8

Object Tracking and Indexing in H.264/AVC Bitstream Domains

Muhammad Syah Houari Sabirin and Munchurl Kim
Korea Advanced Institute of Science and Technology
Korea

1. Introduction

The use of surveillance video recording is mainly based on the activity of monitoring the areas in which the surveillance cameras are located. Consequently, one of the motivations of observing the footage from surveillance video is to locate and identify regions of interest (e.g. suspected moving objects) in the covered areas. The visual information of objects such as color, shape, texture and motion etc. enables users to easily navigate and browse regions of interest in the surveillance video data. However, this requires the semi-automatic or automatic analysis and description of visual features for surveillance video data, which is usually very challengeable and time consuming.

To describe the visual information of the surveillance video contents in a structured way for fast and accurate retrieval over visual queries, MPEG has standardized a set of visual descriptors in forms of metadata schema. The MPEG-7 Visual descriptors, as Part 3 of MPEG-7 (Multimedia Content Description Interface, standardized under ISO/IEC 15938), provide the standardized descriptions of visual features that enable users to identify, categorize or filter the image or video data (MPEG57/N4358). The MPEG-7 Visual descriptors enable the structured descriptions of color, texture, shape, motion, location and face recognition features. Moreover, since they are described in low-level, the retrieval process can be relatively easier by comparing a given query with the pre-generated descriptions of MPEG-7 Visual descriptors in database.

In surveillance video contents, MPEG-7 Visual descriptor enables the description of the characteristics of objects. The motion trajectory can be used to describe the behavior of objects. It can track and identify when and where the objects move. The motion information for an object enables the retrieval of the object by querying the intensity of the motion or the moving direction.

The automatic extraction of visual features from video contents has also become an emerging issue with MPEG-7 Visual descriptors. Recently, H.264/AVC has become a popular video compression tool for video surveillance due to its high coding efficiency and the availability of its real-time encoding devices. By incorporating content analysis into the surveillance applications, intelligent surveillance can be possible for efficient searching and retrieval of surveillance video data. In this case, the content analysis for the surveillance video data is usually required to be performed in real-time for metadata generation. For real-time generation of metadata, the content analysis is usually performed in the bitstream domain of compressed video instead of its raw pixel domain. In this chapter, we explain a feature extraction of motion trajectory which detects and tracks moving objects (ROI's) and

also propose improvement to the feature extraction of motion activity of the moving objects in H.264/AVC bistreams. Our target is fast indexing of object retrieval in the metadata. Object retrieval can be performed only in interested time frame in which the metadata describes the existence of an object with high motion activity. Thus the motion trajectory of the object can be detected in the given time frame, without having to search the entire metadata, localizing objects of interest by effectively limiting the scope of searching subjects, thus reducing computation complexity in similarity matching process

Much research has been made object detection and tracking in MPEG-1 or MPEG-2 compressed video, and recently some related works (Zeng, 2005; Thilak 2004; You 2007; You 2009) have been tried for moving object detection and tracking in H.264/AVC bitstream domains. Nevertheless the methods do not consider identifying the objects into different identity, thus the objects cannot be consistently tracked along the sequence. To overcome this problem, a novel technique is introduced which uses motion information and quantized residual coefficients in every 4x4 block partition units of a macroblock. Using smallest block partition, our method can detect small sized object. Furthermore, by observing both motion and residue coefficients, we develop a similarity model to keep track of object when occlusion occurred.

We use motion activity of detected object as a point of interest in indexing the tracking using MPEG-7 descriptor. The MPEG-7 motion activity descriptor can describe the degree of motion activity caused by moving objects in the video. It has a motion intensity element to describe how much active the moving objects are. The normative of MPEG-7 motion activity descriptor only expresses the motion intensity into 5 levels of motion activity. However, how to categorize an amount of motion intensity into one of the five motion activity levels is non-normative and is left for application vendors.

In this chapter, the proposed motion activity categorization method takes into account the moving and non-moving parts in each frame. The non-moving parts are usually stationary background, which are more likely to be encoded in the partition of 16x16 macroblock size. On the other hand, the moving objects are likely to be encoded in sub-macroblock partitions due to their motion. Therefore weighting factors are assigned on encoded macroblock types by giving more emphasis on sub-macroblock partitions. Details of motion activity description are described in Section 2.2.

This chapter is organized as follows: We first describe the proposed method of extracting visual features from H.264/AVC bitstream domain in Section 2; the experimental results are described in Section 3; the example of conceptual structure of MPEG-7 metadata with the descriptions of extracted features is presented in Section 4; finally, the conclusion and future works are described in Section 5.

2. Feature extraction for MPEG-7 visual descriptor

2.1 Feature extraction of MPEG-7 motion trajectory descriptor

The MPEG-7 motion trajectory descriptor describes the transition of the relative coordinates of moving region in a frame during a sequence. The object detection and tracking method will provide the coordinates of the detected object for each frame. Feature extraction of motion trajectory is performed by observing each of 4x4 block partition units in a macrbolock in the bitstream. We take into account only the blocks having nonzero motion vectors or nonzero quantized residue coefficient by assumption that the camera is static so the observed blocks have high probability of being the region of moving objects.

Define B_{ij} as a 4x4 block partition in frame f having nonzero motion vectors and nonzero quantized residue coefficient values, where $\{i \in I, j \in J\}$ is the location of the block in the 4×4

block-unit of a $4I \times 4J$-sized frame. Thus by definition, all blocks having zero motion vectors are spatially segmented into background and $K \prec 4I \times 4J$ denotes the number of 4x4 block partition that belongs to foreground that contains a nonzero motion vector and nonzero quantized residual coefficient values. We then define a block group C_f as the group of 4×4 block partitions that are adjacent to each other (we will use the term "group" to refer to as the block group hereafter). Each group is representation of individual objects to be tracked.

Let $\hat{O}_f^{\ell} = \left\{ C_f^u ; u \in K \right\}$ be a set of groups where ℓ denotes the index of group and $u \in U$ denotes the block partition index. A group may have U member block partitions, where $U \in K$ but not necessarily in order of K (that is, one group may contain any arbitrary block partition regardless its index order). For implementation, we keep the maximum number of groups to be recognized to ten objects. For each group, the parameters are classified into two categories: static parameters and dynamic parameters. Static parameters are the parameters that have been defined by the bitstreams and its values will not be changed by the algorithm throughout the sequence. Dynamic parameters are the parameters that will be updated adaptively to the change of the static parameters.

