Net comparison: an adaptive and effective method for scene change detection[†]

Wei Xiong, John Chung-Mong Lee, Ding-Gang Shen *

Technical Report HKUST-CS95-12
April 13, 1995

*Department of Computer Science
The Hong Kong University of Science and Technology
Clear Water Bay, Kowloon, Hong Kong

[†]This research was supported by Sino Software Research Centre under grants SSRC92/93.016 and SSRC93/94.EG16.

Abstract

As video information proliferates, managing video sources becomes increasingly important. Automatic video partitioning is a prerequisite for organizing and indexing video sources. Several methods have been introduced to tackle this problem, e.g. pairwise and histogram comparisons. Each has advantages, but all are slow because they entail inspection of entire images. Furthermore none of these methods have been able to define camera break and gradual transition, which are basic concepts for partitioning.

In this paper, we attempt to define camera break. Then, based on our definition and probability analysis, we propose a new video partitioning algorithm, called NET Comparison (NC), which compares the pixels along predefined net lines. In this way, only part of the image is inspected during classification. We compare the effectiveness of our method with other algorithms such as pairwise, likelihood and histogram comparisons, evaluating them on the basis of a large set of varied image sequences that include camera movements, zooming, moving objects, deformed objects and video with degraded image quality. Both gray-level and HSV images were tested and our method out-performed existing approaches in speed and accuracy. On average, our method processes images two to three time faster than the best existing approach.

Keywords: camera break, color image sequences, multimedia, video indexing, video classification

1 INTRODUCTION

The information time brings us an enormous number of video sources. Video cassette tapes and laser-discs are used for many purposes such as entertainment, instructions, scientific record, art storage and so on. To make good use of the video information, we need to analyze and index video tapes and laser-discs. Video partitioning is one of the key issues for video index, video analysis and video communication. It is tedious, time-consuming and even impossible to do the job manually. Automatic video partitioning is therefore attracting more and more researchers' attention.

The partitioning process involves the detection of boundaries between uninterrupted segments (camera shots) of screen time, space or graphic configurations. These boundaries are commonly known as transitions. Transitions can be classified into two categories, gradual and instantaneous. The most common transition is camera break. Although video partitioning research has been carried out for many years, basic concepts like camera break and gradual transition have never been well defined.

Several methods such as pairwise comparison, likelihood comparison and histogram comparison have been introduced [1, 2]. These methods have their merits and limitations. For example, histogram comparison is quite simple but ignores spatial information and, therefore fails to detect camera breaks in some cases[3]. Both pairwise comparison and likelihood comparison make use of spatial information but the former one is too sensitive in some cases and easily causes false alarms while the latter one has complex computational problem. One common problem of all of the existing methods is time-consuming. This is because these methods are based on idea of inspecting the entire image.

In this paper, we attempt to give a reasonable definition for camera break. Then, based on our definition and probability analysis, we propose a new video partitioning algorithm called Net Comparison (NC), which compares the pixels along the predefined net lines. In this way, only part of the image is inspected during the classification process. In our experimental results, we shall contrast the effectiveness of our method with other existing algorithms such as pairwise, likelihood and histogram comparisons. Our method has been evaluated based on a large set of various video sequences that include camera movements, zooming, moving objects, deformed objects and degraded quality images. Both gray-level and HSV images were tested and our method out-performed existing approaches in speed and accuracy. On average, our method method processes an image two to three times faster than the best existing approach.

2 RELATED METHODS

Conventional camera break detection algorithms can be divided into two basic categories: Pairwise comparison and Histogram comparison.

1. Pair-wise Comparison

   There are two basic algorithms in this category:

   (a) Pairwise

      In Pairwise Comparison, successive frames are examined point by point. Each pixel is compared with the corresponding pixel in the successive frame. If the frame-to-frame difference of gray levels exceeds a threshold, a camera break is declared. The frame-to-frame difference d(i,i+1) of frame i and frame i+1 is computed as follows

$$d(i, i+1) = \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} |g(i,x,y) - g(i+1,x,y)|$$

where g(i,x,y) denotes the gray level of the pixel at (x,y) in frame i, W and H are the width and height of a frame respectively.

