H.264/AVC 비트스트림을 활용한 감시 비디오 내의 그래프 기반 객체 검출 및 추적

*호와리 **김문철

한국과학기술원

*houari@kaist.ac.kr **mkim@ee.kaist.ac.kr

Graph-based Object Detection and Tracking in H.264/AVC bitstream for Surveillance Video

*Houari Sabirin **Kim, Munchurl

Korea Advanced Institute of Science and Technology

Abstract

In this paper we propose a method of detecting moving object in H.264/AVC bitstream by representing the 4x4 block partition units as nodes of graph. By constructing hierarchical graph by taking into account the relation between nodes and the spatial-temporal relations between graphs in frames, we are able to track small objects, distinguish two occluded objects, and identify objects that move and stop alternatively.

1. Introduction

The challenge of object detection and tracking in bitstream domain is to utilize limited resources, usually in forms of motion vectors, residual data or the size in bytes of a macroblock. Many techniques have been developed to detect and track moving object in compressed domain, especially in H.264/AVC bitstreams. Some techniques are able to detect objects by observing the motion vectors and partially decoded data such as in [1]~[4]. However, those techniques are unable to give exact identification of the objects thus might be failed in conditions such as occlusion.

In this paper we propose a method to detect moving object by observing the 4x4 block partition unit of a macroblock. Each block having nonzero motion vectors or quantized coefficient of residual data is then defined as a node of a graph. Thus one frame may contains set of nodes representing a graph of moving objects. We further observe the temporal consistency of the graph to remove noise and determine the object identity by constructing a hierarchical graph to find the relation between graphs in spatial and temporal domain. Finally we present our result in detecting and identify the moving objects.

2. Graph Definition

Let an undirected attributed graph $G=(V,E,a)$ represents a frame in which its vertices $V=\{v_1,...,v_P\}, P=I\times J$ represent the blocks in the frame of $4I$ width and $4J$ height size and the edge $E(u,v)=\{1,0\}$ denotes the presence of adjacent blocks. The attribute $a(v_{p\in P})=\{c,D,M,e\}$ is defined from the block parameters. Attribute $c(v)=\{i,j;i\in I,j\in J\}$ denotes the coordinate of the block relative to the top-left edge of a frame. The direction of motion vector $D(v)=\tan^{-1}\left(\frac{mv_{ij}^{y}}{mv_{ij}^{x}}\right)$ is calculated from the x and y component of motion vector in block location of i,j, thus magnitude of motion vector is defined as $M(v)=|mv_{ij}|$. Attribute $e(v)=\frac{1}{K}\sum_{k}\left((r_k)_{ij}\right)^2$ denotes the energy of the quantized coefficient of residue ("residue") of the block where $(r_k)_{ij}$ is the residue of the k-th pixel in i,j block and K is the number of pixel in that block where the residue is not zero.

The base of our hierarchical graph is constructed by defining a subgraphs $H\subset G$ where $H=\{H_1,H_2,...,H_N\}$ and N is the number of moving objects in a graph. The second level of the hierarchy is constructed by defining a weighted supergraph $G^*=\{V^*,E^*,w\}$ where the vertices are the representation of subgraphs in five consecutive frames, as shown in Fig. 1, defined by

$$V^*=\left\{v_1^{f-4},...,v_{N_{-4}}^{f-4},v_1^{f-3},...,v_{N_{-3}}^{f-3},v_1^{f-2},...,v_{N_{-2}}^{f-2},v_1^{f-1},...,v_{N_{-1}}^{f-1},v_1^{f},...,v_{N_0}^{f}\right\} \quad (1)$$

and the edges are the relation between two subgraphs in the five frames. defined as

$$E^*\in\left\{\left(v_{n_{-4}}^{f-4},v_{n_{-3}}^{f-3}\right),\left(v_{n_{-3}}^{f-3},v_{n_{-2}}^{f-2}\right),\left(v_{n_{-2}}^{f-2},v_{n_{-1}}^{f-1}\right),\left(v_{n_{-1}}^{f-1},v_{n_0}^{f}\right)\right\}. \quad (2)$$

The weight of the edge is determined by calculating the similarity in distance between two vertices as follow

$$w\left(v_{n_{-N}}^{f-N}, v_{n_{-(N+1)}}^{f-(N+1)}\right) = \| c\left(v_{n_{-N}}^{f-N}\right) - c\left(v_{n_{-(N+1)}}^{f-(N+1)}\right) \| \quad (3)$$

where N_0, N_{-1}, N_{-2}, N_{-3}, and N_{-4} are the number of vertices in frame f to frame f-4, respectively.

