I-064

動き補償ブロックサイズと順方向動きベクトル情報に基づく H.264/AVC ビデオにおける実時間移動物体追跡

Real-time Moving Object Tracking in H.264/AVC Video based on Motion Compensation Block Size and Forward Motion Vector Information

翁 素文† ルイス ロヨラ‡ バスカラ ヌグラハ† 森田 啓義†
Suwen Weng† Luis Loyola‡ I G. B. Baskara Nugraha† Hiroyoshi Morita†

1. Introduction

Moving object tracking is one of the most important tools for many video applications, such as video indexing, video surveillance system, video coding, etc. Many researchers have attained good results on performing tracking in pixel domain. However, due to the computational complexity of pixel domain method, researchers have also proposed object tracking technologies in compressed domain that allow us to process compressed video stream, e.g. from surveillance camera or a server, without fully decoding the stream.

Generally object tracking system consists of two steps: object detection and object correspondence. Every tracking method requires an object detection mechanism in every frame. Recently research on moving object detection in compressed domain has been focused on H.264/AVC video stream, which is widely used in many applications and devices. Most of the detection algorithms for H.264/AVC videos are based on detection methods for MPEG-2 video, which can be basically classified into two categories: motion vector (MV) based and residual information based algorithms. MV-based methods classify motion vector into background and foreground parts, which usually have high computational complexity. On the other hand, residual information based algorithm cannot be directly implemented in H.264/AVC video because of intra-prediction in I frame and variable size of Group of Pictures (GOP). Some researchers have proposed moving objects detection methods in H.264/AVC video using the skip type macroblock (MB) [8], and the size of encoded data of each MB [9]. Both of these methods only support baseline profile H.264/AVC video, but not the main profile format, which is widely used in services such as high definition TV and online video website, e.g. YouTube. Therefore there is a need of build real-time moving object detection algorithm for main profile H.264/AVC video.

The aim of object correspondence is to generate the trajectory of an object over time. A few moving object correspondence algorithms have been proposed for H.264/AVC compressed video. A conventional method for MPEG-2 compressed video uses DCT coefficient or color information that is decoded from I frame for objects correspondence. However, H.264/AVC compressed video has a variable GOP size. The number of I frames is inversely proportional to the size of GOP. Besides, intra-prediction is used in I frame, thus a conventional object correspondence algorithm for MPEG-2 video cannot be directly applied for H.264/AVC video.

In order to meet different needs for variety of multimedia applications, in this paper we proposed a detection method using variable motion compensation block size and motion vector information to segment the moving object from the background, which has average 90% precision and 86% recall rate for all type of H.264/AVC profiles.

We also proposed one real-time tracking system based on Kalman filter and forward motion vectors in P and B frames to keep tracking the trajectory of multiple objects even in presence of occlusion situation. Experimental results show that our proposed tracking system has 92% of precision rate and 96% recall rate in average in three outdoor surveillance scenarios. The average processing time of proposed tracking system is 8.73ms for each frame.

The outline of this paper is described as follows. Section 2 describes the related work, while section 3 addresses the moving object detection algorithm. An object tracking method is presented in section 4. Experimental results are shown in section 5, and finally the conclusion and future work of this research are in section 6.

2. Related Work

The H.264/AVC standard is the most widely used codec in the present time. It can achieve higher compression and better quality than any of the previous standards, which benefit online multi-media applications, such as HDTV delivery, online video uploading and downloading as well as video sharing and recording.

Most of moving object detection methods for H.264/AVC video come from previous MPEG-2-based algorithm, which can be classified into two groups: a motion vector based method and a residual information based method.

Motion vector based methods uses only motion vector information. H. Zen in [3] detected objects based on the magnitudes of motion vectors. Direction of motion vector is used for grouping objects. Ashraf et al. [4] detected objects based on the value of motion vectors after Gaussian and median filtering. Ritch and Canagarajah [5] detected moving objects by removing global motion vectors. Most of the moving object detection algorithms in H.264/ACVC are based on MV fields. Thilak [6] proposed a system to track moving objects in H.264/AVC video based on the magnitude of motion vectors with a prior knowledge of objects' size. However, the main problem with this approach is that motion vectors with different directions may be grouped into the same object. Liu [12] uses a median filter to remove the noise and smooth the input MV field. A binary partition tree filtering is used to segment moving objects. Zeng et al. [7] classified motion vectors into foreground, background and noisy MVs to detect moving objects in H.264/AVC compressed domain. Markova labeling is used to track objects. Liu [12] and Zeng et al. [7] have high detection accuracy, but they need several threshold value to segment moving objects for each video.