(b) Likelihood

A frame is divided into uniform regions. The mean and variance of each region are computed, and the likelihood ratio of each corresponding regions is then computed. If the frame-to-frame difference exceeds a threshold, a camera break is declared. The frame-to-frame difference d(i,i+1) of frame i and frame i+1 is computed as follows

$$d(i, i+1) = \sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} I(i, i+1, p, q)$$

$$I(i, i+1, p, q) = \langle \begin{matrix} 1 \quad if \ L(i, i+1, p, q) > T \\ \\ 0 \qquad\qquad\qquad otherwise \end{matrix}$$

$$L(i, i+1, p, q) = \frac{[\frac{v(i,p,q)+v(i+1,p,q)}{2} + (\frac{m(i,p,q)-m(i+1,p,q)}{2})^2]^2}{v(i,p,q)v(i+1,p,q)}$$

where L(i,i+1,p,q) is likelihood ratio, v(i,p,q) and m(i,p,q) are the mean and variance of intensity values for region(p,q) of frame i. P x Q denotes the number of regions in a frame.

2. Histogram Comparison

There are two basic algorithms in this category:

(a) Global Histogram Comparison

This method makes use of gray level histograms of frames. If the difference between the histograms of two consecutive frames exceeds a threshold, a camera break is declared between the two frames. The frame-to-frame difference d(i,i+1) of frame i and frame i+1 is computed as follows

$$d(i, i+1) = \sum_{l=0}^{G-1} |h(i,l) - h(i+1,l)|$$

where h(i,l) denotes the histogram value at gray level l for frame i, G is the number of possible gray levels.

(b) Local Histogram Comparison

A frame is divided into uniform, non-overlapping regions. Gray level histogram of each region (local histograms) are computed and compared to corresponding regions of the successive frame. If the overall gray-level difference exceeds a threshold, a camera break is declared. The frame-to-frame difference d(i,i+1) of frame i and frame i+1 is computed as follows

$$d(i, i+1) = \sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} H_{dif}(i, i+1, p, q)$$

$$H_{dif}(i, i+1, p, q) = \sum_{l=0}^{G-1} |h(i,p,q,l) - h(i+1,p,q,l)|$$

where h(i,p,q,l) denotes the histogram value at gray level l for region(p,q) of frame i, G is the number of possible gray levels.

These algorithms can be used for either gray level images or color images[1, 4]. Several (6) bits of RGB are recommended for representing the color images[5].

3 NET COMPARISON

Before discussing the camera break detection algorithm, we must make clear what camera break is. As pointed out earlier, camera break is instantaneous transition. As this concept may cause ambiguity in some cases, we try to give a formal definition for camera break.

Suppose we uniformly and consecutively divide the image into small non-overlapping square areas(see Figure 1), we call these areas base windows, denoted by B_{ij} (i, j=0,1,2,...).

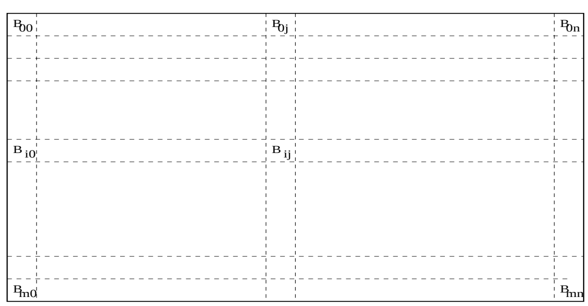

Figure 1: Base windows

Let D_n denotes the mean-value difference between the corresponding base windows of two consecutive images in the non-break case while D_c in the camera break case. Our purpose is try to make the difference between D_n and D_c as large as possible. If the size of the base windows(L) is too small, both D_n and D_c will be large while both D_n and D_c

may be small if L is too large. So L should be large enough not to be sensitive to the non-break change between two images and small enough to contain the spatial information as much as possible. So the decision of base window's size is a kind of trade off. The size of the base windows(L) depends on the maximum non-break velocity (δ) [6], which is the largest movement between two images. The movement may be caused by object moving, zooming, panning, etc.. Obviously, we do not need to consider the movement of any small object in the image such as bullet, rain points, which speeds are usually very high. So velocity (δ) here means the movement of a big enough part of the image between two consecutive frames. From experiments, we know δ is rarely larger than 10 pixels. If we let L=2δ, we use 20 pixels as the windows size. In this way, we may get following conclusion. In most cases, at least a quarter number of pixels are same between the corresponding windows of two consecutive images if no camera break occurs between the two images.

Because of continuity of the images, the actual number of the same (or very similar) value pixels is usually much greater than one quarter number of pixels in base window.