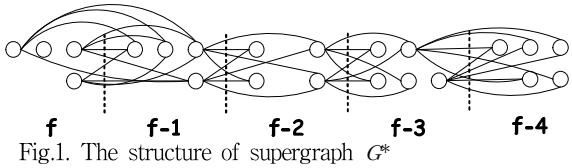
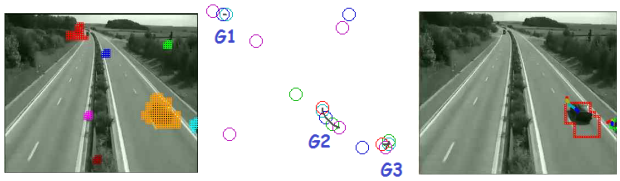


Fig.1. The structure of supergraph G^*

3. Graph Pruning

Quite often motion vectors are also produced by abrupt intensity change, non-object movement or even video signal noise that can made a vertex is not part of the real moving object, as shown in Fig. 2(a). To overcome this problem we perform temporal filtering, in term of graph, by removing subgraphs in H that are not real moving object. Graph pruning is performed between subgraph in current frame and subgraph in previous frame by observing the weight between two subgraphs within the supergraph G^*. If the weight is larger than threshold value specified by the size of the block (4.0), the subgraph is then pruned. Figure 2(c) shows the result of graph pruning.



(a) (b) (c)

Fig. 2. (a) Snapshot from *Speedway* sequence showing subgraphs H representing real objects and noise. (b) The struture of nodes from supergraph G* showing nodes of object and nodes of noise. (c) The result of graph pruning of (a).

Due to improper motion compensation or the difference between two frames is insignificant, graph pruning process sometimes suffers from missing blocks of moving object. As a result, the pruning process removes all vertices and the moving object becomes undetected. To overcome this problem, simple graph projection is performed to recover the missing vertices.

The graph projection is performed when the number of object is decreasing or becomes zero from previous to current frame. Let N_0 and N_{-1} be the number of vertices in current and previous frame, respectively. Define $v_m^{f-1}, m \in N_{-1}$ as a vertex in frame f-1, the corresponding missing vertex to be found by projection in frame f is defined as $\hat{v}_n^f, n \in N_0$. Vertices v_m^{f-1} and $\hat{v}_n^f$ are matched by their vertex identity n=m. Thus if vertex n is cannot be found, then $\hat{v}_n^f$ is signalled as the missing vertex and consequently $\hat{v}_n^f = v_m^{f-1}$ where the center attribute is calculated as

$$c\left(\hat{v}_n^f\right) = c\left(v_m^{f-1}\right) + \alpha . mv\left(v_m^{f-1}\right) \quad (4)$$

where $mv(v)$ is the motion vector of v and $\alpha = 0.5$ is a regulator coefficient to avoid the projected vertex shifted too far from the actual object position.

4. Graph Tracking

Object tracking is performed in supergraph level by finding the similarity between two vertices of supergraph from one frame and its reference frame, and consistently give the correct identity for each vertex. Assuming that the moving object does not move very fast (approximately less than 16 pixels per frame), we can match the identity of the object by simply searching for the corresponding object having similar location in the reference frame.