Residual information based methos uses DCT coefficient or color information which partially decoded from I frame. Schonfeld and Lelescu method [1] acquired objects from I frame by using template matching. While Manerba et al. [2] also detected moving objects by removing global motion vectors on P-frames and additionally use DCT coefficients to obtain objects on I-frames. However, because of the intra-prediction in I frame for H.264/AVC video, a prediction block is formed based on previously encoded and reconstructed blocks. Transform coefficients cannot represent the pixel information. Moreover, due to the variable size of GOP, the number of I frames in H.264/AVC will vary with the changes of compression rate, thus I frame is not reliable for object detection and tracking.

Some researchers are using the features of H.264/AVC to achieve moving object detection. Wonsang et al. [8] uses skip MB to remove background MBs, and uses spatial filtering and temporal filtering to remove noisy foreground MBs. However the number of skip MB is dependent on the resolution of the video and also the type of frames. Since P frames contain less number of skip MBs than B frames do, and skip MB is also used to encode homogeneous color area within a large size of moving object, the method by Wonsang et al. [8] cannot be directly used for main profile H.264/AVC video. Size (in bits) of MB and

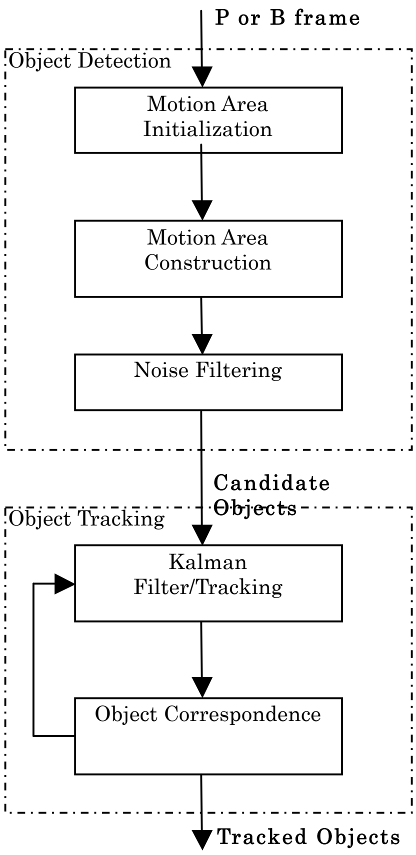


Fig.1 Object Tracking System Diagram

transform coefficients are used in Chris [9]. It can achieve detection in compressed domain, but it needs several predefined threshold, which depend on the resolution of each video. Motion vectors associated to the MBs and motion compensation modes are used for segmentation in [10]. The algorithm uses fuzzy logic and allows describing position, velocity and size of the detected regions, demanding high computational complexity.

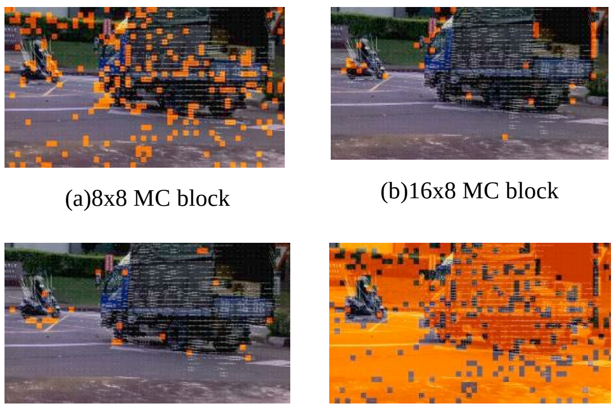
Noticing that previous detection algorithm for H.264/AVC cannot meet the need of real-time processing for all type profiles of H.264/AVC video, in this paper we propose a detection method using motion compensation block size and motion vector information.

3. Moving Object Detection

The whole object tracking system can be divided into 2 sub-systems: detection and tracking as shown in fig.1. Moving objects can be detected by segmenting the foreground from the rather static background (with static camera). Here propose a moving object detection approach using variable motion compensation (MC) block size and motion vector information.

3.1 Variable Motion Compensation Block Size

H.264/AVC advanced coding supports more flexibility in the selection of motion compensation block sizes than the previous standards. Each MB (16x16 samples) can be split up in four ways for motion compensation either as one 16x16 MB partition, two 16x8 partitions, two 8x16 partitions, or four 8x8 partitions. If the 8x8 mode is chosen, each of the four 8x8 sub-MBs within the MB may be split in a further 4 ways. This method of partitioning MBs into motion compensation sub-blocks of variable size is known as the tree structured motion compensation [13]. Figure 2 shows an example of the distribution of each MC block size of MB in one frame. We can observe that 16x16 MB partition is most used in frame, and the number 16x8 and 8x16 MB partitions are fewer than the number of 8x8 MB partition.



