Methods of Video Object Segmentation in Compressed Domain

Cheng Quan Jia

Abstract

Traditional video object segmentation methods operate on the pixel domain, which require every frame in the video sequence to be decoded into raw data. This incurs additional processing and storage overhead which is unfavourable in real-time application. Recently, video object segmentation in the Compressed Domain, i.e. video compressed using Motion Compensation, Discrete Cosine Transform(DCT) and Quantization, have gained attention because it only requires the compressed video to be parsed to obtain the motion vectors and DCT coefficients for segmentation. This paper outlines the features present in the Compressed Domain that can be used in video object segmentation, segmentation methods published by recent researchers, and possible research areas in the future.

1 Introduction

Traditional video object segmentation is performed in the pixel domain, in which pixel data are obtain from full decoding of the video bitstream. The motion flow is extracted by comparing consecutive frames and the basic data used is intensity value from each pixel. Although the traditional approaches have reached certain maturity and give reasonable segmentation results, the processing and storage overhead in decoding every frame from a video sequence prevents these methods from application in real-time applications. Recently, Compressed Domain segmentation have gained attention not only because it does not require full decoding of the compressed video, but the motion information that are already present in the Compressed Domain could help video object segmentation.

The term Compressed Domain in literature refers to video compression methods in which motion compensation and Discrete Cosine Transform (DCT) are used to reduce the number of bits required to represent a video, examples include MPEG-1/-2/-4, H.261 and H.263. All of these compression standards achieve compression by exploiting two observations. Firstly, it is unusual for intensity values to change frequently over a small area(spatial redundancy). Secondly, consecutive frames along time-ordered sequence of frames are similar(temporal redundancy). The Compressed Domain address the first observation with DCT and Quantization and the second with Motion Compensation, both of which are described in the following two sections.

1.1 Intraframe Compression

Suppose we have a picture frame. Based on the first observation, the frame is divided into 8×8 macroblocks. Then DCT is performed on each macroblock. Discrete Cosine Transform turns the intensity values in a macroblock into representation of sums of cosine functions oscillating at different frequencies. The transform used in video compression is a two-dimensional one, which transforms the array of spatial intensity values to an array of DCT coefficients, each coefficient denoting the value of vertical and horizontal frequencies, i.e. frequencies of intensity change, in the macroblock. The DCT coefficients are arranged in the array in a way that the DC coefficient(constant) lies at the top-left element of the array and elements further to the right and down contains AC coefficients of cosine functions with higher horizontal and vertical frequencies respectively.

Quantization is employed on each DCT-ed macroblock in order to remove the high spatial frequency components. Each coefficient in the macroblock is quantized by a rounded division with a quantization value. To achieve bias against high-frequency components, a quantization table with higher quantization values towards the lower right corner is used. The resulting macroblock usually has the high-frequency components discarded. Finally, the DCT coefficients are subjected to Entropy Coding.

1.2 Motion Compensation

According to the second observation, temporal redundancy exists between consecutive frames in a video sequence. Even more bits can be saved using Motion Compensation rather than encoding frames as a whole image (as in Intraframe Compression). Again, the current frame is divided into macroblocks. Each macroblock is compared against the reference frame (a reconstructed frame previous to the current frame) within a small neighbourhood search window for the best match, i.e. the position with the least difference between current and reference macroblock. The result is a predicted motion vector and a predicted macroblock from the reference frame. Since the best match may not be identical to the current macroblock, the difference between the current macroblock and the predicted macroblock gives a difference macroblock to denote the prediction error. The difference macroblock undergoes DCT and Quantization. Finally the motion vector and the difference macroblock undergo Entropy Coding.

Note that to avoid propagation of error, the I-frame is sent after a number of P-frames. The forms Group of Pictures(GOP) structures of IPPP... frames. Also, MPEG standards employ Bidirectional Motion Compensation in addition to forward prediction. Another type of frame, the B-frame, is introduced in addition to the P-frame so that the GOP structure becomes IBB...PBB...PB... A B-frame is constructed by predicting from its previous I/P-frame and its next I-/P-frame such that two sets of motion vector and predicted macroblock are found. Both the predicted macroblock and motion vectors are averaged and compared against the current frame to generate the difference macroblock.