Definition 3.1. *If and only if more than about 50%(P_c) corresponding base windows are different in two consecutive frames, the camera break occurs between the two images . In term of different, we mean the difference between the mean values (either gray-level or color value) of two corresponding base windows is greater than a threshold(T_b).*

As previously pointed out, camera break is instantaneous change. That means there is a big jump or difference between two images. In terms of image processing, it means the gray value (or color) of most area of two images are different. So definition is consistent with what we usually mean. The reason why we describe this kind of change with base windows instead of pixels is that pixel value is too sensitive. The corresponding pixels may completely different just because the camera slightly moving or even shaking. The purpose we use T_b to decide whether the two base windows are different is to remove the influence of some factors such as light changing, different irradiance.

LEMMA 3.2. *To detect a camera break, we may just check small number of base windows and gain a very low error rate.*

Proof: We may assume that the change between two images have a uniform distribution. In other words, each base window has the same probability to be changed.

Suppose: N– number of the base windows in an image. N_c – number of changed base windows between two images. M – number of the checked base windows in a image. M_c – number of checked changed base windows. Er – Error rate. Pr – Probability.

If the purpose is just to detect camera break, we may get the error rate with respect to the different M in the following way:

- If no camera break occurs,

$$Er_1 = Pr(\frac{M_c}{M} > P_c | \frac{N_c}{N} < (1 - P_c)) * Pr(\frac{N_c}{N} < (1 - P_c))$$

- If camera break occurs,

$$Er_2 = Pr(\frac{M_c}{M} < (1 - P_c) | \frac{N_c}{N} >= P_c) * Pr(\frac{N_c}{N} >= P_c)$$

In this case, we have Er=$Er_1 + Er_2$. $Pr(\frac{N_c}{N} < (1 - P_c))$ and $Pr(\frac{N_c}{N} >= P_c)$ are not known priori. We just know $Pr(\frac{N_c}{N} < (1 - P_c)) + Pr(\frac{N_c}{N} >= P_c) \doteq 1$ and $Pr(\frac{N_c}{N} < (1 - P_c)) << Pr(\frac{N_c}{N} >= P_c)$. Now let us introduce loss functions $L_{ij}[]$. Assume $L_{ii} =$

$0; L_{ij} > 0 (i <> j)$. Total loss is therefore : $L = Er_1 * L_{bn} + Er_2 * L_{nb}$; (where b denotes $\frac{N_c}{N} < (1 - P_c)$; n denotes $\frac{N_c}{N} >= P_c$;)

Because our purpose is detecting camera break, natually, we should define $L_{nb} > L_{bn}$. If we simply define $L_{bn} = \frac{1}{Pr(\frac{N_c}{N} < 1 - P_c)}; L_{bn} = \frac{1}{Pr(\frac{N_c}{N} >= P_c)}$;. Our formula can be simplified as:

$$L = Pr(\frac{M_c}{M} > P_c | \frac{N_c}{N} < (1 - P_c)) * Pr(\frac{N_c}{N} < (1 - P_c)) * L_{nb} + Pr(\frac{M_c}{M} < (1 - P_c) | \frac{N_c}{N} >= P_c) * Pr(\frac{N_c}{N} >= P_c) * L_{bn}$$

$$= Pr(\frac{M_c}{M} > P_c | \frac{N_c}{N} < (1 - P_c)) + Pr(\frac{M_c}{M} < (1 - P_c) | \frac{N_c}{N} >= P_c)$$

Where

$$Pr(\frac{M_c}{M} > P_c | \frac{N_c}{N} < (1 - P_c)) = \frac{Pr(\frac{M_c}{M} > P_c \bigcap \frac{N_c}{N} < (1 - P_c))}{Pr(\frac{N_c}{N} < (1 - P_c))}$$

$$= \langle \begin{array}{ll} \frac{\sum_{N_c = MP_c}^{N(1-P_c)} \sum_{M_c = MP_c}^{N_c} C_M^{M_c} C_{N-M}^{N_c - M_c}}{\sum_{N_c = 0}^{N(1-P_c)-1} C_N^{N_c}} & M > N(1 - P_c) \\ \frac{\sum_{N_c = MP_c}^{M} \sum_{M_c = MP_c}^{N_c} C_M^{M_c} C_{N-M}^{N_c - M_c} + \sum_{N_c = M+1}^{M} \sum_{M_c = MP_c}^{M} C_M^{M_c} C_{N-M}^{N_c - M_c}}{\sum_{N_c = 0}^{N(1-P_c)-1} C_N^{N_c}} & otherwise \end{array}$$