(a)8x8 MC block (b)16x8 MC block

(c)8x16 MC bock (d)16x16 MC block

Fig.2 MBs with different MC block size

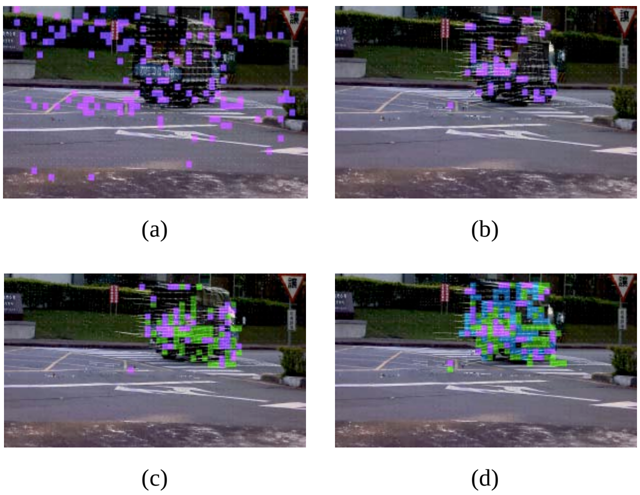
Each partition requires a separate motion vectors. Choosing a large partition size means that a small number of bits are required to transmit the choice of motion vector(s) and the type of partition but the motion compensated residual may contain a significant amount of energy in frame area with high detail. Choosing a small partition size may give a low-energy residual after motion compensation but requires a large number of bits to transmit the motion vectors and choice of partition(s). A large partition size is appropriate for homogeneous area of the frame while a small partition size may be beneficial for detail (non homogeneous) area such as edges/boundaries or rapid movement. The encoder will selects the best partition size for each part of the frame to minimize the amount of information to be sent. The boundary of moving object usually requires smaller size of partition because of the detail edge information, which can be used for moving object detection.

3.2 Motion area initialization

Since the most used partitions in motion areas have a size of 8x8, the invention locates all these 8x8 partition of MBs in a single frame, as initial step for moving object detection. Most of the 8x8 partition of MBs is related with moving object. However, due to motion estimation errors, shadows and luminance changes, many 8x8 partitions of MBs may be found that do not belong to moving objects. Thus, to further segment the moving object area, the solution removes the unrelated MBs that also use a partition size of 8x8.

MBs with 8x8 partitions may contain four motion vectors. Each motion vector has horizontal and vertical value; the size of motion vector is defined as the norm of horizontal and vertical value.

Therefore, the 8x8 partitions of MB that contain non-zero size of motion vector are defined as motion area, while the 8x8 partitions that have zero size of motion vector are discarded, as they are usually caused by noise such as shadow and luminance change, making them unrelated with the moving object. If there are 4 motion vectors with non-zero norm value in a MB, then we decide this MB belongs to moving object.



(a) (b)

(c) (d)

Fig.3 Motion Area Construction

3.3 Motion area construction

Motion area initial step locates part of moving object area. In order to detect whole moving object, we need expand the motion area from initial 8x8 partition MBs into its neighboring bigger partition size MBs.

Direction and size of motion vector will be used to group MBs for moving object. Each motion vector has a direction in degree. Instead of using exact degree for motion vector's direction, we quantize motion vector directions into 8 partitions. The direction of motion vector for each partition is defined as Fig. 4. This partition will be used as a guideline for grouping the motion vector starts that from 8x8 partitions. We assume that two blocks have the same direction if their direction partitions are neighbors to each other, e.g., Partitions 1 and 7 are neighbors to partition 0.

Additionally, each MB has 8 neighboring MBs and each MB has distinct number of motion vectors. For example, if the MB is divided into two 16x8 or 8x16 partitions, there may be two motion vectors. We will take the average size and direction of all the motion vectors inside each MB. If the neighboring MB of 16x8 and 8x16 partitions has non-zero size, it will be grouped into the same motion area. If the neighboring MB with 16x16 partitions has non-zero size and same quantized direction range with the center MB (see Fig.4), it will be decided as part of the motion area too.

After locating the 8x8 partition of MBs with non-near zero size of motion vector (Fig. 3(b)), we can construct motion area by doing the expansion to its neighboring MBs. Firstly, motion area will be expanded by adding MBs with 16x8 and 8x16 partitions (Fig. 3(c)). The further expansion will add MBs of 16x16 partition that contains non-zero motion vector and have the same direction range with center MBs (Fig. 3(d)). The figure 3 is the illustration for the each step and the result of motion area construction processing.