4.1. Adjacent graph tracking

Recall the vertices and edges of weighted supergraph G^* as defined in (1) and (2), we take its subset for two adjacent frames and defined it as tracking graph G^T where the vertices are defined as

$$V^T = \left\{v_1^{f-1}, \ldots, v_{N_{-1}}^{f-1}, v_1^f, \ldots, v_{N_0}^f\right\} \quad (5)$$

and the edges are defined as

$$E^T \in \left\{v_{n_{-1}}^{f-1}, v_{n_0}^f\right\} \quad (6)$$

and the weight between two edges is defined as

$$w^T\left(v_{n_0}^f, v_{n_{-1}}^{f-1}\right) = \| c\left(v_{n_0}^f\right) - c\left(v_{n_{-1}}^{f-1}\right) \| \quad (7)$$

Therefore the matching of two vertices in two adjacent frames can be determined as similar object having the following condition

$$v_n^f = v_m^{f-1} \Leftrightarrow w^T\left(v_n^f, v_m^{f-1}\right) \le \rho \quad (8)$$

where $n \in N_0, m \in N_{-1}$ and threshold $\rho = 4.0 + M\left(v_m^{f-1}\right)$. The magnitude attribute is added as regulator for fast moving objects.

4.2. Conditional graph tracking

In case when the object cannot be continuously detected in two adjacent frames, such as during occlusion or the object moves and stops alternatively, additional parameters should be taking into account in performing vertex matching. Here, we determine the vertices existence and reference frames as the parameters to handle such problem.

Define a condition of vertices existence as $\gamma \in \{-1, 0, 1\}$ where -1 denotes the number of vertices is decreasing, 0 denotes no changes in the number of vertices, and 1 denotes the number of vertices is increasing. Based on this condition we observe the state of a vertex, defined as $S(v) = \{S_0, S_1\}, S \in \{0,1\}$, where S_0 is the default state of the vertex and S_1 is the occlusion state. A restriction is given that a vertex can only have one state per frame. Thus the states of vertex in frame f are determine as follows:

- Default state S_0=1 is initially set a the beginning of a frame
- Occlusion state S_1=1 is set if the position of two vertices in

previous frame is less than specified threshold $\rho = 4.0$ and $\gamma = -1$

- Disocclusion state S1=0 is set if the position of two vertices in current frame is less than $\rho = 4.0$ and $\gamma = 1$.

The reference frame for the conditional graph tracking is determined by the state of vertex. Let θ be the index of reference frame determined by the state, we define weighted conditional supergraph G^{θ} where the vertices and edges are redefined from (5) and (6) by subtituting f-1 with f-Θ. The conditional similarity is performed over vertex in current frame and vertex in frame f-Θ. Thus the weight is now defined as

$$w^{\theta}\left(v_{n_0}^{f}, v_{n_{-\theta}}^{f-\theta}\right) = S\left(v_{n_{-\theta}}^{f-\theta}\right)_0 \| c\left(v_{n_0}^{f}\right) - c\left(v_{n_{-\theta}}^{f-\theta}\right) \| + S\left(v_{n_{-\theta}}^{f-\theta}\right)_1 \left(\| D\left(v_{n_0}^{f}\right) - D\left(v_{n_{-\theta}}^{f-\theta}\right) \| + \| e\left(v_{n_0}^{f}\right) - e\left(v_{n_{-\theta}}^{f-\theta}\right) \| \right) \quad (9)$$

where $S\left(v_{n_{-\theta}}^{f-\theta}\right)_0$ is the state of $v_{n_{-\theta}}^{f-\theta}$ in default state and $S\left(v_{n_{-\theta}}^{f-\theta}\right)_1$ is the state of $v_{n_{-\theta}}^{f-\theta}$ in occlusion state. Here the weight of edge is now considering the direction and the energy of the vertices as the similarity feature. The value of Θ is determined as the frame index when: 1) the last frame when the vertex appear, for object that moves and stops alternatively; and 2) the last frame when the objects were occluded, for occlusion case.

5. ROI Refinement

In this stage, we define a region of interest (ROI) of object by constructing rectangle that encloses the group of blocks and refine its size by controlling the width and height of the ROI so it would not change too vary due to the detected moving objects never have consistent shape or representing the real objects' shape. This method is performed in the lowest level of the graph hierarchy by observing the parameters of subgraph in five consecutive frames.

Recall subgraph H_n as the graph representing the n-th object in frame f, we define the ROI as $O_n^f = \{\alpha, \beta, a(H_n)\}$ where α is the width of ROI determined from the number of vertices along the horizontal direction of H_n, β is the height determined from the vertical direction of H_n, and the attribute of H_n. Thus the ROI is the rectangle that tightly encapsulates the vertices of H_n. Fig, 3 shows the ROI of a subgraph. At this stage, the size of ROI is changing according to the number of vertices in the subgraph, thus a stable representation of the object cannot be obtained.

