Target Detection Algorithm Based on the Movement of Codebook Model

Kaige Chen, Xiaojun Han & Tenghao Huang
College of Electronic and Information Engineering, Tianjin Polytechnic University
Tianjin, 300387, China
E-mail: hanxiaojun_tj@126.com

Received: December 5, 2011 Accepted: December 20, 2011 Published: March 1, 2012
doi:10.5539/cis.v5n2p49 URL: http://dx.doi.org/10.5539/cis.v5n2p49

Abstract

Movement target detection is the researchful emphasis and aporia in the fields of computer vision, model recognition and video coding. In order to extract moving targets from the complex background scenes, this paper puts forward to calculate the color distortion degree by means of getting every pixel or a group of pixels of time series model, and converting pixels from RGB space to HSV space. Based on the background Codebook model of the target motion detection algorithm, the experiment shows that the method can achieve better target detection quality.

Keywords: Codebook, Motion target detection, Computer vision, Distortion degree

1. Introduction

With the development of computer technology, using computer to realize the human visual function through a computer has become one of the most popular subjects in computer field. One important aspect of computer vision research is motion target detection. It can be widely applied in the fields of intelligent video monitoring, human-computer interaction, virtual reality, robot navigation, traffic detection and many other areas. Light flow method (McKenna Setal *et al.*, 2000) is rarely used because of the computational complexity, and frame differential method can deal with the slow background change. However, the detected objects easily appear so rupture and empty that the complete motion target is hard to get. Therefore, people tend to use the method of reducing background more and more.

At present, the problem on moving target detection has not been solved completely in complex environment. This is mainly because the background changes happen at any time whether indoors or outdoors including illumination change, background disturbance and the background change caused by the motion target itself, etc. Meanwhile, there are some shortcomings of the present algorithm in background renewal speed, threshold selection flexibility, memory storage and accuracy. For example, finite difference method (C. Bregler, 1997) and mean filter method (C. Wren *et al.*, 1997) call for large storage space and have poor adaptive ability. And the unimodal distribution model (D. Koller *et al.*, 1993) cannot model the background of the shaking tree branches. Although gaussian mixture model can be applied in background modeling with complex or dynamic background noise(such as the background raindrops and the shaking tree branches), it is difficult to use several limited gaussian model to describe the rapidly changing background accurately. And it is hard to keep the learning efficacy of gaussian mixture model in an ideal state.

Based on the above reasons, this paper proposes the future inspection method on the basis of Codebook finite difference method, which compares the color distortion by converting pixels from RGB space to HSV (hue, saturation, value) space. And through HSV color sampling study on three axis, every pixel can get a code book. According to the distribution characteristics of the prospects and background points in the image point, the background code words are separated and can be used for background modeling. The results show that although the conversion to HSV color model introduces floating point arithmetic which makes the processing speed slower compared to the method of Kim (Kim K *et al.*, 2004), more accurate detection effect can be obtained that the foreground objects of the complex background can be detected successfully.

2. Codebook Background Model

Codebook model algorithm is to obtain each pixel yards based on the color standard and brightness boundaries clustering, by means of using quantitative and clustering techniques to build background model and training each pixel. Each pixel has a different codebook, each codebook contains several code words, and all the pixels of codebook make up the codebook model.

If $X=\{x_1,x_2,\cdots,x_N\}$ is the sequence sample value of a pixel, among which x_t (t=1,…, N) is RGB vector . If $C=\{c_1,c_2,\cdots,c_L\}$ is the codebook of this pixel, each code word x_i (i=1,…, L) is defined as binary group structure $v_i=(\overline{R_i},\overline{G_i},\overline{B_i}),u_i=<I_i^{\min},I_i^{\max},f_i,\lambda_i,p_i,q_i>$, among which $I^{\min}$ and $I^{\max}$ is the corresponding pixel minimum and maximum brightness values of the code words. f can show the frequency of the codebook and λ can show the maximum time interval of the code words. Besides, p and q can be used to show the matching time of the first and last time after its appearance which can be set as a simple frame.