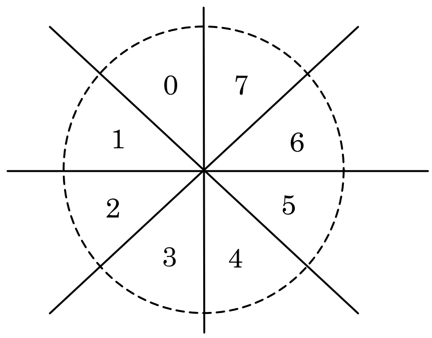


Fig.4 MV Direction Partition

3.4 Noise filtering

After the process of motion area construction, the detected moving object, which is called as candidate in following paper, includes some "noise" area and "real" moving object of interest, e.g., pedestrian, car, bike, etc. In order to clean the noise we apply simple noise filtering using morphological filter (dilation and erosion) to remove small stand-alone motion area and fill the gap or join two or more motion area that are close to each other. We also defined the moving object of interest should be bigger than a threshold value, which is 4 MB size in our case.

4. Moving Object Tracking

In order to perform real-time tracking of the moving object, we apply Kalman filter [11] with observation of motion vector information. For surveillance video, there are many difficult scenarios such as occlusion (same moving direction and opposite moving direction), sudden stop, disappear, occluded by background. We proposed a method that use forward motion vector as reference to handling these situations (especially occlusion).

4.1 Kalman filter

While tracking the trajectory of multiple objects, each object is given a distinct label. To ensure that any object is assigned with the correct label, its position and moving direction in the next frame must be estimated as precisely as possible. We use discrete Kalman filter since it is suited to a non-stationary case when the precise nature of the modeled system is unknown. Moreover, comparing with particle filter and multiple hypotheses tracking, Kalman filter requires less computational complexity, which is suitable for real-time application. A lot of works on tracking multiple objects using Kalman filter have been reported so far [14][15]. However, most of them are based on pixel level processing, and most of the works assume that only position information can be observed from the system. In surveillance scenario, the velocity of moving object is not always constant and each object could have different speed. Hence, static motion model is not adequate to produce precise prediction. We use motion vector information to update motion model automatically, which is proposed in our previous work [16]. First of all, to explain our methodology, we consider the case where there is only one moving object in a frame. Let x_k be a four-dimensional column vector $(x_k, y_k, \Delta x_k, \Delta y_k)$, called state, related to a moving object in a frame at step k where (x_k, y_k) are pair of x-y coordinates of its center of gravity in Cartesian coordinate and $(\Delta x_k, \Delta y_k) = (x_k - x_{k-1,} y_k - y_{k-1})$. We assume that an object has constant velocity within two sequential frames, and the result from the detection processing is the real location of moving object as following laws:

$$x_{k+1} = Ax_k + w_k$$

with a measurement:

$$z_k = Hx_k + v_k$$

where:

- A = the state transition matrix
- H = a measurement matrix
- w_k = a normal random variable with zero mean and covariance Q_k
- v_k = a normal random variable with zero mean and covariance R_k

And we also assume that the random variables in each of both sequences $\{v_k\}$ and $\{w_k\}$ are independent, respectively.

At each step k , the Kalman filter outputs the estimated state $\hat{x}_k$ and covariance $\hat{P}_k$ of the estimated error $x_k - \hat{x}_k$ using two operations: prediction and update. The prediction part is performed as follows:

$$\tilde{x}_k = A\hat{x}_{k-1} \tag{1}$$
$$\tilde{P}_k = A\hat{P}_{k-1}A^T + Q_k \tag{2}$$

When a measurement z_k is obtained, $\hat{x}_k$ and $\hat{P}_k$ are given by the following “update” equations:

$$K_k = \tilde{P}_k H^T (H\tilde{P}_k H^T + R_k)^{-1} \tag{3}$$
$$\hat{x}_k = \tilde{x}_k + K_k(z_k - H\tilde{x}_k) \tag{4}$$
$$\hat{P}_k = (I - K_k H)\tilde{R}_k \tag{5}$$

where $\tilde{x}_k$ is a priori state estimate at step k and matrix K_k is called Kalman gain.

The state transition matrix A and a measurement matrix H are defined as:

$$A = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

and then calculate Eqs (1) and (2). The values of $(\Delta x_k, \Delta y_k)$ are taken from the average of motion vectors of forward predicted MBs within the region related to a moving object.

In case of multiple moving objects, after the prediction part in Eqs (1) and (2), each measurement should be correctly associated with the corresponding object before the estimation is updated in Eqs (3) to (5). This matching procedure is done as follows:

1. Suppose that we have J moving objects in a frame and a priori state $\tilde{x}_k^{(j)}$ has been estimated for the j^{th} object at step k .
2. For each candidate object detected at step k we estimate its moving vector region $d_k^{(j)}$ based on the most dominant motion vectors' direction within the object's bounding box.
3. To a measurement, we assign $j*$ such that $\tilde{x}_k^{(j*)}$ is the most similar to it among $\tilde{x}_k^{(j)}$ $1 \le j \le J$ and $|\, d_k^{(j*)} - d_{k-1}^{(j)} | \le 1$. The similarity will be calculated by object correspondence.

