Proceedings of the 6th World Congress on Intelligent Control
and Automation, June 21 - 23, 2006, Dalian, China

A Novel Binary Image Filtering Algorithm Based on Information Entropy[*]

Junyi Zuo, Chunhui Zhao, Quan Pan and Wei Lian

Department of Automation
Northwestern Polytechnical University
Xi'an, China, 710072

zuojunyi@163.com

***Abstract* - The results of moving object detection in image sequences are often represented as binary image sequences containing both masks of moving objects and false positives, which make it necessary to quickly eliminate false positives and preserve the interested moving objects as completely as possible online. Traditional approaches such as morphological filter and area threshold filter, sometimes, are unfit for the job because of unsatisfactory results and heavy computational load. In light of the fact that foreground pixels density caused by moving objects is higher than that of caused by disturbances, we propose a novel binary image filtering method based on discrete information entropy(BIFBE) for the task. Experimental results show that BIFBE is more effective and has less computational load than traditional methods.**

Index Terms - Binary image filtering, Information entropy, Moving object detection.

I. Introduction

Detecting moving objects in image sequences captured by static cameras is one of the fundamental tasks in computer vision because it provides information for higher level operations such as objects tracking and identification etc. In application, the results of moving object detection are influenced by both video gathering devices and environments. Especially in outdoor scene, considerable amounts of disturbing motions exist in the background of video sequence, such as fountains, ocean ripples, raindrop or snowflake in the air, swaying trees in the wind, which strongly influence the detection accuracy. In addition, electronic thermal noise and camera shaking are another two sources of disturbance. To address these challenges, many detecting algorithms have been proposed, and they can be divided into several categories: optical flowing, temporal differencing, background subtraction or the combinations of them. Among them, background subtraction, the key step of which is building a proper background model that can reflect the changes of real scene, is most extensively researched in the past two decades and many background models or their variations have been proposed, such as temporal average adaptive background model (TAM) [1], single Gaussian model [2], Gaussian mixture model (GMM) [3] [4], nonparametric kernel density estimation background model [5] [6] etc. Different algorithm has different computational complexity and different detecting accuracy. Generally speaking, high performance algorithms usually have the drawback of heavy computational load and vice versa, so we have to make a compromise between them. Furthermore, approaches with higher detection accuracy, such as nonparametric model, may fail to give satisfactory results in the case that pixel values have more complex distributions, All these considerations make it necessary for us to apply more powerful post-processing strategy to make up for the weakness of existing detection algorithm.

Morphological filter [7] and area threshold filter [8] have been widely applied to the post-processing of binary image sequences in literatures. The former uses open operation to eliminate false alarms and may distort object shapes if its structure elements are not properly selected. The latter signs all the connected regions and calculates the area of each region, and then eliminates the foreground points in the regions whose areas are less than a given threshold value. The computation speed of this method heavily depends on the results of motion detection. So this approach can not meet the needs for real time application when clutter level is high in binary image sequence.

Usually, areas in binary image occupied by real moving objects have higher foreground point density than areas occupied by background. According to this rule which is hold in most cases, in this paper we propose a novel binary image filtering method based on information entropy (BIFBE), it aimed at eliminate false alarms via modeling the density of foreground pixels and constructing discriminant to determine which foreground pixel should be removed and which should not be. The algorithm was carried out in two steps: firstly, when a new frame of binary image was obtained, we divide it into some small sub-regions. For each sub-region, we apply a pre-designed discriminant to determine which sub-region belongs to background, and which belongs to moving objects. In this step, most of the false alarms can be eliminated, which makes it convenient for us to focus our attention on regions which are more likely to belong to moving objects. Yet, object shapes may be distorted in this step because some pixels on objects, especially on object borders, could be eliminated also. So in next step, we retrieve them by using the same discriminant to search the border of the regions preserved in step 1.

II. Discrete Information Entropy

[*] This work is supported by Natural Science Foundation of China, #60372085。

The following information theoretic definitions and inequalities are fundamental to develop the most basic ideas of this paper.