The static parameters are position, direction, and energy. Position of the group is denoted by x and y, where $x \le width$ and $y \le height$. The position value is defined from the leftmost x and topmost y positions of the member block partitions of the group. For the *l*-th group in frame f, its position is defined as x_f^{ℓ}, y_f^{ℓ}.

The direction of the group, $D \in \Re, 0 \le D \le 360$, is calculated from the motion vector average of member blocks of the group. For the *l*-th in frame f, its direction is defined as

$$D_f^{\ell} = \frac{1}{U} \sum_{u \in U} \tan^{-1} \left(\frac{(mv_y)_f^u}{(mv_x)_f^u} \right) \tag{1}$$

where $(mv_x)_f^u$ and $(mv_y)_f^u$ are the motion vectors of C_f^u for each block partition u. For the *l*-th group in frame f its energy is defined as

$$e_f^{\ell} = \frac{1}{U} \sum_{u \in U} \left(\frac{1}{16} \sum_{j \in 16} (R_f^{uj})^2 \right) \tag{2}$$

where R_f^{uj} is the residue data of C_f^u for each j-th pixel in block partition u.

The dynamic parameters are event status and label. The status of events occurred in the group is given by $S = \{0,1\}$. The status events defined for a group are: (1) *occlusion*, (2) *out of frame*, (3) *stop*, and (4) *in to frame*. Additionally we add default status (0) if none of the mentioned status occurred to the group. For the *l*-th group in frame f its status is defined as $S_f^{\ell}(0)$, $S_f^{\ell}(1)$, $S_f^{\ell}(2)$, $S_f^{\ell}(3)$ and $S_f^{\ell}(4)$ according to the event respectively. Restriction applies to the event status where one object can only have at most one status in a frame.

The label of the group is denoted as l with $l > 0$. For the *l*-th group in frame f its label is defined as l_f^{ℓ}. Note that label l is not the same index as ℓ, although it is possible that a group may have the same value for the label and index, for example, in initial frame where we set the label equal to the index, or if the relative positions of objects are interchanged over the sequence (for example, if the objects are moving toward the same direction). Label

l defines the actual identification of groups as object candidates regardless their positions in frames, while index ℓ defines the order of groups for which the top-leftmost group has the smaller index value. Similar to groups, we limit the maximum number of objects to ten objects.

To this point, we have object candidates as groups with static and dynamic parameters which are then to be correctly identified frame to frame, i.e. tracking the objects by correctly recognize whether the candidate of objects in the current frame are the same as the identified object in the previous frame. In the next section we will describe the method of tracking the groups and set their label appropriately using a similarity model.

Object tracking process begins with pre-processing the groups in order for inconsistent groups to prevent from incorrect tracking. In addition, temporal and spatial filtering is applied to remove noise that resulted from inconsistent, isolated and dangled block. In the temporal refinement, we filter (remove) the groups that inconsistently appear in frames and interpolate (insert) the groups that inconsistently disappear over frames.

Let h be the number of frames where a group $\hat{O}^l$ exists over five consecutive frames including the current frame, two frames prior and two frames, thus $0 \le h \le 5$. The filtering is simply to remove the groups that appears only in 2 frames or less as given by

$$\hat{O}_f^\ell \begin{cases} keep & if\ h > 2 \\ remove & else \end{cases} \tag{3}$$

Similarly, $\hat{O}_f^\ell$ is interpolated when there is missing groups in frame f by taking the average value of static parameters of groups in frames $f' = \{f-2, f-1, f, f+1, f+2\}$ so the interpolated $\hat{O}_f^\ell$ will have average values of positions, directions and energies. The interpolation is defined by

$$\hat{O}_f^\ell = \varnothing \Leftrightarrow \hat{O}_f^\ell = \frac{1}{h}\sum_{f'} \hat{O}_{f'}^l, \hat{O}_{f'}^l \neq \varnothing \tag{4}$$

From the filtering and interpolating process, we expect to have consistent groups (object candidates) ready to be tracked. In this process we track the object candidates by updating their dynamic parameters. At the end of the process, the object candidates will be determined as the real objects O_f^ℓ.

The first step of tracking is to observe the difference (increment or decrement) of number of groups from one frame to the next. Let L_f be the total number of groups in the current frame and L_{f-1} be the total number of groups in the previous frame. Then we define the difference as γ and set its value as follows

$$\gamma = \begin{cases} 0 & \Leftrightarrow L_{f-1} = L_f \\ 1 & \Leftrightarrow L_{f-1} < L_f \\ 2 & \Leftrightarrow L_{f-1} > L_f \end{cases} \tag{5}$$

Assessing the changes in the number of groups in the first step is performed when a new frame is tracked. Following the assessment we may then update the status event of every candidate of objects in a frame. The status event will determine the way of finding the

similarity of an object candidate in the current frame with its corresponding one in the previous frame.

The status event is updated when the number of groups in the current frame is larger than the number of groups in the previous frame. For the other γ we set status events of all groups as 0, that is, $S_f^{\ell}(0)=1$. Thus for γ =2 we update the status event of a group based on the position of the group relative to the other groups or to the the edge of the frame, or based on its existence in the current and previous frames.

The relative position of a group to other groups is determined based on the Euclidean distance of its parameters. We define

$$d(O_{f-1}^{\ell},O_{f-1}^{m})_{xy}=\left(\left(x_{f-1}^{\ell}-x_{f-1}^{m}\right)^2+\left(y_{f-1}^{\ell}-y_{f-1}^{m}\right)^2\right)^{1/2} \tag{6}$$

as the distance between object ℓ and object m in frame $f-1$,

$$d(O_{f-1}^{\ell},O_{f-1}^{m})_{x}=\left(\left(x_{f-1}^{\ell}-x_{f-1}^{m}\right)^2\right)^{1/2} \tag{7}$$

as the distance between position x of object ℓ and object m in frame $f-1$, and

$$d(O_{f-1}^{\ell},O_{f-1}^{m})_{y}=\left(\left(y_{f-1}^{\ell}-y_{f-1}^{m}\right)^2\right)^{1/2} \tag{8}$$

as the distance between position y of object ℓ and object m in frame $f-1$; $m\in L$ denotes the pair index of object ℓ in frame $f-1$ where $\ell\neq m$.