4.2 Objects correspondence

After getting the prediction result for Kalman filter, we need to establish the corresponding relationship between candidate objects and previously tracked objects. The simplest method is based on nearest distance. However, if there is occlusion or multiple objects are moving close to each other, distance information is not enough to decide the corresponding. Although color information must be useful, it need decode the frames to get pixel information in H.264/AVC video. Therefore, we propose another approach using forward motion vectors in P and B frames.

The forward motion vector can represent the corresponding relationship between MB in the current frame and previous frame. It starts from current MB and points to the “best match” MB in reference frames (Figure 5). If there are a lot of motion vectors inside candidate objects referring from previously tracked object, this candidate object can be decided has relation/correspondence with that previously tracked object. Thus we can find the correspondence based on the percentage of reference forward motion vectors.

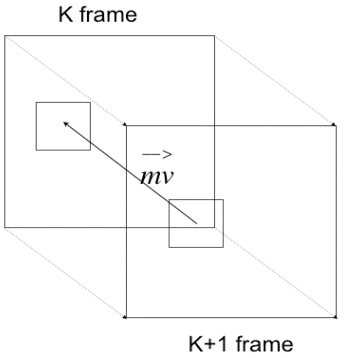


Fig.5 Forward Motion Vector

For each candidate object, we check the reference percentage (P) of motion vectors that point to the area of tracked object. We check which pair of candidate and tracked objects has the higher percentage, then the candidate object will be decided that has corresponding relationship with the tracked object. The reference percentage will be defined as follows:

$$P = \frac{N_{f\,mv}}{N_{mv}}$$

where $N_{f\,mv}$ is the number of motion vectors belonging to candidate object, which point to the tracked object's area. N_{mv} is the total number of motion vectors inside the candidate object. Figure 6 shows the example of objects correspondence using

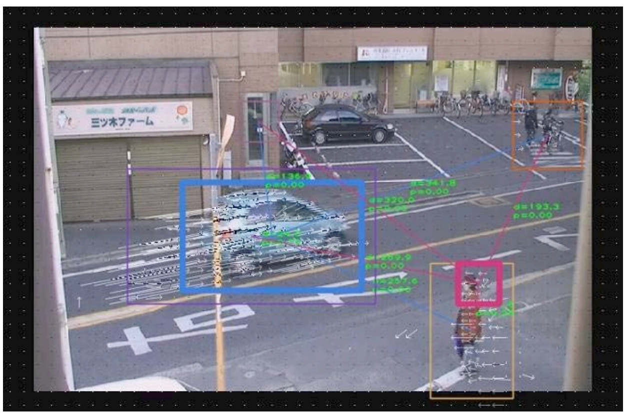


Fig.6 Objects Correspondence

reference percentage, there are three candidate objects (thin boundary line) inside current frame, and two tracked objects (thick boundary line). The line between candidate object and tracked object shows the distance and reference percentage between those objects.

4.3 Objects History

The trajectory of the object will be kept as long as the object is still in the scene, and can be detected within some duration of times (we chose 45 frames).

After object correspondence above, there might be detected a new candidate object that possibly was a noise area. And an object that only appears once and never appears for 5 frames is defined as noise object. Object that alternately appears and disappears quickly or never disappears in 5 frames is counted as a non-noise object.

5. Experiment Results

In our experiments, we used three different outdoor surveillance H.264/AVC videos. Video 1 is encoded using ffmpeg with three different profiles. The resolution of video is 720x480, and contains 1775 frames. These sequences contain difficult scenarios such as noise, shadows, luminance change and occlusions. Video 1 contains different speed and varying size of moving objects, video 2 contains most times of occlusion scenarios, while moving objects in video 3 are close to each other. The proposed method is implemented and testing with Apple's Macbook Pro computer with 2.6 GHz –Intel Core 2 Duo processor.

5.1 Processing time

Processing time is one the most important issues for real-time moving object tracking system. We measured the processing time required in our detection and tracking algorithm to track all of the moving objects in a frame. Table 1 depicts the processing time for each testing video. It shows that our method can work much faster than the presentation time of a video frame, which is about 8.73ms average. The average processing time is almost constant regardless of the number of the moving objects.

