Moving Region Segmentation from Compressed Video Using Global Motion Estimation and Markov Random Fields

Yue-Meng Chen, Ivan V. Bajić, *Member*, *IEEE*, and Parvaneh Saeedi, *Member*, *IEEE*

***Abstract*—In this paper, we propose an unsupervised segmentation algorithm for extracting moving regions from compressed video using Global Motion Estimation (GME) and Markov Random Field (MRF) classification. First, motion vectors (MVs) are compensated from global motion and quantized into several representative classes, from which MRF priors are estimated. Then, a coarse segmentation map of the MV field is obtained using a maximum a posteriori estimate of the MRF label process. Finally, the boundaries of segmented moving regions are refined using color and edge information. The algorithm has been validated on a number of test sequences, and experimental results are provided to demonstrate its superiority over state-of-the-art methods.**

***Index Terms*— Motion segmentation, global motion estimation, global motion compensation, Markov Random Field, compressed video**

I. Introduction

Moving object segmentation is an important problem in a variety of applications such as video surveillance, video database browsing, object-based video transcoding, etc. During the last two decades, a number of approaches have been proposed to tackle this problem. Especially interesting is the problem of moving object segmentation in compressed video, due to the abundance of compressed video content.

State-of-the-art object segmentation methods can be broadly grouped into pixel-domain approaches (e.g., [1–3]) and compressed-domain approaches (e.g., [4-11]). The former extract objects by exploiting visual features such as shape, color and texture. In this case, the compressed video has to be fully decoded prior to segmentation. The high computational load and over-segmentation are two major drawbacks of these methods. On the other hand, compressed-domain methods exploit compressed-domain data, such as motion vectors (MVs) and DCT coefficients, to facilitate segmentation. Some methods [4–5] operate directly on sparse (block-based) MV field. These methods have low complexity, but often suffer from poor localization of object boundaries, and inconsistency in the number of segmented regions from frame to frame. The presence of camera motion often worsens the performance of these segmentation approaches, and objects may be over-segmented due to motion bias introduced by camera movement [14]. Alternatively, one can create a dense (pixel-based) MV filed by interpolation, and then run segmentation on the dense field, at the cost of significantly higher complexity [6–8]. Combinations of compressed-domain and pixel-domain operations have also been proposed to balance complexity and accuracy [9–11]. These methods first create a coarse segmentation from the sparse MV field, and then refine it in the pixel domain. Although these methods generally offer higher segmentation accuracy near object boundaries than purely compressed-domain approaches, maintaining a consistent number of segmented regions across frames can still be a challenge.

The segmentation method proposed in this paper extends our earlier work on combined-compressed domain and pixel-domain segmentation [15] by incorporating global motion estimation (GME) and global motion compensation (GMC). Briefly, our method proceeds as follows. First, GME and GMC are employed to remove the influence of camera motion on the MV field. Then, MV vector quantization (VQ) based on local motion similarity is used to find the most likely number of moving regions. The statistics of the VQ clusters are used to initialize prior probabilities for subsequent Markov Random Field (MRF) classification, which produces a coarse segmentation map. Finally, a coarse-to-fine strategy is utilized to refine region boundaries. While each of these components has been employed in previous segmentation approaches, we believe that the complete solution incorporating all the listed components is novel, and represents the main contribution of this work. Through such comprehensive strategy, the proposed segmentation framework is able to overcome some of the difficulties faced by previous methods, such as over-segmentation [1-3], under-segmentation [4], and segmented region inconsistency [9-11]. Further, coarse-to-fine boundary refinement yields more accurate region boundaries than compressed-domain methods [4-5], while still maintaining a much lower complexity than pixel-domain methods [6-8].

The paper is organized as follows. In Section II, we describe GME, GMC, and MV quantization. The segmentation

This work was supported in part by the NSERC/CCA New Media Initiative Grant STPGP 350740. The authors are with the School of Engineering Science, Simon Fraser University, Burnaby, BC, V5A 1S6, Canada. Tel: 1-778-782-7159; Fax: 1-778-782-4951; E-mail: yuemengc@sfu.ca, ibajic@ensc.sfu.ca, psaeedi@sfu.ca.

framework and its major components are elaborated in Section III, followed by the experimental results in Sections IV. The conclusions are drawn in Section V.

II. Global Motion Compensation And MV Quantization

The main difficulty in MRF segmentation is to determine the parameters that specify the MRF, particularly the number of moving regions and their statistics. Our approach is to first perform vector quantization (VQ) of MVs in order to estimate these parameters. However, directly performing VQ on MV field may lead to inaccurate parameter estimates due to two reasons. First, MVs are usually generated to maximize the coding efficiency, rather than represent true motion. Second, MVs are biased by the camera motion (a.k.a., global motion) associated with the sequence, which may cause erroneous region clustering. Thus, to achieve robust VQ, we use GMC to remove global motion. We also try to suppress the influence of possibly inaccurate MVs by examining the smoothness of the MV field.

A. Global Motion Compensation

The first step in GMC is the estimation of global motion parameters from coarsely sampled MV fields extracted from the compressed video [16-18]. We use a perspective model with eight parameters, $\mathbf{m} = [m_0, \ldots, m_7]$, to represent global motion. This model describes the 2-D projection of the 3-D motion of a planar surface, so it is often used to model the motion of the background, which is assumed far from the camera. Given (x, y) and (x', y') as the coordinates in the current and the reference frame, respectively, the perspective transformation is given by:

$$x' = \frac{m_0 x + m_1 y + m_2}{m_6 x + m_7 y + 1}, \quad y' = \frac{m_3 x + m_4 y + m_5}{m_6 x + m_7 y + 1}. \tag{1}$$

The X- and Y- components of the MV at (x, y) in the current frame corresponding to this motion model are given by:

$$\mathbf{MV}^X(x, y; \mathbf{m}) = x' - x, \quad \mathbf{MV}^Y(x, y; \mathbf{m}) = y' - y. \tag{2}$$

Estimating global motion parameters ($\mathbf{m}$) from noisy MV fields involves two steps, often performed iteratively: outlier removal and parameter estimation. Previous work on this topic includes three prominent approaches: the iterative gradient descent [16], the least square solution with an M-Estimator (LSS-ME) [17], and RANdom SAmple Consensus (RANSAC) [19]. In this work, we use LSS-ME for estimating $\mathbf{m}$ due to its superior performance over other two approaches when 8-parameter motion model is applied [18].