Given a discrete probability distribution:

$$[X, P_X] = [x_k, p_k \mid k = 1,2,\cdots,K] \quad (1)$$

The information entropy of discrete random variable X is defined as:

$$H(X) = -\sum_{k=1}^{K} p_k \log p_k \quad (2)$$

Because of $\lim_{x \to 0} x \log x = 0$, so we define $0 \log 0 = 0$ in (2). The logarithms are taken in base 2 and entropy is expressed in bits. Discrete information entropy is nonnegative and has an extremum.

Nonnegativity: $H(X) \geq 0$,

if and only if $\max\{p_i \mid i = 1,2\cdots K\} = 1$, then $H(X) = 0$

extremum: $\log K \geq H(X)$,

if and only if $\forall \ i \in 1,\cdots K, \ p_i = 1/K$, then $H(X) = \log K$.

For discrete random variables, larger entropy means larger uncertainty which depends on the range and uniformity of the distribution, i.e., the wider the distribution range is, the larger the entropy is. If ranges of two distributions are equal to each other, the more uniform the distribution is, the larger the entropy is. This can be illustrated by Fig. 1.

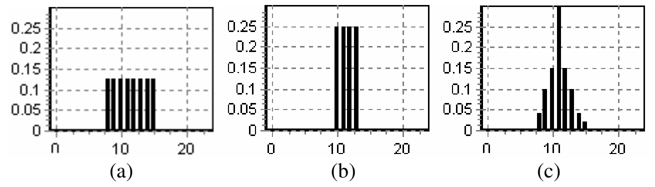

Fig.1 The relationship between discrete distributions and their entropies. The range of distribution in (a) is wider than that in (b), so (a) has larger entropy, the discrete distributions of (a) and (c) have the same range, (a) obeys uniform distribution, and (c) obeys normal distribution approximately. So the former has larger entropy.

III. CONSTRUCTING DISCRIMINANT

For the convenience of illustrating our algorithm, without loss of generality, we assume that white pixels stand for foreground pixels in original binary image sequences and black ones for background pixels. The white pixels maybe caused by real moving objects or maybe caused by disturbances mentioned in introduction. Our destination is designing an efficient algorithm to eliminate white pixels caused by false alarms and preserve the ones caused by moving objects as completely as possible.

Assume in a sub-region size of $n \times n$ pixels in binary image, the number of white pixels in it is large than 0 (if it equals to 0, we will let it along), we project white pixels onto horizontal and vertical axes respectively, and then normalize them to get the white pixels discrete distributions along the two axes:

$$[X, P_X] = [x_k, p_{xk} \mid k = 1,2,\cdots n] \quad (3)$$

$$[Y, P_Y] = [y_k, p_{yk} \mid k = 1,2,\cdots n] \quad (4)$$

Entropies of X and Y denoted as $H(X)$ and $H(Y)$ respectively, are measures of uncertainties of white pixel value's distributions along these two directions, i.e., for a sub-region with fixed number of white pixels, $H(X)$ with larger value means white pixels in the sub-region have a larger distribution range and more uniform distribution along horizontal axis. The same conclusion is for $H(Y)$. Obviously, $H(X) \in [0, \log n], H(Y) \in [0, \log n]$. In order to preclude the influence of n, we normalize them according to nonnegativity and extremum of entropy described above:

$$H_N(X) = -\sum_{k=1}^{n} p_{yk} \log p_{yk} / \log n \quad (5)$$

$$H_N(X) = -\sum_{k=1}^{n} p_{xk} \log p_{xk} / \log n \quad (6)$$

here, $H_N(X) \in [0,1], H_N(Y) \in [0,1]$

We denote the number of white pixels as T, and define $r = T / n^2$, so r is the proportion between the white pixel number and the total pixel number in a sub-region size of $n \times n$ pixels. When T doesn't vary with time, The T white pixels spread more widely in the sub-region, normalized entropies along two directions will be larger at the same time. So their product:

$$f(p) = \frac{-\sum_{k=1}^{n} p_{yk} \log p_{yk}}{\log n} \times \frac{-\sum_{k=1}^{n} p_{xk} \log p_{xk}}{\log n} \quad (7)$$

has larger value too. Obviously, $f(p) \in [0,1]$. When $r \approx 0.5$, we can use (7) as discriminant. The smaller the $f(p)$ is, the more densely the white pixels spread, and vice versa. So given a threshold value $th \in (0,1)$, if $f(p) \geq th$, then white pixels in the sub-region will be regarded as false alarms and will be eliminated. if $f(p) < th$, they will be regarded as pixels belonging to moving objects and will be preserved.