$$Pr(\frac{M_c}{M} < (1 - P_c) | \frac{N_c}{N} >= P_c) = \frac{Pr(\frac{M_c}{M} < (1 - P_c) \bigcap \frac{N_c}{N} >= P_c)}{Pr(\frac{N_c}{N} >= P_c)}$$

$$= \frac{Pr(\frac{M_c}{M} < (1 - P_c)) - Pr(\frac{M_c}{M} < (1 - P_c) \bigcap \frac{N_c}{N} < (1 - P_c))}{Pr(\frac{N_c}{N} >= P_c)}$$

$$= \frac{\sum_{M_c = 0}^{M(1-P_c)-1} C_M^{M_c}}{\sum_{M_c = 0}^{M} C_M^{M_c}} * \frac{\sum_{N_c = 0}^{N} C_N^{N_c}}{\sum_{N_c = NP_c}^{N} C_N^{N_c}} -$$

$$\frac{\sum_{N_c = 0}^{M(1-P_c)-1} \sum_{M_c = 0}^{N_c} C_M^{M_c} C_{N-M}^{N_c - M_c} + \sum_{N_c = M(1-P_c)}^{NP_c - 1} \sum_{M_c = 0}^{M(1-P_c)-1} C_M^{M_c} C_{N-M}^{N_c - M_c}}{\sum_{N_c = NP_c}^{N} C_N^{N_c}}$$

If the resolution of our images is 318X238, the relationship of M and L is shown in Figure 2. With this resolution, N is about 192, and we can find that we may get a very low error rate (or loss) while $M < N$, so the lemma holds.

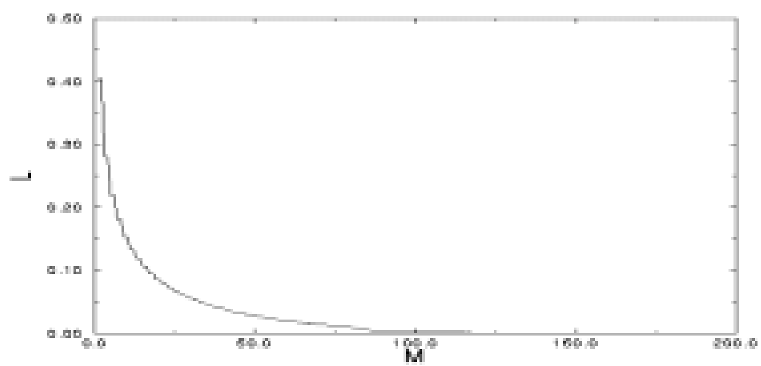

Figure 2: The relationship between L and M for detecting camera break

We will say the M samples represent the real situation, if $0.9\frac{N_c}{N} <= \frac{M_c}{M} <= 1.1\frac{N_c}{N}$. So if the purpose is to actually represent the real situation, the error rate can be computed in a way similar to the above,

$$L = 1 - Pr(0.9\frac{N_c}{N} <= \frac{M_c}{M} <= 1.1\frac{N_c}{N})$$

$$
= \left\langle \begin{array}{ll}
1 - \dfrac{\sum_{N_c=0}^{N/1.1} \sum_{M_c=0.9M\frac{N_c}{N}}^{1.1M\frac{N_c}{N}} C_M^{M_c} C_{N-M}^{N_c-M_c} + \sum_{N_c=\frac{N_c}{1.1}+1}^{N} \sum_{M_c=0.9M\frac{N_c}{N}}^{M} C_M^{M_c} C_{N-M}^{N_c-M_c}}{\sum_{N_c=0}^{N} C_N^{N_c}} & M <= \frac{N}{1.1} \\
1 - \dfrac{\sum_{N_c=0}^{M} \sum_{M_c=0.9M\frac{N_c}{N}}^{N_c} C_M^{M_c} C_{N-M}^{N_c-M_c} + \sum_{N_c=M}^{N} \sum_{M_c=0.9M\frac{N_c}{N}}^{M} C_M^{M_c} C_{N-M}^{N_c-M_c}}{\sum_{N_c=0}^{N} C_N^{N_c}} & M > \frac{N}{1.1}
\end{array} \right.
$$

Figure 3 shows the relationship between M and L when N is 192.

