MOVING OBJECT DETECTION FROM MPEG BIT STREAM

Zhonghua LIANG[1] Ping WANG[1] Zheng TAN[1]

School of Elec. & Info. Eng., Xi'an Jiaotong University. 710049 P.R.C.

Abstract: In this paper we'll discuss how to detect the moving objects with scene changing directly from MPEG bit stream and propose an efficient method to eliminate some pseudo "background-motion" due to camera operations. A rectifying operator is introduced to deal with pseudo "background-motion". Experimental results have shown that the proposed approach can be competent for certain application occasions such as real-time video surveillance, object tracking system.

1. Introduction

Coupled with the significant advance in computer technology and the growth of Internet, multimedia information has become the mostly available and widely used media in people's daily life. Applications such as network transmission, VCD, DVD, interactive TV, distance learning and digital video broadcast usually generate and use a mass of video data. All these information have already been processed by multimedia compression technologies, in which MPEG1 and MPEG2 are the most widespread international standards now.

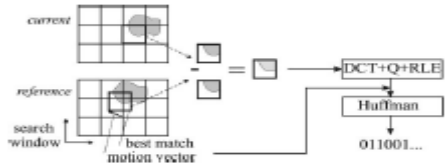
As nowadays video is increasingly stored and moved in compressed format (e.g. PEG 1, 2), it is highly desirable to develop methods that can operate directly on the MPEG coded stream. So working in the compressed domain can be of great practical significance. Compared to those working in the uncompressed domain, these approaches have the following characters. First, by not having to perform decoding/re-decoding, processing video directly in the compressed domain can save the whole processing time greatly. Secondly, the quantity of video bit-stream data is reduced than the data of original, so the computational complexity is reduced and higher processing-efficiency can be obtained. Thirdly, the video bit-stream already contains certain useful information, such as motion vectors (MV), DC terms and so on, that are suitable for video analyzing and processing [2]. Fourthly and unfortunately, MPEG video bit-streams have lost spatial characters of image which are familiar to us; they are only a succession of data, which will bring some trouble and inconvenience to video analyzing and processing.

In the following content, our main topic will be focused on information analyzing and processing in video bit-streams especially moving object tracking and detecting. All experiments are designed with MPEG2 standard, and considering the compatibility of PS (Program Stream) of MPEG2 to MPEG1, we use both (MPEG1 video stream and MPEG2 PS) in our experiments.

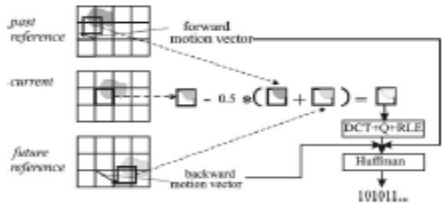
2. Information detection towards moving object

2.1 Motion information contained in the coded MPEG stream

We know that MPEG standard manifests perfectly in information redundancy compression, and it uses the following two key-techniques: 1) In transform domain, utilize block-based compression, namely Intra coding, to capture spatial redundancy, and Intra (*I*) frames are coded by this means; 2) In temporal domain, introduce MB-based motion prediction and compensation to reduce temporal redundancy. *P* (*predicted*) frames and *B* (*Bi-directional*) frames usually use this technique. Later we'll see that it can also provide us advantageous condition to process coded video stream directly.



a. Forward prediction for *P* frames



b. Interpolated prediction for *B* frames

Fig. 1. Two types of motion prediction coding

As Figure 1 shown, *P* frames are coded with forward motion compensation using the nearest previous reference (I *or P) pictures*. *B* frames are also motion compensated with respect to both past and future reference frames. In the case of motion compensation, for each 16×16 MB (macroblock) of the current frame the encoder finds the best matching block in the respective reference frame(s), calculates and DCT-encodes the residual error and also transmits one or two motion vectors. When the best match block is found, the residual error between the current coded MB and the best match block should be the least.

The expression of MV(motion vector) is：

$$MV(H,V) = \underset{h,v}{Min} \sum_{y=1}^{16}\sum_{x=1}^{16} \left| C(x,y) - P(x+h, y+v) \right| \quad \cdots(1)$$

And residual error is defined by the following expression**:**

$$\Delta MB_{xy} = C(x, y) - P(x + H, y + V) \quad \cdots (2)$$

Where $C(x, y)$ represents the data of the current MB, and $P(x+h, y+v)$ is the data of candidate MB in reference frame(s), MV means displacement between the current MB and the best match MB.