Table 1. Processing Time of Proposed Algorithm

Video	#frames	Processing Time
Video1(baseline)	1775	10.54 ms/frame
Video1(main)	1775	7.83 ms/frame
Video1(high)	1775	8.20 ms/frame
Video2	517	9.05 ms/frame
Video3	839	8.01 ms/frame

5.2 Detection evaluation

To detect moving object in H.264/AVC video, most algorithms use motion vector and skip MB information. Skip MBs can represent background area. Wonsang et al. [8] removes the entire skip MBs that belong to background firstly, and then filters the isolated and zero IT coefficient MBs in P frames of baseline H.264/AVC video. The remaining areas are detected as moving objects. Motion vector is also widely used for detecting moving object [6][7][12]. Previous works in [6][7][12] obtains good detection result but they are too complex for real-time processing. Thus we develop a real-time MV-based method [4] use magnitude of motion vector in each MB to distinguish foreground and background object. Firstly median filter is applied to smooth input motion vector field. Secondly a predefined threshold is utilized to segment moving object from background. If the magnitude of motion vector in MB is greater than threshold, current MB is detected as motion object area.

Proposed method uses motion compensation block size information to detect moving object. The Skip MB-based method also uses MB information that is same with proposed method. The MV-based method can be applied for all types of profile H.264/AVC video like the proposed method. Furthermore, both the MV-based method and the Skip MB-based method can achieve real-time processing. Therefore we compare the accuracy of proposed detection algorithm with these two methods based on an Object-based ground truth. The threshold value in the MV-based method is experimentally determined for each video sequence.

Table 2. Detection Evaluation

Video (Profile)	Proposed		MV-based[4]		Skip-MB[8]	
	P(%)	R(%)	P(%)	R(%)	P(%)	R(%)
Video1 (base)	**90.4**	83.2	64.0	79.4	86.0	**91.0**
Video1 (main)	**91.0**	84.0	60.0	76.0	82.0	**90.0**
Video1 (high)	**93.0**	86.0	66.0	67.0	76.0	**90.4**
Video2 (main)	**92.0**	**92.4**	69.0	73.4	74.0	87.0
Video3 (main)	**90.0**	**87.0**	72.4	76.0	72.0	84.6

P: Precision; **R**: Recall

Two metrics called precision and recall are used to describe the accuracy of detection. Precision and recall are defined as following equations:

$$\text{precision} = \frac{\#\,\text{of correct detections}}{\#\,\text{of correct detections} + \#\,\text{of false detections}}$$

$$\text{recall} = \frac{\#\,\text{of correct detections}}{\#\,\text{of correct detections} + \#\,\text{of miss detections}}$$

Table 2 shows the precision and recall obtained by proposed method and the other two methods. It can be observed that the MV-based method has the lowest precision and recall in all videos. Most compressed videos contain a lot of noisy motion vectors caused by decoding error, luminance changes or shadow. It is difficult to remove the entire noisy motion vectors in real-time processing. Therefore motion vector information is not reliable to detect moving objects. The Skip-MB based method achieves the highest recall for video 1 (baseline), while its precision is relatively low in other videos. Skip MBs can be used for removing a part of background area, however the remaining areas could contain lot of noises and moving objects, which led to the low precision results for the skip MB-based method. Furthermore, B frames contain less skip type MBs than P frames in main and high profile video, thus removing skip MB and the isolated MB in main and high profile video cannot segment moving objects from background area, which is the reason that the main and high profile video have worse precision result than the baseline profile video. Table 2 also shows that our proposed method achieves the highest precision than other approaches, and it also has almost similar precision and recall in different resolution. As results, motion compensation block size is more suitable for real-time moving object detection for all types of profile H.264/AVC video.

5.3 Tracking evaluation

We also use precision and recall rate to measure the tracking accuracy [17]. A correct tracking is defined as all moving objects should be tracked with distinguish label/ID related with previous frames. False tracking is that there is wrong assignment of ID or falsely track noise area as moving object. Miss tracking is that system only can track some of the moving object in frame.

Table 3. Tracking Evaluation

	Video1 (main)	Video2 (main)	Video3 (main)
Precision	95.0%	92.0%	90.7%
Recall	96.7%	98.0%	96.0%

Table 3 shows the evaluation result for tracking in three videos. The size of moving objects in Video 1 is different with each other, and most of objects such as car, motorbike have high velocity, which are the reasons that Video 1 have the highest precision and satisfying recall. Video3 has lowest precision and recall rate since Video 3 contain many objects moving toward same direction. Beside, most of objects inside video 3 are close to each other, which led to some of objects are falsely detected and tracked as one object. Video 2 have highest recall but low recall because of the occlusions In order to improve the result, we can add moving direction and size of moving objects as features for determining object correspondence.

5.4 Discussion

Figure 7-10 shows several samples of experimental results for the proposed tracking algorithm. It can be observed that the proposed algorithm can keep tracking the trajectory for different size and different speed. Even when the speed of moving object is too slow to be detected, we can predict the position based on previous velocity and position.