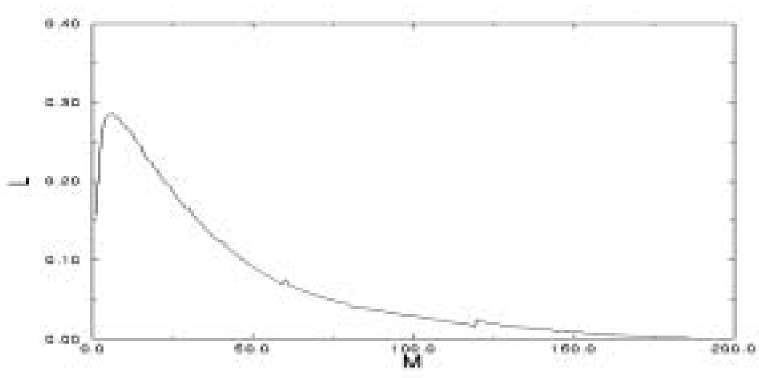

Figure 3: The relationship between L and M for representing the real situation

Based on above analysis, we get the net comparison algorithm as follows:

1. Uniformly take M base windows on the image.

2. Compare the corresponding regions of the successive images by computing the difference D_1 between the mean values of the regions' gray-level(or RGB, HSV,...).

3. If D_1 is greater than threshold T_b, we shall say that the region is changed between the two frames, otherwise it is unchanged.

4. If the number of changed regions is greater than threshold P_c, a camera break is declared.

So the number of changed base windows is computed as follows

$$D = \sum_{i=1,k=1}^{i=m,k=p} A(i,k) \ + \sum_{j=1,l=1}^{j=n,l=q} A(j,l)$$

$$A(i,j) = \left\langle \begin{array}{ll} 1 & |ave(f,i,j) - ave(f+1,i,j)| > T_1 \\ & \\ 0 & otherwise \end{array} \right.$$

where $ave(f,i,j)$ is the mean value of the base window B_{ij} on frame f.

Doing further analysis, we find the number of checked base windows (M) may be even smaller in some cases. So the number of checked base windows is changable , or say, adaptive. That is, M is rather small initially and increased only when necessarily. Whenever $\frac{M_c}{M} > P_m$ or $\frac{1-M_c}{M} > P_m$ the algorithm will stop and know it is camera break or not, otherwise M will be increased unless $M >= N$. Using this algorithm, we need to know the relationship among M, P_m and error rate (or loss). Similar to the above analysis, we have,

$$L = Pr(\frac{M_c}{M} > P_m | \frac{N_c}{N} < (1 - P_c)) + Pr(\frac{M_c}{M} < (1 - P_m) | \frac{N_c}{N} >= P_c)$$

Where

$$Pr(\frac{M_c}{M} > P_m | \frac{N_c}{N} < (1-P_c)) = \frac{Pr(\frac{M_c}{M} > P_m \bigcap \frac{N_c}{N} < (1-P_c))}{Pr(\frac{N_c}{N} < (1-P_c))}$$

$$= \langle \begin{array}{ll} \frac{\sum_{N_c=MP_m}^{N(1-P_c)} \sum_{M_c=MP_m}^{N_c} C_M^{M_c} C_{N-M}^{N_c-M_c}}{\sum_{N_c=0}^{N(1-P_c)-1} C_N^{N_c}} & M > N(1-P_c) \\ \frac{\sum_{N_c=MP_m}^{M} \sum_{M_c=MP_m}^{N_c} C_M^{M_c} C_{N-M}^{N_c-M_c} + \sum_{N_c=M+1}^{M} \sum_{M_c=MP_m}^{M} C_M^{M_c} C_{N-M}^{N_c-M_c}}{\sum_{N_c=0}^{N(1-P_c)-1} C_N^{N_c}} & M <= N(1-P_c) \end{array}$$

$$Pr(\frac{M_c}{M} < (1-P_m) | \frac{N_c}{N} >= P_c) = \frac{Pr(\frac{M_c}{M} < (1-P_m) \bigcap \frac{N_c}{N} >= P_c)}{Pr(\frac{N_c}{N} >= P_c)}$$

$$= \frac{Pr(\frac{M_c}{M} < (1-P_m)) - Pr(\frac{M_c}{M} < (1-P_m) \bigcap \frac{N_c}{N} < (1-P_c))}{Pr(\frac{N_c}{N} >= P_c)}$$

$$= \frac{\sum_{M_c=0}^{M(1-P_m)-1} C_M^{M_c}}{\sum_{M_c=0}^{M} C_M^{M_c}} * \frac{\sum_{N_c=0}^{N} C_N^{N_c}}{\sum_{N_c=NP_c}^{N} C_N^{N_c}} -$$

$$\frac{\sum_{N_c=0}^{M(1-P_m)-1} \sum_{M_c=0}^{N_c} C_M^{M_c} C_{N-M}^{N_c-M_c} + \sum_{N_c=M(1-P_m)}^{NP_c-1} \sum_{M_c=0}^{M(1-P_m)-1} C_M^{M_c} C_{N-M}^{N_c-M_c}}{\sum_{N_c=NP_c}^{N} C_N^{N_c}}$$