So we can gain the current MB's position with best matching strategy and we are also inspired by the fact that expression (1) contains the variety of moving object's position in different *P* (or *B*) frames, in other wards, it is possible to gain the feature information of moving object with only decoding MB's, MVs and locations from the compressed video stream.

2.2 Moving Object detecting algorithm based on MVs(motion vectors)

According to the discussions above, we propose an algorithm to detect moving object based on MVs and the flow chart is showed in Figure 2. The algorithm includes three main parts: the first is extracting moving character information directly from compressed video stream, then analyzing and processing these character information, and last marking the moving object's region. In our experiments, we only decode character information contained in *P* frames. Figure 3 shows an example of application occasions by MV patterns; here we can see that each MV pattern has marked the moving regions in a picture. In this case, the most important feature of video is that background is stationary and the camera is fixed (or no camera operations). Under this condition, MV pattern can achieve favorable performance. So MV patterns can be applied directly in certain occasions such as real-time visual supervision.

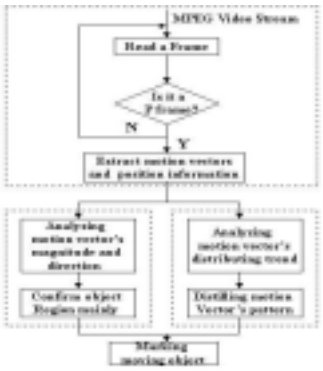


Fig.2. Moving Object detecting Algorithm flow chart

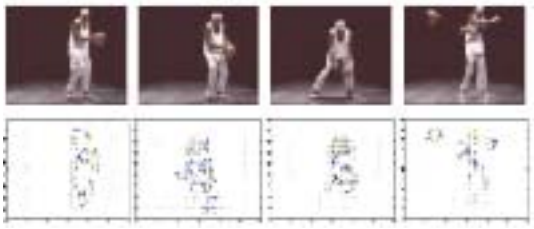


Fig.3. An illustration of MV patterns

3. Camera operation's negative effect upon MV pattern

In most cases, camera operations frequently occur during the course of shooting. Therefore, in order to enhance the practicability of the proposed algorithm, it is necessary to resolve some problems such as how to identify camera operations occur in the video stream and how to eliminate the negative effect caused by them.

Basic camera operations are presented in Figure 4. Generally camera operations are classified as following [1]:

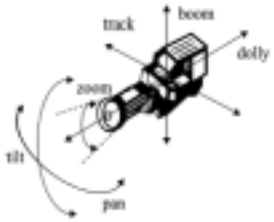


Fig.4. Basic camera operations

Fixed (the camera is stationary and focal length is invariable);

Zooming (focal length change of a stationary camera);

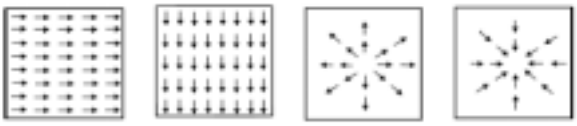
Panning/tilting (camera rotation around its horizontal/vertical axis);

Tracking/booming (horizontal/vertical transverse movement);

Dollying (horizontal lateral movement).

Accordingly, camera movements exhibit specific patterns in the field of MVs, as shown in Figure 5[1].

As Figure 5 shown, stationary background may become "moving" region due to camera operations. In fact the actual position of background has not been changed. It is camera movements that "change" the stationary background's position in a shot. So the feature information of moving object may be submerged in the noise of background's "moving" and that will cause negative effect upon our task of moving object analyzing and detecting. We call this negative effect "pseudo motion"



Left pan **Up tilt** **Zoom in** **Zoom out**

Fig.5. MV patterns resulting from several camera operations

.3.1 Camera operation recognition

When we analyze a MV pattern, for the sake of analyzing motion vectors well and truly, we should detect camera operations and take some measures to eliminate their effect before detecting the moving object.

Zhang et al.[1] apply rules based on the analysis of MV field to detect pan/tilt and zoom in/zoom out. As Figure 5 shown, during a pan most of the MVs will be parallel to a modal vector that corresponds to the movement of the camera. This is expressed by the following inequality:

$$\sum_{b=1}^{N} \left| \theta_b - \theta_m \right| \le T \quad \cdots (3)$$

Where θ_b is the direction of the MV for block *b*, θ_m is the direction of the modal vector, *N* is the total number of blocks into which the frames is partitioned and *T* is a threshold near zero.

