}\|^2)\kappa(\|\frac{|u_i-v_j|+|p_i-q_j|}{h}\|^2) \tag{9}$$

In equation (9), u and v are 1-dimension feature combining hue, saturation and value with different weights(see section 3.1) for model image and target image, respectively.

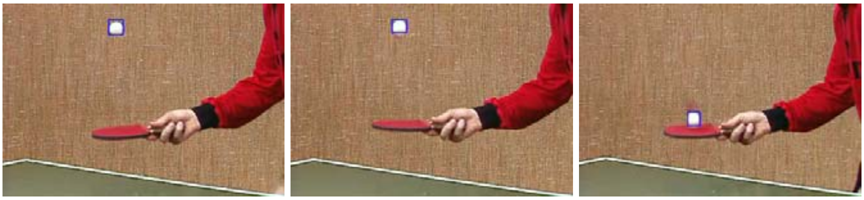
Secondly, computing method of tracking algorithm in literature[10] is a typical time-consuming iteration, which perhaps leads to local optimum. It can be assumed that the tracked object will appear in the position with large weight in weight image because the color change of object remains smoothly and slowly in several successive frames intervals. Thus we compute similarity measure in these pixels with large weight achieving global optimum and reducing time consumption.

Thirdly, since width and height of the object is not same usually, the bandwidth τ for x coordinate and y coordinate should be also different. Actually in our approach, computing of inscribed circle from square in Yang's method[10] is changed into computing of inscribed eclipse from rectangle.

4 Experiments

Both simulated experiments on MPEG testing sequences and real-time outdoor experiments are performed to verify the proposed approach. In all experiments, image gradient, which is obtained by the employment of *Sobel* operator after a Guassian smoothing operation, is the horizontal and vertical gradient of L in equation (2).

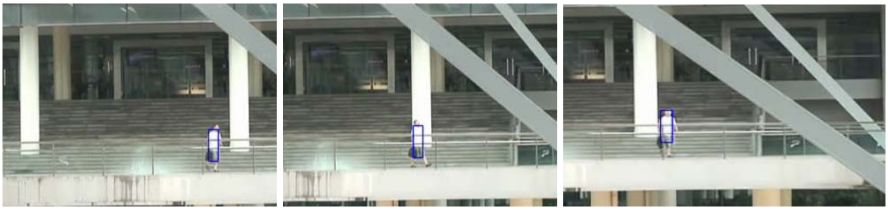
The first experiment, the *Ball* sequence for MPEG is tested, the contrast between object(white ball) and background(brown wall) is high. The object is initialized with a manually selected rectangular region in frame 0. The tracking results for frame 3, 16 and 26 are shown in Fig. 4.



Frame 3 Frame 16 Frame 26

Fig. 4. The results of simulated experiment on *Ball* sequence

In the second experiment, a complex video sequence *Library* is used to test. The size of image is 720×480 pixels. The contrast between object and background changes in a wide range and occlusion exists. Especially, contrast is rather low when pedestrian in white passed by white pillar. The object is automatically detected by Gaussian Mixture Modeling (GMM) of background. Fig. 5 shows the tracking results when pedestrian passed by white pillar. Though being similar in color to white pillar, the pedestrian is located accurately.



Frame 269 Frame 287 Frame 312

Fig. 5. The results of simulated experiments on the *Library* sequence

In the third experiment, we test the proposed surveillance system for two real-time scenes named by *Bridge* and *Lawn*. The distance between camera and object for *Bridge* scene is about 70m. The size of panoramic and close-up frame is the same, 640×480 pixels. Fig. 6 shows the overall results of our system for Bridge scene with software interface. Image on the left is from panorama camera, while that in the right is from close-up camera. Although it seems very small in image, the object is tracked accurately(see blue rectangle in Fig. 6). Adaboost algorithm[15] is used to detect human face(see red rectangle in Fig. 6). Certainly, many other works, e.g., face recognition and gait recognition, can be tried on the close-up image.