Unfortunately, when $r - 0.5$ or $r - 0$, the discriminant expressed in (7) doesn't work well. For example, if all the pixels in the sub-region are white, $f(p)$ reach the maximum, i.e., $f(p) = 1$, in this case, white pixels in the sub-region will be removed incorrectly. In addition, assume just only one pixel in one sub-region is white, $f(p)$ reach the minimum, i.e. $f(p) = 0$ and all white pixels in the sub-region will be preserved incorrectly. So whether the sub-region belongs to moving objects or background depends on both $f(p)$ and r, a revising item $\varphi(r)$ should be introduced to (7) to modify the discriminant. Here we define:

$$\varphi(r) = \left(\frac{1 - r}{r}\right)^{\alpha} \quad (8)$$

where, a is a constant power which determine how much r influences the final decision. Obviously, $\varphi(r)$ satisfies three relations:

$$\lim_{r\to 0^+}\varphi(r)=+\infty,\quad \lim_{r\to 0.5}\varphi(r)=1,\quad \lim_{r\to 1^-}\varphi(r)=0$$

So the final discrinant expressed as:

$$d(p,r)=\frac{-\sum_{k=1}^{n}p_{yk}\log p_{yk}}{\log n}\times\frac{-\sum_{k=1}^{n}p_{xk}\log p_{xk}}{\log n}\times\left(\frac{1-r}{r}\right)^{\alpha} \qquad (9)$$

from (9), we can see when $r\approx 0.5$, the final decisions are mainly determined by (7); when r — 0 or r — 1, the final decision are mainly determined by (8).

There is a special case need to be careful. When $f(p)=0$, the revising item $\varphi(r)$ in (9) is invalidated. In this case, we can conclude:

$$-\sum_{k=1}^{n}p_{yk}\log p_{yk}=0 \text{ or } -\sum_{k=1}^{n}p_{xk}\log p_{xk}=0$$

From the properties of discrete entropy, we know the number of white pixels in the sub-region is not more than n and all of them lie in the same row or column in this situation, i.e., we can't find two white pixels in the sub-region which lie in different row and different column at the same time. In this case, r is not more than $1/n$, so we classify the sub-region as background directly. Such approximation scheme can result in inaccuracy results sometimes, so in step2, we execute a refine procedure to recover the legal white pixels.

IV. Implementation of BIFBE

The algorithm was carried out in two steps.

The first step:

Divide original binary image, size of $m\times l$ pixels, into $(m/n)\times(l/n)$ small sub-regions size of $n\times n$ pixels. Here, m,l can be divided exactly by n. For each sub-region $R_{i,j}$, according to (7), we get $f_{R_{i,j}}(p)$.

if $f_{R_{i,j}}(p)=0$, then $R_{i,j}\in A_b$

if $f_{R_{i,j}}(p)\neq 0$, according to (9), we calculate $d_{R_{i,j}}(p,r)$

if $d_{R_{i,j}}(p,r)<th$, then $R_{i,j}\in \overline{A_b}$

if $d_{R_{i,j}}(p,r)\geq th$, then $R_{i,j}\in A_b$,

finally, for any $R_{i,j}$, if $R_{i,j}\in A_b$, then change the white pixels in $R_{i,j}$ into black pixels.

where A_b and $\overline{A_b}$ denote areas occupied by background and moving objects respectively. th is a threshold value.

The Second step:

In the first step, we have signed each sub-region a tag indicating the sub-region belongs to background or moving objects and most of the false alarms will have been eliminated, at the same time some parts of object, especially in the neighbourhoods of object borders, will be cut off also, which make the object shapes unstable from frame to frame. In step 2, we will apply refine procedure to recover them and stabilize the object shapes.

The complete refine procedure includes the following substeps:

Substep1: Scan all the $(m/n)\times(l/n)$ sub-regions, if the tag of (i,j) th sub-region $R_{i,j}$ indicates that $R_{i,j}$ belongs to background, then let it along and continue to scan next sub-region $R_{i,j+1}$, else go to substep2.