In the case of zooming, the field of MVs has focus of expansion (zoom in) or focus of contraction (zoom out). Zooming is determined on the basis of "periphrastic vision", i.e. by comparing the vertical components v_k of the MVs for

the top and bottom rows of a frame, since during a zoom they have opposite directions. In addition, the horizontal components u_k of the MVs for the left-most and right-most columns are analyzed in the same way. Mathematically these two conditions can be expressed in the following way:

$$\left|v_k^{top} - v_k^{bottom}\right| \geq \max\left(\left|v_k^{top}\right|, \left|v_k^{bottom}\right|\right)$$
$$\left|u_k^{left} - u_k^{right}\right| \geq \max\left(\left|u_k^{left}\right|, \left|u_k^{right}\right|\right) \quad \cdots (4)$$

When both conditions are satisfied, a zooming operation is declared.

By analyzing panning/tilting and tracking/booming operations, it is easy to find the common ground they act on MV field, that is they both exert a "pseudo motion field" to the whole MV field in invariable directions; the only difference between them lies in the fact that the "pseudo motion field" caused by panning/tilting is on the whole a "invariable field" in which each vector's magnitude is equal to another's, and the "pseudo motion field" caused by tracking/booming does not have the feature. This is because when panning/tilting operation occurs, the tangent speed of each point on the arc formed by the camera's rotation is unique, and during tracking/booming operations, there is no homologous feature. In reality, most directions of panning/tilting and tracking/booming are generally parallel to horizontal/vertical direction. For instance, during panning or tracking operation, the camera rotates or moves in a horizontal plane, so the direction of modal vector $\theta_m \approx 0$, and with geometry knowledge we know when θ_m is very little, the expression $tg\theta_m = v_k/u_k \approx \theta_m$ comes into existence.

In the case of panning, because the magnitude of "pseudo motion field" is unique, surely $v_k/u_k \approx 0$; And in the case of tracking, although the magnitude of "pseudo motion field" is variable, due to the feature of moving horizontally, the expression $v_k/u_k << 1$ is applicable all the same.

Therefore, panning and tracking operation can be both detected by select an appropriate threshold near zero. Similarly, in the case of tilting or booming, there exists homologous condition ($u_k/v_k \approx 0$ or $u_k/v_k << 1$).

As stated previously, we propose a more applicable rule to detect the camera's panning and tracking operation, and the rule is defined as following:

Step 1: analyze all MBs of a *P* frame and count the number of prediction-coded MBs *N*;

Step 2: introduce variables T_k and *M*, and define u_k as the horizontal component of a prediction-coded MB's MV, and v_k the vertical component, here ϖ is a positive threshold near zero.

Step 3:

$$T_k = \begin{cases} 1 . \text{当} \left|\frac{v_k}{u_k}\right| \leq \omega \\ 0 . \text{当} \left|\frac{v_k}{u_k}\right| > \omega \end{cases} \quad k = 1,2,....., N \qquad (5)$$

Step 4: ; $M = \sum_{k=1}^{N} T_k$

Step 5:

When $M/N \geq \delta$, panning or tracking is declared.

We can also detect tilting or booming operation with similar rule by substituting u_k/v_k for v_k/u_k (shown in Step 3). Here the value of δ can be set on the basis of practical conditions. In most cases, the moving object is small comparing to the background, so δ can get a value near 1, such as 0.7 or 0.8.

3.2 eliminating the negative effect due to "pseudo motion"

According to the rule proposed previously, we can detect panning/tilting or tracking/booming operation, then we should eliminate the negative effect upon MVs field. Therefore, we introduce a rectifying operator based on prediction-coded MBs, namely, each prediction-coded MB in a *P* frame will be processed by following method (here we assume that panning/tracking operation occurs):

$$u_k = \begin{cases} G \cdot u_k, \left|\frac{v_k}{u_k}\right| > \omega \\ 0, \left|\frac{v_k}{u_k}\right| \leq \varpi \end{cases} \qquad v_k = \begin{cases} G \cdot v_k, \left|\frac{v_k}{u_k}\right| > \omega \\ 0, \left|\frac{v_k}{u_k}\right| \leq \varpi \end{cases} \qquad (6)$$

Where *G* is the gain factor and generally *G* = 1. After processing MV patterns with the rectifying operator, we are able to basically eliminate the "pseudo motion filed" due to panning/tracking operation by adjusting ω to an appropriate value. And in the case of tilting/booming operation, so can we do (remember substituting u_k/v_k for v_k/u_k).

Next we'll discuss the final effect of the moving object detecting algorithm including rectifying operator with experimental results.

4. Experimental results

We have applied the proposed algorithm to several sorts of video stream. Figure 6 illustrates the result of one sample video. The pictures in row 1 are extracted from original video stream, and the results in row 2 represent the corresponding MV patterns without using rectifying operator, then the results processed by rectifying operator are shown in row 3.