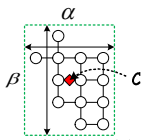


Fig. 3. Determining the width and height of ROI of a subgraph based on the vertices.

The ROI refinement is performed by observing a batch of ROI in five consecutive frames. At the every fifth frame, the average width and height of ROI in this batch are computed and compared to the ROI in the previous batch. Let b denotes the index of a batch, we define $\overline{O}_n^b = \{\overline{\alpha}, \overline{\beta}\}$ as the average size of ROI of the n-th object and refine the ROI according to the following conditions

$$\overline{O}_n^{b+1} = \begin{cases} \overline{O}_n^b : \overline{O}_n^{b+1} \le 3:4 & \frac{3}{4}\overline{O}_n^{b+1} \\ \overline{O}_n^b : \overline{O}_n^{b+1} \ge 4:3 & \frac{3}{4}\overline{O}_n^{b} \\ otherwise & \frac{1}{2}\left(\overline{O}_n^b + \overline{O}_n^{b+1}\right) \end{cases} . \quad (10)$$

Thus the ROI in every frame of the batch is updated to $O_n^f = \left\{\overline{O}_n^{b_f}, a(H_n)\right\}$ where b_f is the batch in which the frame f is included.

In case of occlusion, the subgraphs of the two objects are merged and thus the ROI. Therefore we need to reconstruct the ROI of both objects. To reconstruct the ROI of occluded objects, we use the similarity of attribute of vertices in merged subgraphs before and during the occlusion. Let $C' = \{C_1', ..., C_T'\}$ be the set of mean of the centroid of groups of vertices in merged subgraph H^0 during occlusion having the same attributes, where T is the finite number denoting the number of cluster in H^0. Thus the similarity matching is seeking for the member of H (subgraphs in one frame prior to occlusion) having the most similar attribute with member of H^0, and update $C(H_t^0) = C_t', t \in T$ while the size of the reconstructed ROIs are the same as the size of ROI prior to occlusion. Fig. 4 illustrates the reconstruction method.

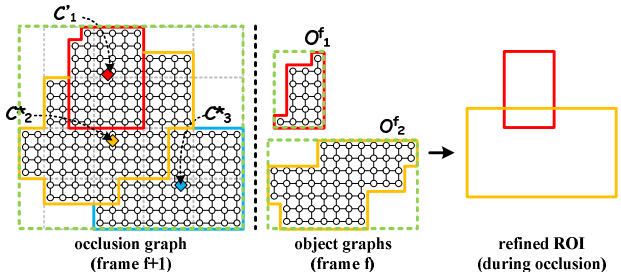


Fig. 4. Three clusters of group of vertices having similar attribute in a merged subgraph. The similarity is calculated and the new location of reconstructed ROI are determined.

6. Experiment Results

We perform our experiments in three sequences: S*peedway* is an outdoor surveillance recording of a roadway in CIF size, *PETS2001* is an outdoor surveillance recording of a school way in QPAL size and S*hinji* is an indoor surveillance recording of a corridor of building in QVGA size. The *Speedway* sequence has the case of small moving objects, the object that moves and stops alternatively, and the two objects that dismerged. The *PETS2001* sequence has the case of occlusion between two objects. The *Shinji* sequence has the case when the object moves toward the direction of the camera thus the motion vectors and residue data sometimes

unable to give sufficient information of the moving objects.

From the experiments in *Speedway* sequence, our method enable to detect object up to smallest size (top-right in Fig. 5). It can also detect the object that unmerged into two objects (bottom-right in Fig. 5). Object 2 is the object that moves and stops alternatively.