1.3 Features for Motion Segmentation

Since image change detection could be seen as a classification problem [7], this section lists the features available in the Compressed Domain. While full decoding is not required in the Compressed Domain, the motion vector(in interframes) and DCT coefficients(of image data in intraframes and difference image in interframes) are required for video object tracking. Parsing, in which the input binary bitstream is first entropy decoded then inverse quantized, extracts these necessary information [6]. Therefore, the information available in the Compressed Domain is the DCT coefficients of the picture macroblocks in the intraframes and difference macroblocks in the interframes, and motion vectors associated with the predictive macroblocks. Of particular importance is the motion vectors and top row and and left column DCT coefficients in the I-frames that denotes vertical and horizontal spatial frequency respectively [5, 3].

Note that the motion vectors mentioned above are generated for the best match in the reference frame rather than generated to denote video object motion. Including motion vectors that are uncorrelated to true motion degrades segmentation accuracy, which is discussed in the next section. Several assumptions should hold for the following discussion. Firstly, smooth and relatively small motion should exist in the input video otherwise most of the macroblocks would be intracoded instead of intercoded and we would have few motion vectors to work with. Secondly, the motion in the input video should span over more than one Group of Pictures.

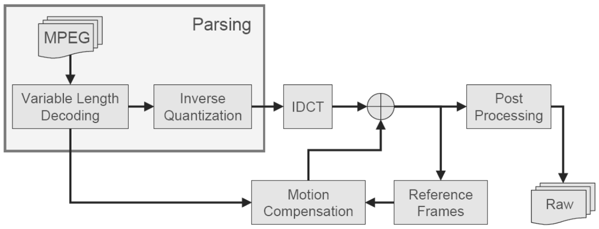

Figure 1. Parsing from MEPG bitstream[6]

2 Related Work

2.1 Video object segmentation using motion vectors

Since motion information in a video sequence is critical in video object segmentation, sparse motion vectors present in predictive macroblocks are used by many researchers for video object segmentation. Their methods involve constructing dense motion field by accumulating sparse motion vectors or obtaining average motion vectors over a series of P- or B-frames. As the motion vectors in encoded video sequences do not reflect object motion, outliers do occur. Many segmentation algorithms([1], [5], [3]) remedies this by median filtering or other methods.

Liu et al. [5] uses motion vectors to locate coarse video object regions then DCT coefficients as spatial feature to identify similar macroblocks. Firstly, the motion vectors are accumulated and median-filtered to obtain a motion field. Secondly, based on the motion field, the macroblocks in the object edge area of non-zero motion vectors are rectified by similarities of of DCT blocks. Rectification removes pseudo moving blocks(blocks out of a video object but within non-zero motion field) from the segmentation area and include pseudo still blocks(blocks within a video object but out of non-zero motion field) in the segmented video area and uses the belief that pseudo moving blocks and pseudo still blocks appear mainly in edge areas to aid rectification. It selects the moving blocks that appears in the edge area as the center moving block and finds similar blocks in the four-neighbourhood, if they do not exist then the center moving block is deemed pseudo moving block and be discarded from segmentation area. Otherwise, for each still block from the four-neighbourbood of blocks, calculate the distance between the features(the mean, horizontal edge, vertical edge and diagonal edge) of the center moving block and the chosen moving block and the distance between the center moving block and the neighbouring moving blocks; if the still block is closer to the center moving block than the neighbouring moving blocks, it is deemed as a pseudo still block and be included in the segmentation area.

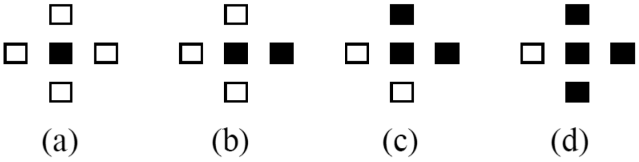

Figure 2. Four cases of edge area (a)Four still blocks around a moving block (b)Three still blocks around a moving block (c)Two still blocks around a moving block (d)One still block beside a moving block[5]

The experimental results of [5] show that the segmentation masks, while envelop the desired video objects, tend to be a few macroblocks wider than ground truth. The reason behind this is the inclusion of similar macroblocks(in terms of DCT coefficients) as pseudo still blocks.