The relative position of a group to the frame edge is determined by its outer boundaries. We define $p_{\min}(\hat{O}_f^{\ell})_{xy}$ as the leftmost x and topmost y positions of member block partitions of group ℓ in frame f, $p_{\max}(\hat{O}_f^{\ell})_{x}$ as the leftmost x position of member block partitions of group ℓ in frame f, and $p_{\max}(\hat{O}_f^{\ell})_{y}$ as the topmost y position of member block partitions of group ℓ in frame f.

Finally, the criterion is given by

$$\begin{aligned}
S_f^{\ell}(1)&=\begin{cases}1 & if\ d(O_{f-1}^{\ell},O_{f-1}^{m})_{xy}<\left(d(O_{f-1}^{\ell},O_{f-1}^{m})_{x}\wedge d(O_{f-1}^{\ell},O_{f-1}^{m})_{y}\right)\\ 0 & otherwise\end{cases}\\
S_f^{\ell}(2)&=\begin{cases}1 & if\ \left(p_{\min}(\hat{O}_f^{\ell})_{xy}\le 8\right)\wedge\left(p_{\max}(\hat{O}_f^{\ell})_{x}\ge width-8\wedge p_{\max}(\hat{O}_f^{\ell})_{y}\ge height-8\right)\vee O_{f-1}^{\ell}\neq\varnothing\\ 0 & otherwise\end{cases}\\
S_f^{\ell}(3)&=\begin{cases}1 & if\ S_{f-1}^{\ell}(1)=0\vee S_{f-1}^{\ell}(2)=0\vee\hat{O}_f^{\ell}=\varnothing\\ 0 & otherwise\end{cases}\\
S_f^{\ell}(4)&=\begin{cases}1 & if\ \left(p_{\min}(\hat{O}_f^{\ell})_{xy}\le 8\right)\wedge\left(p_{\max}(\hat{O}_f^{\ell})_{x}\ge width-8\wedge p_{\max}(\hat{O}_f^{\ell})_{y}\ge height-8\right)\vee O_{f-1}^{\ell}=\varnothing\\ 0 & otherwise\end{cases}
\end{aligned} \tag{9}$$

where O_{f-1}^{ℓ} denotes object ℓ in the previous frame. Based on γ and the status event, the next step of tracking is to update the label l_f^{ℓ} by finding the most similar object between two

frames and to set the same label for the most similar objects. The similarity is computed by finding index ℓ that gives the minimum distance between the group in the current frame $\hat{O}_f^\ell$ and the object in the previous frame O_{f-1}^ℓ for given previous frame's event S and the changes of group γ. We define the similarity model as

$$l_f^\ell = \left\{ l_{f-1}^\ell \mid \min_\ell \left(\Delta\left(O_{f-1}^\ell, \hat{O}_f^\ell, S, \gamma \right) \right) \right\} \tag{10}$$

where $\Delta\left(O_{f-1}^\ell, \hat{O}_f^\ell, S, \gamma\right)$ denotes the similarity function between O_{f-1}^ℓ and $\hat{O}_f^\ell$ for given S and γ. When γ =0, the similarity is defined as

$$\Delta\left(O_{f-1}^\ell, \hat{O}_f^\ell, S, \gamma\right) = d(O_{f-1}^\ell, \hat{O}_f^\ell)_{xy} \Gamma(\hat{O}_f^\ell) \tag{11}$$

where $\Gamma(\hat{O}_f^\ell) = \{0,1\}$ denotes the existence of group l in the current frame. If γ =1, the similarity is defined as

$$\begin{aligned} \Delta\left(O_{f-1}^\ell, \hat{O}_f^\ell, S, \gamma\right) &= d(O_{f-1}^\ell, \hat{O}_f^\ell)_{xy} S_f^\ell(0) \\ &\quad + d(O_{f-t}^\ell, \hat{O}_f^\ell)_{De} S_f^\ell(1) + l_f^{\ell^{new}} S_f^\ell(4) \end{aligned} \tag{12}$$

where

$$d(O_{f-t}^\ell, \hat{O}_f^\ell)_{De} = \left(\left(\tilde{D}_{f-t}^\ell - \tilde{D}_f^\ell \right)^2 + \left(\tilde{e}_{f-t}^\ell - \tilde{e}_f^\ell \right)^2 \right)^{1/2} \tag{13}$$

which denotes the similarity between O_{f-t}^ℓ and $\hat{O}_f^\ell$ based on their direction and energy. t denotes number of elapsed frames since $S_f^\ell(1)$ is set (i.e. when occlusion has occurred), and $l_f^{\ell^{new}}$ is a new label.

In finding the distance in direction and energy between two objects, instead of using the actual values, we normalize the parameter values in logarithmic scale, where $\tilde{D} = \log_{10}(D)$ and $\tilde{e} = \log_{10}(e)$, to make fair comparison of two values in different ranges (as can be seen from (1) and (2), the direction D has possible values from 0 to 360 while energy e has possible values from 0 to virtually unlimited). New label $l_f^{\ell^{new}}$ is determined by observing the list of labels already used or currently being used in the frame so the used labels or the current labels are not defined as new labels. When γ =2,

$$\Delta\left(O_{f-1}^\ell, \hat{O}_f^\ell, S, \gamma\right) = d(O_{f-1}^\ell, \hat{O}_f^\ell)_{xy} \Gamma(O_{f-1}^\ell) \tag{14}$$

where $\Gamma(O_{f-1}^\ell) = \{0,1\}$ denotes the existence of the object in the previous frame.

By this step, the groups are now labeled with the same label of the most similar object in the previous frame and the groups are then defined as the real objects for which their parameter values will then be used in the next frame to identify the candidate of objects.

2.2 Feature extraction of MPEG-7 motion activity descriptor

The MPEG-7 motion activity descriptor describes the degree of motion activity in a frame in terms of the intensity, direction, spatial distribution, spatial localization, and temporal

distribution features of motion activity. Here, we especially take into consideration the intensity feature of the descriptor because surveillance observation tends to give more attention to the intensity of motion. Furthermore, the usage of intensity descriptor is mandatory in annotating MPEG-7 motion activity.

The intensity feature indicates the degree of motion into five levels (*very low*, *low*, *medium*, *high*, and *very high*). However, the feature extraction itself is outside the standard. So, in this chapter, we propose a feature extraction scheme for the computation of such feature values in the H.264/AVC bitstream domain.

Feature extraction of motion intensity in a frame can be performed by computing the standard deviation of the magnitudes of motion vectors of the macroblocks. The standard deviation can then be quantized into one of the five levels in which the quantization parameters are determined experimentally by the training data of motion vectors from H.264/AVC bitstreams. In this chapter, we use the quantization parameters in Table 1 as defined in (Manjunath, 2002).