Substep2: If the $(i-1,j)$th sub-region $R_{i-1,j}$ which lie above of $R_{i,j}$ belongs to background, then Move $R_{i,j}$ up a pixel line to get a new sub-region. If the new sub-region belongs to background, then let it along, at the same time stop moving the new sub-region up. If the new sub-region belongs to foreground, then recover pixels at the top line in the new sub-region with original values, and continue to move the new sub-region up a line to recover object shapes in the same way. Obviously, we can move the sub-region up $n-1$ times at most.

Substep3: Just like substep 2, we continue to recover the object shapes by moving $R_{i,j}$ to other 7 directions such as right, left, top left, top right etc.

The computational load for applying the algorithm described above directly is very high because there is large number of logarithmic operations. Fortunately, the parameter n is fixed when application. The set containing probable value of $-p_i\log p_i$ which is necessary for calculation sub-region's $H(X)$ and $H(Y)$, is independent on the number of white pixels and their distribution in the sub-region. That is to say, every necessary item $-p_i\log p_i \quad i=1,2,\cdots n$ used for calculation $H(X)$ and $H(Y)$ must come from the following set Φ

$$\Phi=\{x\mid x=-\frac{i}{j}\log\frac{i}{j},\quad i=0,1,2,\cdots n,\quad j=1,2\cdots n^2\}\cup\{0\} \qquad (10)$$

Obviously, the cardinality of Φ is limited. In order to avoid online calculating entropies, we calculate every elements of Φ and store them in a data table size of $(n^2+1)\times(n+1)$ in advance and recall them online if necessary.

Tab.1 The structure of offline data table when $n=3$

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	/	/	/
1	/	*	*	*	*	*	*	*	/	/
2	/	/	*	*	*	*	*	*	*	/
3	/	/	/	*	*	*	*	*	*	*

Without loss generality, in order to illustrate how to construct the data table, we take example for $n=3$ (see Tab1). The index of first row stands for the probable number of white pixels in the sub-region, while the index of first column stands for the probable number of white pixels in a certain row/column of the sub-region. Obviously, some combinations of row index and column index will not occur and the data in corresponding grids which have been filled with "/" in Tab.1 will not be recalled for ever in application. If the row index and column index of a"*" are i and j respectively, then the data represented by the "*"equals to - $(i/j)\log(i/j)$. With the help of online data table, we easily avoid large amounts of logarithmic operations.

V. EXPERIMENTAL RESULTS AND ANALYSIS

In order to validate the proposed algorithm, two different types of scenes were considered, the first one is a challenging traffic scene which involves shaking camera, swaying leaves and their dynamic shadows, a frame of the scene is shown in the left of Fig.2 (a). The right of Fig.2 (a) shows one frame of scene 2 where there is a clump of bushes swaying in the wind. Temporal average adaptive background model [1] whose mean value learning rate and threshold value is set as 0.002 and 30 respectively was applied to detect moving objects; the detecting results are shown in Fig.2 (b), obviously, the false alarm rates are high.

There are 3 parameters needed to choose for the execution of BIFBE algorithm. The first one is the size of sub-region denoted as n. Choosing a too large n will cause the object shapes unstable from frame to frame and tend to miss small objects; while choosing a too small n will cause the entropy subject to the influence of the local density of white pixels. The constant power α determines how much r influences the final decision. We assume that the sub-region either belongs to background or belongs to moving objects when $r = 0.5$, so the threshold value th must be within (0, 1). In our experiment, the three parameters are set as follows: $n = 6, \alpha = 2, th = 0.4$. Structure element was set as 2×2 for morphological operation and the threshold value was set as 20 for area threshold filter.

Results of post-processing of several approaches are shown in Fig. 2 as well, we can see morphological close operation can fill most of the holes in moving objects, but fail to remove false alarms, so it is unfit for the job and will be out of consideration later. On the contrary, morphological open operation can remove most of the false alarms but distort object shapes slightly and sometimes make holes in objects even bigger. From Fig3 (b) we can see there are still some shape distortedness even if we use small structure element. So selecting proper structure element is important for morphological filter, big structure elements usually can remove more false alarms but tend to distort the object shapes, on the contrary, small structure elements are in favor of preserving object shapes but do poor job in removing false alarms. It's difficult to make a compromise between them. Area threshold filter can remove most false alarms as well and preserve object shapes. When bigger threshold value is chosen, it can remove more false alarms, but when processing high clutter level binary image sequence just like Fig2 (b), heavy computational load makes it unfeasible for real time application. In contrast, BIFBE can do a better job both in removing false alarms and in preserving object shapes with low computational load , this can be seen from Fig2 (f) and Fig3 (d).

