Automatic threshold decision of background registration technique for video segmentation

Yu-Wen Huang, Shao-Yi Chien, Bing-Yu Hsieh, and Liang-Gee Chen

DSP/IC Design Laboratory, Graduate Institute of Electronics Engineering
National Taiwan University
1, Sec. 4, Roosevelt Road, Taipei 106, Taiwan

ABSTRACT

Background registration technique is useful to solve still object problem and uncovered background problem for video segmentation. However, it is hard to automatically decide the threshold of the frame difference for background registration to make it more feasible for real-time applications. Many previous works made efforts on automatic threshold decision for change detection. In this paper, we propose a new method of automatic threshold decision algorithm in a totally different viewpoint. Not only change detection but also the quantization effect in discrete domain is concerned. A Gaussianity test is first applied to find the standard deviation of Gaussian noise from the camera. Then, the quantization effect in discrete domain is taken into consideration to derive the relation between the standard deviation and the optimal threshold value. A couple MPEG-4 sequences and experimental sequences are tested as examples. Simulation results show that the calculated threshold values are suitable for background registration to give good segmentation results.

Keywords: automatic threshold decision, background registration, change detection, video segmentation, object extraction.

1. INTRODUCTION

MPEG-4 [1] is the latest video coding standard and will be applied to many real-time applications, such as videophone, videoconference, and video camera. Since one of the key functions of MPEG-4 is content-based coding, the shape information is indispensable for shape coding, motion estimation/compensation, and texture coding [2]. Video segmentation is the methodology that extracts video objects from video sequences and provides MPEG-4 encoding systems with shape information. Therefore, video segmentation is the fundamental for content-based real-time MPEG-4 camera systems.

Despite of its importance, video segmentation still remains an unsolved problem. Many video segmentation algorithms have been proposed. In [3][4][5], a change detection based algorithm is proposed. Global motion compensation is first applied to compensate the camera motion, and scene change detection is used to reset the algorithm. Automatic threshold decision for change detection is described. The threshold value can be calculated as long as the standard deviation and the distribution model of camera noise are known. Boundary relaxation is adopted to depress the noise, and a temporal filter that can improve the temporal coherence of the objects is developed. Motion estimation is applied to eliminate the uncovered background. Finally, object masks are adapted to luminance edges. In [6][7], Canny edge operator, combined with change detection or a morphological motion filter, is used to extract a portion of object boundaries, followed by a filling technique and boundary refinement. In [8], instead of the temporal segmentation scheme like change detection, a spatial segmentation scheme is adopted. Each frame is first intra-segmented by watershed algorithm using multi-scale morphological gradient filter to deal with ramp edges and to alleviate the over-segmentation problem. Next, region merging and temporal tracking, which include weak edge detection and region merging, respectively, are used to get the final segmentation results. Color segmentation is another spatial segmentation algorithm. In [9][10], image intensities are classified into several clusters and homogenous regions are labeled by connected component algorithm. Then, regions can be gathered to form objects. In [11], a hybrid scheme is adopted. Watershed algorithm, as well as region merging, is applied. Change detection is combined to determine the object shape region by region. Region tracking is also used to increase the temporal coherence. In [12], a hybrid scheme is also adopted. Foreground regions are found by temporal segmentation based on change detection. Then, regions are created by spatial segmentation based on K-Means with connectivity constraints. Although the above algorithms give good segmentation results in some cases, their computation may be too high for real-time applications.

It is not easy to meet the real-time requirements for computationally expensive operations, such as motion estimation, K-means, watershed, boundary relaxation, …etc. Therefore, change detection is selected as the basic operation of our segmentation algorithm [13][14][15]. A background registration technique, combined with pre-processing and post-processing, is developed to solve the still object problem and uncovered background problem encountered by the conventional change detection. The implementation is optimized on SIMD architecture by sub-word parallelism and MMX instructions. In [16], predictive watershed is proposed to cooperate with background registration and to provide video objects with more accurate boundaries with reduced computational load, compared with watershed algorithm.

Change detection and background registration technique have been provided that they are useful for video segmentation. Although our previous works can quickly give good segmentation results under a fixed camera, the threshold value of frame difference for background registration is empirically obtained. In order to make the segmentation algorithm more promising for real-time applications, automatic threshold decision is urgently needed. In this paper, a new method of automatic threshold decision of background registration for video segmentation is proposed. It can be divided into two main parts: Gaussianity test and scaling factor mapping. First, a criterion can be applied in each frame to find the blocks with Gaussian-distributed frame differences. In fact, these blocks with Gaussian-distributed frame differences are very likely to be stationary background, and their frame differences are resulted from Gaussian camera noise. Therefore, the standard deviation of these blocks is computed. Probability theories and the quantization effect in discrete domain are also taken into consideration to derive the relation between the standard deviation of Gaussian camera noise and the optimal threshold value. Finally, the standard deviation can be easily mapped to an optimal threshold by a table look-up method. Simulation results show that the calculated thresholds are suitable for background registration to give good segmentation results.

The rest of this paper is organized as follows. An efficient video segmentation algorithm under a fixed camera is reviewed in Section 2. Automatic threshold decision of background registration is proposed in Section 3. Simulation results are shown in Section 4, and the conclusion is drawn in Section 5.

2. VIDEO SEGMENTATION ALGORITHM

The flowchart of our video segmentation algorithm is shown in Fig. 1. The basic idea is change detection. A background registration technique is developed to solve the still object problem and uncovered background problem. The background difference, which is the difference between current frame and background buffer, is combined with frame difference to determine the initial object mask, which is refined by the post-processing. Morphological gradient filter can be optionally turned on to suppress the influence of shadows. Predictive watershed can also be combined to get more accurate boundaries.