Let $\mathbf{m}_t$ be the vector of estimated GM parameters from LSS-ME in frame t. The global motion can be compensated from the MV at location (x, y) in frame t by:

$$\mathbf{MV}^{res}(x, y, t) = \mathbf{MV}(x, y, t) - \mathbf{MV}(x, y; \mathbf{m}_t), \tag{3}$$

where $\mathbf{MV}^{res}(x, y, t)$ is the compensated MV at location (x, y), and $\mathbf{MV}(x, y; \mathbf{m}_t)$ is obtained as in (1) and (2). Motion quantization is then conducted on $\mathbf{MV}^{res}(x, y, t)$.

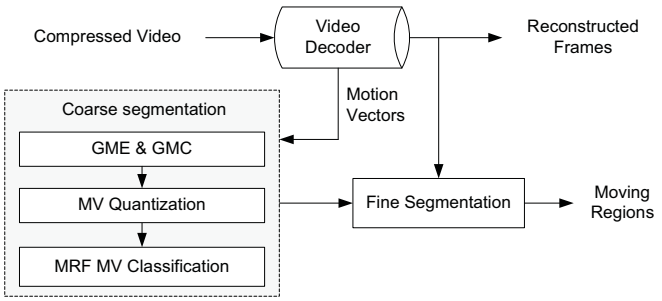


Figure 1: Overview of the MRF moving region segmentation system.

B. MV Quantization

In our scheme, a MV that is very different from its neighbors, and therefore suspected to be inaccurate, will have less influence on the resulting quantization. A similar idea was studied in [2] in the context of color quantization. We first apply a 3×3 vector median filter to the global motion-compensated MV field. Then, for each motion vector $\mathbf{MV}_j^{res}$, we find the maximum Euclidean distance $D_{MAX,j}$ from its 8-adjacent neighbors, and assign it the weight $W_j = \exp(-D_{MAX,j})$. Using these weights, we run a generalized Lloyd algorithm for vector quantization:

1) Start with a single cluster (all MVs in the frame), compute its centroid $\mathbf{MV}_{cent}$ as
$$\mathbf{MV}_{cent} = \left(\sum_j W_j \mathbf{MV}_j^{res}\right) / \left(\sum_j W_j\right), \tag{4}$$
then split it into two clusters by deriving two new centroids as $\mathbf{MV}_{cent} \pm \mathbf{MV}_{cent}/2$.
2) Quantize all MVs in the frame into existing clusters using the nearest neighbor criterion. Then, for the i-th cluster C_i, update the centroid MV as
$$\mathbf{MV}_{cent}^{C_i} = \left(\sum_{\mathbf{MV}_n^{res} \in C_i} W_n \mathbf{MV}_n^{res}\right) / \left(\sum_{\mathbf{MV}_n^{res} \in C_i} W_n\right). \tag{5}$$
3) Compute the weighted distortion of each cluster C_i:
$$WD^{C_i} = \sum_{\mathbf{MV}_n^{res} \in C_i} W_n \left\| \mathbf{MV}_n^{res} - \mathbf{MV}_{cent}^{C_i} \right\|. \tag{6}$$
Let C_k be the cluster with the maximum weighted distortion, and let $X_{\max}$, $X_{\min}$, $Y_{\max}$, and $Y_{\min}$ be, respectively, the maximum and minimum horizontal and vertical components among the centroids. Split cluster C_k into two clusters with centroids $\mathbf{MV}_{cent}^{C_k} \pm \mathbf{P}$, where
$$\mathbf{P} = \left(\frac{X_{\max} - X_{\min}}{2(N-1)}, \frac{Y_{\max} - Y_{\min}}{2(N-1)} \right), \tag{7}$$
and N is the total number of clusters prior to splitting.
4) Repeat steps 2) and 3) until the total weighted distortion (sum of all WD^{Ci}) becomes less than a given threshold (in our experiments, 5% of its initial value in step 1), or the smallest cluster size becomes less than another threshold (in our experiments, 5% of the total MV field size).

Upon completion, a preliminary segmentation map is obtained: global motion-compensated MVs in cluster C_i will be used to compute MRF priors.

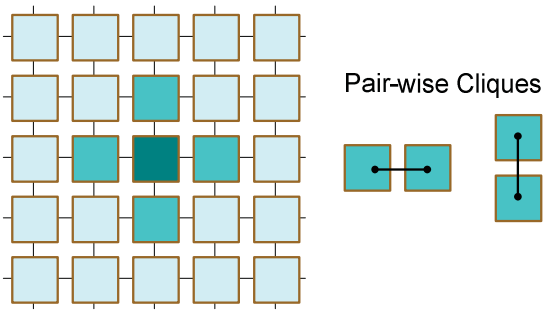


Figure 2: First-order MRF system and clique configuration.

III. Markov Random Field Motion Segmentation

Block diagram of the proposed segmentation system is shown in Fig. 1. The framework uses a coarse-to-fine segmentation strategy with two major components: coarse segmentation from motion, which is carried out in the compressed domain, and fine segmentation, performed in the pixel domain near moving region boundaries. GME and GMC, along with MV quantization, serve as the foundations for coarse segmentation, and facilitate the computation of the priors for Markov Random Field (MRF) MV classification.

A. Markov Random Field Motion Model

Our approach to coarse motion segmentation is based on a Markov Random Field (MRF) motion model [1], [3], [8]. In this model, motion vectors **MV** = (MV^X, MV^Y) within a given moving region ω follow a conditional distribution $P(\mathbf{MV} \mid \omega)$, while region labels ($\omega$s) follow a 2-D MRF distribution based on a given neighborhood system. The goal is to infer region labels (ωs) from the observed MV field.

