Background Subtraction Based on a Robust Consensus Method

Hanzi Wang and David Suter

Institute for Vision Systems Engineering
Department of Electrical and Computer Systems Engineering
Monash University, Clayton Vic. 3800, Australia
{hanzi.wang, d.suter}@eng.monash.edu.au

Abstract

Statistical background modeling is a fundamental and important part of many visual tracking systems and of other computer vision applications. In this paper, we presents an effective and adaptive background modeling method for detecting foreground objects in both static and dynamic scenes. The proposed method computes SAmple CONsensus (SACON) of the background samples and estimates a statistical model per pixel. Numerous experiments on both indoor and outdoor video sequences show that the proposed method, compared with several state-of-the-art methods, can achieve very promising performance.

1. Introduction

Background modeling is crucial and fundamental for many computer vision applications. There are numerous background models appearing in the literature in recent years. For example, Pfinder [1] assumes that the pixels over time window at a particular image location are Gaussian distributed. W^4 [2] models the background by maximum and minimum intensity values, and the maximum intensity difference between consecutive frames in the training stage. Although these methods work well when the background includes only a static scene, they may fail if background pixels are multi-modal distributed.

Several methods have been proposed to deal with multiple-modal distributed background pixels. For example, Wallflower [3] employs a linear Wiener filter to learn and predict background changes. Tracey [4] models foreground and background by codebook vectors; CB [5] which quantizes and compresses background samples at each pixel into codebooks; In [6], "cooccurrence" of image variations at neighboring image blocks is employed for modeling a dynamic background. The pixel-level Mixture of Gaussians (MOG) background model [7, 8] is used to model multiple-modal distributed backgrounds. Elgammal et. al. [9] presented a non-parametric background model to model dynamic background.

In this paper, we propose a new efficient background modeling method, SAmple CONsensus (SACON), and we apply it to background subtraction. SACON gathers background samples and computes sample consensus to estimate a statistical model at each pixel. SACON is easy to perform but highly effective in background modeling and subtraction. Numerous quantitative experiments show the advantages of SACON over several other popular methods in background modeling/subtraction.

The organization of the remainder of this paper is as follows: in section 2, we present the SACON method. In section 3, we propose a framework for applying SACON to background subtraction. Experiments showing the advantages of our method over other popular methods are provided in section 4. We summarize in section 5.

2. The method of SACON

We now propose our background model: Sample Consensus:

Let N be the number of background samples at each pixel. For an observation at pixel m at time t—$x_t(m)$, we need to classify it into either a background pixel or a foreground pixel according to the background samples $\{x_i(m) \mid i = 1,...,N\}$ at that pixel location. Each observation $x_t = (x_t^{C_1},...,x_t^{C_k})$ has k channels (e.g., in RGB color space, each observation is expressed by three channels of R, G, B). Let:

$$\Gamma_i^c(m) = \begin{cases} 1 & if \left| x_i^c(m) - x_t^c(m) \right| \le T_r \\ 0 & otherwise \end{cases} \quad (1)$$

where T_r is a constant value reflecting the error tolerance.

For each observation at time t, we form a binary mask B_t *defining the sample consensus classification* (with "one" for a background pixel and "zero" for a foreground pixel):

$$B_t(m) = \begin{cases} 1 & \sum_{i=1}^{N} \Gamma_i^c(m) \ge T_n \ \forall c \in \{C_1,...C_k\} \\ 0 & otherwise \end{cases} \quad (2)$$

where T_n is a value thresholding the number of data points that are within the error tolerance T_r of a mode.

T_n should reflect both the sample size N and the error tolerance T_r: when the larger value N is and the larger value T_r is set, the larger value T_n should be and

vice versa. Thus, T_n can be approximately set to $\tau\ T_r N$ where τ is a constant.

3. Framework

In this section, using SACON as a core step, we present a complete framework for background subtraction.

The major components are shown in Figure 1. Key additional elements include a Foreground Mask (FM) and a Time Out Map (TOM). FM is used to mark foreground (FG) pixels. TOM is used to record the consecutive times that a pixel is marked as a FG pixel. As Figure 1 shows, the proposed framework mainly contains three phases. In the first phase, the adjacent frame difference method [3] is employed to extract possible foreground pixels. However, if the background includes dynamic parts, these pixels may belong to the background. This issue will be resolved in the next phase. In the second phase, we feed the possible foreground pixels, and also the pixels whose TOM values are high, as well as the background samples, to SACON. The output is the detected foreground (FG) pixels. Only the FM of the pixels from the first stage is updated at this stage. However, there may be holes inside the foreground regions. In the third phase, we validate the pixels inside the holes of the detected FG regions, and we update the background samples and the TOM.