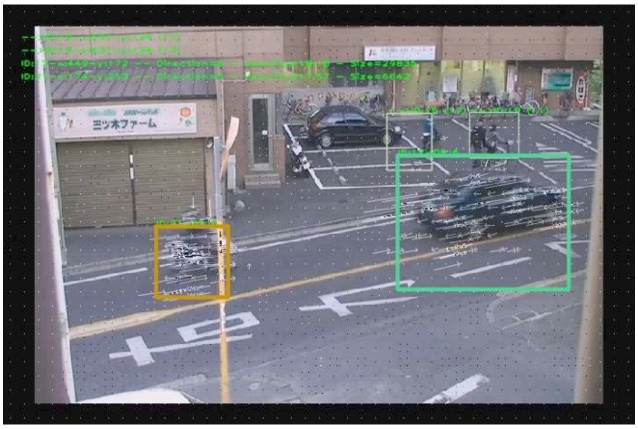


Fig.7 Tracking for Different Speed Objects

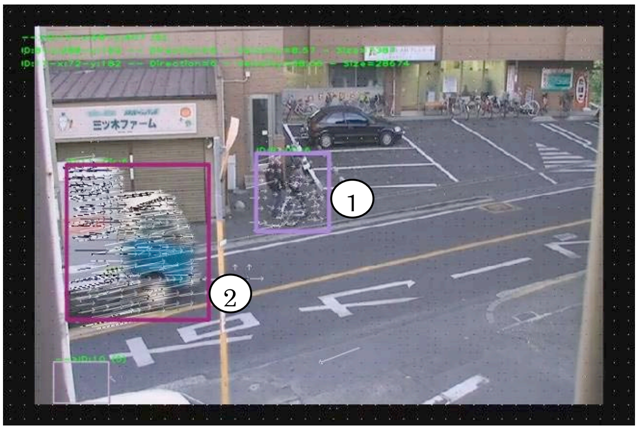


Fig.8 Before Occlusion

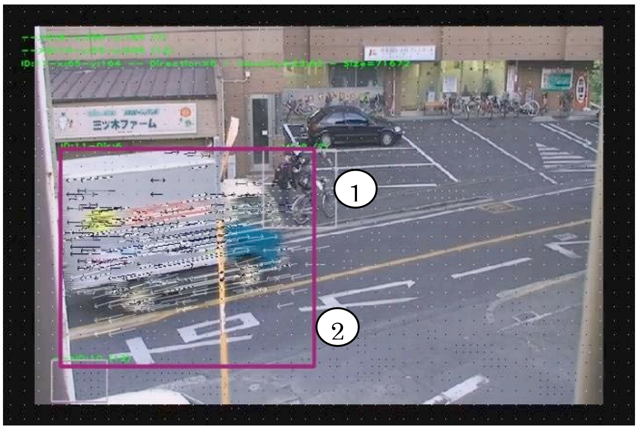


Fig.9 During Occlusion

Proposed tracking system can solve some occlusion situation (see Fig.8 to Fig.10). If the sizes of moving objects are different, each object will have different forward motion vector reference percentage during occlusion, which can be used to distinguish those objects after occlusion.

Figure 7 shows an example of four detected and tracked objects (pedestrian, riding bicycle, motor bike, and car). Each object has different speed and size, but the propose method can detect and track them properly.

Figure 8 show two detected and tracked objects (pedestrian and truck) some times before the occlusion. Figure 9 shows the moment just before occlusion happened, which the truck is in front of the pedestrian can be detected and tracked (thick boundary line), while the pedestrian cannot be detected but it state (position and velocity) is predicted (thin boundary line) by the system (Kalman filter).

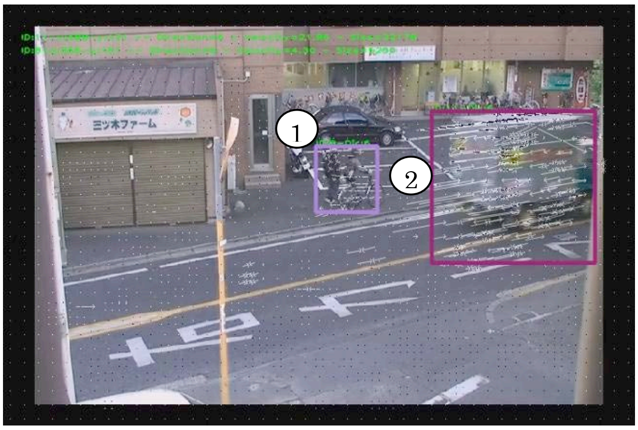


Fig.10 After Occlusion

Figure 10 shows the moment after the occlusion, which the occluded pedestrian still can be detected and tracked after the truck pass through the pedestrian.