To simplify calculations, we assume that within each region, MVs form an independent bivariate Gaussian process. Under this assumption, the likelihood function for the j-th block in the frame is

$$P(\mathbf{MV}_j \mid \omega_j) = \frac{1}{\sqrt{2\pi}\sigma^X_{\omega_j}\sigma^Y_{\omega_j}} \cdot \exp\left(-\frac{1}{2}\left(\frac{(MV^X_j - m^X_{\omega_j})^2}{(\sigma^X_{\omega_j})^2} + \frac{(MV^Y_j - m^Y_{\omega_j})^2}{(\sigma^Y_{\omega_j})^2}\right)\right), \quad (8)$$

where $m^X_{\omega_j}$ and $m^Y_{\omega_j}$ are the means of the horizontal and vertical MV components within the region labeled ω_j, while $\sigma^X_{\omega_j}$ and $\sigma^Y_{\omega_j}$ are the corresponding standard deviations. The dependence among the labels of neighboring blocks is modeled by a MRF which follows the Gibbs distribution:

$$P(\omega_j) = \frac{1}{Z}\prod_C \exp(-V(C)), \quad (9)$$

where Z is the normalizing constant ensuring that $\Sigma P(\omega_j) = 1$, C is a *clique* (a set of neighboring blocks) and $V(C)$ is the *clique potential*. We only consider 4-adjacency cliques. In other words, two blocks form a clique if one is immediately to the North, South, East, or West of the other, as shown in Fig.2.

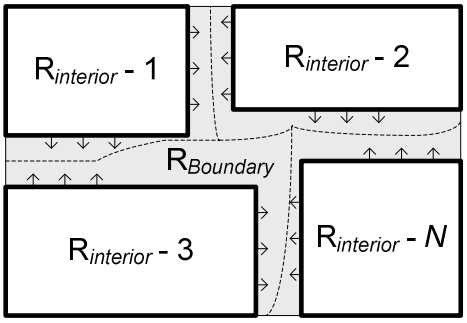


Figure 3: Interior regions grow within boundary regions.

If ω_1 and ω_2 are the region labels of the two blocks in the clique C, the potential of C is defined to be

$$V(C) = \begin{cases} -\beta, & \text{if } \omega_1 = \omega_2, \\ +\beta, & \text{otherwise.} \end{cases} \quad (10)$$

where $\beta > 0$ is a parameter controlling the homogeneity of the regions. Based on (9) and (10), nearest neighbors are more likely to have the same region label.

The MRF priors, i.e., the number of regions, $m^X_{\omega_i}$, $m^Y_{\omega_i}$, $\sigma^X_{\omega_i}$, $\sigma^Y_{\omega_i}$ and $P(\omega_i)$, are determined after MV quantization. The MVs in each cluster C_i obtained from quantization are given the region label ω_i, and the means and standard deviations are computed from MVs in each cluster.

B. MRF Motion Segmentation

For block j, based on the Bayes' theorem, the posterior probability $P(\omega_j \mid \mathbf{MV}_j)$ is proportional to $P(\mathbf{MV}_j \mid \omega_j)P(\omega_j)$, so the Maximum A Posteriori (MAP) estimate of ω_j is given by:

$$\hat{\omega}_j = \arg\max_{\omega_j} P(\mathbf{MV}_j \mid \omega_j)P(\omega_j), \quad (11)$$

where $P(\mathbf{MV}_j \mid \omega_j)$ is computed as in (8) and $P(\omega_j)$ as in (9)–(10). The MAP segmentation for the entire MV field corresponds to maximizing:

$$\prod_j P(\mathbf{MV}_j \mid \omega_j)P(\omega_j), \quad (12)$$

and is obtained using the method of Iterated Conditional Modes (ICM) [13], by iteratively solving (12) for each block in the frame. We use the ICM implementation from [3] (modified for MV segmentation instead of pixel segmentation), with six iterations. The final step is to identify small regions whose size is less than 2% of the total MV field, and group each block in those regions to the neighboring large region with the closest centroid MV.

C. Boundary Refinement

Segmentation map obtained from the coarse segmentation is block-based. Since real region boundaries rarely follow block boundaries, segmentation map must be refined. As shown in Fig. 3, based on motion consistency along the coarsely segmented regions, we identify the blocks that likely contain region boundaries, and apply a region growing procedure to obtain pixel-wise boundaries using features such as edges and color.

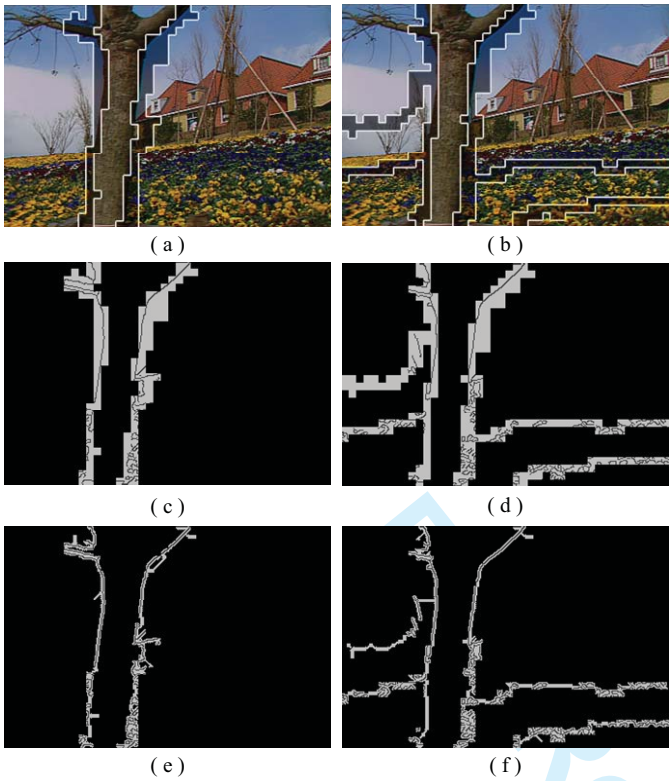


Figure 4: [Top]: Coarse segmentation and identified boundary blocks, [Middle]: Initial boundary regions and edges within them, [Bottom]: Result of interior region growing. Segmentation with GMC is employed on the left, while segmentation without GMC is employed on the right.

