

Multi-camera tracking algorithm study based on information fusion

Guoqiang Wang^a, Shangfu Li and Xue Wen

School of Electronic Engineering, Heilongjiang University, Harbin 150000, China.

Abstract. Intelligent video surveillance technology is a part of pattern recognition used for analyzing, extracting and recognizing behavior characteristics of a moving target basing on computer algorithms. The target tracking algorithm combines particle filter and Mean-shift in overlapping multi-camera environment on the intelligent video surveillance system. Multi-camera tracking is studied using a fusion of SURF characteristics, color characteristics and geometric characteristic in Matlab. Experimental results show that the method of tracking between multiple cameras has good accuracy and stability.

Keywords: characteristics fusion; target tracking; pedestrian matching.

1 Introduction

Intelligent video surveillance technology is one aspect of pattern recognition, which uses a number of computer algorithms [1], analysis, extraction and recognition of moving objects within the video behavioral characteristics, and we use the computers to determine whether the behavioral characteristics of the moving object are defined as "suspicious behavior". During the fifty years of research and development, the target tracking has been largely used in various areas [2]. One camera can detect limited area, it is difficult to track target with multiple perspectives, and long time and wide range, so using multiple cameras become inevitable. The advantages of multi-camera tracking are the wide field of view, comprehensive perspective and large monitoring area, which makes the multi-camera tracking become popular.

2 Method summary

Every feature of object has its limitations when we match the object, so it is unable to reflect the characteristics of the target only by using one feature, and it is also difficult to match a target. Thus the robustness would be improved if we use a various type of features. This paper uses SURF, color and geometric characteristics of pedestrian profile to complete matching [3]. This method is based on detecting, we should work out the SURF key descriptor, color and geometric characteristics of the profile which are located in the divided target area, and then we calculate the size of the distance between the feature matrixes, as a result of those above, we could complete the pedestrian matching. The basic process of this paper shows below [4]. First of all, in order to handle the video sequences

^a Corresponding author : wangguoqiang@hlju.edu.cn

from different cameras, we should use background subtraction to finish the motion segmentation and obtain the binary image of the target. The result is showed in Figure 1.



Figure 1. The results of motion segmentation.

3 Feature extractions

3.1 SURF feature extraction

SURF feature points matching method is an improved method of SIFT, mainly in terms of speed, and its efficiency is higher than the SIFT. SURF algorithm processes integral images and convolution that is only related to previous image, the down sampling is depending on applied increasing size of its core [5]. SURF algorithm can be analyzed at the same time several layers of image scale space, and eliminating the need for two image sampling process, so the execution time of the algorithm can be improved. SURF algorithm can be divided into the following four steps: (1) Describe Gaussian pyramid structure scale space; (2) Locate feature points Position; (3) Determine the direction of the feature point; (4) Calculate SURF descriptors.

(1) Describe Gaussian pyramid structure scale space

The most important point of the reduced running time for SURF algorithm is that it uses the integral image. The formula (1) is given:

$$I_{\sum(x,y)} = \sum_{i=0}^{i<x} \sum_{j=0}^{j<y} I(x, y) \tag{1}$$

The using filter is different between SURF and SIFT when they build the multi-scale space, SURF uses box filter which is shown in Figure 2.

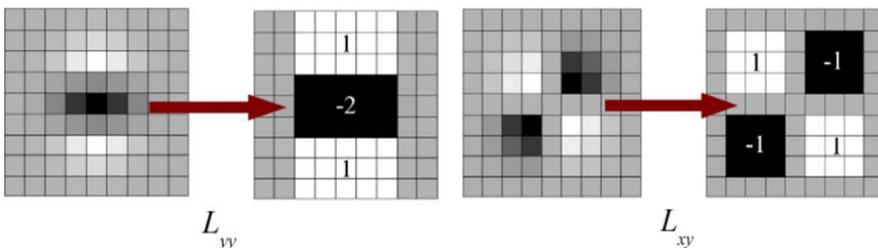


Figure 2. Box filter.