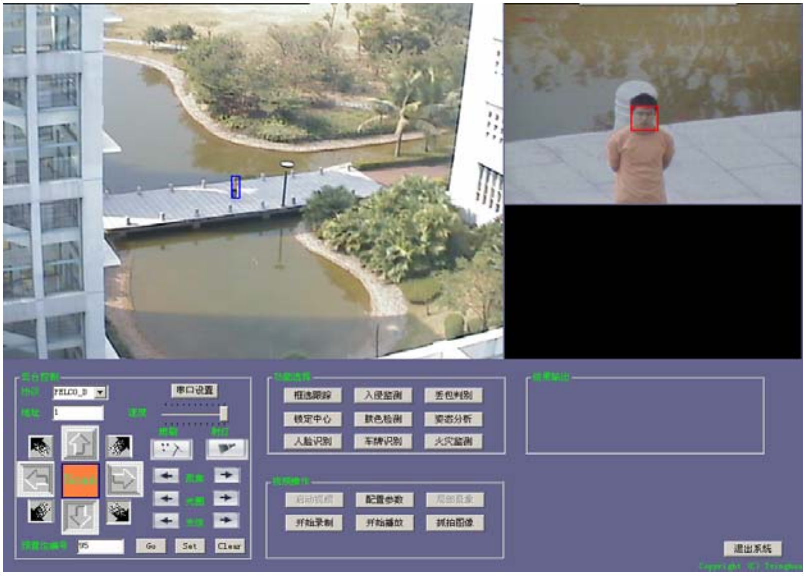


Fig. 6. The real-time outdoor experiments results for *Bridge* scene

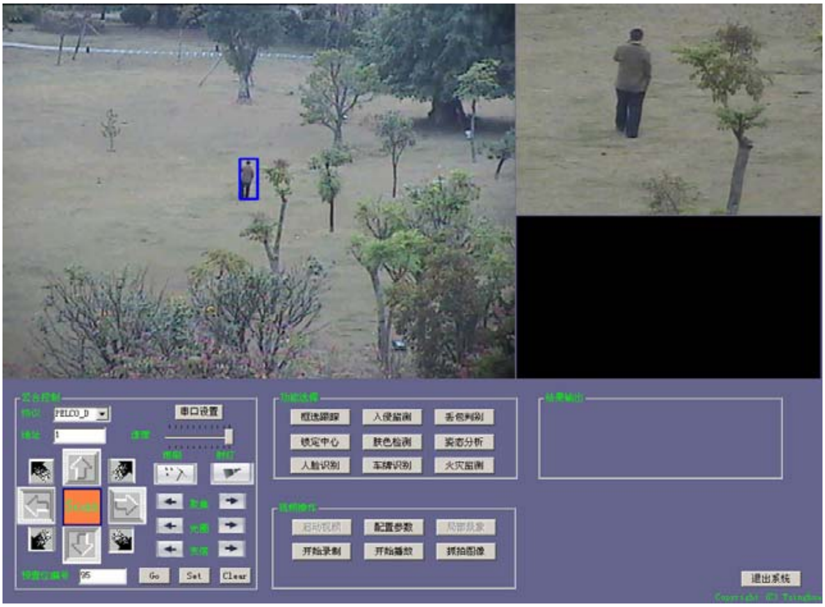


Fig. 7. The real-time outdoor experiments results for *Lawn* scene

The distance between camera and object for *Lawn* scene is about 150m. Fig. 7 gives a tracking result for frame 21. Our system provides a framework and platform for object recognition and content analysis, etc in video surveillance for remote scene.

Please note that the pan&tilt is rotated by controller to make object remain in close-up view, therefore the object is almost at the same position in panoramic view all the time.