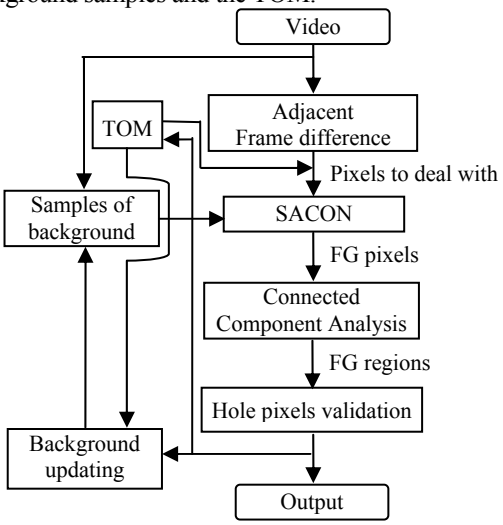


Figure 1. Block diagram of the complete framework.

3.1 Suppressing the influence of shadow

We employ the normalized (r, g, I) color space similar to [9, 10, 11], where r=R/(R+G+B), g=G/(R+G+B), I=(R+G+B)/3. (Note: in our case, we scale r, g and I to be in the range [0, 255]).

Let (r_b, g_b, I_b) be the observed value of a background pixel x_b and (r_t, g_t, I_t) be the observed value at this pixel (i.e., x_t) in frame t. The shadow can be suppressed if the following two conditions hold:

$$\begin{cases} \left| x_t^c - x_b^c \right| \le T_r & \forall c \in \{r, g\} \\ \beta \le x_t^c / x_b^c \le \gamma & c \in \{I\} \end{cases} \tag{3}$$

where β, γ are constant and are chosen empirically (in our case, we set β=0.6, γ=1.5).

If the background is dynamic and includes multiple modes, we compute the sample consensus by checking how many data points of the N background samples satisfy Equation (3), if the number is larger than T_n, we label $B_t(m)$ in Equation (2) with value 1.

We find that when the intensity I is small, the estimated normalized color r, and g can be very noisy. In this case, we use I only instead of using (r, g, I).

3.2 Validation of pixels inside holes

When a foreground object has similar color to the background scene, the foreground pixels may be wrongly labelled as background pixels. Let us consider Equation (3), although $\beta \le x_t^I / x_b^I \le \gamma$ can be used to suppress shadows, the intensity information is also "damaged" to some extent. If the chromaticity component of foreground pixels is similar to that of background pixels, the difference of the intensity part is large but still within the range of $\beta \le x_t^I / x_b^I \le \gamma$ (this is notable especially when x_b^I is large), the pixels are wrongly marked. We use a validation procedure to recheck the pixels inside the holes. For these pixels inside the holes, we use $\left| x_t^I - x_b^I \right| \le T_I$ (where we set T_I to 7).

3.3 Background updating

We use a selective update mechanism to update the background samples. To incorporate the moved/inserted background object or static foreground object into the background model, we use a Time Out Map (TOM). Let $\mathrm{TOM_t}(m)$ be the TOM at pixel m at frame t. We have:

$$\begin{cases} TOM_t(m) = TOM_{t-1}(m) + 1 & if\ B_t(m) = 0 \\ TOM_t(m) = 0 & if\ B_t(m) \ne 0 \end{cases} \tag{4}$$

We update the background samples at both pixel level and blob level. At pixel level, when a pixel of an object has remained in place too long (i.e., the value of TOM of the pixel is high), that pixel will be assigned to the background. For pixels whose 4-connected pixel numbers are high, we treat these pixels at blob level. If an object (i.e., a blob) is static, we increase the TOM value of all pixels of that object by one; otherwise, we set the TOM values of the pixels to zero. If the TOM value of an object is high, we add the all pixels of the object to the background samples.