The boundary refinement process consists of three steps: boundary block identification (Fig. 4 (a)-(b)), edge detection (Fig. 4 (c)-(d)), and interior region growing (Fig. 4 (e)-(f)). Boundary blocks are identified in the segmentation map from Section III-B using the Region Motion Deviation (RMD) map. The RMD value I_j^C of $\mathbf{MV}_j$ within region C is the normalized deviation of $\mathbf{MV}_j$ from the centroid MV of region C:

$$I_j^C = 255 \times \left(D_j^C / D_{\max}^C \right), \tag{13}$$

where

$$D_j^C = \left\| \mathbf{MV}_{cent}^C - \mathbf{MV}_j \right\|, \; D_{\max}^C = \max_j D_j^C . \tag{14}$$

A two-pass procedure is employed to classify a block as either a *boundary block* or *interior block*. In the first pass, we scan all the blocks in the raster scan order, and for each block we check its East (E), South (S), and South-East (SE) neighboring blocks, if available. If any of these blocks belong to a different region than the one the current block belongs to, we compare the RMD values of all four blocks (current, E, S, SE), and label the block with the highest RMD value as a boundary block. In the second pass, we seek to extend the boundary to be at least 2 blocks (16 pixels) wide, to improve the chance that the real region boundaries lie within boundary blocks. To do this, we check 4-adjacency neighbors of all boundary blocks found so far, and check if they have at least one horizontal (vertical) neighbor classified as a boundary block. If not, we label the horizontal (vertical) neighbor with the higher RMD value as a boundary block. At the end, all blocks not classified as boundary blocks are labeled as interior blocks.

Canny edge detector on the Y-component is used to identify edges within boundary blocks as shown in Fig. 4(c). Then, interior regions are grown towards each other via morphological erosion of the boundary blocks using a 3×3 structuring element. The structuring element is not allowed to cross an edge. Hence, this restricted erosion will move the interior region boundaries up to the nearest edge(s). In this process, some boundaries of neighboring interior regions may meet, in which case the pixel-wise boundary between these regions is identified. In other cases, boundaries do not meet due to a complicated edge pattern between them, so we further employ region growing based on color, as in [11], to finalize region boundaries.

In Fig. 4, we illustrate the refinement procedure on frame #22 from *Flower Garden*. Figs. 4 (a)-(b) show boundary block identification, Figs. 4 (c)-(d) show detected edges in boundary regions, and Figs. 4 (e)-(f) show interior region growing. In this example, the comparison is made between the segmentation produced by the proposed algorithm that incorporates GMC (Fig. 4(a), (c), (e)), and the one produced by our previous work [15] (Fig. 4(b), (d), (f)), which does not include GMC. In both cases, the tree trunk is well-segmented, but the method from [15] ends up with a higher number of regions in the background due to its lack of GMC.

IV. Results and Discussion

The proposed segmentation algorithm has been tested on several standard YUV 4:2:0 sequences at CIF (352×288) and SIF (352×240) resolution, all with a frame rate of 30 frames per second. We employed the XviD MPEG-4 codec (http://www.xvid.org/) for compression, using the IPPP… GOP structure, at 512 kbps. We point out that the segmentation framework is generic and easily adapted to other video compression standards. The MVs extracted from the bitstream are normalized to form a uniformly sampled MV field, where each MV corresponds to an 8×8 block.

A. LSS-ME tuning

One of the key parameters in LSS-ME [17] is the tuning constant C, which is used to reject outlier MVs by assigning the weight w, computed as:

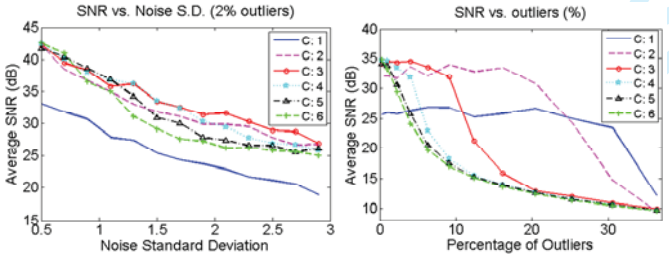
$$w(\varepsilon) = \begin{cases} \left(1 - \left(\varepsilon / C \cdot \mu\right)\right)^2, & \varepsilon < C \cdot \mu, \\ 0, & \varepsilon > C \cdot \mu, \end{cases} \tag{15}$$

where ε is the L1 error between the estimated $\mathbf{MV}(x, y; \mathbf{m})$ and the observed $\mathbf{MV}(x, y)$, and μ is the average L1 error over all MV's in the field..

The performance of LSS-ME depends on the value of C, and we investigate its effect on the accuracy of GME using synthetic noisy MV fields. We identify two types of MV noise: (1) outliers, such as moving objects, and (2) MV estimation noise caused by imperfect motion estimation in the encoder. We synthesize four MV fields using four sets of global motion parameters in [16], and then corrupt them by varying amounts

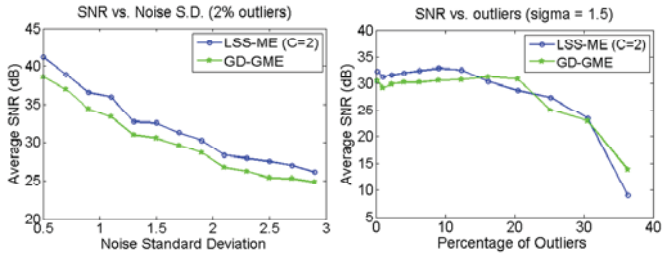
of both types of noise. The estimation performance criterion is the signal-to-noise ratio (SNR) between the MV field generated by the estimated parameters $\hat{\mathbf{m}}$ and the ground truth MV field, as in [16].

Fig. 5 shows the effect of C on LSS-ME accuracy. Fig. 5(a) demonstrates the GME performance on MV field with 2% MV outliers while the noise standard deviation σ varies from 0.5 to 3. We see that LSS-ME performance with $2 \le C \le 4$ is, on average, better than with C outside of this range. Fig. 5(b) illustrates the GME performance on MV field corrupted by noise with $\sigma = 1.5$ while the outlier percentage varies from 0% to 40%. We observe that with $C = 2$, LSS-ME maintains a relatively consistent performance up to outlier percentage of 20%. These experiments indicate that the optimal value of C depends on the amount of noise and outliers in the MV field. However, these statistics are generally not available in the compressed stream. We therefore set $C = 2$ in all our subsequent experiments, because this value seems to work well for different amounts of noise and outliers. With $C = 2$, we compare the performance of LSS-ME to gradient-descent GME (GD-GME) [16] using the same four sets of global motion parameters. The results are shown in Fig. 6, where we see that LSS-ME achieves about 1.5 - 2 dB SNR improvement over GD-GME with 2% outliers in Fig. 6(a), and slight SNR gain with MV noise ($\sigma = 1.5$), as shown in Fig. 6(b).