Hariharakrishnan and Schonfeld [4] used only motion vectors as their segmentation feature. In [4], an adaptive block matching algorithm is used. From the initial object mask, macroblocks that lie entirely in the object are chosen as seed motion blocks when the other macroblocks in the object mask are labeled uncertain blocks. Then, from every three frames before the current frame, backward motion estimation is performed so that the average motion between consecutive frames is obtained. In case the average motion is higher than a threshold, motion is estimated from the previous frame instead. Then the object mask for the current frame is derived by motion compensated from the motion obtained in the previous step, and detect for occlusion and disocclusion. For disocclusion detection, the regions that will be uncovered in the current frame is estimated from previous frames using motion compensation. The uncovered regions are tested against the object for motion consistency, by first clustering motion vectors in the region using k-means clustering then comparing it with the motion vector of the object. If the difference between the two motion vectors is smaller than a threshold the uncovered region is treated as disocclusion and included in the object. [4] views occlusion and disocclusion as dual events, therefore occlusion detection is similar to disocclusion, except covered regions are obtained from the next frame and are tested for motion dissimilarity.

Yokoyama et al. [10] uses Vector-featured Images obtained in the MPEG sequence to discover and track moving video objects. In each image, macroblocks are classified into five types: the Current Block, the Reference Block, the Background Block, the Moving Block and the Unmoving Block. The Reference Block is a macroblock in the I-frame associated to an initial point of a motion vector; the Current Block is a macroblock associated to the end of a motion vector; the Moving Block is created at the overlap of Current Block and Reference Block and indicates a moving region; the Unmoving Block serves to keep track of regions that momentarily stop. An Unmoving Block is created when a Current Block generates zero motion, and decrease its brightness on successive frames, until its maximum life expires. When the object resumes motion, if the Moving Blocks overlaps with the Unmoving Blocks, the Unmoving Blocks are updated to Moving Blocks. Under this scheme, motion is detected if from the Moving Blocks if the Moving Region is large enough, otherwise the union of Moving and Unmoving Blocks is used instead. The moving object candidate is compared with previously registered objects to achieve object tracking.

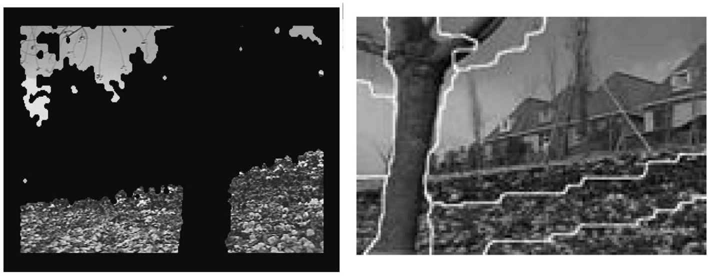

Figure 3. Extracted flower bed object from [1] (above) and segmentation result from [2] (below). A portion of the sky is included in their segmentation masks.

It is observed from the experimental results of [1] and [2] that the segmentation masks tend to cover areas other than the video objects. This is more prominent in areas with similar texture. It is because the encoder looks for the best match in the block matching process rather than true object motion. In addition, the motion field in P-frames, due to larger temporal distance to reference frames, are less reliable [8]. Therefore, contaminated with erroneous motion vectors, the aggregated motion field would not give a correct boundary. The system in [4] avoids this slightly since it has an initial segmentation mask to work with(it uses a four-band multi-valued segmentation followed by a lattice partition operator) and it switches to calculating the motion from the next frame when the average motion over the next three frames is higher than a threshold, and [10] imposes a minimum bounding rectangle to prevent inclusion of uncorrelated motion. Still, there should be some form of confidence measure that ensure a motion vector is approximated to real object motion, which is addressed in the following subsections.

2.2 Porikli et al.'s investigation in the Compressed Domain

The previous assertion that motion vectors alone do not suffice in accurate video object segmentation finds ground in an article by Porikili et al[6]. They proposed a video object segmentation system that experimented with almost all of the information present in the Compressed. The system uses the DCT coefficients in the I-frame motion vectors of P-frames in a Group of Pictures to construct Frequency-Temporal(FT) data structures for each macroblock in the GOP. The FT data consists of the following:

- The DC parameters(for Y, U, V channels) of the I-frame
- A subset of low vertical and horizontal frequency AC values
- A spatial energy term(total magnitude of AC coefficients) measuring spatial variance
- Aggregated motion flow of the corresponding macroblock