SURF algorithm uses the Hessian matrix determinant approximation image. Formula (2) is the Hessian matrix of one image pixel:

$$H(f(x, y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} \quad (2)$$

That is, each pixel can order out a Hessian matrix. But because the feature point scale need to have scale irrelevance, we need to execute Gaussian filter first, and then build Hessian matrix. The formula is given:

$$H(X, \delta) = \begin{bmatrix} L_{xx}(X, \delta) & L_{xy}(X, \delta) \\ L_{xy}(X, \delta) & L_{yy}(X, \delta) \end{bmatrix} \quad (3)$$

Where, $L_{xx}(X, \delta) = G(\delta) * I(X)$, $G(\delta) = \frac{\partial^2 \delta}{\partial x^2}$, δ is the variance. $L_{xy}(X, \delta)$ is the image of a point of second-order filter with Gaussian convolution. $L_{xy}(X, \delta)$, $L_{yy}(X, \delta)$ is similar as above.

Using mean filter to solve the second derivative of the Gaussian in SURF, the approximation coefficients is $w = 0.9$, and then Hessian matrix can be get by the formula above, the formula is:

$$\det(H) = D_{xx}D_{yy} - (0.9 \times D_{xy})^2 \quad (4)$$

(2) Locate feature points Position

The principle of positioning feature points accurately is that we should abandon all values which are lower than the setting extreme value, so that the number of discoverable feature points can be reduced and ultimately we can receive some points which have the most obvious features.

(3) Determine the direction of the feature point

After detecting the feature points, we need to determine the main direction of the feature point, to ensure that the feature points have a characteristic of rotation and scale invariance.

We should select a random area corresponding to the regional scale in the vicinity of the points, and identify the main orientation. SURF selects a circle area, and determines the main orientation by using a method of activity sector, the main orientation of a line can be realized.

(4) Calculate SURF descriptors

The method of getting SURF descriptors: Select a $20s$ size of window along the main orientation of the feature point position, and divide it equally into a 4×4 rectangle, then calculate the Haar wavelet response of the feature points in each 5×5 rectangle space separately, we can see that in Figure 3, d_x is the Haar wavelet response of x orientation, d_y is the Haar wavelet response of y orientation. Firstly, we should set a Gaussian-weighted for d_x and d_y in the vicinity of the center of interested points. ($\delta = 3.3s$) Then, we should calculate the cumulative sum of each rectangle areas d_x , d_y , $|d_x|$ and $|d_y|$. After this, each rectangle area has a 4-dimensional intensity which represents a vector, its vector can be described as $V_s = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$, and finally we can get a 64-dimensional feature vector. It can improve the robustness of the geometric distortion and positioning errors.

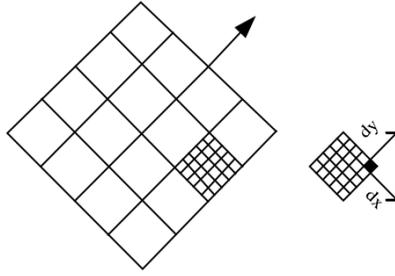


Figure 3. The key descriptors constitute.

The similarity between the two images is determined by using the Euclidean distance of the key feature vector in SURF algorithm [6]. We take a critical point within the first image and search the former two feature points in the second image which has a minimum Euclidean distance with the critical point. If the value that the minimum distance divides the second-smallest distance is lower than the set threshold value, we admit the two matched points [7].

3.2 Color feature extraction

Color feature has an important role in most object tracking methods because of its advantages, for example, it has the invariance. It is changeless with object's changing dimensions as well as rotation. Firstly, we translate the target image from the RGB color space to HSV color space. Then, after getting SURF key point coordinates (x, y) , we should calculate the H , S and V mean value of a 3×3 neighborhood of the key point. Finally, the pixel values of key point (x, y) have been translated from (R, G, B) space into HSV space and described as the mean values (h, s, v) , which is 3D color feature vector $V_c = [h, s, v]$.