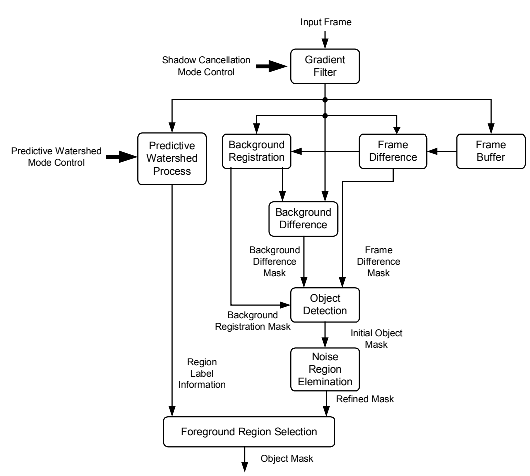


Figure 1. The flowchart of our video segmentation algorithm

2.1 Background registration technique

2.1.1 Frame difference mask generation

The absolute difference between current frame and the previous frame is calculated. Pixels with frame difference larger than a predefined threshold value are regarded as foreground while those with frame difference smaller than the threshold are viewed as background. Thus, a frame difference mask (*FDM*) can be generated as follows:

$$FDM(x,y)=\begin{cases}1 & \text{if } |I_t(x,y)-I_{t-1}(x,y)|\geq TH_{FD}\\ 0 & \text{else}\end{cases} \quad (1)$$

where I_t is the intensity of current frame at position (x, y), I_{t-1} is the intensity of previous frame at position (x, y).

2.1.2 Background registration

A stationary index (SI) is to register if the current pixel belongs to background. The registration process can be shown as:

$$SI_t(x,y)=\begin{cases}SI_{t-1}(x,y)+1 & \text{if FDM}(x,y)=0\\ 0 & \text{else}\end{cases} \quad (2)$$

$$Background_t(x,y)=I_t(x,y) \text{ and } SI_t(x,y)++ \quad \text{if } SI_t(x,y)=sth$$

where $SI_t(x, y)$ is the stationary index at time t, $Background_t(x, y)$ is the background buffer at time t, and sth is the stationary threshold. If a pixel is detected as unchanged in the past consecutive sth frames, it will be included into background buffer. Otherwise, the stationary index of the pixel will be reset to zero.

2.1.3 Background difference mask generation

This step generates a background difference mask (*BDM*) by thresholding the difference between the current frame and the background data stored in the background buffer. It can be described as the following equation:

$$BDM(x,y)=\begin{cases}1 & \text{if } |I_t(x,y)-Background_{t-1}(x,y)|\geq TH_{BD}\\ 0 & \text{else}\end{cases} \quad (3)$$

2.1.4 Initial object mask generation

If the background exists, that is, $SI(x, y)$ is larger than sth, we let object mask, $OM(x, y)$, equal to $BDM(x, y)$. Otherwise, $OM(x, y)$ is equal to $FDM(x, y)$. Table 1 lists the criteria for initial object mask generation. |FD| and |BD| mean the absolute frame difference and the absolute background difference, respectively. For the first two cases listed in table 1, the background is not yet registered, and the frame difference is used. In case 3 to case 6, the background is available, so the object mask is determined by background difference. In case 3 and 4, our results are the same as conventional change detection. However, conventional change detection makes wrong decisions in case 5 and 6. Conventional change detection is not capable of dealing with the still object problem (case 5). It loses tracking of still objects. The uncovered background is another problem (case 6) for conventional change detection. Motion estimation is adopted in [4][5] to solve the uncovered background problem. Pixels with large motion compensated errors are regarded as uncovered background. There are two main drawbacks of this method. One is that motion estimation is too computationally expensive, and the other is that if the object shape is changed, motion estimation cannot be precise, and the uncovered background cannot be correctly removed. It is clearly seen from table 1 that our algorithm easily solves the still object problem (case 5) and uncovered background problem (case 6) with the help of background registration technique to avoid heavy computation.

Case	Background Difference	Frame Difference	Region Description	Object
1	Not Available	$\lvert FD\rvert > TH_{FD}$	Moving	Yes
2	Not Available	$\lvert FD\rvert \leq TH_{FD}$	Stationary	No
3	$\lvert BD\rvert > TH_{BD}$	$\lvert FD\rvert > TH_{FD}$	Moving Object	Yes
4	$\lvert BD\rvert \leq TH_{BD}$	$\lvert FD\rvert \leq TH_{FD}$	Background	No
5	$\lvert BD\rvert > TH_{BD}$	$\lvert FD\rvert \leq TH_{FD}$	Still Object	Yes
6	$\lvert BD\rvert \leq TH_{BD}$	$\lvert FD\rvert > TH_{FD}$	Uncovered Background	No

Table 1. Initial object mask generation

2.1.5 Post-processing

Sometimes the initial object mask is very noisy because of the camera noise. Instead of adopting the heavy computation like boundary relaxation [3][4][5], we simply eliminate too small regions using connected component algorithm to calculate the area of each region. Then, a temporal filter is optionally applied to improve the temporal coherency. The main concept is that if a certain pixel is determined as foreground object now by the initial mask, the possibility that this pixel belongs to the object for the next L frames should be very large. Thus, no matter how the initial mask will be, the pixel will be labeled as object for the next L frames. Finally, binary morphological closing and opening are applied to smooth object boundaries.