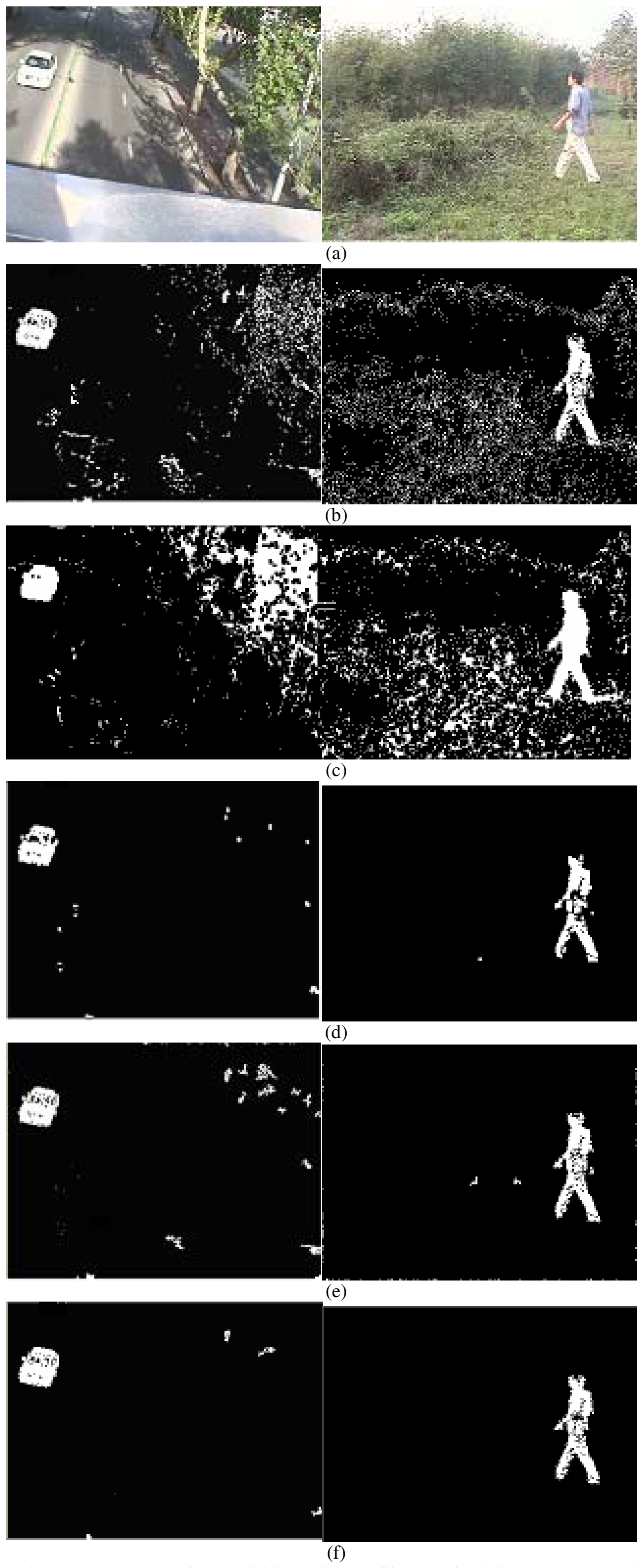

Fig.2 Comparison of several binary image filter methods in two scenes. The right column is scene 1 and its results, the right column are scene 2 and its results. (a) two frame in original video sequences.(b)the detecting results by applying temporal average model to the two scenes .(c)the results of post-processing using morphological close operation (d) the results of post-processing using open operation.(e) the results of post-processing using area threshold filter.(f) the results of post-processing using BIFBE