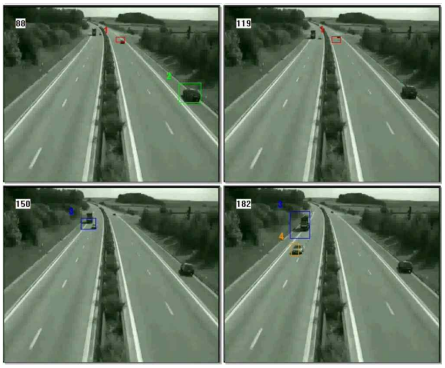
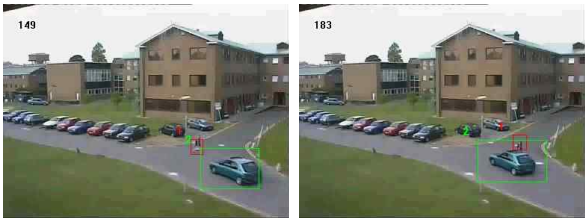


Fig. 5. The result in *Speedway* sequence.

In Fig. 6, we present the result of constructing the ROI of two occluded objects. It is shown that the method can approximate the ROI of object 1 (a person) and object 2 (a car).



(a) (b)

Fig. 6. ROI of two occluded objects in (a)149th and (b) 183rd frame of *PETS2001* sequence.

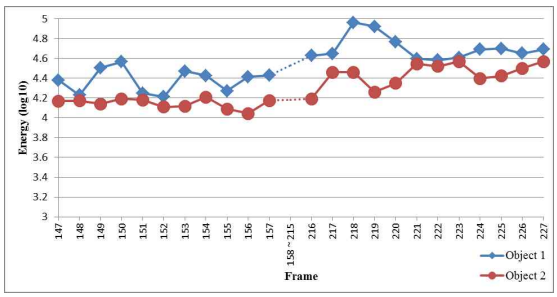


Fig. 7. The graph of energy of residue of two objects.

The graph in Fig. 7 shows the relation between object 1 and object 2 in *PETS2001* sequence in terms of energy. As mentioned, we use direction and energy as similarity feature to distinguish two occluded objects. In Fig. 7 it is shown that two objects can be distinguished by observing its energy value from frame to frame. The energy of object 1 is relatively different from that of object 2, therefore we can compare the energy similarity of each object prior and after occlusion thus obtain the correct identity of both objects after the occlusion.

Lastly, we present the result in case when the object is moving toward the same direction of the camera. In this case, the motion vector is small or sometimes missing that make the object hard to be detected constantly. We shown in Fig. 8 that the object can be detected along the sequence by observing the batch of vertices in supergraph as well as projecting the missing vertices.

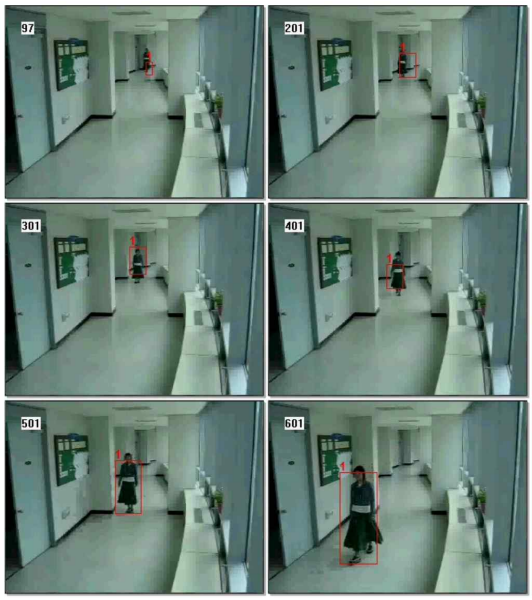


Fig. 8. The result in *Shinji* sequence.

7. Conclusions

In this paper we present our method of object detection and tracking in H.264/AVC bitstream using graph-based method and show good results. Some enhancements to more precisely approximate the objects' shape and to detect object in moving camera will be the future work

References

[1] W. Zeng, J. Du, W. Gao, and Q. Huang, "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.

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

[3] W. You, M. S. H. Sabirin, and M. Kim, "Real-time Detection and Tracking of Multiple Objects with Partial Decoding in H.264-AVC Bitstream Domain," 21st IST/SPIE Symposium on Electronic Imaging: Real-Time Image and Video Processing, San Jose, USA, January 2009.

[4] C. Poppe, S. De Bruyne, T. Paridaens, P. Lambert, and R. Van de Walle, "Moving object detection in the H.264/AVC compressed domain for video surveillance applications", J. Vis. Commun. Image R., vol. 20, 2009, pp. 428-437.