2.2 Shadow cancellation mode

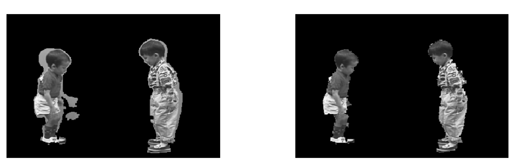
Moving objects may result in moving shadows in background regions, and the moving shadows will be detected as foreground by conventional change detection. A morphological gradient filter is applied as pre-processing to partially solve this problem. A typical gradient image is shown in Fig. 2(c). The edges are emphasized while the homogeneous regions are depressed. The segmentation result in the original domain is shown in Fig. 2(b), and the one in the gradient domain is shown in Fig. 2(d). Extra binary erosion is applied on the final mask to compensate the edge enhancement of the gradient filter.

(a) (b) (c) (d)

Figure 2. (a) original frame, (b) segmentation result in original domain, (c) gradient image, (d) segmentation result in gradient domain.

2.3 Predictive watershed

Watershed algorithm can be combined with temporal segmentation to give more accurate boundaries. However, it wastes a lot of time on segmenting the background. Instead of recalculating the watersheds of the whole frame, predictive watershed algorithm updates the watersheds in changed regions. In the I-frame, original watershed algorithm is applied. In the P-frames, the watersheds where the frame difference is smaller than a threshold are kept the same, and a modified watershed algorithm is applied on the changed regions. Thus, the background regions are not segmented again and again, and a lot of computation is saved. The original segmentation result and the significant improvement in *Children* sequence are shown in Fig. 3(a) and Fig. 3(b), respectively. This part is also optional.



(a) (b)

Figure 3. (a) Segmentation result of the original algorithm, (b) segmentation result with predictive watershed.

3. AUTOMATIC THRESHOLD DECISION

In [3, pp.168], the frame difference threshold is determined by the significant test. The standard deviation of camera noise is computed off-line or recursively on-line from unchanged regions. Then, the threshold can be mapped from the standard deviation according to the distribution model of camera noise. In our algorithm, the calculation of TH_{BD} in (3) is also determined by the significant test, while TH_{FD} is derived in a different view, that is, background registration. The proposed automatic threshold decision algorithm consists of three main stages: Gaussianity test, calculation of the standard deviation of camera noise, and scaling factor mapping. The flowchart is illustrated in Fig. 4.

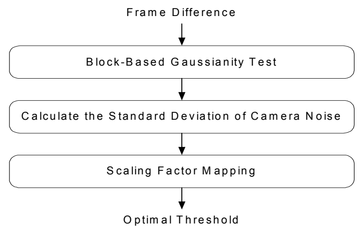


Figure 4. The flowchart of automatic threshold decision.

3.1 Gaussianity test

It is reasonable to say that the frame difference of stationary background resulted from camera noise is Gaussian-distributed, for the assumption of camera noise with Gaussian distribution is valid in many situations. Besides, the frame difference of moving objects is often not Gaussian-distributed. Therefore, a Gaussianity test is adopted to roughly separate the stationary background and the moving object. First, each frame is divided into $M \times N$ blocks. Then, a criterion can be applied to tell whether the frame difference of a block is Gaussian-distributed or not. The criterion that has been used to detect small moving object well in noisy image [17] is shown as follows:

$$I_k = \frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N} d^k[m,n] \tag{4}$$

$$H_{small}(I_1, I_2, I_3, I_4) = I_3 + I_4 - 3I_1(I_2 - I_1^2) - 3I_2^2 - I_1^3 + 2I_1^4 \tag{5}$$

$$H_{l\arg e}(I_1, I_2, I_3) = I_3 - 3I_1(I_2 - I_1^2) - I_1^3 \tag{6}$$

where (4) is the k-th moment of the $M \times N$ block and d is the frame difference. According to the block size, we can use (5) or (6) to determine whether this block is Gaussian-distributed or not. Criterion (5) is suitable for smaller blocks, while (6) is suitable for larger blocks. In our simulation, the block size is 16×16, and (5) is adopted. If the block is Gaussian-distributed, H is close to zero; otherwise, it is far from zero. Therefore, we define a fixed threshold for Gaussianity test, not requiring other steps to determine it. The typical value is $1 \sim 5$ and is not critical in our algorithm.

3.2 Calculate the standard deviation of camera noise

The standard deviation of frame difference in Gaussian blocks is calculated in this stage. It is believed that the standard deviation of frame difference in Gaussian blocks is very close to that of camera noise. It is mapped to an optimal threshold using the step described in the next subsection. Notice that except the assumption of Gaussian camera noise, no other prior knowledge, such as the object shape or the position of background, is required as an input to calculate the standard deviation of camera noise. If the position of background is known, Gaussianity test can be skipped. For example, if it is guaranteed that the blocks at the top left or top right corners always belong to background, these blocks can be directly used to calculate the standard deviation of camera noise without Gaussianity test.

3.3 Scaling factor mapping

The decision of TH_{BD} is simply obtained by the standard deviation multiplied by 2.5. The reason is the same as significance test in [3]. As for the other threshold TH_{FD}, background registration, instead of change detection, is our main consideration. We let the frame difference of a certain pixel in background region be a random variable d. The probability density function under the assumption of Gaussian camera noise is shown as follows:

$$f(d) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-d^2}{2\sigma^2}} \;,\; -\infty < d < \infty \tag{7}$$