All sample video streams adopted in our experiments are compressed with MEG1/2 encoders and the lengths of them are generally from 2 *s* to 15 *s* (about 50 ~ 400 frames). As Table 1 shown, these sample video can be classified by video contents and sources.

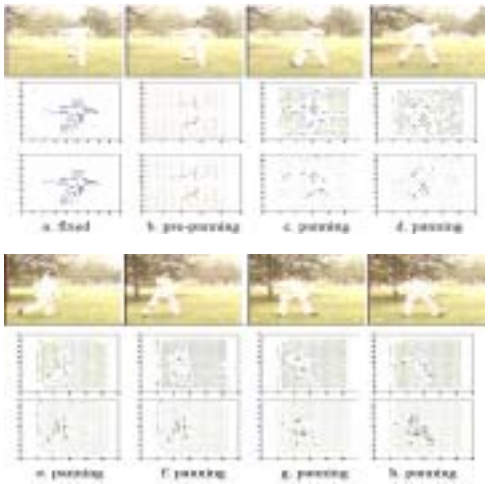


Fig.6. The elimination of "Pseudo motion field"

Table1 the Classifications of sample video streams

Video Sort	Video Content	Video source	Quantity (segment)
A	News,interviews, entertainments. etc	Web download, VCD	20
B	Music, or other performance	Web download, VCD	20
C	Sports games such as basketball. etc	Web download, VCD	20
D	Video surveillance or tracking	Captured by camera	40

The main characters of sample video streams are summarized in Table 2.

Table 2 the Characters of sample video streams

Video Sort	Characters of camera operations						Moving characters	
	pan	tilt	track	boom	zoom	dolly	Object size	moving speed
A	√	○	√	○	○	○	moderate	low
B	√	√	√	○	○	○	moderate	moderate
C	√ √	√	√√	√	√√	√	small	high
D	√	○	√	○	○	○	moderate	high

The meaning of each symbol in Table 2：○—seldom occurs or absence ；√√—frequently occurs ；√— occurs but not so frequently.

As Table 2 shown, the main characters of video sort C are more complex than others. Moreover, in the case of multi-object, the phenomenon of overlapping between objects occurs frequently in video sort C. Therefore, the selection conditions to video sort C are more rigorous, for example, we can select those video streams in which camera operations are simple namely only panning/tilting or tracking/booming operations occur and the moving objects' size are moderate. To another three sorts of video streams, we have got satisfying effects with the proposed algorithm.

Table 3 The Adaptability of the proposed algorithm to camera operations in various video streams

Video	Characters of camera operations						general
	pan	tilt	track	boom	zoom	dolly	
A	√√	○	√√	○	○	○	good
B	√√	√√	√√	○	○	○	good
C	√	√	√	√	W	W	Need improved
D	√√	○	√√	○	○	○	good

The meaning of each symbol in Table 3：√√— better ；√— good ；W — worse ；○— without grade

When we processed the sample video streams with the proposed algorithm, the whole processing time of each video stream was less than the length of the stream itself. Therefore, the proposed algorithm is Real-time, and the rates of detecting accurately using the algorithm to various video streams are listed in following: Sort A 91%，Sort B 90%，Sort C 75% and Sort D 93%.

5. Conclusions

The moving object detecting algorithm proposed in this paper is based on compressed video and has achieved detecting the moving object by extracting the moving character information from coded video stream. In the cases of camera operation, a rectifying operator is introduced to deal with pseudo "background-motion". Therefore, the proposed algorithm can be applied in certain application occasions such as Real-time surveillance, object tracking etc.

However, the adaptability of the proposed algorithm to various camera operations, such as zoom or dolly, need to be improved. Moreover, how to match multi-object in each frame and achieve tracking multi-object simultaneously, along with the potential approaches namely taking full advantage of the MPEG 4 standard [3], will be emphasized in our future researches.

6. References

[1] Irena Koprinska，Sergio Carrato；Temporal Video Segmentation：A Survey；Signal Processing：Image Communication Volume：16，Issue：5，January，2001，pp. 477-500

[2] A.Benzougar，P.Bouthemy，R.Fablet；MRF-based moving object detection from MPEG coded video；Image Processing, 2001. Proceedings. 2001 International Conference on，Volume: 2 , 2001 Page(s): 402 -405 vol.3

[3] Yiwei Wang，J.F.Doherty，R.E.Van Dyck；Moving object tracking in video；Applied Imagery Pattern