6. Conclusion and Future Works

This paper presents real-time moving object detection and tracking method in H.264/AVC compressed video, which is completely based on the compressed data. We proposed object detection method based on variable motion compensation block size and motion vector information. The proposed method can achieve average 92% precision rate and 85% recall rate for all type of video profile. Compared to the skip MB-based and MV-based methods, proposed method has equal performance regardless of the video profile type.

We applied the Kalman filter uses the position of the object and the velocity of the object that is calculated from the average value of motion vectors of moving objects.

Forward motion vectors are used for calculating the reference percentage to determine the object correspondence between previous frames and current frame. This simple correspondence metrics can achieved high precision and recall rate even there were occlusion.

Our experiments also show that the processing time for the proposed algorithm compare to presentation time is fast enough (8.73ms/frame) real-time implementation.

Currently we used only forward motion vector information to calculate the reference percentage as a feature for object correspondence. In the future, we consider including the object's moving direction, speed and size of object as features for objects correspondence.

References

[1] Dan Schonfeld and Dan Lelescu, "VORTEX: Video retrieval and tracking from compressed multimedia databases–multiple object tracking from MPEG-2 bit stream," Journal of Visual Communication and Image Representation, vol. 11, no. 2, pp. 154–182, 2000.

[2] Francesca Manerba, Jenny Benois-Pineau, Riccardo Leonardi, and Boris Mansencal, "Multiple moving object detection for fast video content description in compressed domain," EURASIP J. Adv. Signal Process, vol. 2008, no. 1, pp. 1–13, 2008.

[3] H. Zen, T. Hasegawa, and S. Ozawa, "Moving object detection from MPEG coded picture," Image Processing,1999. ICIP 99. Proceedings. 1999 International Conference on, vol. 4, pp. 25–29 vol.4, 1999.

[4] Ashraf M.A. Ahmad, Duan-Yu Chen, and Shu-Yin Lee, "Robust object detection using cascade filter in mpeg videos," in Proceedings of the IEEE 5th International Symposium on Multimedia Software Engineering (ISMSE), 2003, pp. 196–203.

[5] M. Ritch and N. Canagarajah, "Motion-based video object tracking in the compressed domain," IEEE International Conference on Image Processing (ICIP) 2007, vol. 6, pp. VI – 301–VI – 304, 2007.

[6] V.Thilak,Creusere,"Tracking of extended size targets in H.264 compressed video using the probabilistic data association filter", Processing of the European Signal Processing Conference ,281-284, (2004) .

[7] W.Zeng, J.Du, W.Gao, Q .Huang, "Robust moving object segmentation on H.264 compressed video using the block-based MRF model ", Real-Time Imaging 11, 290-299,(2003)

[8] Wonsang You, M.S.Houari Sabirin, Munchurl Kim, "Real-time detection and tracking of multiple objects with partial decoding in H.264/AVC bitstream domain", Proceedings of SPIE(2009),Vol.7244, (2009)

[9] Chris Poppe, Sarah De, Bruyne, Tom Paridaens, Peter Lambert, Rik Van de, Walle, "Moving object detection in the H.264/AVC compressed domain for video surveillance application", Journal of Visual Communication and Image Representation, Vol.20,Issue 6, Page 428, (2009)

[10] C.Solana-Cipres, G.Gernandez-Escribano, L.Rodriguez-Benitez, J.Moreno-Garcia, L.Jimenez-Linares, "Real-time moving object segmentation in H.264 compressed domain based on approximate reasoning ", International Journal of Approximate Reasoning, Vol.51,Issue.1, Page 99-114, (2009)

[11] R. E. Kalman, "A new approach to linear filtering and prediction problems," Journal of Basic Engineering, vol. 82-D, pp. 35–45, 1960.

[12] Zhi Liu, Zhao yang, and Liquan Shen, "Moving object segmentation in the H.264 compressed domain, " Opt. Eng., Vol. 46, 2007.

[13] Iain E. Richardson, "H.264 and MPEG-4 Video Compression Video Coding for Next-generation Multimedia", 2003.

[14] Fei HAO, Zhenjiang MIAO, Ping GUO, Zhan Xu, "Real Time Multiple Object Tracking Using Tracking Matrix", Proceedings of the 2009 International Conference on Computational Science and Engineering, Vol. 02, 2009

[15] H. Wang, H.S. Stone, and S-F Chang, "Facetrack: Tracking and summarizing faces from compressed video," SPIE Multimedia Storage and Archiving System IV, pp. 19–22, September 1999.

[16] I G. B. Baskara Nugraha, Suwen Weng and Hiroyoshi Morita, "Multiple object tracking on static surveillance video using field-based prediction information in MPEG-2 video", ICIP 2010.

[17] A. Yilmaz, O. Javed, and M.Shah, "Object Tracking: A Survey," ACM Computing Survey, vol.38, no.4, pp. 1-40,2006.