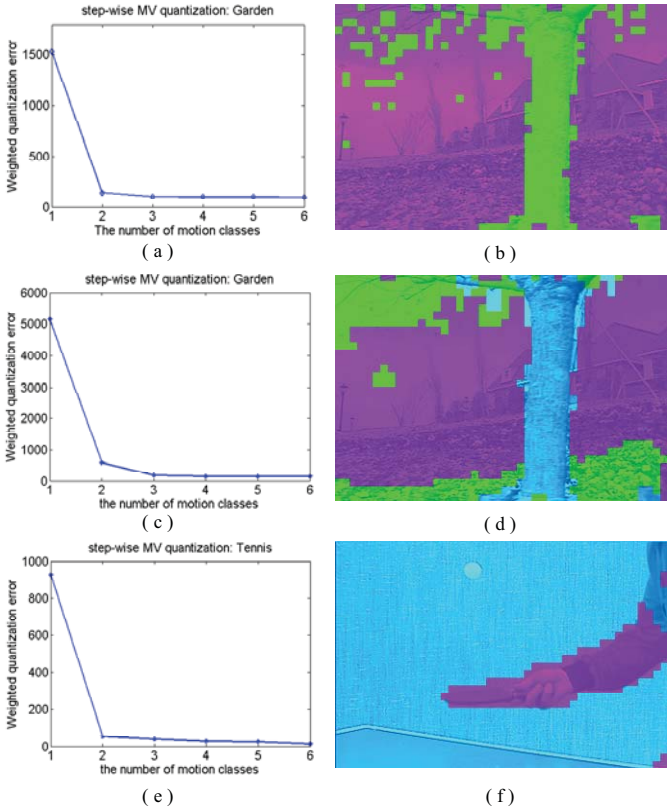
(a) (b)

Figure 5: (a): LSS-ME performance with MV field corrupted by 2% outliers and noise with different standard deviations, (b): LSS-ME performance with MV field corrupted by noise ($\sigma = 1.0$) and various outlier percentages.



(a) (b)

Figure 6: (a): LSS-ME (C=2) vs. GD-GME, MV field corrupted by 2% outliers and noise with different standard deviations, (b): LSS-ME vs. GD-GME, MV field corrupted by noise ($\sigma = 1.0$) and various outlier percentages.



(a) (b) (c) (d) (e) (f)

Figure 7: [Left]: Weighted quantization error vs. the number of motion classes, [right]: the corresponding segmentation map after MV quantization, where segments are distinguished by different colors. From top to bottom: *Flower Garden* with GMC, *Flower Garden* without GMC, and *Table Tennis*.

B. *Estimation of the number of MRF classes*

Next, we evaluate MV quantization as a way to determine the number of MRF classes and provide the initial segmentation map. Evaluation results on two sequences are presented in Fig. 7, *Flower Garden* (frame #2) with a moving camera, and initial portion of *Table Tennis* (frame #4) with a relatively static camera.

We demonstrate two approaches on *Flower Garden*: the proposed MV quantization after GMC, shown in the top row (Fig. 7 (a)-(b)), and direct MV quantization [15] without GMC, shown in the middle row (Fig. 7 (c)-(d)). The sub-figures on the left show how the weighted quantization distortion changes as a function of the number of clusters (classes). The knee of this curve indicates that three classes seem to be appropriate for the frame #2 of *Flower Garden* without GMC (Fig. 7(c)), while two classes are appropriate if GMC is performed prior to quantization (Fig. 7(a)). This makes sense, since the tree in this sequence is much closer to the camera than the other objects (garden and houses), and appears to be the only foreground object in the scene. GMC seems to be a critical factor to mitigate background over-segmentation when the sequence contains camera motion. The corresponding initial coarse segmentation maps are shown in Figs. 7(b) and (d). In Fig. 7(b), the tree trunk is the main segmented object; also, some of blocks in the branches are associated with the tree trunk. Meanwhile, in Fig. 7(d), the

background is separated into multiple regions due to the lack of GMC. While the difference between GMC and non-GMC quantization is obvious on a sequence like *Flower Garden*, the two approaches differ little on a sequence with relatively static camera. We show the quantization result of the proposed approach on the initial portion of *Table Tennis* at the bottom (Fig. 7 (e)-(f)), which is similar to the result presented in [15].

C. MRF motion segmentation and boundary refinement

In this subsection we evaluate MRF segmentation, especially the number of ICM iterations and the role of parameter β in (10), which influences the spatial structure of the MRF. In Fig. 8, on sequence *Flower Garden*, we show the normalized posteriori energy (the sum of potentials in (10) of all cliques in the field) vs. the number of iterations of ICM implementation from [3], when $\beta \in \{1.5, 2.5, 3.5\}$. The graph indicates that 4-6 iterations are sufficient for convergence for this range of β, as suggested in [3]. Hence, we used 6 iterations in all our experiments. Also note that as β increases, the number of iterations needed for convergence is reduced.

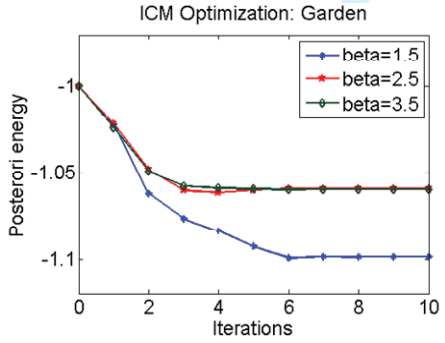


Figure 8: Posteriori energy vs. iterations in MRF motion classification

In Fig. 9, we show the segmentation of frame #2 of *Flower Garden* obtained after 6 iterations, where β is set to 0, 1.5, and 3.5, from top to bottom, respectively. The segmentation with GMC (left side of the figure) and without GMC as in [15] (right side) are both shown for comparison. When $\beta = 0$, no spatial constraints are imposed on the MRF, so the segmentation does not change from its initial layout obtained by MV quantization (Fig. 7(b) and (d)). As β increases, neighboring blocks are more likely to be in the same region, so region boundaries end up being more compact. Our experiments indicate that $\beta = 3.5$ provides a good balance between boundary compactness and segmentation accuracy, so we use this value in the remaining experiments.