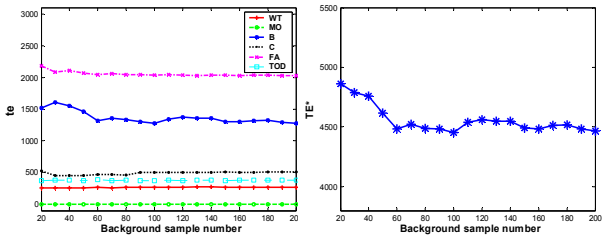
4. Experiments

In this section, we evaluate our method using the *Wallflower* sequences (Toyama et. al. [3]) and compare with several other methods. A brief description of the *Wallflower* image sequences follows: **Moved Object (MO):** A person enters into a room, makes a phone call, and leaves. The phone and the chair are left in a different position. **Time of Day (TOD):** The light in a room gradually changes from dark to bright. Then, a person enters into the room and sits down. **Light Switch (LS):** A room scene begins with the lights on. Then a person enters the room and turns off the lights for a long period. Later, a person walks in the room, switches on the light, and moves the chair, while the door is closed. The camera sees the room with lights both on and off during the training stage. **Waving Trees (WT):** A tree is swaying and a person walks in front of the tree. **Camouflage (C**): A person walks in front of a monitor, which has rolling interference bars on the screen. The bars include color similar to the person's clothing. **Bootstrapping (B):** The image sequence shows a busy cafeteria and each frame contains people. **Foreground Aperture (FA):** A person with uniformly colored shirt wakes up and begins to move slowly.

For the evaluation of performance against each image sequence, we use three terms: False Positive (FP), False Negative (FN), and total error (*te*). FP is the number of background pixels that are wrongly marked as foreground; FN is the number of foreground pixels that are wrongly marked as background; *te* is the sum of FP and FN for each image sequence. For the evaluation of overall performance, we use TE (the sum of total error for all seven image sequences) and TE* (the sum of total error excluding the light switch image sequence). For each resulting image, we eliminated the foreground pixels whose 4-connected foreground pixels number less than 8.

Figure 2 and Table 1 show the results obtained by SACON, and several other state-of-the-art methods. From Figure 2 and Table 1, we can see that our method achieves the most accurate overall performance on TE and TE* among the competitive methods. SACON also obtains the best results of total error (*te*) for image sequences of MO, TOD, WT, C and B. For the LS sequence, *Wallflower* (which maintains the background at frame level) achieves the least total error. For the FA sequence, Wallflower achieves the most accurate result based on the measure *te*. However, the authors of [3] used a region-level processing as a post-processing step. In contrast, SACON validates pixels inside the foreground holes at pixel level only.

Because our method is based on sample consensus, we also investigate the influence of the background sample number *N* on the results. We use the same set of image sequences but various numbers of background samples, and *N* changes from 20 to 200, with interval 10. From Figure 3 (a), we can see that the influence of the sample number is small on most image sequences, except for image sequence B, where there is relatively large fluctuation in total error (*te*). Figure 3 (b) shows the overall performance on various *N*. We can see that when *N* is larger than 50, TE* remains relatively stable. When *N* is less that 50, TE* increases with the decrease of *N*.



(a) (b)

Figure 3. Plot of (a) total error (*te*) and (b) TE* vs different background sample number *N*.

We perform our method in MATLAB language (interfaced with C MEX) on a Laptop with Pentium M processor 1.6GHZ. The averaged processing time for the seven *Wallflower* image sequences (120x160 color images) is about 10 fps when we set *N* equal to 100, and 6 fps when *N* is set to 200.

5. Conclusion

In this paper, an effective and robust background modeling method (SACON) is proposed. An effective framework is also proposed to apply SACON to background subtraction. The proposed method has been tested and validated by a significant number of experiments. SACON has proved to be robust in various environments (including indoor and outdoor scenes) and different types of background scenes such as dynamic or static scenes. We also numerically evaluate the performance of SACON with the Wallflower benchmarks and compare its results with those of several popular background modelling methods. The comparisons show that SACON achieves promising results.

Acknowledgements

This work is supported by ARC grant DP0452416.

References

1. Wren, C.R., et al., *Pfinder: real-time tracking of the human body*. PAMI, 1997. **19**(7): p. 780-785.
2. Haritaoglu, I., D. Harwood, and L.S. Davis, *W4: Real-Time Surveillance of People and Their Activities.* PAMI, 2000. **22**(8): p. 809-830.