3.3 Geometric characteristics of the target profile

The geometric characteristics of the pedestrian's profile are described by the contour of the pedestrian and the positional relationship of curve. In that case, we can extract several special points through the pedestrian contour edge, calculate Euclidean distance between SURF feature points and extracted points, and define the distance as the geometric characteristics of the pedestrian profile.

We use Canny operator to acquire the edge of the pedestrian profile image, then use the results of binary image segmentation to remove the background image of pedestrian area and use a black background to replace the original background, so we ensure basically that the Canny edge extraction operator is integrate, which is shown in Figure 4.



Figure 4. Geometric features and its contour map.

Select several special points from the edge of the image outline to extract the geometrical features of a pedestrian (shown in Figure 4):

- (1) The maximum distance of the pedestrian outline is defined as the long axis AD , which

represents the height of a pedestrian profile;

(2) According to AD , we define a horizontal axis BC which is perpendicular to the longitudinal axis AD and place it at $1/4 AD$, which represents the width of the pedestrian contour;

(3) Assume any SURF key point P and the distance from P to A, D, B, C which constitutes the object contour feature vector V_p :

$$V_p = \left(\frac{PA}{AD}, \frac{PD}{AD}, \frac{PB}{AD}, \frac{PC}{AD} \right) \quad (5)$$

Because the various features of V_p is performed by calculating the ratio, V_p possesses the invariance property when the image is in a condition of rotation, translation and scale change. Detected feature vectors have the same feature point in any pedestrian image, and each feature point is a stable feature descriptor, which consists of the following parts: SURF descriptor V_s represents local features characteristic points, color vector V_c represents the color characteristics, geometric vector V_p can remove local similar feature. So each point descriptor vector can be expressed as:

$$F = [\alpha V_s, \beta V_c, (1 - \alpha - \beta) V_p] \quad (6)$$

In formula (6), α and β are weight and balance features, we adopt $\alpha = 0.3, \beta = 0.4$.

4 Target tracking based on fusion

We suppose that at t moment, there are M targets in the field of Camera i , which are segmented and marked; there are N targets in the field of Camera j , which are segmented and marked, M and N perhaps be equal.

We define the x -pedestrian in Camera i is O_x^i , the number of SURF key points obtained is a ; the y -pedestrian in Camera j is O_y^j , the number of SURF key points obtained is b ; calculate V_c and V_p which are in the $3*3$ vicinity of the SURF key points at the same time and get the feature vector matrix valued $[a*135]$ and $[b*135]$. Give two feature descriptors F_i and F_j , then D can be got by the formula (7):

$$D = |F_i - F_j| = \alpha \sqrt{\sum_{k=0}^{63} (V_{sik} - V_{sjk})^2} + \beta d_{ij} + (1 - \alpha - \beta) \sqrt{\sum_{k=0}^3 (V_{pik} - V_{pj k})} \quad (7)$$

where,

$$d_{ij} = \frac{\sqrt{(h_{x1} - h_{y1})^2 + (s_{x1} - s_{y1})^2}}{\sqrt{h_{x1}^2 + s_{x1}^2} + \sqrt{h_{y1}^2 + s_{y1}^2}} \quad (8)$$

The similarity of two matched is given:

$$D(O_x^i, O_y^j) = \frac{\sum (D_1, D_2, \dots, D_a)}{a} \quad (9)$$

The step of the algorithm is as follows:

(1) The quantitative distribution of the whole pedestrian areas in Camera i is $1 \sim M$, and the quantitative distribution of the whole pedestrian areas in Camera j is $1 \sim N$.

(2) We select a pedestrian area p and a multi-feature vector descriptor z in the field of Camera i , and calculate the distance D by formula (7) with all the descriptors in the whole pedestrian area q of Camera j . Search the minimum similarity distance D between two descriptors and get the similarity distance between p and q by formula (9).