The aggregated motion flow of the macroblock is the mean of aggregated pixel motion vectors, which is obtained by interpolation of the filtered motion vector in a macroblock for all P-frames in a GOP then back-propagate from the last frame to the first. Porikli et. al [6] includes two segmentation approaches in their article. One uses FT volume growing, in which macroblock with the lowest local variance(derived from the energy in the local spatial and temporal neighbourhood) is chosen as the seed block and the volume is grown in both 2D spatial and temporal dimensions. The other employs Multi-Kernel Mean-Shift Segmentation, one in spacial-temporal dimension, one in aggregated motion vector space, one in DCT coefficient space. The process shifts these kernels at the same time computes their gradient, until their sink points are found. It goes on linking sink points closer than than a preset value from each other in the joint domain, to form clusters of sink points. The points in the clusters constitutes a segmentation volume. Both approaches would have volumes of negligible size removed and remaining volumes inflated. Finally, hierarchical clustering is performed and a partition tree is built by iteratively merging pairs of most similar volumes.

Porikli et. al's experimental results show that both segmentation approaches produce similar results. Also, a slight over segmentation using DCT coefficients followed by aggregated motion based clustering produces more accurate boundaries than single stage joint segmentation. They attribute this to the fact that motion boundaries tend to be deformed and erroneous. Also, using all of the DCT coefficients do not necessarily provide a stable segmentation in that the mean-shift algorithm becomes sensitive when AC components and spatial energy term are included. Note that the best combination stated above renders the system to segment video objects with similar average intensity value and texture, which in turn sensitive to intensity differences; in addition, the proposed algorithm favours moderate motion since spatial-temporal volumes would be disjoint in the presence of large motion.

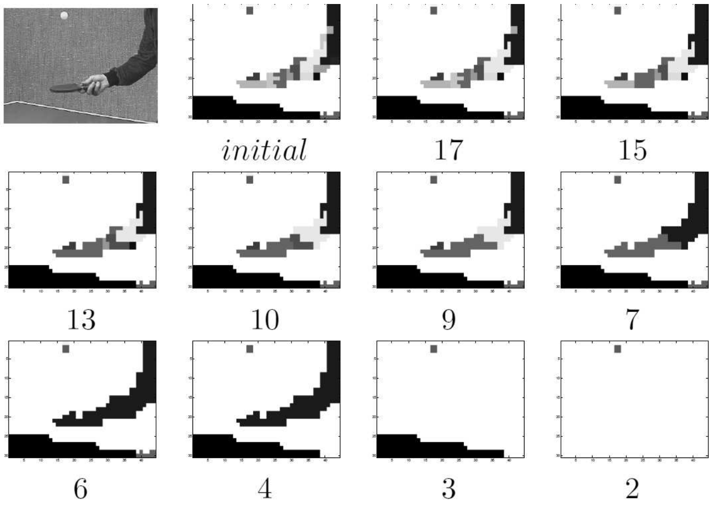

Figure 4. Porikli et al.'s segmentation results at the corresponding clustering levels. Note the volume growing process could not blend the lower part of the arm into other regions since its DCT coefficients were also significantly different.[6]

2.3 Approximation of Optical Flow and confidence measure

Coimbra [3] tries to approximate optical flow from motion vectors and DCT coefficients in MPEG-2 Compressed Domain, in comparison to the Lucas-Kanade algorithm. It reasoned that the optical flow and eigenvalues for confidence measure is parallel to motion vectors and AC coefficients found in Compressed Domain respectively. The proposed method obtains a smooth dense motion field from a Group of Pictures, in which motion vector magnitude is normalized and motion vector from the previous macroblock is interpolated to macroblocks that have no motion information. The motion obtained from this method is smoothed by median filtering to remove isolated motion vectors. Then the first vertical AC coefficient and the first horizontal AC coefficient (AC[1] and AC[8] in a macroblock) in the I-frame macroblocks is used as a confidence map. The confidence update step will have a 8×8 macroblock referencing a 16×16 image block in the I-frame, and the confidence of the motion vector of the macroblock is the weighted average of confidence in the 16×16 window. The optical flow estimation technique in [3] is employed in [9] to estimate motion for each macroblock in a video sequence to achieve action recognition and localization.