In the process of training the initial Codebook, Let X is defined as a single pixel consisting of N RGB-vectors: $X=\{x_1,x_2,\cdots,x_N\}$. Every sample value x_t represents the sample at the moment of t. It can be compared to the present yard so as to determine which yard c_m can be matched with. Then the matching yard is used as code estimate of the sample. In order to ascertain the best match of the yard, the color distortion and brightness are used as the reference for the measure boundary. When pixel sample values surpass the existing Codebook border, new Codebook entry is created. And when the pixel value is in the existing border area, the code sample border will increase. However, if a pixel is beyond the boundary distance of the code sample, a new code entry is created and a high or low threshold is set. And every code entry is examined to check whether pixel is in the code border. Besides, if this pixel value is within the border, the maximum and minimum threshold is adjusted to make sure it is included in the code border. Meanwhile, the last updated time is set as the current time and the statistics access frequency of each code item is estimated. In addition, when the code cannot be assessed for a long time, it may be due to the noise or moving prospect target. As time goes on, it becomes an old code. Therefore, it is necessary to update the needed entry codebook and delete the unwanted entry code. The main steps of the algorithm (A. Elgammal *et al*., 2000) are described as:

```
Ⅰ. Set $L \leftarrow 0$, $c \leftarrow \phi$;
Ⅱ. FOR $t=1$ TO N DO
    ⅰ. $x_t=(R,G,B),\ I \leftarrow \sqrt{R^2+G^2+B^2}$        (1)
    ⅱ. Search code $c_m$ in $C=\{c_m \mid 1 \le m \le L\}$ to match the sample $x_t$ to satisfy:
        brightness boundary color distance $colordist(x_t,v_m) \le \varepsilon_1$,
        brightness boundary $brightness(I,<I_m^{\min},I_m^{\max}>,)=ture$.
    ⅲ. If $c \leftarrow \phi$ cannot find the matched code, $L=L+1$ and a new code entry $c_m$ is added:
        $v_L \leftarrow (R,G,B),\ u_L \leftarrow <I,I,1,t-1,t,t>$;        (2)
    ⅳ. Updated matching code $c_m$ includes: Updating $u_m=<I_m^{\min},I_m^{\max},f_m,p_m,q_m>$ and
        $v_m=(\overline{R_m},\overline{G_m},\overline{B_m})$:
        $v_m=((f_m\overline{R_m}+R_t)/(f_m+1),(f_m\overline{G_m}+G_t)/(f_m+1),(f_m\overline{B_m}+B_t)/(f_m+1))$
        $u_m \leftarrow <\min\{I,I_m^{\min}\},\max\{I,I_m^{\max}\},f_m+1,\max\{t-q_m,\lambda_m\},p_m,t>$        (3)
  END FOR
Ⅲ. For every code $c_i$, $i=1,\cdots,L$ : set $\lambda_i \leftarrow \max\{\lambda_i,(N-q_i+p_i-1)\}$        (4)
```

Ⅳ. Using λ to eliminate redundant code words, the accurate and initial codebook with the representative background is found out M (K is the index of code word): $M = \{c_k | c_k \in C, \lambda_k \le T_M\}$.

Steps in the algorithm, ε_1 is global threshold variable which must make suitable adjustment for the application. Threshold T_M usually take half the number of frames, namely $N/2$ all the code words with the representative background must appear within the frame.

Among these, the 2 conditions of step Ⅱ are that the colors of x_t and c_m are very similar and that the brightness of x_t must be within the acceptable brightness of c_m. Here it is necessary to find the code word that makes the first condition satisfy the second condition. The reason of introducing time rule λ is that redundant codes are obtained in the process of training among which some represent the prospect moving target and noisy code words. Using step Ⅳ can separate these code words in probability sense which allows the existence of moving target in the initial training process.

3. Determining the Scope of Color Distortion Degree and Brightness Degree

In order to deal with the illumination changes of shadows and highlights, RGB model is often used. But it is bad for the color effect of the dark areas because the pixel differences in the dark areas have higher stability than that in the bright areas. For example, L and D represent the two dark pixels and a lighter pixel, then their color distortion degree (CDD) is calculated respectively as shown in Table 1.

From Table 1 we can see that the color change is larger in low light than that in high light. Thus high brightness of the detection sensitivity will be sacrificed.

Theoretically, RGB color space has bigger differences than eye perception, while HSV color space is consistent with human eye perception characteristics and it has been widely used in the video image retrieval. Through HSV color space, human eyes can have the separate perception of the color change. And the perceived color difference is proportional to the amount of Euclid distance. Therefore this paper will calculate the color distortion degree by converting pixels from RGB space to HSV space, so as to judge the changes between prospect pixel and background pixel (Greiffenhagen M. *et al*., 2000) agilely.