Activity value	Range of σ
1	$0 \leq \sigma < 3.9$
2	$3.9 \leq \sigma < 10.7$
3	$10.7 \leq \sigma < 17.1$
4	$17.1 \leq \sigma < 32$
5	$32 \leq \sigma$

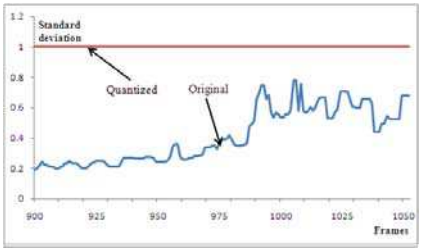
Table 1. Quantization threshold of motion intensity

The result of using the method described in (MPEG57/N4358) for surveillance video recording has drawback to describe the information of object activity. This justification is depicted in the following case. Fig. 1(a) shows the curve of standard deviation of the magnitudes of motion vectors for a surveillance video sequence with its corresponding quantized motion intensity, while Fig. 1(b) shows one frame from the sequence with its representation of motion vectors. While the frame shows the objects actively moving, the resulting level of motion activity corresponds to very low activity.

This problem is caused by the size of the objects that are relatively small, compared to the resolution of the frame. Even though the method described in (MPEG57/N4358) takes into account the resolution of frame for the computation of standard deviation, it is still hard to correctly categorize such motion. Therefore, small objects with high activity cannot be identified as having high activity motion. On the other hand, it is also possible that large objects (e.g., the object that located very near to the camera) with slow activity will be identified as having high activity motion.

To overcome this problem, we propose a method of preprocessing the standard deviation values before it is quantized. Our method utilizes the macroblock partition types (16x16, 16x8, 8x16, 8x8, 8x4, 4x8, 4x4 block types) and applies different values of the weighting factor for different block classes. Our method is applicable for surveillance video contents that are commonly captured by still cameras. These surveillance video usually has still background (i.e., no abrupt changes in intensity) and all motions are likely to happen by object movement. The weighting factor for each MB partition is defined based on the size of the partition itself. For example, an MB with 4x4 block types can have 16 motion vectors, so it can contribute more to the standard deviation of motion intensity, compared to the MB's

with a smaller number of block partitions. For such an MB, we multiply a larger value of weighting factor to its magnitude of motion vectors so the resulting standard deviation increases large enough to represent the motion activity of the object.



(a)

(b)

Fig. 1. (a) The curve of standard deviation of the magnitudes of motion vectors for a surveillance video sequence and its corresponding quantized value identifies the sequence has very slow activity of objects (intensity = 1) (b) a snapshot frame from the same sequence with its corresponding motion vectors of macroblocks

However, a slowly moving object close to a surveillance camera can be identified as having high motion intensity due to the large portion of the object size, compared to the background. Therefore, we need to reflect the partition of the object into the weighting factor by estimating the object size with respect to its distance to the camera. That is, the larger the estimated object size is, the larger a penalty factor value is taken into account in the weighting factor. So, the weighting factor is defined as

$$w_i = M - r\sum_{k=1}^{n_{PB}} e^{k \cdot r} \tag{15}$$

Here M reflects the size of the i-th partition block or macroblock, assuming that the square-sized blocks (4x4 and 8x8) contribute more to the motion activity due to more numbers of those block types in each MB, while the remaining block types (4x8, 8x4, 16x8, 8x16) are contribute less. Therefore M can be calculated as follows:

$$M = \begin{cases} m \times n, \\ if\ mb\_type = class\ 1\ (8\times 8\ or\ 4\times 4) \\ \\ \sqrt{m \times n}, \\ if\ mb\_type = class\ 2\ (16\times 8, 8\times 16, 8\times 4\ or\ 4\times 8) \end{cases} \tag{16}$$

where m and n indicate the width and height of one partitioned block in the i-th partition block or MB. r is a multiplication factor of the penalty term in (15) based on the partition ratio of partition block, with similar assumptions to M where square-sized blocks have smaller penalty than the other block type. Thus r is given by

$$r = \begin{cases} \frac{1}{4}, & if\ mb\_type = 8\times 8\ or\ 4\times 4 \\ \frac{1}{2}, & if\ mb\_type = 16\times 8, 8\times 16, 8\times 4\ or\ 4\times 8 \end{cases} \tag{17}$$

n_{PB} is the accumulated number of all the partition blocks for a same class of the block partition types to the i-th partition block or macroblock. In (15), the penalty term contains exponents in summation. This is to put more penalties on the partition blocks of a same class coming in the subsequent MB's. In this regard, smaller values of the weighting factor are then applied for the partition blocks of having the same class with more frequent occurrence. By doing so, the slowly moving object of a large size can be regarded as having low motion activity.

In (16), the square block types are more emphasized for the calculation of standard deviation than the non-square block types because the square block type allows for more sub-block partitions which results in more motion vectors. Therefore, with the weighting factor in (15), the standard deviation of the intensity in motion activity can be calculated as

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \tag{18}$$

where x_i is a weighted magnitude of motion vector and is given by

$$x_i = w_i \times |mv_i| \tag{19}$$

mv_i is the motion vector of the i-th partition block or macroblock and N is the total number of the partition blocks and macroblocks (no partitions) in a frame.

Sometimes, due to the noise of motion caused by camera shaking, abrupt light intensity change or unexpected shadow, the standard deviation undergoes abrupt change in a certain frame over time. Therefore, we remove such outliers by applying a median filter on standard deviation values for every five frames. Finally the median filtered standard deviation values are quantized into the five levels of motion activity.

3. Experimental results

3.1 Feature extraction of motion trajectory

(a) Three sequences were used in our experiments: *Corridor* and *Speedway* for low and high camera, respectively; and *PETS2001* for occlusion handling experiments. *Corridor* sequence is a QVGA (320x240), 670 frames, indoor surveillance video with single object (a person) moves toward the camera. *Speedway* sequence is a CIF (352x288), 198 frames, outdoor surveillance video of traffic in a speedway with several vehicles. *PETS2001* is a QPAL (384x288), 600 frames, outdoor surveillance video in a building complex, with several objects and group of objects and having occlusions between some of them.