Only the absolute value of frame difference is concerned in background. Thus, the above equation can be rewritten as:

$$f(d) = \frac{2}{\sigma\sqrt{2\pi}} e^{\frac{-d^2}{2\sigma^2}} \;,\; 0 < d < \infty \tag{8}$$

$$F(d) = \int_0^d \frac{2}{\sigma\sqrt{2\pi}} e^{\frac{-\omega^2}{2\sigma^2}} d\omega$$

where *F(d)* is the probability accumulation function of *f(d)*. The background registration technique is to register the pixels labeled as unchanged in consecutive N frames into the background buffer. This process is similar to a random variable transformation from d to X_k shown as:

$$X_k = Kth(f(d_1), f(d_2), ..., f(d_n)) \tag{9}$$

where *Kth* is an order statistics operator [18][19] that always selects the *k-th* largest element. The accumulation function and probability density function of X_k are derived as:

$$\begin{aligned}
&F_{X_k} = P(X_k \le x) = \sum_{j=k}^{N} C_j^N F^j(x)(1-F(x))^{N-j} \\
&\Rightarrow \frac{dF_{X_k}}{dx} = f_{X_k}(x) \\
&= \sum_{j=k}^{N} \frac{N!}{(j-1)!(N-J)!} f(x)F^{j-1}(x)(1-F(x))^{N-j} - \sum_{j=k}^{N-1} \frac{N!}{j!(N-j-1)!} f(x)F^{j}(x)(1-F(x))^{N-j-1} \\
&= \sum_{j=k}^{N} \frac{N!}{(j-1)!(N-J)!} f(x)F^{j-1}(x)(1-F(x))^{N-j} - \sum_{j=k+1}^{N} \frac{N!}{(j-1)!(N-j)!} f(x)F^{j-1}(x)(1-F(x))^{N-j} \\
&\Rightarrow f_{X_k}(x) = \frac{N!}{(k-1)!(N-k)!} f(x)F^{j-1}(x)(1-F(x))^{N-k}
\end{aligned} \tag{10}$$

A pixel will be registered into background buffer if every random sample, d, in consecutive N frame is lower than the threshold. Therefore, the optimal threshold should be as the maximum of the N samples, that is, $k = N$. We define the optimal threshold as the expectation value of the maximum among N frame differences in background regions. In this way, we can both alleviate miss penalty and avoid over-registration. A pixel of moving object will be registered into background buffer if its frame difference is lower than the tiny camera noise for consecutively N times. Since the probability is quite low, the false alarm penalty is not significant. Finally, the optimal threshold is shown as follows:

$$\begin{aligned}
&TH_{FD} \equiv E(X_N) = N\int_0^{\infty} y\frac{2}{\sigma\sqrt{2\pi}} e^{\frac{-y^2}{2\sigma^2}} [\int_0^y \frac{2}{\sigma\sqrt{2\pi}} e^{\frac{-x_1^2}{2\sigma^2}} dx_1 \int_0^y \frac{2}{\sigma\sqrt{2\pi}} e^{\frac{-x_2^2}{2\sigma^2}} dx_2 ..... \int_0^y \frac{2}{\sigma\sqrt{2\pi}} e^{\frac{-x_N^2}{2\sigma^2}} dx_n]dy \\
&\Rightarrow E(X_N) = N\int_0^{\infty} y\frac{2}{\sigma\sqrt{2\pi}} e^{\frac{-y^2}{2\sigma^2}} [\int_0^y \frac{2}{\sigma\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma^2}} dx]^{N-1} dy
\end{aligned} \tag{11}$$

The characteristic curves of calculated threshold values to standard deviation for different N are drawn in Fig. 5. However, they are somewhat different from the optimal thresholds that are empirically obtained. The reason is that the standard deviation of camera noise computed via the steps in subsection 3.1 and 3.2 is discrete while the σ in the derivation of (11) is continuous. Therefore, the quantization effect in discrete domain is then taken into consideration to modify the mapping. Each value of the standard deviation in continuous domain can be mapped to its corresponding standard deviation in discrete domain, and the standard deviation in discrete domain can further be mapped to the optimal threshold.

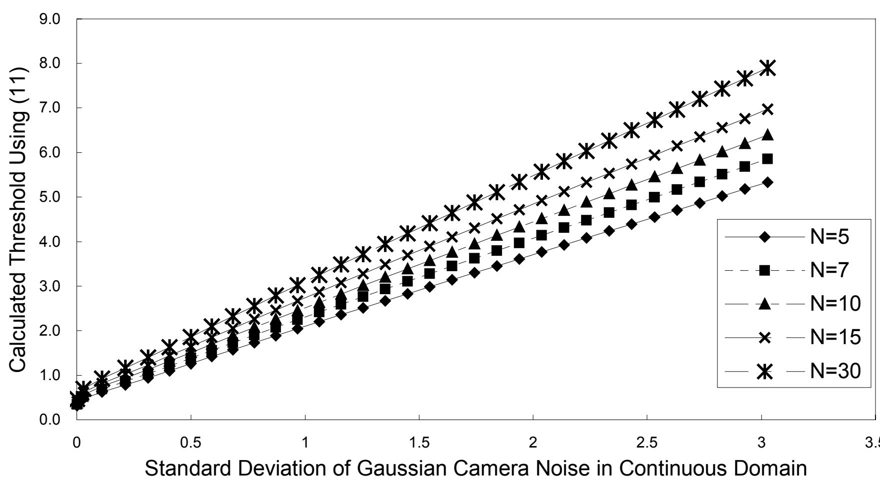


Figure 5. Calculated threshold in equation (11) versus the standard deviation of Gaussian camera noise in continuous domain.