In Fig. 10(a), we illustrate the final MRF segmentation of frame #2 of *Flower Garden* (after merging blocks from small regions to neighboring regions), and in Fig. 10(b) we show the boundary refinement results. The corresponding results (MRF segmentation and boundary refinement) from [15] are shown in Fig. 10 (c) and (d) for comparison. As we can see, the proposed segmentation correctly separates the foreground from the background, while the algorithm from [15], which does not include GMC, over-segments the background into several regions due to global motion bias.

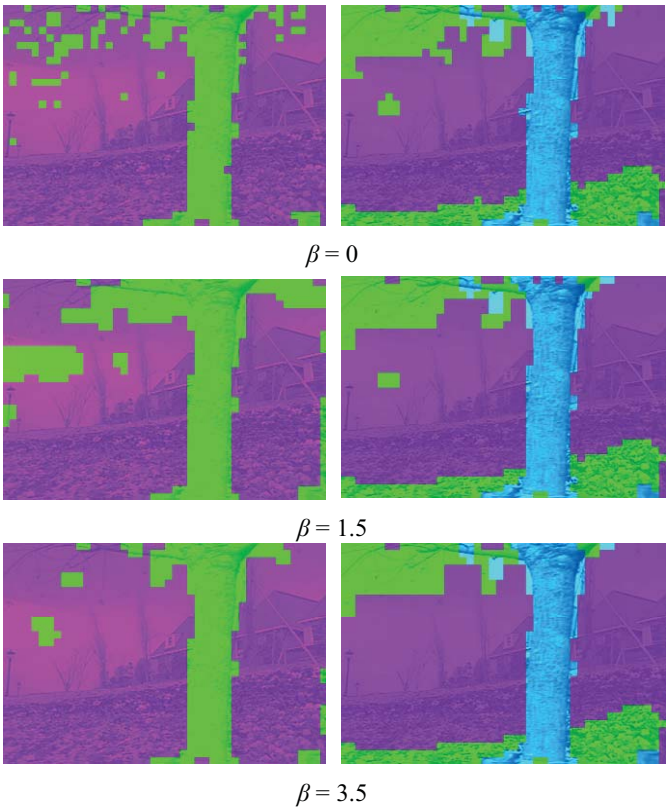


Figure 9: [Left]: MRF segmentation with proposed coarse segmentation incorporating GMC, [Right]: MRF segmentation from [15]. Top row to bottom row, the MRF segmentation with parameter β set to {0, 1.5, 3.5}.

We also show the results from four other state-of-the-art segmentation algorithms: [2], [4], [7], and [11], for comparison. Fig. 10(e) shows the segmentation result using the algorithm from [2], which is image-based, and does not use motion information, thus resulting in over-segmentation. This problem has been mitigated to some extent by the method proposed in [11], shown in Fig. 10(f), which utilizes *k*-means clustering and motion consistency. However, the scene is still over-segmented. Fig. 10(g) shows the result of using the method from [4], which is based on two-class MRF (background and foreground) without GMC. The main problem here is the accuracy, since part of the background (garden) is included in the same segment as the foreground (tree trunk). Finally, Fig. 10(h) shows the segmentation result from [7], which is a MV-based approach using the Expectation Maximization algorithm on a dense MV field. This method ends up with the same number of moving regions as [15], and segments the background scene into multiple regions due to its lack of GMC. Also, the segmented moving regions are less compact than in our case, and some are not even spatially connected.

Finally, in Fig. 11 we demonstrate the final boundary-refined segmentation results of our method for several other sequences: *Table Tennis*, *Stefan*, *Coastguard*, and *Hall Monitor*, where both *Stefan* and *Coastguard* involve a moving camera, while *Hall Monitor* and the initial portion of *Table Tennis* have a static camera.

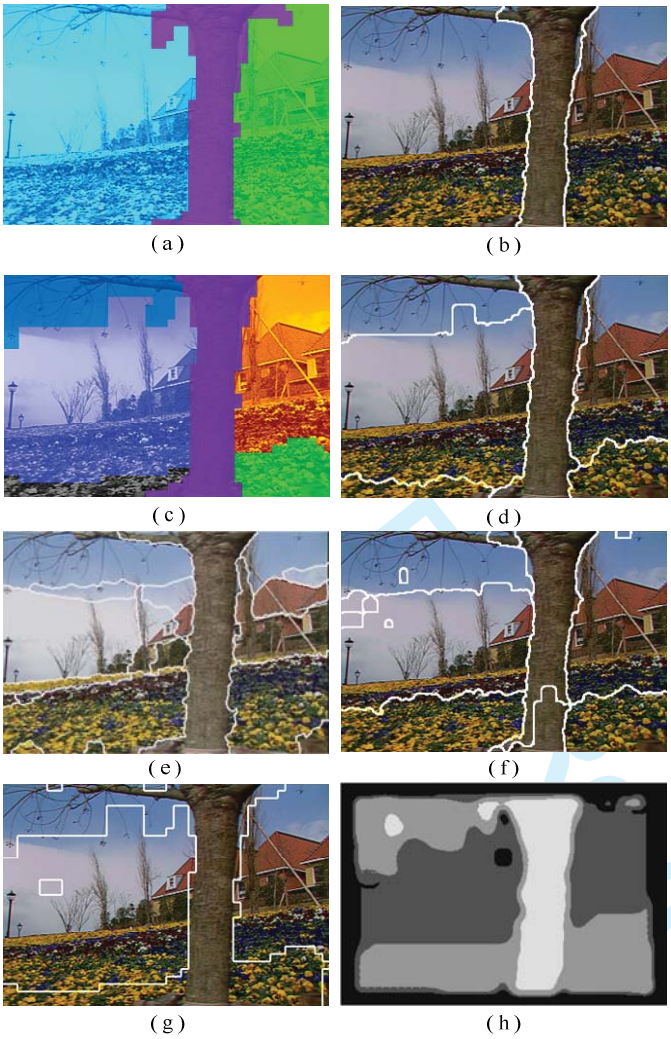


Figure 10: (a): Coarse MRF segmentation, (b):boundary refinement., (c) Coarse MRF segmentation from [3], (d) boundary refinement from [3], (e, f, g, h): segmentation result from Ref. [2], [11], [4] and [7], respectively.