We first present the observation of using energy of object to determine the similarity between objects before and after occlusion as defined in (12). Fig. 2 shows the example of

occlusion and disocclusion occurred in *PETS2001* sequence with the graphs showing the energy value for occluded objects before and after occlusion.
In Fig. 2(a), object 1 and object 2 starts to occlude in frame 157 and disocclude in frame 216. As depicted by the graph in Fig. 2(c) we can see that the energy value of each object is distributed fairly different so we can easily match the energy of object 1 and object 2 before and after occlusion. However, there is also case when the energy value alone is not enough to determine the similarity of object. As shown in Fig. 2(d), the energy value of object 4 fluctuates and in some points overwhelming the energy value of object 3. Furthermore, it makes the energy value almost has no significant differences for the similarity calculation to find similarity (and dissimilarity) of both objects after occlusion in frame 567, although energy value of both objects fairly different just before occlusion in frame 490. For this reason we add direction value term in the similarity model as the weighting factor to control the energy value for the model to not depend only one feature of object. Fig. 2(b) shows the result of handling occlusion for object 3 and object 4.

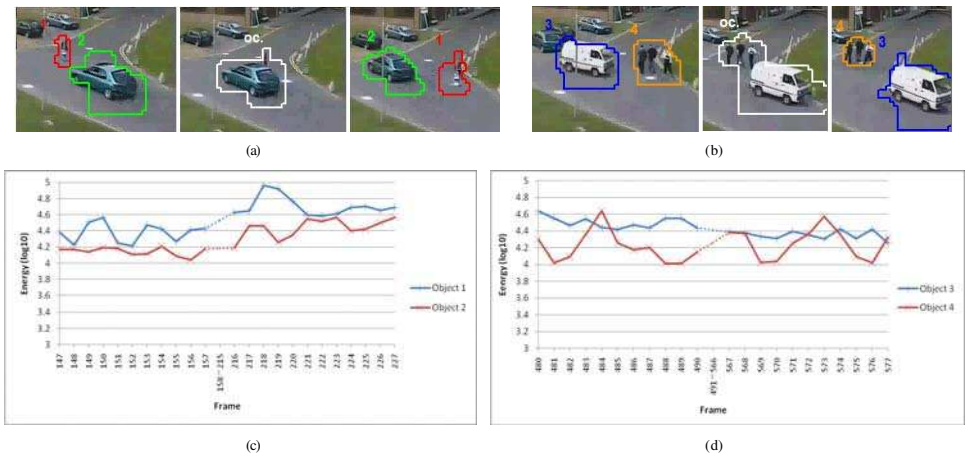


Fig. 2. Computing energy similarity of two occluded objects in PETS2001 sequence (a) between object 1 and object 2, (b) between object 3 and object 4. Graphs show the value of energy of (c) objects 1 and 2, (d) object 3 and 4. Dashed lines in the graphs imply the occlusion.

The results from of overall object detection and tracking process are shown. Fig. 3 shows the result of overall object detection and tracking process for *Corridor* sequence. The sequence is taken from video surveillance camera located near the ceiling of a corridor and features one person moving toward the camera. In this sequence the object's size changes gradually from small to large until the person finally walks approximately below the camera. As shown in Fig. 3, when the object starts to appear in the frame, its motion and residue information is available and the algorithm can recognize it correctly until the object is large and located near the camera.
Fig. 4 shows the result of overall object detection and tracking process for *Speedway* sequence. The sequence is taken from video surveillance camera located in high position and features four vehicles: two vehicles move away from camera and two vehicles move toward the camera. In this sequence the objects' size changes rapidly from large to small for

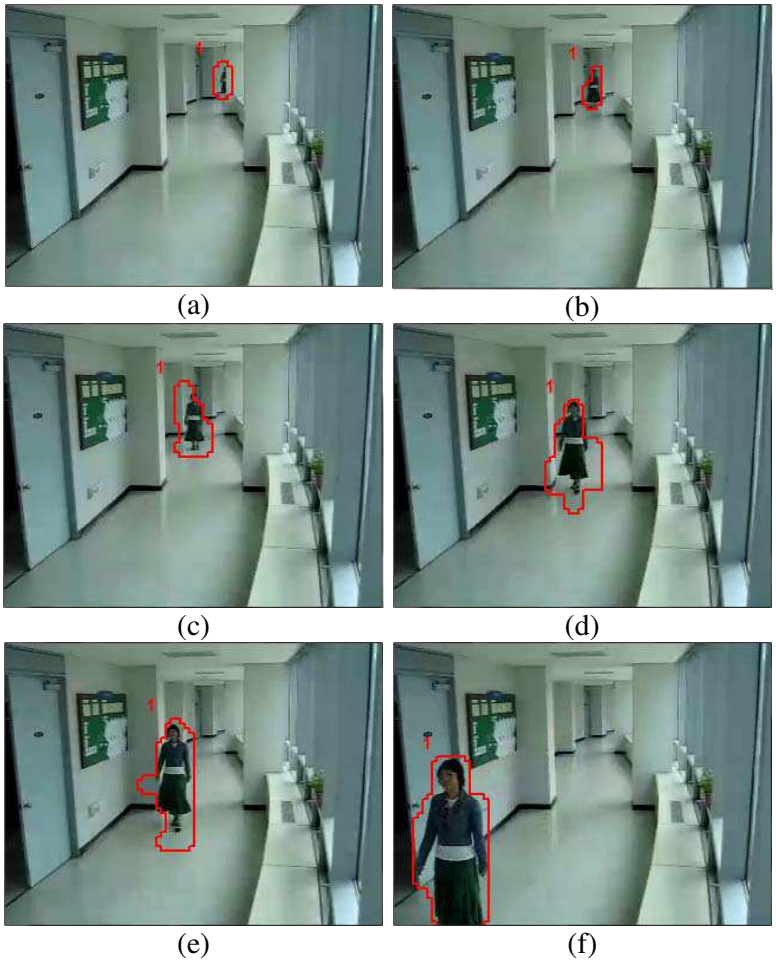


Fig. 3. Object detection and tracking results for 18th (a), 298th (b), 447th (c), 498th (d), 535th (e) and 650th (f) frame of Corridor sequence.

the objects that move away from camera, and vice versa. In the bitstreams, the small objects are too small to have motion and residue information. As a result, those objects cannot be recognized by the segmentation process, as can be seen in Fig. 4(a). In frame 54, shown in Fig. 4(b), the small objects moving toward the camera are first recognized into single object because their motion and residue information are represented in small amount of adjacent block partitions. As the frame advances, the two objects become large and finally can be recognized as two objects in the frame 186 as shown in Fig. 4(f). Similarly, object one (red) is getting small as the frame advances and finally unable to be recognized by the algorithm (Fig. 4(e)). From this experiment it can be seen that the algorithm is able to detect small object with block partitions take up less than one macroblock as long as its appearance is consistent along several frames in the sequence.