First, we define the histogram of frame difference in background regions as follows:

$$h(d) = \#\{(r,c) \mid FD(r,c) = d, (r,c) \in background\} \tag{12}$$

where # is an operator that counts the number of pixels, (r, c) is the position of pixel, and $FD(r, c)$ is the frame difference. The probability of a certain value of frame difference in background regions can be written as:

$$P(d) = \frac{h(d)}{\#\{(r,c) \mid (r,c) \in background\}} \tag{13}$$

Therefore, we can rewrite (8) into a discrete form as shown in Fig. 6 and define P_k as the probability $P(d = k)$ as follows:

$$P_k = \begin{cases} \displaystyle\int_{-0.5}^{0.5} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma^2}} dx = \int_{0}^{0.5} \frac{2}{\sigma\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma^2}} dx & for \quad k = 0 \quad ; \\ \displaystyle\int_{k-0.5}^{k+0.5} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma^2}} dx + \int_{-k-0.5}^{-k+0.5} \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma^2}} dx = \int_{k-0.5}^{k+0.5} \frac{2}{\sigma\sqrt{2\pi}} e^{\frac{-x^2}{2\sigma^2}} dx & for \quad k = 1, \dots 255. \end{cases} \tag{14}$$

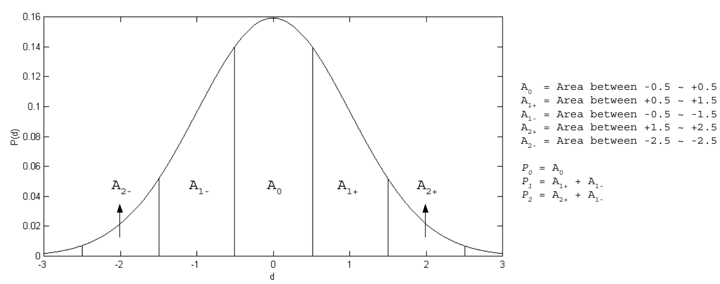


Figure 6. An example of mapping $P(d)$ in continuous domain to P_k in discrete domain for $\sigma = 1$.

Note that the probability density function for $|d|>255.5$ is almost zero, and thus P_k for $k>255$ is discarded. Then, the standard deviation and the corresponding optimal threshold in discrete domain can be defined as:

$$\sigma^{discrete} = \sqrt{\sum_{k=0}^{255} k^2 P_k} \tag{15}$$

$$TH_{FD} = E(X_N^{discrete}) = N \sum_{k=0}^{255} k P_k \left[\sum_{m=0}^{k} P_m \right]^{N-1} \tag{16}$$

Thus, the characteristic curve of the optimal threshold to the estimated standard deviation of Gaussian camera noise in discrete domain can be drawn in this way and is shown in Fig. 7. Now if the standard deviation of camera noise is known, TH_{FD} can be obtained by a table-lookup method.

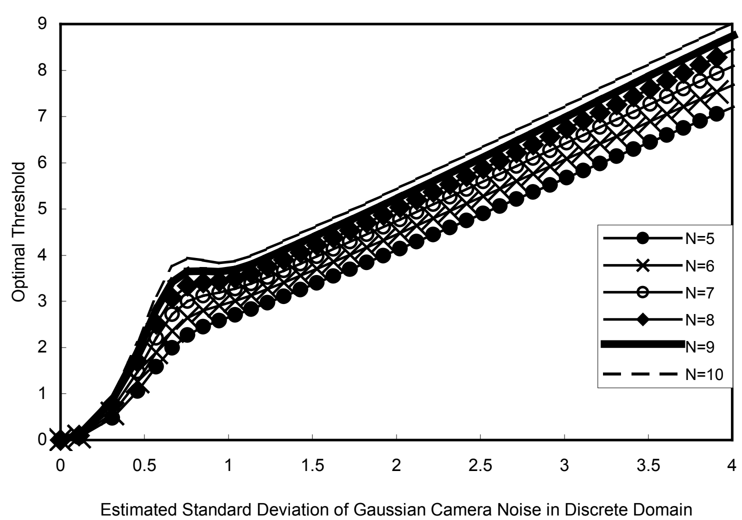


Figure 7. Optimal threshold in equation (16) versus estimated standard deviation of Gaussian camera noise in discrete domain.

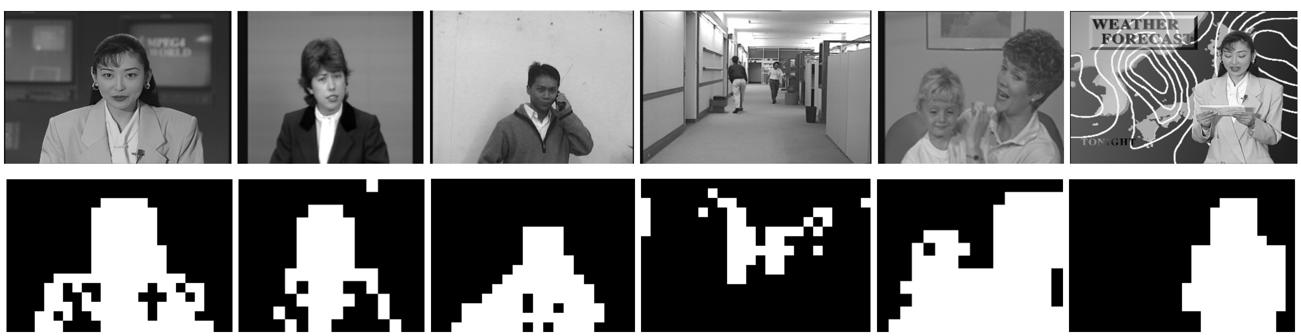
As shown in Fig. 7, the larger N becomes, the bigger the optimal threshold will be. This fact is taken for granted since N implies the extent of difficulty to registration. That is, the optimal threshold is defined as the expectation value of the largest frame difference of a certain pixel in the background among the N frame differences of the same pixel, and thus the optimal threshold should increase as N does. Also, we may note that there is a local maximum. This is because probability density within $-0.5 \leqq d < 0.5$, which is quite a large portion of total probability, is contributed to d = 0 (P_0), as shown in Fig. 6. For example, when $\sigma = 0.5$ $\sigma = 1.0$, P_0 is 38.29% and 68.27%, respectively. Note that P_0 contributes nothing to the optimal threshold TH_{FD}, as shown in (16).