Suppose a function RGB2HSV ( ) which is converted pixel RGB value to HSV value, h, s, v are three vectors in HSV color model, $h \in [0,2\pi], s \in [0,1], v \in [0,1]$.For pixel $x_t = (R_t, G_t, B_t)$ and $y_m = (\overline{R_m}, \overline{G_m}, \overline{B_m})$.

$$p_t = RGB2HSV(x_t), p_m = RGB2HSV(y_m)$$

$$colordist(x_t, y_m) =$$

$$(p_t.s^* \sin(p_t.h) - p_m.s^* \sin(p_m.h))^2 +$$

$$(p_t.s^* \cos(p_t.h) - p_m.s^* \cos(p_m.h))^2 +$$

$$(p_t.v - p_m.v)^2 \tag{5}$$

In order to define dark and highlight areas, it is necessary to define the brightness change range of the moving target. And for each code word, its scope is defined as $[I_{low}, I_{hi}]$:

$$I_{low} = \alpha I^{\max},\ I_{hi} = \min\{\beta I^{\max}, I^{\min}/\alpha\} \tag{6}$$

Among them, $\alpha < 1, \beta > 1$. $0.4 < \alpha < 0.7$, the smaller α is, the greater the brightness range is. $1.1 < \beta < 1.5$, the range of the code is stable in the process of updating. Brightness function is defined as

$$brightness \quad (I, \langle I^{\min}, I^{\max} \rangle) = \begin{cases} ture & , I_{low} \le \|x_t\| \le I_{hi} \\ false & , others \end{cases} \tag{7}$$

4. The Prospect of Movement Target Detection Based on Codebook

For detection technology to reduce background motion, the most direct method is to subtract the current frame from the background model. This paper will judge whether pixel sampling value matches with the corresponding code. If they match, the point is taken as the background point. Otherwise, it is taken as the prospect point. For the new input pixel $x = (R, G, B)$ and the corresponding code M in the process of movement target detection, reducing background operation algorithm $BGS(x)$ is shown in Figure 1.

5. Updating the Background Model in the Testing Process

In order to be suitable for the real-time accuracy of the video monitoring system, updating the code is necessary in the process of target detection. Much attention should be paid to illumination change and moving target itself (such as stop or traffic) caused by the changes of the background model updating. And it includes two parts, namely the background areas and the coverage areas of the frame object. If the sample value of a pixel cannot match with the present code M , a new code word will be created.

First, updating the background pixel and finding the new input pixel $x = (R, G, B)$ matched with code word c_m from M, then updating $u_m =< I_m^{\min}, I_m^{\max}, f_m, \lambda_m, p_m, q_m >$ and $v_m = (\overline{R_m}, \overline{G_m}, \overline{B_m})$ to

$$u_m = (\min < I, I_m^{\min} >, \max < I, I_m^{\max} >, f_m + 1, \max(t - q_m, \lambda_m), p_m, t)$$

$$v_m = ((f_m \overline{R_m} + R)/(f_m + 1), (f_m \overline{G_m} + G)/(f_m + 1), (f_m \overline{B_m} + B)/(f_m + 1)) \tag{8}$$

At the same time, a frame for moving objects coverage area is considered. The updating method will incorporate it into the background model at high updating rate and update the matched code words c_m , u_m and $v_m = (\overline{R_m}, \overline{G_m}, \overline{B_m})$ to

$$v_m = ((f_m \overline{R_m} + 2R)/(f_m + 2), (f_m \overline{G_m} + 2G)/(f_m + 2), (f_m \overline{B_m} + 2B)/(f_m + 2))$$

$$u_m = (\min < I, I_m^{\min} >, \max < I, I_m^{\max} >, f_m + 1, \max(t - q_m, \lambda_m), p_m, t) \tag{9}$$

6. Analyzing Experimental Results

In order to evaluate the proposed algorithm, Visual C++6.0 and open source machine vision library OpenCV1.0 are used. And in order to detect the prospect target, a series of videos realize the background modeling and background model is regularly updated. Moreover, the results show that as to outdoor video, every pixel needs an average six yards to get background model, while indoor video just needs one or two background value. This algorithm needs to be trained once and the repeated training cannot enhance the detection ability. And the time standards λ can be used to distinguish the code words having real background from the moving foreground code words. And it makes no difference between considering the frequency f and λ of the code words and only considering the inspection ability of λ . The experimental results are obtained after the simple morphology processing by using the algorithm for a group of indoor and outdoor videos. They are shown as Figure 2 and Figure 3. Parameter settings in the experimental process are shown in Table 2.

In Table 2, $\varepsilon_1, \varepsilon_2$ are threshold variables and α , β are brightness coefficients, N is for training the frame number. In the process of background modeling, the existence of the moving target cannot be allowed to store the previous pixel numerical value, so the small memory is needed. From above we can see that because the brightness and color difference are considered in the method of this paper, the dark and highlight areas are dealt with effectively.

7. Conclusion

In this paper, adaptive background minus algorithm is used to obtain effective background model through simple training which can limit the size of the memory used and can be well adapted to the background movement and light changes. Because the color distortion is compared by converting the pixel from RGB space to HSV space, more accurate target detection effect is achieved than the method of Kim. However, due to the introduction of floating point arithmetic, the algorithm processing speed is becoming slower. Therefore, in the future research, more attention should be paid on the limited strategy of the code size and the ordering strategy of the code words, which can reduce the memory match the time of the codebook and further improve the processing speed.