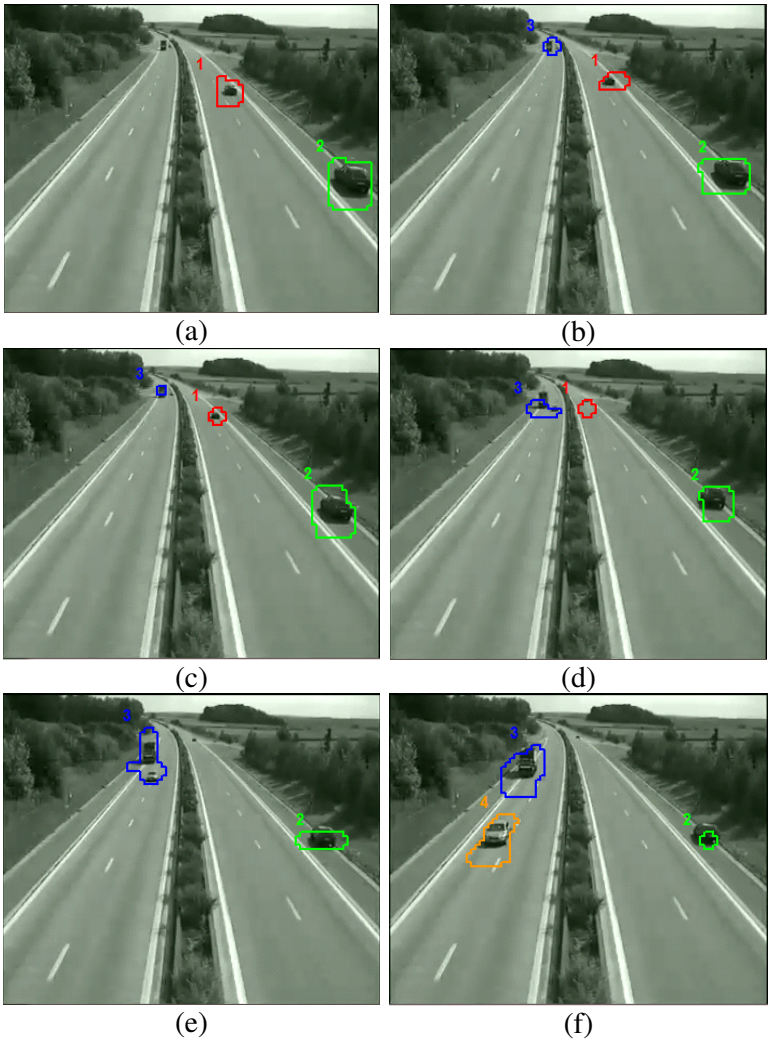


Fig. 4. Object detection and tracking results 43rd (a), 54th (b), 70th (c), 92nd (d), 138th (e) and 186th (f) frame of Speedway sequence.

Fig. 5 shows the result of overall object detection and tracking process for *PETS2001* sequence. The sequence features five objects: two persons, two vehicles, and one group of persons where there are two times occlusions between the objects. The first occlusion is occurred between a person and a vehicle (Fig. 15(b)). During the occlusion, both objects are identified as single object and labeled as "occluded objects". Because we kept their information prior the occlusion, the label information correctly identified both object to their original label after the occlusion. Similar result also occurred in the second occlusion when another vehicle occluded with group of persons, as shown in Fig. 5(e). The identification of all objects can be correctly detected prior to occlusion (Fig. 5(d)) and after occlusion (Fig. 5(f)).

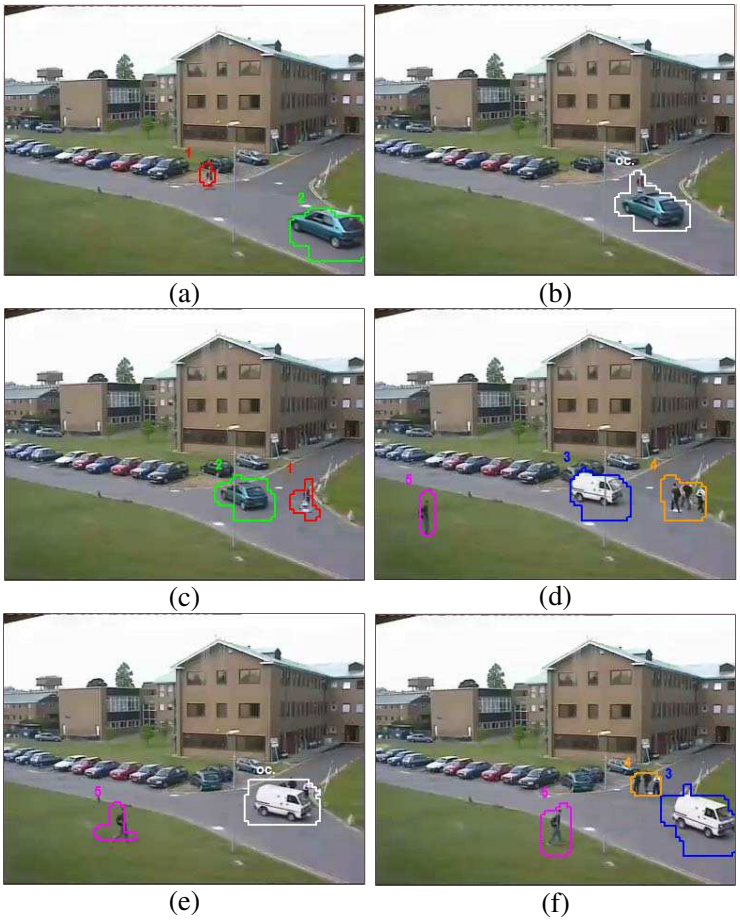


Fig. 5. Object detection and tracking results for 104th (a), 165th (b), 207th (c), 462nd (d), 510th (e) and 556th (f) frame of PETS2001 sequence.

3.2 Feature extraction of motion activity

In this section we present the result of the extraction of motion activity using the proposed method with a weighting factor and its comparison with the one without the weighting factor. The experiments were performed for surveillance video of QVGA size encoded at 30 fps by an H.264/AVC baseline profile encoder.