4. SIMULATION RESULTS

The characteristic curves of optimal threshold values versus the standard deviation of Gaussian camera noise under different background registration conditions have been shown in the previous section. Once the standard deviation is obtained, the optimal threshold can be calculated. Many sequences under different noise conditions are tested using the proposed automatic threshold decision algorithm and the previously proposed segmentation algorithm. The test sequences include the sequences provided by MPEG-4 standard and those recorded by us. Different levels of additive Gaussian noise are also intentionally added to test our algorithm.

4.1 Gaussianity test

The main purpose of Gaussianity test is to roughly find the background regions. As you can see the results of Gaussianity test for different sequences in Fig. 8, the correlation between the Gaussian blocks drawn in black and the stationary background is high. Thus, the standard deviation of Gaussian camera noise can be estimated as the standard deviation of frame difference in detected Gaussian blocks without any prior knowledge of the positions of background regions.



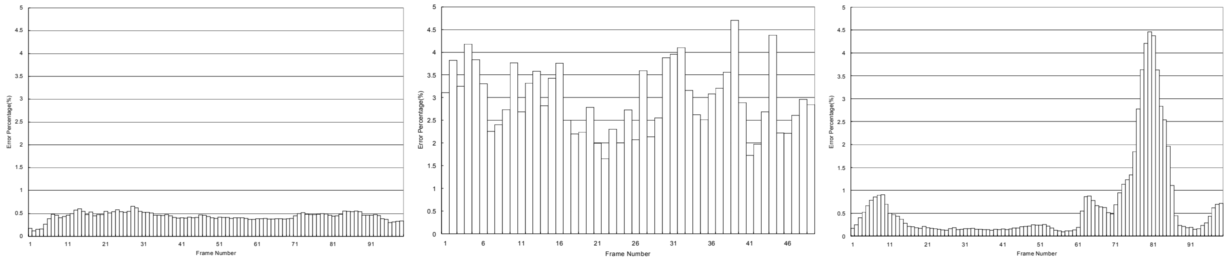
(a) (b) (c) (d) (e) (f)

Figure 8. The original frames and the results of Gaussianity test for (a) *Akiyo* #50, (b) *Claire* #50, (c) *Frank* #50, (d) *Hall Monitor* #50, (e) *Mother and Daughter* #50, (f) *Weather* #50.

4.2 Objective evaluation

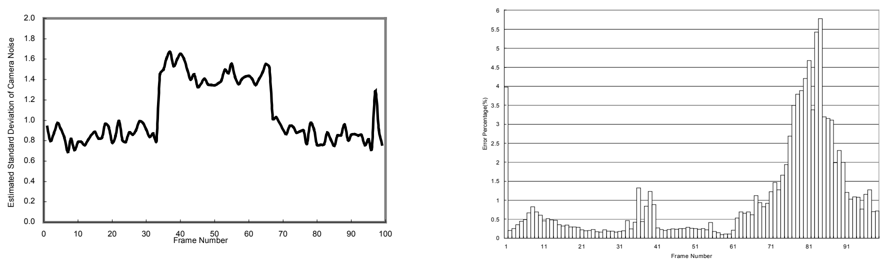
The criterion for objective evaluation is defined as the number of error pixels divided by the number of total pixels in the frame. Ground-true masks are available for some MPEG-4 test sequences. The results for *Akiyo*, *Children*, and *Weather* are shown in Fig. 9. The error percentage of *Akiyo* is below 0.7% for all frames. As for *Children*, gradient filter and predictive watershed are turned on. The average error percentage of *Children* is about 3% due to the large motion of the two children and the ball. The error percentage of *Weather* is below 1% at most of the frames, except a sudden rise to 4.5% near frame 75. It is because of the sudden step back of the weather forecaster. After the uncovered background regions are registered into background buffer, the error rate quickly drops back to 0.5%.

In order to further test the capability of the proposed automatic threshold decision algorithm, different levels of additive Gaussian noise is added into *Weather*. The standard deviation of Gaussian noise is 0.5, 1.2, and 0.7 during frame 1 ~ 33, frame 34 ~66, and frame 67 ~ 100, respectively. The optimal threshold is adaptively adjusted according to the camera noise. In Fig. 10(a), the calculated standard deviation for each frame is shown. The optimal threshold can be mapped using the curves in Fig. 7 (N = 5). The error percentage is shown in Fig. 10(b). The error rate rises a little, compared with Fig. 9(c).



(a) (b) (c)

Figure 9. The error percentage of (a) *Akiyo*, (b) *Children*, (c) *Weather*.



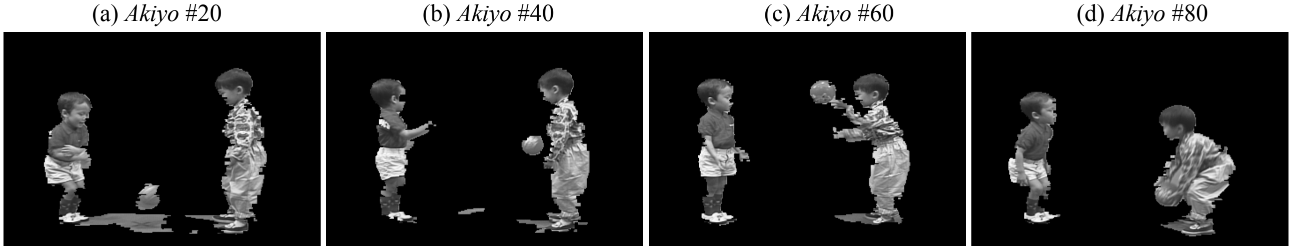
(a) (b)

Figure 10. Three different levels of Gaussian noise are added into *Weather*. The levels are 0.5, 1.2, and 0.7. (a) Estimated *s*tandard deviation of camera noise from Gaussianity test versus frame. (b) Error percentage versus frame.