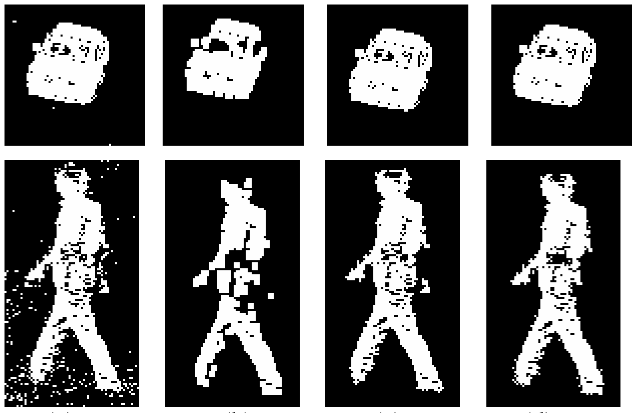

Fig.3 BIFBE and area threshold filter are superior to morphological filter in preserving object shapes. (a)The detecting results of objects without applying post-processing. (b) The result by applying morphological open operation. (c) The result by applying area threshold. (d) The result by applying BIFBE

In order to compare the speeds of the three algorithms, we tested them on a 2.0GHz Intel Pentium 4 processor with 256MB RAM. Frame rate of original video, size of 320×240 pixels, is 25 fps. We applied temporal average background model to scene 2 to obtain binary image sequences, and then use morphological open operation, area threshold, and BIFBE for post-processing. BIFBE runs at 25fps that is the original video frame rate, and CPU usage is below 50%; Morphological open operation runs at 24fps and CPU usage reaches 100%; area threshold filter runs at 3fps, the CPU usage reaches 100%, so it is unfit for area threshold filter to process binary image sequence like Fig2 (b) directly.

Further research shows that BIFBE can process binary image sequences with different clutter levels almost at the same speed. While the processing speed of area threshold filter is terribly affected by cluster levels, that is, it can process low clutter levels binary image sequence at a decent speed, but its speed become unacceptable for high clutter level ones such as Fig2 (b). So the above analysis can explain the phenomenon found in our experiment: Despite that GMM's processing speed is much lower than TAM, GMM's false alarm rate is lower than TAM for those complex scenes such as Fig2 (a), which make the post-processing by area threshold filter become rather easier. So as a result, the combination of GMM and area threshold filter turns out to be more efficient than the combination of TAM and area threshold filter when overall execution time was considered.

VI. Conclusion

For the task of binary image sequence filtering, the BIFBE algorithm proposed in this paper is superior to morphological filter and area threshold filter when computational load is considered. As for keeping object shapes, BIFBE and area threshold filter perform better than morphological filter sometimes. Although BIFBE and morphological filter are not as good as area threshold filter in preserving small objects sometimes, yet if all the factors are taken into account, BIFBE deserves to be regarded as a high performance, efficient binary image filter algorithm.

References

[1] N. Friedman and S. Russell, "Image segmentation in video sequences: probabilistic approach," *in Proc. 13th Conf. Uncertainty in Artificial intelligence*, 1997.

[2] C. R. Wren, A. Azarbayejani, T. Darrell, and A. Pentland, "Pfinder: Real-time tracking of the human body." *IEEE Trans. on PAMI*, vol. 19, no. 7, pp. 780.785, 1997.

[3] C. Stauffer and W. Grimson," Adaptive background mixture models for real-time tracking," *IEEE Conf. on CVPR*, pp. 246-252, 1999.

[4] Zoran Zivkovic. "Improved Adaptive Gaussian Mixture Model for Background Subtraction" *In Proc. ICPR*, 2004.

[5] A. Elgammal, D. Harwoodl and Larry. Davis, "Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance," *Proceedings of IEEE*. Vol. 90, pp. 1152-1163 2002.

[6] A. Mittal and N. Paragios. "Motion-Based Background Subtraction using Adaptive Kernel Density Estimation," *IEEE Conf. on CVPR*, vol. 2, pp. 302–309, 2004.

[7] Jianpeng Zhou and Jack Hoang. "Real Time Robust Human Detection and Tracking System," *IEEE Conf. on CVPR,* 2005.

[8] C. Conaire, E. Cooke, N. O'Connor, N. Murphy, and A.Smeaton, "Background Modeling in Infrared and Visible Spectrum Video for People Tracking," *IEEE Conf. on CVPR,* 2005.