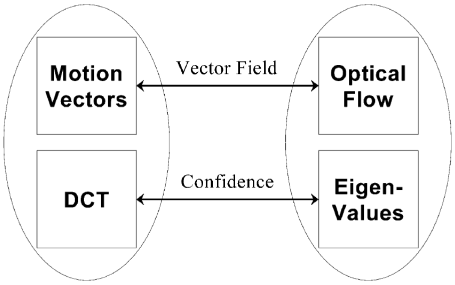

Figure 5. Comparison of the LK and MPEG-2 system[3]

The experimental results in [3] clearly demonstrates the effects of employing confidence measure on the obtained motion field. The motion vectors after harmonization step and median filtering have motion information uncorrelated to object displacement caused by illumination noise and aperture problem. After the confidence update step motion vectors with high confidence measure(usually those at boundary areas) are kept.

2.4 Pixel precision edge refinement in raw domain

While they are not working entirely in the Compressed Domain, the systems proposed by Babu et al. [1] and Chen and Bajic [2] can further refine their coarse segmentation results to achieve segmentation with pixel precision. Both algorithms first identify the edge macroblocks within the segmented area, then perform edge refinery to obtain the precise object boundaries. Both algorithm requires the edge macroblocks to be fully decoded in the edge refinement step. The algorithms still perform faster than raw domain segmentation algorithms, despite the fact that full decoding is performed the blocks.

The algorithm in [1] performs coarse segmentation by first obtaining a dense motion field by accumulating motion vectors over a few frames forward and backward then applying a 2D median filter and a Gaussian filter. Only reliable motion vectors i.e. vectors that correspond to macroblocks that have total DCT error energy less than a threshold is used in the process. If the macroblock is unreliable or intracoded, motion corresponding to the block is interpolated from the neighbouring blocks. Next it uses the Expectation Maximization(EM) Algorithm to compute the likelihood of each pixel to a number affine motion models, which is determined using k-means clustering. The result of the EM Algorithm is the coarse segmentation result. Edge refinement in [1] involves decoding the edge blocks and their eight neighbourhood, computing representative motion vectors for the block, defining a search range based on the motion vector, and matching a small pixel block against the previous frame within the search region.

Unlike [1], [2] obtains dense motion field using Motion Vector Integration, in which coherent motion integration(backward summing up of normalized MVs over frames) and incoherent motion integration(adding of MV magnitude over successive coherent integrations) are combined. Coarse segmentation is achieved using k-means clustering and blocks are identified as a boundary block if the block's eight neighbourhood are from more than one region. [2]'s edge refinement involves Pixel-based MV Integration, in which the pixels in a macroblock acquire the MV of their corresponding macroblock in the previous frame, to interpolate MV into boundary regions in order to increase their MV density, Canny Edge detector and morphological operations to obtain object boundary in pixel accuracy.

3 Observations

Several observations can be obtained from the above segmentation methods. To use motion vectors as a suitable indiction of object motion, a dense motion field should be constructed from the sparse motion vectors present in inter-coded macroblocks. [1], [2], [5], [6] and [3] either use motion accumulation or use motion aggregation to deduce the resulting motion field over frames. In addition, [1], [5], [6] and [3] clearly state that Motion Compensation in the encoding process only gives motion vectors of the best match macroblock but not the actual motion in the video sequence. This is particularly prominent when compensating large objects with similar interior texture. Also, the Motion Compensation step is suspectable to aperture problem as the process concerns only with displacement of texture in the macroblock. While median filtering removes outlying motion vectors, the filtering is also blind. The confidence measure introduced in [3] ensures to some degree that the motion vectors in macroblocks with high vertical and horizontal edge energy is trustable, and [1] chooses the motion vectors of inter-coded macroblocks that has DCT error energy lower than a threshold to be reliable motion vectors. For macroblocks that are intra-coded, motion vectors are interpolated to them from negihbouring regions.

While motion accumulation, motion vector interpolation and filtering guarantee a dense motion field, repetitive motion over large sequences of frames leads to motion cancelation. [2] migitate this by summing motion vector magnitudes over a series of frames. The segmentation method in [10] does not face this problem since it uses vector images to memorize object trajectory, but loses the benefit of filtering out unreliable motion vectors.