5 Discussion and Conclusions

In this paper, we established a remote scene surveillance system with a panoramic view for object tracking as well as a close-up view for local appearance capturing, aiming at enhancing the practical functionalities of video surveillance system. In tracking algorithm, we proposed a modified object tracking approach integrating color feature combination with feature pixel spatial distribution. Namely, The HSV color value of pixels and gradient of pixels both are introduced into kernel-based tracking framework. A method to compute the position relationship of these two views is proposed. In addition, a new formula of computing weight image is explored to avoid the interference from cluttered background. Both simulated and real-time experimental results demonstrated that the proposed tracking approach and surveillance system work well.

We will focus on improving the quality of close-up image and the operationality of pan&tilt in future research. Also, real-time online face recognition, gait analysis and recognition will be tried on the basis of our remote scene video surveillance framework. In addition, we set manually several fixed bandwidths and thresholds in experiments, their adaptive selection will be studied in the future work.

Acknowledgments. The research leading to this paper was partially supported by China Postdoctoral Science Foundation(No.20090450353) and Guangdong Nature Science Foundation of China(No.9451805702003923). The authors would like to thank the three anonymous referees whose comments and suggestions have greatly improved this paper.

References

1. Foresti, G.L.: Object Recognition and Tracking for Remote Video Surveillance. IEEE Transactions on Circuits and Systems for Video Technology 9, 1045–1062 (1999)
2. Bue, A.D., Comaniciu, D., Ramesh, V., Regazzoni, C.: Smart cameras with real-time video object generation. In: Proceedings of International Conference on Image Processing, pp. 429–432 (2002)
3. Chen, T.-W., Hsu, S.-C., Chien, S.-Y.: Automatic Feature-based Face Scoring in Surveillance Systems. In: IEEE International Symposium on Multimedia, pp. 139–146 (2007)
4. Liang, D., Huang, Q., Jiang, S., et al.: Mean-shift Blob Tracking with Adaptive Feature Selection and Scale Adaptation. In: IEEE International Conference on Image Processing, San Antonio, United States, pp. 369–372 (2007)
5. Chang, C., Ansari, R., Khokhar, A.: Multiple Object Tracking with Kernel Particle Filter. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, vol. 1, pp. 566–573 (2005)

6. Shiu, Y., Kuo, C.-C.J.: A Modified Kalman Filtering Approach to On-Line Musical Beat Tracking. In: IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 2, pp. 765–768 (2007)
7. Lee, S.-W., Kang, J., Shin, J., et al.: Hierarchical Active Shape Model with Motion Prediction for Real-time Tracking of Non-rigid Objects. IET Comput. Vis. 1(1), 17–24 (2007)
8. Comaniciu, D., Meer, P.: Mean Shift: A Robust Approach Toward Feature Space Analysis. IEEE Transaction on Pattern Analysis and machine Intelligence 24(5), 603–619 (2002)
9. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based Object Tracking. IEEE Transaction on Pattern Analysis and machine Intelligence 25(5), 564–577 (2003)
10. Yang, C., Duraiswami, R., Davis, L.: Efficient Mean-Shift Tracking via a New Similarity Measure. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, vol. 1, pp. 176–183 (2005)
11. Collins, R.T., Liu, Y., Leordeanu, M.: Online selection of discriminative tracking features. IEEE Transaction on Pattern Analysis and Machine Intelligence 27(10), 1631–1643 (2005)
12. Perez, P., Hue, C., Vermaak, J., Gangnet, M.: Color-Based Probabilistic Tracking. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2350, pp. 661–675. Springer, Heidelberg (2002)
13. Maggio, E., Cavallaro, A.: Multi-Part Target Representation for Color Tracking. In: Proceedings of International Conference on Image Processing, pp. 729–732 (2005)
14. Salembier, P., Oliveras, A., Garrido, L.: Antiextensive Connected Operators for Image and Sequence Processing. IEEE Transactions on Image Processing 7, 555–570 (1998)
15. Jones, M., Viola, P.: Rapid Object Detection using a Boosted Cascade of Simple Features. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 511–518 (2001)