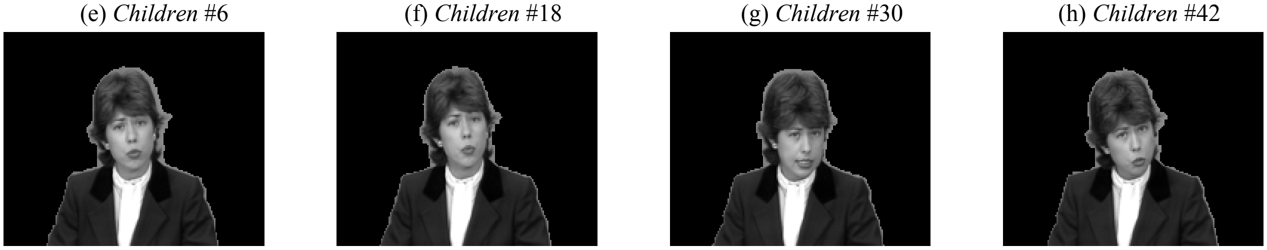
4.3 Subjective view

In Fig. 11, segmentation results of various sequences are shown in subjective views. The frame rate of the original sequence is 10fps. *N* is set to 5 in these cases. Frame #20, #40, #60, and #80 are shown for all the sequences, except for *Children*. There are only 50 frames in *Children*, and frame #6, #18, #30, and #42 are selected. Gradient filter is turned on for *Children*, *Claire*, *Frank*, *Hall Monitor*, *Shaoyi*, and *Silent*. Predictive watershed is turned on for *Children*.

(a) *Akiyo* #20 (b) *Akiyo* #40 (c) *Akiyo* #60 (d) *Akiyo* #80



(e) *Children* #6 (f) *Children* #18 (g) *Children* #30 (h) *Children* #42



(i) *Claire* #20 (j) *Claire* #40 (k) *Claire* #60 (l) *Claire* #80

(m) *Frank* #20 (n) *Frank* #40 (o) *Frank* #60 (p) *Frank* #80

(q) *Hall Monitor* #20 (r) *Hall Monitor* #40 (s) *Hall Monitor* #60 (t) *Hall Monitor* #80

(u) *Mother and Daughter* #20 (v) *Mother and Daughter* #40 (w) *Mother and Daughter* #60 (x) *Mother and Daughter* #80

(y) *Shaoyi* #20 (z) *Shaoyi* #40 (α) *Shaoyi* #60 (β) *Shaoyi* #80

(γ) *Silent* #20 (δ) *Silent* #40 (ε) *Silent* #60 (ζ) *Silent* #80

(η) *Weather* #20 (θ) *Weather* #40 (ι) *Weather* #60 (κ) *Weather* #80

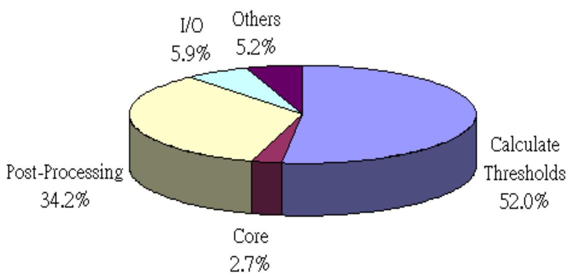
Figure 11. Segmentation results of various sequences.

4.4 Run-time analysis

The run-time analysis is also shown in this paper. The test environment is a PC with Intel Pentium III 800MHz CPU running Microsoft Windows 2000 operating system. The profile of the segmentation algorithm is shown in Fig. 12(a).

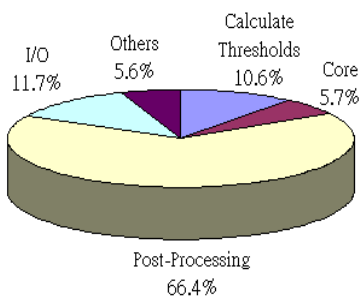
The optional morphological gradient filter and predictive watershed are turned off. The core process, which includes frame difference, background registration, background difference, and initial mask generation, is optimized by MMX instructions. The post-processing includes connected component operations, a temporal filter, and binary morphological operations. The binary morphological operations are speeded up by sub-word parallelism. The calculation of thresholds occupies the most computation. The processing speed is 20 QCIF (176×144) frames per second.

The automatic threshold decision algorithm is modified to reduce the heavy computation. Gaussianity test is only applied on the first several frames. For the following frames, the standard deviation of camera noise is computed from the blocks that are not neighboring to any block containing part of objects in the previous frame. The profile is shown in Fig. 12(b). Now the processing speed is 38 QCIF frames per second.



Mother and Daughter, QCIF, 100 frames	
Operations	Run-Time
Calculate Thresholds	2554.993 ms
Core	134.493 ms
Post-Processing	1681.264 ms
I/O	285.162 ms
Others	257.753 ms
Total	4913.665 ms
20.35 frames per second	

(a)



Mother and Daughter, QCIF, 100 frames	
Operations	Run-Time
Calculate Thresholds	270.228 ms
Core	143.997 ms
Post-Processing	1689.703 ms
I/O	297.535 ms
Others	167.656 ms
Total	2569.119 ms
38.92 frames per second	

(b)

Figure 12. The run-time analysis of *Mother and Daughter*. Gaussianity test is turned on for (a) every frame, (b) only the first 7 frames.

If the frame size is larger or when gradient filter or predictive watershed is turned on, the computation will be too heavy for general-purpose processors to attain real-time performance. Fortunately, most of the operations in predictive watershed, pre-processing, and post-processing can be mapped to morphological operations. They are simple and regular, and thus, are suitable for hardware acceleration. Currently, we are working on these related issues, and some achievements have been made [20][21].