The relationship among M, P_m and L is shown in figure 4.

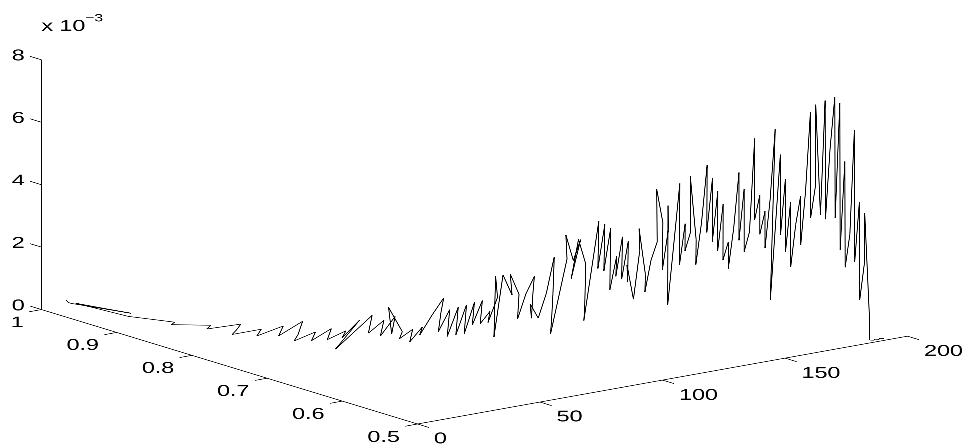

Figure 4: The relationship among M, P_m and L for representing the real situation

4 EXPERIMENT

Now, we have five different methods: Pairwise(PW), Likelihood(LK), Global histogram(GH), Local histogram(LH) and Net comparison(NC). Each of them can be applied on either gray-level images or color images. Many researchers have evaluated the various color models[7, 8, 9]. It is believed that the perceptual color space HSV is a good choice [10, 11, 12]. In our experiment we use HSV (Hue, Saturation, Value) to represent color information. As the LK method is too slow to be practical for color image detection, we use 5 methods for gray-level and 4 methods for HSV images. The methods are used on several typical episodes from the movie "A better tomorrow" and run on a computer MAC Quatra 700. The images are originally in RGB format and achieved from the PIONEER laser-disc player(LD-V8000) through the RasterOps Media-Time frame grabber board. The experimental results are listed in Table 1 Table 2. Because of the limitation of RAM capacity, each frame is actually transfered twice. If the conversion can be implemented by hardware, the running time of every algorithm will be considerably improved, especially for HSV images. The 11 typical episodes are demonstrated from Figure 5 to Figure 15.

From Table 1, we may find that the performance of PW and NC are both very good except for very dark sequences. The algorithms LK, GH and LH work poorly for three episodes ((4),(5) and (11)) which are either in darkness or have much noise, and work fine for the other 8 episodes. Obviously, NC is much faster than the others. From Table 2, we can see that the performance of NC for HSV images is almost perfect(with optimized threshold). LH has also very good performance except for the dark sequences. PW is quite good but has a little more false alarms. GH works fine in most cases but

Table 1: For gray level images

Episodes	PW	LK	GH	LH	NC
(1) fade out	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)
(2) moving object (a train)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)
(3) zooming with panning	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)
(4) conversation indoor	M(0)/F(3)	M(3)/F(19)	M(5)/F(43)	M(1)/F(16)	M(0)/F(0)
(5) gun fight	M(0)/F(1)	M(3)/F(10)	M(5)/F(17)	M(2)/F(14)	M(0)/F(0)
(6) outdoor scene	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)
(7) raining	M(0)/F(0)	M(0)/F(1)	M(0)/F(1)	M(0)/F(1)	M(0)/F(0)
(8) gradual transition	M(0)/F(0)	M(0)/F(2)	M(0)/F(2)	M(0)/F(2)	M(0)/F(0)
(9) moving car with panning	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)
(10) road scene	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)
(11) fight in the darkness	M(5)/F(0)	M(4)/F(16)	M(4)/F(15)	M(4)/F(33)	M(3)/F(1)
Time (seconds/frame)	0.19	10.67	0.14	0.38	0.06
*M(n) denotes missing n breaks. *F(n) denotes n false alarms.					