As shown in Fig. 1, the quantization of standard deviation of motion vector magnitudes cannot appropriately describe the real motion activity. In this experiment, the surveillance video camera covers a large area, i.e., the mounted camera is far away from the objects. As the result, quantization of the standard deviation without the weighting factor results in incorrect classification of motion intensity.

Fig. 6 shows the quantization of the standard deviation of the motion vector magnitudes using a weighting factor. As a result we can identify the motion of the object. The frames that have relatively high motion are quantized as higher intensity level.

We present another result where the position of the camera is relatively low from the floor, which means there is a possibility that the size of objects becomes large so that it might lead to incorrect motion activity quantization.

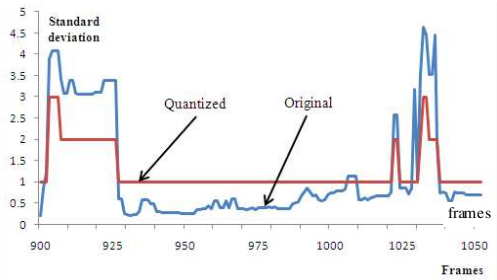
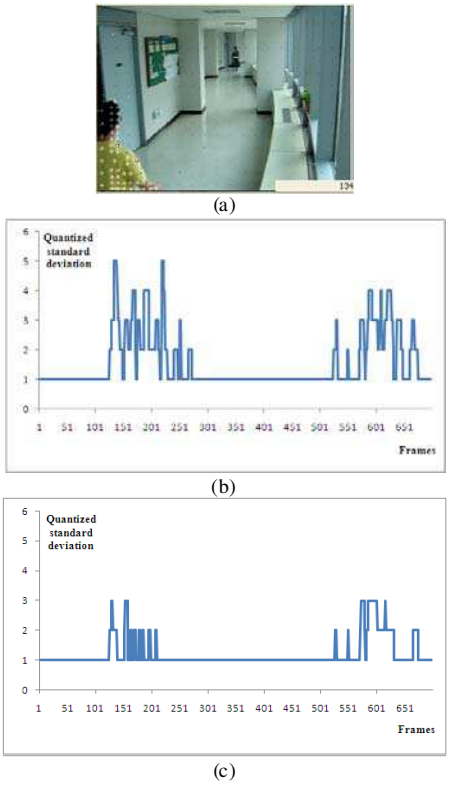


Fig. 6. Curves of original and quantized standard deviations

In order to see the effect of assigning penalty value in computing the standard deviation of the motion vector magnitudes, we present a screenshot with an object that moves slowly near the camera as shown in Fig 7(a).



(a)

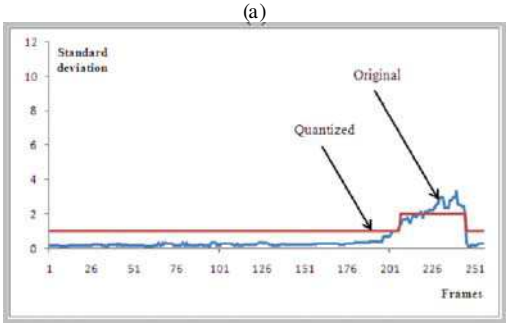
(b)

(c)

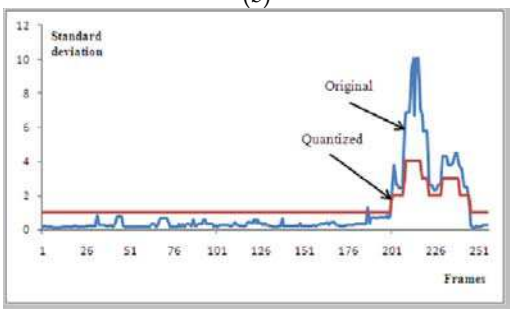
Fig. 7. The effect of assigning penalty value: (a) A snapshot frame (b) The quantized standard deviation without the penalty value and (c) The quantized standard deviation with penalty value.

For this, the quantized standard deviation without penalty values results in higher intensity levels as shown in Fig 7(b), compared to those with penalty values in Fig. 7(c). The object motion activity is abnormally high as having level 4 and 5 in the interval between 134th frame and 230th frame, and the interval between 570th frame to 670th frame. Assigning penalty values for this sequence gives better quantization result for the same frame as shown in Fig 7(c).

(a)



(b)



(c)

Fig. 8. (a) A snapshot frame of surveillance video sequence shows high motion activity. (b) Quantized standard deviation without weighting factor and (c) Quantized standard deviation with weighting factor.

Fig 8(a) shows the screenshot of the sequence in which an object moves fast. Fig. 8(b) shows the quantization results of the standard deviation of the magnitude of motion vectors without applying the weighting factor. The quantized standard deviation seems to correctly represent the activity of the objects. However, the video sequence actually consists of the scene where an object moves fast. Fig. 8(c) shows the result of the quantization by applying the weighting factor and penalty values, which results in relatively more appealing quantization results of motion intensity.

4. Implementations

Here we present the conceptual structure of MPEG-7 Visual for describing the result of motion trajectory and motion activity description to illustrate the practical usage of our method.