References

A. Elgammal, D. Harwood, & L. S. Davis. (2000). Non-parametric model for backgroun subtraction. *In proceedings of European Conference on Computer Vision*, 2, 751-767.

C. Bregler. (1997). Learning and recognizing human dynamics in video sequences. In *Proceedings of IEEE*

CVPR 97, 568-574. http://dx.doi.org/10.1109/CVPR.1997.609382

C. Wren, A. Azarbayejani, T. Darrell, & A. Pentland. (1997). Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7), 780-785. http://dx.doi.org/10.1109/AFGR.1996.557243

D. Koller, K. Danilidis, & H. H. Nagel. (1993). Model-based object tracking in monocular image sequences of road traffic scenes. *International Journal of Computer Vision*, 10(3), 257-281.

Greiffenhagen, M., Ramesh, V., Comaniciu, D. *et al*. (2000). Statistical modeling and performance characterization of a realtime dual camera surveillance system. *IEEE CVPR*, 335-342. http://dx.doi.org/10.1109/CVPR.2000.854840

Kim, K., Chalidabhongse, T. H., Harwood, D., *et a1*. (2004). Background modeling and subtraction by codebook construction. Proc. Int. Conf. Image Processing, Singapore, 3061-3064. http://dx.doi.org/10.1109/ICIP.2004.1421759

McKenna Setal. (2000). Tracking groups of people. *Computer Vision and Image Understanding*, 80(1), 42-56.

Table 1. Color distortion degree

L(RGB)	D(RGB)	CDD
(10,10,10)	(8,10,12)	4/30
(200,200,200)	(198,200,202)	4/600

Table 2. Parameter setting

ε_1	ε_2	α	β	N
0.97	0.991	0.45	1.57	30

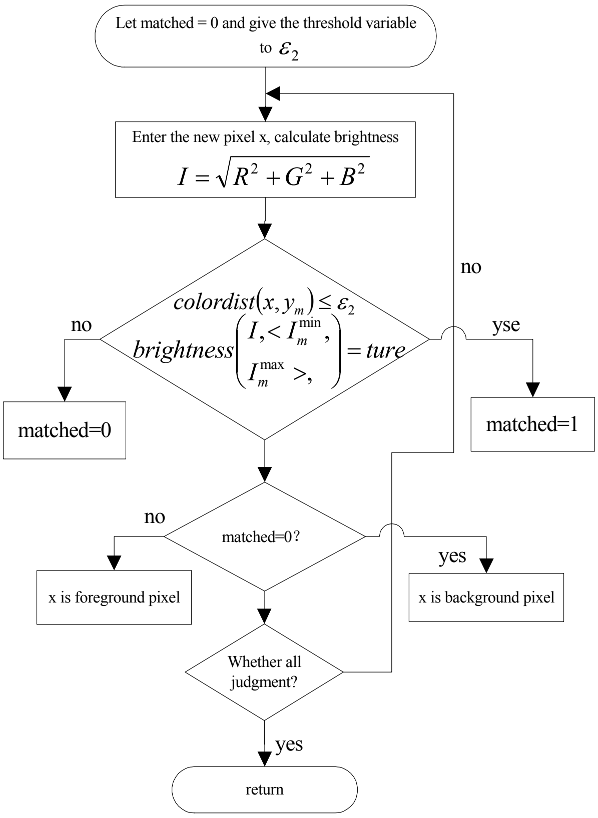
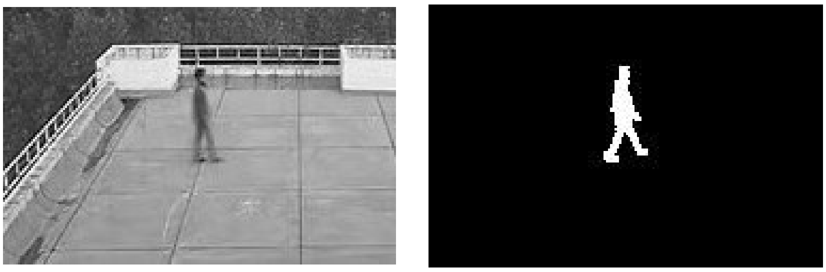


Figure 1. Flow chart of algorithm



a. outdoor video screenshot b. outdoor target detection results

Figure 2. Outdoor contrast experiment results

a. indoor video screenshot b. outdoor target detection results

Figure 3. Indoor contrast experiment results