5. CONCLUSION

A new method of automatic threshold decision of the frame difference for background registration is proposed. Gaussianity test is first applied to find the standard deviation of Gaussian camera noise. Then the optimal threshold value can be obtained from the standard deviation multiplied by a scaling factor. Probability theories and the quantization effect in discrete domain are taken into considerations to derive the relation between the standard deviation of Gaussian camera noise and the optimal threshold value. Simulation results show that the automatic threshold decision algorithm, combined with the previously proposed video segmentation algorithm, can extract the video objects well under a fixed camera. Furthermore, the processing speed is fast enough to meet real-time requirements. When the gradient filter and predictive watershed are turned off, 20 and 38 QCIF frames per second can be achieved with and without Gaussianity test, respectively, on a personal computer with Intel Pentium III 800MHz CPU running Microsoft Windows 2000 operating system.

REFERENCES

[1] Thomas Sikora, “The MPEG-4 video standard verification model,” *IEEE Tran. on Circuit and Systems for Video Technology*, vol. 7, no. 1, pp.19-31, February 1997.

[2] MPEG Video Group, “The MPEG-4 video standard verification model version 15.0,” *ISO/IEC JTC 1/SC 29/WG 11 N3093*

[3] Til Aach, Andre Kaup, and Rudolf Mester, “Statistical model-based change detection in moving video,” *Signal Processing*, vol. 31, pp.165-180, March 1993.

[4] Roland Mech and Michael Wollborn, “A noise robust method for 2D shape estimation of moving objects in video sequences considering a moving camera,” *Signal Processing*, vol. 66, pp.203-217, April 1998.

[5] Roland Mech, “Robust 2D shape estimation of moving objects considering spatial and temporal coherency in one map detection rule,” in *Proceedings of International Conference on Image Processing*, vol. 1, pp.331-334, 2000.

[6] Thomas Meier and King N. Ngan, “Automatic segmentation of moving objects for video object plane generation,” *IEEE Transactions on Circuit and Systems for Video Technology*, vol. 8, no. 5, December 1998.

[7] Thomas Meier and King N. Ngan, “Video segmentation for content-based coding,” *IEEE Transactions on Circuit and Systems for Video Technology*, vol. 9, no. 8, December 1999.

[8] Demin Wang, “Unsupervised video segmentation based on watersheds and temporal tracking,” *IEEE Transactions on Circuit and Systems for Video Technology*, vol. 8, no. 5, pp.539-546, September 1998.

[9] Ju Guo, Jongwong Kim and C. -C. Jay Kuo, "Fast video object segmentation using affine motion and gradient-based color clustering," in *Proceedings of IEEE Workshop on Multimedia Signal Processing*, Los Angeles, California, December 1998.

[10] Luis Salgado, Narciso Garcia, Jose M. Menendez, and Enrique Rendon, “Efficient image segmentation for region-based motion estimation and compensation,” *IEEE Transactions on Circuit and Systems for Video Technology*, vol. 10, no. 7, pp.1029-1039, October 2000.

[11] Munchurl Kim, Jae Gark Choi, Daehee Kim, Hyung Lee, Myoung Ho Lee, Chieteuk Ahn, and Yo-Sung Ho, “A VOP generation tool: automatic segmentation of moving objects in image sequences based on spatial-temporal information,” *IEEE Transactions on Circuit and Systems for Video Technology*, vol. 9, no. 8, pp.1216-1226, December 1999.

[12] Ioannis Kompatsiaris and Michael Gerassimos Strinzis, “Spatiotemporal segmentation and tracking of objects for visualization of videoconference image sequences,” *IEEE Transactions on Circuit and Systems for Video Technology*, vol. 10, no. 8, pp.1388-1402, December 2000.

[13] Shao-Yi Chien, Shyh-Yih Ma, and Liang-Gee Chen, “An efficient video segmentation algorithm for real-time MPEG-4 camera system,” in *Proceedings of Visual Communication and Image Processing*, vol. 4067, pp. 1087-1098, 2000.

[14] Shao-Yi Chien, Yu-Wen Huang, Shyh-Yih Ma, and Liang-Gee. Chen, “Efficient video segmentation on SIMD architecture for real-time MPEG-4 systems”, in *Proceedings of Workshop on Consumer Electronics*, pp. 22-26, Taipei, Taiwan, October 2000.

[15] Shao-Yi Chien, Yu-Wen Huang, Shyh-Yih Ma, and Liang-Gee Chen, “A real-time practical video segmentation algorithm for MPEG-4 camera systems”, in *Proceedings of International Conference on Consumer Electronics*, pp. 142-143, Los Angeles, USA, June 2001.

[16] Shao-Yi Chien, Yu-Wen Huang, Shyh-Yih Ma, and Liang-Gee Chen, “Automatic video segmentation for MPEG-4 using predictive watershed”, in *Proceedings of International Conference on Multimedia and Expo*, Tokyo, Japan, August, 2001.

[17] Rabi Zaibi, ”Small moving object detection in video sequences,” in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, 2000.

[18] Hoel and Paul Gerhard, *Introduction to Prabibility theory*, 1971.

[19] N.Balakrishnan and C.R.Rao, *Order Statistics: Applications*, 1998.

[20] Shao-Yi Chien, Yu-Wen Huang, Shyh-Yih Ma, and Liang-Gee Chen, "A hybrid morphological processing units architecture for real-time video segmentation systems", in *Proceedings of IEEE International Symposium on Circuits and Systems*, Sydney, Australia, May 2001.

[21] Shao-Yi Chien, Shyh-Yih Ma, and Liang-Gee Chen, "A partial-result-reuse architecture and its design technique for morphological operations", in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 1185-1188, May 2001.