poor for the three episodes ((4),(5) and (11)). For every algorithm, the performance for HSV images is better than the one for the gray images and the price is time consuming. As HSV model is more sensitive to light change than the gray value, false alarms of using HSV model are little more than the ones of using gray level while the images have a lot of light changing, like the episode 5 (gun fight). For both HSV and gray images, Net comparison has the best performance and is two to three times faster than the others. From the experiments, we may also observe that although histogram has several advantages [13], and color histograms of multicolored objects can provide a robust and efficient way for representing the color objects [14], it is not good for dark or noisy video sequences representation.

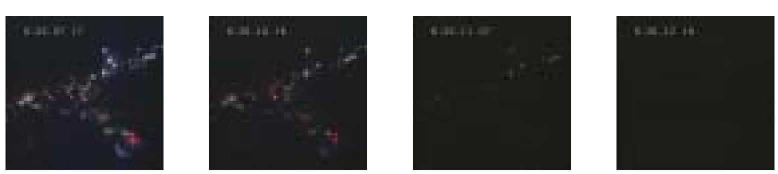

Figure 5: Fade out

Table 2: For color(HSV) images

Episodes	PW	GH	LH	NC
(1) fade out	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)
(2) moving object (a train)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)
(3) zooming with panning	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)
(4) conversation indoor	M(0)/F(4)	M(2)/F(7)	M(0)/F(0)	M(0)/F(0)
(5) gun fight	M(0)/F(10)	M(5)/F(21)	M(0)/F(0)	M(0)/F(0)
(6) outdoor scene	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)
(7) raining	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)
(8) gradual transition	M(0)/F(0)	M(0)/F(5)	M(0)/F(1)	M(0)/F(0)
(9) moving car with panning	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)
(10) road scene	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)	M(0)/F(0)
(11) fight in the darkness	M(2)/F(1)	M(2)/F(13)	M(3)/F(1)	M(0)/F(0)
Time (seconds/frame)	16.43	18.72	19.51	4.98
*M(n) denotes missing n breaks. *F(n) denotes n false alarms.				

5 CONCLUSION

We have discussed the various algorithms and the experimental results. We found that the new method, Net comparison, has the best performance for either gray-level or color images although it checks only part of the images. The reason is that the method choose a reasonable size of base windows to make use of the spatial information. Actually, pairwise, likelihood and local histogram are all trying to use the spatial information, the areas they used are either too small (even one pixel) to avoid sensitive or too large to keep the most spatial information. As the algorithm, Net comparison, is much faster than the others and so more practical than them. Further work includes three aspects. (1) Test the algorithm NC on various movies to find a way of automatically setting the thresholds and other parameters used in the algorithm. According to [3], the adaptive and multiple thresholds instead of the single threshold which we used in our experiments will improve the results. (2) Test NC on merely intensity which is considered as the most important color feature[15, 8]. (3) Test the algorithm on the uniform color space [16, 17].

References

[1] A. Nagasaka and Y. Tanaka, "Automatic video indexing and full-video search for object appearances," in *IFIP Transactions A-7, Visual Database System II*, (North-Holland), pp. 113–127, Elsevier Science Publishers B.V., 1992. Edited by E. Knuth and L. M. Wegner.

[2] C. M. Lee, S. W. Cheng, and M. C. Ip, "Camera break detection algorithms and their evaluation," tech. rep., HKUST-CS93-10,Dept. of Computer Science, Hong Kong University of Science and Technology, 1993.

[3] C. M. Lee and M. C. Ip, "Camera break detection in color video sequence," in *IPAR workshop on Machine Vision Application*, (Kawasaki, Japan), pp. 502–505, Dec. 1994.

[4] H. J. Zhang, A. Kankanhalli, and S. Smoliar, "Automatic Video Partitioning and Indexing," in *IFAC'93 world congress*, pp. 18–23, 1993.

[5] H. J. Zhang and S. W. S. Atreyi Kankanhalli, "Automatic partitioning of full-motion video," *Multimatia Systems*, vol. 1, no. 1, pp. 10–28, 1993.