3. Toyama, K., et al. *Wallflower: Principles and Practice of Background Maintenance*. ICCV. 1999. p. 255-261.
4. Kottow, D., M. Koppen, and J. Ruiz-del-Solar. *A Background Maintenance Model in the Spatial-Range Domain*. in *Workshop on Statistical Methods in Video Processing*. 2004.
5. Kim, K., et al., *Real-time Foreground-background Segmentation Using Codebook Model.* Real-Time Imaging, 2005. **11**(3): p. 172-185.
6. Seki, M. and T.F. Wada, H.Sumi, K. *Background Subtraction Based on Cooccurrence of Image Variations*. CVPR. 2003. p. 65-72.
7. Stauffer, C. and W.E.L. Grimson. *Adaptive Background Mixture Models for Real-time Tracking*. CVPR. 1999 p. 246-252.
8. Friedman, N. and S. Russell. *Image Segmentation in Video Sequences: A Probabilistic Approach*. in *Proc. Conf. on Uncertainty in Artificial Intelligence*. 1997. p. 175-181.
9. Elgammal, A., D. Harwood, and L.S. Davis. *Non-parametric Model for Background Subtraction*. ECCV. 2000. p. 751-767.
10. Mittal, A. and N. Paragios. *Motion-Based Background Subtraction using Adaptive Kernel Density Estimation*. CVPR. 2004. p. 302-309.
11. Mittal, A. and L.S. Davis, *M_2 Tracker: A Multi-View Approach to Segmenting and Tracking People in a Clutter Scene.* IJCV, 2003. **51**(3): p. 189-203.

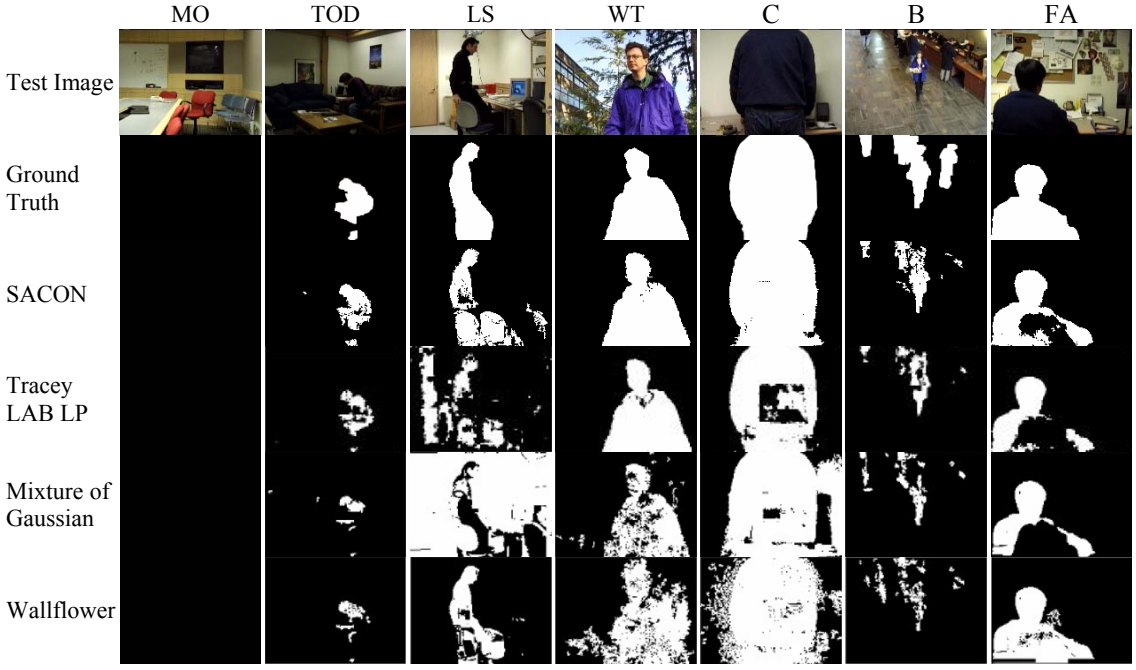


Fig. 2: Experimental results by several methods on the *Wallflower* benchmarks. The top row shows the evaluated frames of each image sequences; the second row shows the hand-segmented ground truth; the third row shows the results of SACON. The fourth row shows the results of Tracey reported in [4]; the fifth to the sixth rows show the results reported in [3].

Methods	ET	MO	TOD	LS	WT	C	B	FA	TE	TE*
SACON	f. neg.	0	236	589	41	47	1150	1508	6087	4467
	f.pos.	0	147	1031	230	462	125	521		
	te	0	383	1620	271	509	1275	2029		
Tracey LAB LP	f. neg.	0	772	1965	191	1998	1974	2403	12035	8046
	f. pos.	1	54	2024	136	69	92	356		
	te	1	826	3989	327	2067	2066	2759		
Mixture of Gaussian	f. neg.	0	1008	1633	1323	398	1874	2442	27053	11251
	f. pos.	0	20	14169	341	3098	217	530		
	te	0	1028	15802	1664	3496	2091	2972		
Wallflower	f. neg.	0	961	947	877	229	2025	320	11478	10156
	f. pos.	0	25	375	1999	2706	365	649		
	te	0	986	1322	2876	2935	2390	969		

Table 1: Experimental results by different methods on *Wallflower* benchmarks.