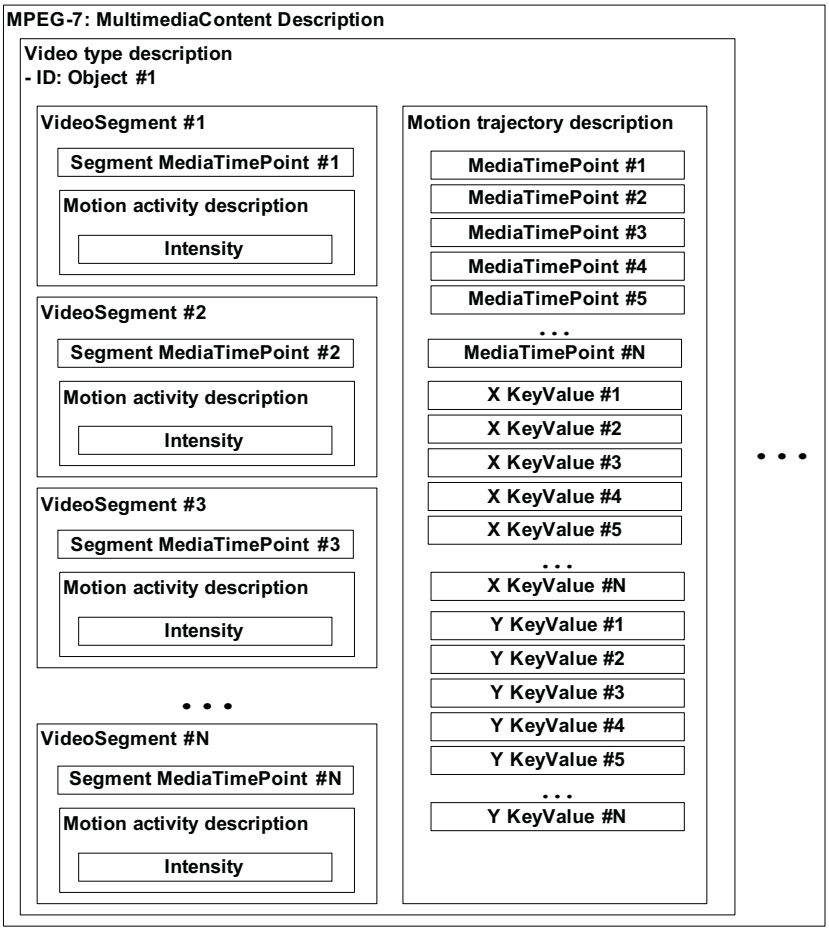


Fig. 9. The example of conceptual structure of MPEG-7 metadata to describe object motion trajectory and motion intensity.

Fig. 9 shows the example of conceptual structure of MPEG-7 metadata. A multimedia content descriptor from MPEG-7 Multimedia Description Scheme (MDS) (MPEG57/N4242) is the container for video type object description that can be used to describe each of detected objects in the bitstream. As shown in Fig. 9, the identity of object, which obtained from labeling process described in section 2.1, can be described by an identifier attributes, thus all descriptions inside the video descriptor belong to this object. A video segment descriptor in a video descriptor is used to index the segment of the video having the motion

intensity description of this object. Each video segment has media time information, describing the frame number where the intensity is annotated. Effective indexing can be generated by describing video segment description for every time frame when the intensity is high. A motion trajectory description describes the coordinate x and y of tracked objects, which can be represented by the interpolation of the centroid of the region resulted by the tracking algorithm. As shown in Fig. 9, each of x and y coordinates of tracked object is referred by each media time point.

Referencing the motion trajectory time point and video segment time point can be performed to find the location of detected object from frame to frame. Suppose the motion intensity description of an object describes a high intensity motion at frame 10, then the corresponding segment media time point of video segment acts as pointer to the media time point of motion trajectory. Then the trajectory of this object can be found by tracing the x and y coordinate values referred by media time point of motion trajectory started from frame 10.

5. Conclusions

In this chapter, a novel approach for extracting the ROI's of multiple moving objects in H.264/AVC bitstream domains is proposed which includes spatial and temporal filtering, similarity model for object detection and tracking and motion activity extraction to enhance indexing.

In order to extract the motion activity feature values, we proposed to utilize the macroblock types of H.264/AVC bitstreams by assigning the weighting factor to the block partitions so that the standard deviation of the magnitudes of the motion vectors can be appropriately quantized to represent the motion activity levels.

Finally, some examples for the usage of the extracted features are shown for visual information retrieval in the form of MPEG-7 structure that can make it easy the detected object indexing and information retrieval. Future study on the current can be extended to deal with occlusion problems during object detection and tracking in H.264/AVC bitstream domains.

6. References

MPEG57/N4358, *ISO/IEC FDIS 15938-3 Visual*, ISO/IEC JTC 1/SC 29/WG 11, July 2001, Sydney

MPEG57/N4242, *ISO/IEC FDIS 15938-5 Multimedia Description Schemes*, ISO/IEC JTC 1/SC 29/WG 11, July 2001, Sydney

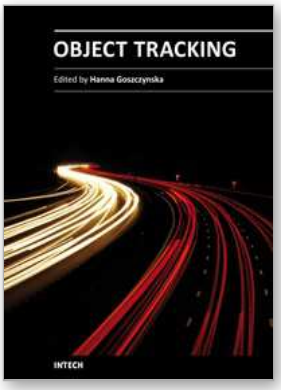
Manjunath, B. S.; Salembier, P. & Sikora, T., *Introduction to MPEG-7, Multimedia Content Description Interface*, John Wiley & Sons, 2002

Zeng, W.; Du, J.; Gao, W. & Huang, Q., Robust moving object segmentation on H.264|AVC compressed video using the block-based MRF model, *Real-Time Imaging*, vol. 11(4), 2005, pp. 290-299

Thilak, V. & Creusere, C. D., Tracking of extended size targets in H.264 compressed video using the probabilistic data association filter, *EUSIPCO 2004*, pp. 281-284

You, W.; Sabirin, M. S. H. & Kim, M., Moving Object Tracking in H.264/AVC bitstream, *MCAM 2007, LNCS*, vol. 4577, pp. 483-492

You, W.; Sabirin, M. S. H. & Kim, M. (2009) Real-time detection and tracking of multiple objects with partial decoding in H.264/AVC bitstream domain, In: *Proc. Of SPIE*, N. Kehtarnavaz & M. F. Carlsohn (Ed.), 72440D-72440D-12, SPIE, San Jose, CA, USA



Object Tracking

Edited by Dr. Hanna Goszczynska

ISBN 978-953-307-360-6
Hard cover, 284 pages
Publisher InTech
Published online 28, February, 2011
Published in print edition February, 2011

Object tracking consists in estimation of trajectory of moving objects in the sequence of images. Automation of the computer object tracking is a difficult task. Dynamics of multiple parameters changes representing features and motion of the objects, and temporary partial or full occlusion of the tracked objects have to be considered. This monograph presents the development of object tracking algorithms, methods and systems. Both, state of the art of object tracking methods and also the new trends in research are described in this book. Fourteen chapters are split into two sections. Section 1 presents new theoretical ideas whereas Section 2 presents real-life applications. Despite the variety of topics contained in this monograph it constitutes a consisted knowledge in the field of computer object tracking. The intention of editor was to follow up the very quick progress in the developing of methods as well as extension of the application.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Muhammad Syah Houari Sabirin and Munchurl Kim (2011). Object Tracking and Indexing in H.264/AVC Bitstream Domains, Object Tracking, Dr. Hanna Goszczynska (Ed.), ISBN: 978-953-307-360-6, InTech, Available from: http://www.intechopen.com/books/object-tracking/object-tracking-and-indexing-in-h-264-avc-bitstream-domains

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821