[6] D. H. Ballard and C. M. Brown, *Computer Vision*. New Jersey: Prentice-Hall, 1982.

[7] M. W. Schwarz, W. B. Cowan, and J. C. Beatty, "An Experimental Comparison of RGB, YIQ, LAB, HSV and Opponent Colour Models," *ACM Transaction on Graphics*, vol. 6, no. 2, pp. 123–158, 1987.

[8] Y. Ohta, T. Kanade, and T. Sakai, "Color Information for Region Segmentation," *Computer Graphics and Image Processing*, vol. 13, pp. 222–241, 1980.

[9] R. C. Gonzalez and R. E. Woods, *Digital Image Processing.* USA: Addison-Wesley Publishing Company, Inc., 1992.

[10] D. C. Tseng and C. H. Chang, "Color Segmentation Using Perceptual Attributes," in *Proceedings. 11th IAPR International Conference on Pattern Recognition. Vol.III. Conference C: Image, Speech and Signal Analysis*, (Los Alamitos, CA, USA), pp. 228–231, IEEE Comput. Soc. Press, 1992.

[11] X. Y. Lin and S. Y. Chen, "Color Image Segmentation Using Modified HSI System for Road Following," in *Proceedings of the 1991 IEEE International Conference on Robotics and Automation*, (Los Alamitos, CA, USA), pp. 1998–2003, IEEE Comput. Soc. Press, 1991.

[12] Z. G. Zhu, G. Y. Xu, S. Y. Chen, and X. Y. Lin, "Dynamic Obstacle Detection through Cooperation of Purposive Visual Modules of Color, Stereo and Motion," in *Proc. IEEE RA'94*, 1994.

[13] M. J. Swain and D. H. Ballard, "Indexing Via Color Histograms," in *Image Understanding Workshop*, pp. 623–630, 1990.

[14] M. J. Swain and D. H. Ballard, "Color Indexing," *International Journal of Computer Vision*, vol. 7, no. 1, pp. 11–32, 1991.

[15] R. I. Taylor and P. H. Lewis, "Color Image Segmentation Using Boundary Relaxation," in *Proceedings. 11th IAPR International Conference on Pattern Recognition. Vol.III. Conference C: Image, Speech and Signal Analysis*, (Los Alamitos, CA, USA), pp. 721–724, IEEE Comput. Soc. Press, 1992.

[16] S. Tominaga, "Color Classification of Color Images Based on Uniform Color Spaces," *Human Vision and Electronic Imaging: Models, Methods, and Application*, vol. 1249, pp. 356–365, 1990.

[17] M. Celenk, "Colour Image Segmentation by Clustering," *IEEE PROCEEDINGS-E*, vol. 138, no. 5, pp. 368–376, 1991.

Figure 6: Moving object (a train)

Figure 7: Zooming with panning

Figure 8: Indoor (dialog)

Figure 9: Gun fight

Figure 10: Outdoor scene

Figure 11: Raining

Figure 12: Gradual transition

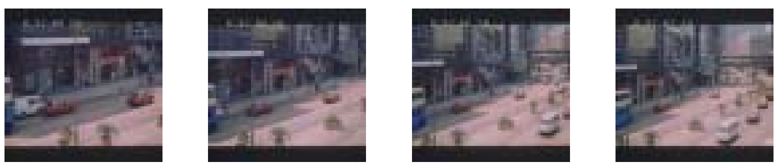

Figure 13: Moving car with panning

Figure 14: Road scene

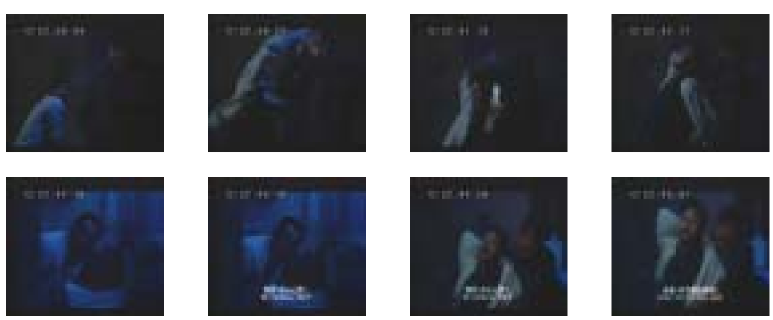

Figure 15: Fighting in dark