D. Quantitative evaluation

In addition to the above visual results, we provide a quantitative evaluation of our method, using the manually segmented sequences *Stefan* CIF and *Table Tennis* SIF (available at: http://www.sfu.ca/~ibajic/datasets.html). We test how accurately the foreground moving regions (player's hand and ball in *Table Tennis,* tennis player in *Stefan*) can be segmented. By counting the pixels correctly identified as moving region pixels (True Positives −TP), the pixels correctly identified as the background (True Negatives – TN), the pixels wrongly identified as moving region pixels (False Positives − FP), and the pixels wrongly identified as background (False Negatives – FN), we can compute several quantities for assessing segmentation accuracy, such as Precision, Recall, and F-measure as the harmonic mean of Precision and Recall [15]. In terms of these quantities, we compare our proposed method to the method from [4], which is a recent work addressing MRF motion segmentation in block-based compressed video. In our implementation, an 8×8 uniformly sampled MV field is utilized.

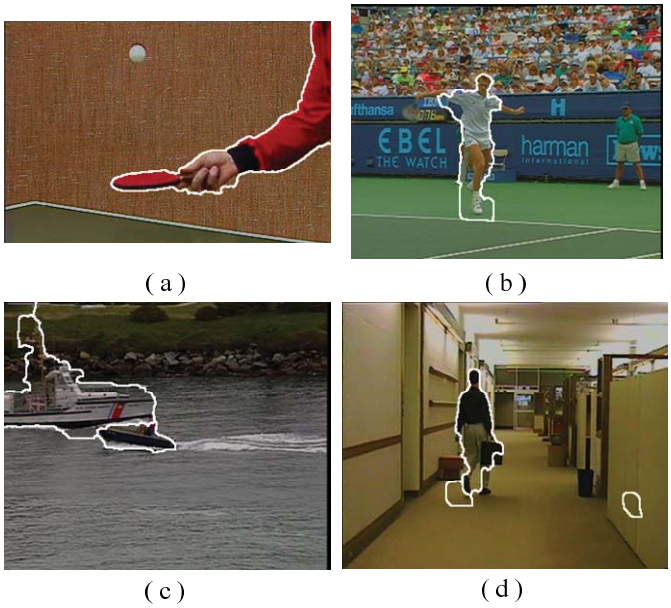


Figure 11: Final segmentation results, (a): Sequence *Table Tennis* (frame # 5), (b): *Stefan* (frame #24), (c): *Coastguard* (frame # 40), (d): *Hall Monitor* (frame # 50).

The top and middle rows in Figs. 12-13 show the segmented objects in *Table Tennis* and *Stefan*, extracted by our method and the one from [4]. TP, TN, FP, and FN pixels are shown in different colors. The last row in both figures shows the quantitative measures for the initial portion of the two sequences, while their averages are listed in Table I. From these figures, we see that the proposed method performs much better in segmenting moving regions than the method from [4]. This is especially true in *Stefan* where, due to the global motion, a large portion of background is inappropriately classified as part of the foreground by the method from [4]. For *Table Tennis*, whose initial portion involves a static camera, the two methods perform similarly. Nonetheless, our boundary refinement yields more accurate boundaries, which again leads to higher precision (0.91 vs. 0.79).

Note that the precision curve of our method in *Stefan* appears lower in first 15 frames, and then rises from frame 16 onwards. The reason is that the player does not move much in the first 15 frames, so he gets classified as the background. However, once his distinct motion starts at frame 16, our segmentation approach picks it up quite easily and separates it rather well from the background. By contrast, the method from [4] erroneously includes large portions of background into its estimate of foreground (shown as blue pixels in Fig. 12), and yields very low precision on this sequence. Another observation from Fig. 12 is that the region boundary (tennis player) is not well localized in comparison to the manually segmented ground truth, because the moving region represents a rather small and flexible object. Hence, coarse segmentation from block-based MVs may miss certain small parts of this object (e.g., head or arms) that do not completely fill an 8×8 block. This problem is common to all block-based coarse segmentation methods.

Finally, note that our segmentation method has a reasonably

low complexity. On a standard desktop PC with Intel Pentium CPU at 3.0 GHz, with 2 GB of RAM, on a CIF sequence, motion segmentation (in MATLAB) takes on average about 105 ms per frame, and boundary refinement (in C/C++) takes about 20 ms.

V. Conclusions

In this paper, we have presented an unsupervised moving region segmentation algorithm for compressed video. The framework consists of camera motion removal through global motion compensation, followed by MRF-based coarse segmentation and boundary refinement using color and edge information. The proposed method delivers a good balance between accuracy and complexity, and compares favorably against other state-of-the-art segmentation methods.

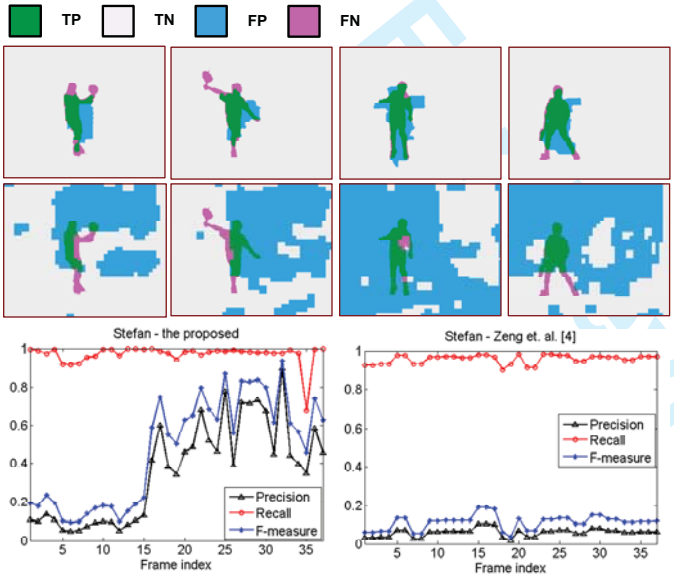


Figure 12: Quantitative evaluation – *Stefan.* [Top]: the proposed method, frame #17, #26, #31, #37, from left to right, [Middle]: corresponding segmentation using method from [4], [Bottom]: the quantitative evaluation for the proposed method (left) and method from [4] (right).