In addition to motion vectors, DCT coefficients from both intra-coded macroblocks and difference macroblocks are chosen as a segmentation feature in some aforementioned methods. [5] uses DC the component and AC edge energy as features for calculation of distance between macroblocks, and [6] uses DC component, AC edge energy and spatial energy to construct a feature vector for each macroblock. Note that the DC coefficient (in both intra-coded and inter-coded macroblocks) is not used as a determining feature since DC coefficient denotes average energy over an 8×8 block. Also, the AC coefficients used in [3], i.e. the first horizontal and vertical AC coefficients, are convincing measure of edge strength and can be used to identify object boundaries.

4 Conclusion

In this paper, we have introduced some recent methods on video object segmentation in the Compressed Domain. Compressed Domain video object segmentation involves first obtaining the motion vectors(in inter-coded frames) and DCT coefficients of macroblocks from the input video sequence by parsing then constructing object areas or boundaries from a dense motion field. To summarize the methods listed in this paper, a Compressed Domain motion segmentation should perform the following: upon receiving the parsed input video sequence, motion is accumulated over several frames so that dense motion are captured, motion vectors that are associated with a macroblock with error higher than a threshold is discarded and motion vector is interpolated from the block's neighbours [1]; the accumulated motion is subjected to confidence thresholding using confidence map obtained by [3]; moving objects are segmented from the initial frame, either by clustering macroblocks into motion models [1] or identify macroblocks that has high horizontal or vertical AC energy as edge block candidates then rectify the boundary by removing pseudo moving/still macroblocks [5]; the coarse object mask is detected for occlusion/disoccusion by discovering covering/uncovering regions and test for motion consistency/inconsistency [4]; finally edge refinement is performed on the coarse segmentation mask, in which the edge blocks and their eight neighbourhood are decoded, computing representative motion vectors for the block, defining a search range based on the motion vector, and matching a small pixel block against the previous frame within the search region [1].

While current Compressed Domain technique greatly reduce the processing and storage complexity of segmentation, due to fact that the block matching process in video compression is sensitive to changes of intensity values and problems associated with optical flow such as the aperture problem. Also not discussed in this paper are method to estimate motion in the presence of camera motion and video object segmentation in the presence of scene cuts. Both are interesting problems that worth investigation.

References

[1] R. V. Babu, K. R. Ramakrishnan, and S. H. Srinivasan. Video Object Segmentation: A Compressed Domain Approach. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(4):462–473, April 2004.

[2] Y.-M. Chen and I. V. Bajic. Compressed-Domain Moving Region Segmentation with Pixel Precision using Motion Integration. In *IEEE Pacific Rim Conference on Computers and Signal Processing, 2009*, pages 442 – 447, August 2009.

[3] M. T. Coimbra and M. Davies. Approximating Optical Flow Within the MPEG-2 Compressed Domain. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(1):103–107, January 2005.

[4] K. Hariharakrishnan and D. Schonfeld. Fast Object Tracking Using Adaptive Block Matching. *IEEE Transactions on Multimedia*, 7(5):853–859, October 2005.

[5] L. Long, F. Xingle, J. Ruirui, and D. Yi. A Moving Object Segmentation in MPEG Compressed Domain Based on Motion Vectors and DCT Coefficients. In *Congress on Image and Signal Processing, 2008*, volume 3, pages 605–609, May 2008.

[6] F. Porikli, F. Bashir, and H. Sun. Compressed Domain Video Object Segmentation. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(1):2–14, January 2010.

[7] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image Change Detection Algorithms: A Systematic Survey. *IEEE Transactions on Image Processing*, 14(3):294–307, March 2005.

[8] R. Wang, H.-J. Zhung, and Y.-Q. Zhang. A confidence measure based moving object extraction system built for compressed domain. In *The 2000 IEEE International Symposium on Circuits and Systems, 2000*, volume 5, 2000.

[9] C. Yeo, P. Ahammad, K. Ramchandran, and S. S. Sastry. High-Speed Action Recognition and Localization in Compressed Domain Videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(8):1006–1015, August 2008.

[10] T. Yokoyama, T. Iwasaki, and T. Watanabe. Motion Vector Based Moving Object Detection and Tracking in the MPEG Compressed Domain. In *Seventh International Workshop on Content-Based Multimedia Indexing, 2009*, pages 201–206, June 2009.