(3) Back to step (2) and stop until we go through all the pedestrian area of Camera i . Get the similarity distance between all the pedestrian areas of Camera i and Camera j .

(4) For any pedestrian area in Camera i , the pedestrian area in Camera j is matched to the one which has the minimum similarity distance.

5 Experimental results and analysis

Figure 5 is one image of Camera i and j , the image size is pixel. There is a smaller overlapping area between Camera 1 and Camera 2. There is only one pedestrian in the field of Camera 2. We calculate the distance between the pedestrian in Camera 2 and the three pedestrians in Camera 1 by formula (7) and (9) respectively, and then we compare them and find that the minimum distance is successfully matched with the camera 1. The match results are shown in Table 1.

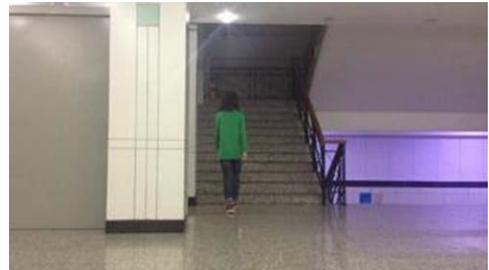
Table 1. The results of matching.

Target in camera 1	1	2	3
The distance	0.5429	0.5359	0.5096

The tracking results are shown in Figure 6, and the operating environment is MATLAB2010a, the computer configuration is Pentium (R) Dual-Core processor TL56 3.00GHz, 2G memory, Windows7 (32-bit) operating systems.



(a) The view of camera 1

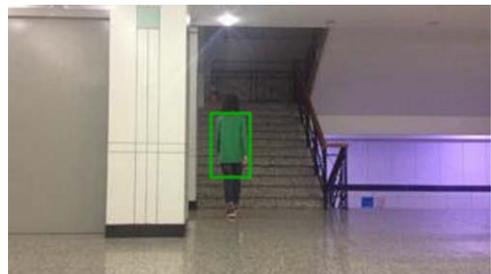


(b) The view of camera 2

Figure 5. The image of Camera 1 and Camera 2.



(a) The results of camera 1



(b) The results of camera 2

Figure 6. The results of tracking.

6 Summary

This paper is in the framework of multi-camera video monitoring and analysis system, we use a fusion of SURF feature descriptors, colors and the geometric characteristics of the pedestrian's profile to match the targets, and aim at improving the accuracy of pedestrian matching. Further study will focus on how to link captured information from each camera and how to improve the efficiency of collaboration between multiple cameras in terms of time and accuracy.

References

1. C. Dorin, Kernel-based object tracking. *IEEE Trans on Patern Analysis and Machine Intelligence*, **25**, 564-577 (2013)
2. J. Czyz, B. Ristic, B. Macq, A particle filter for joint detection and tracking of color objects. *Image Vision Computing*, **25**, 1271-1281(2012)
3. R.Mazzon, P. A.Cavallaro, Multi-camera tracking using a Multi-Goal Social Force Model. *Neurocomputing*, **100**, 41-50(2013)
4. M. Piccardi, E. D. Cheng, Track matching over disjoint camera views based on an incremental major color spectrum histogram. *IEEE conference on Advanced Video and Signal Based Surveillance. Como*, 147-152(2012)
5. G. Y. Lian, J. H. Lain, W. S. Zheng, Spatial-temporal consistent labeling of tracked pedestrians across non-overlapping camera views. *Pattern Recognition*, **44**, 1121 -1136(2010)
6. P. L. Mazzeo, P. Spagnolo, T. D'Orazio, Object Tracking by Non-overlapping Distributed Camera Network., 516-527(ACIVS, 2012)
7. X. G. Wang, Intelligent multi-camera video surveillance: A review. *Pattern Recognition Letters*, **34**, 3-19(2013)