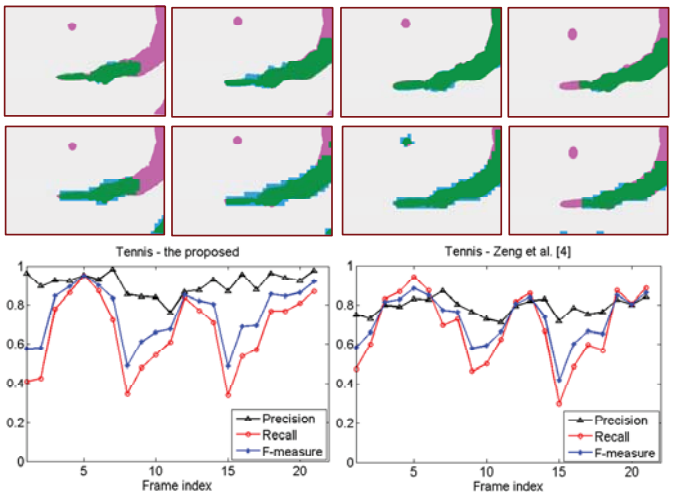


Figure 13: Quantitative evaluation – *Table Tennis.* [Top]: the proposed method, frame #1, #3, #5, #7, from left to right, [Middle]: corresponding segmentation using the method from [4], [Bottom]: the quantitative evaluation for the proposed method (left) and the method from [4] (right).

TABLE I
AVERAGE PRECISION, RECALL, AND F-MEASURE.

Sequence	*Table Tennis*		*Stefan*	
	Proposed	**Ref. [4]**	**Proposed**	**Ref. [4]**
Precision	0.91	0.79	0.39	0.06
Recall	0.67	0.69	0.97	0.96
F-measure	0.75	0.72	0.49	0.11

References

[1] Z. Kato, "Segmentation of color images via reversible jump MCMC sampling," *Image and Vision Computing*, vol. 26, issue 3, pp. 361-371, Mar. 2008.

[2] Y. Deng, and B. S. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE Trans. on Pattern Anal. Mach. Intell.*, vol. 23, issue 8, pp. 800-810, Aug. 2001.

[3] Z. Kato, T. C. Pong, and J. C. M. Lee. "Color Image Segmentation and Parameter Estimation in a Markovian Framework," *Pattern Recognition Letters*, 22(3-4):309--321, March 2001.

[4] 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, pp. 290-299, Jun. 2005.

[5] M. Ritch and N. Canagarajah, "Motion-based video object tracking in the compressed domain," *Proc. IEEE ICIP'07,* vol. 6, pp. VI-301-VI-304, Oct. 2007.

[6] J. Wang and E. Adelson, "Representing Moving Images with Layers," *IEEE Trans. Image Processing*, vol. 3, pp. 625-638, Sept. 1994.

[7] R. V. Babu, K. R. Ramakrishnan, and S. H. Srinivasan, "Video object segmentation: a compressed domain approach," *IEEE Trans. Ciruits Syst. Video Technol*. vol. 14, no. 4 , pp. 462-474, Apr. 2004.

[8] N. Vasconcelos and A. Lippman, "Empirical Bayesian motion segmentation," *IEEE Trans. on Pattern Anal. Mach. Intell*., vol. 2, issue 2, pp. 217-221, Feb. 2001.

[9] D. Zhong and S. F. Chang, "An integrated approach for content-based video object segmentation and retrieval," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 9, no. 8, pp. 1259-1268, Dec. 1999.

[10] X. Shi, Z. Zhang, L. Shen, "Multiresolution segmentation of video objects in the compression domain," *Optical Engineering.,* vol. 46, no. 9, 097401, Sep. 2007.

[11] Y.-M. Chen, I. V. Bajić, and P. Saeedi, "Coarse-to-fine moving region segmentation in compressed video," *Proc. IEEE WIAMIS'09*, pp. 45-48, London, UK, May 2009.

[12] A. Dante and M. Brookes, "Precise real-time outlier removal from motion vector fields for 3D reconstruction," *Proc. IEEE ICIP'03*, pp. 393-396, Sep. 2003.

[13] J. Besag, "On the statistic analysis of dirty pictures," *J. Roy. Statist. Soc. B*. vol. 48, no. 3, pp. 259-302, 1986.

[14] R. Ewerth, M. Schwalb, P. Tessmann, and B. Freisleben, "Segmenting moving objects in MPEG videos in the presents of camera motion," Proc. ICIAP'07, pp. 92-96, Sept. 2007

[15] Y.-M. Chen, I. V. Bajić, and P. Saeedi, "Motion segmentation in compressed video using Markov random fields," *Proc. IEEE ICME'10*, pp. 760-765, Singapore, July 2010

[16] Y. Su, M.-T. Sun, and V. Hsu, "Global motion estimation from coarsely sampled motion vector field and the applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 232-242, Feb. 2005.

[17] A. Smolic, M. Hoeynck, and J.-R. Ohm, "Low-complexity global motion estimation from P-frame motion vectors for MPEG-7 application," *Proc. IEEE ICIP'00*, pp. 271-274, Sep. 2000.

[18] M. Haller, A. Krutz, and T. Sikora, "Evaluation of pixel- and motion vector-based global motion estimation for camera motion characterization," *Proc. WIAMIS'09*, pp. 49-52, May 2009.

[19] M. Fischler and R. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". *Comm. ACM*, vol. 24, no.6, pp. 381-395, Jun. 1981.