OBJECT-ORIENTED VIDEO SEGMENTATION BASED ON MOTION COHERENCE IN A VIDEO SEQUENCE

First Author, A.kiruthika devi, Second Author R.krishnaveni
1, PG student, Electronics and communication department
1, J.J College of Engineering and Technology, Trichirappali, India
1, Email:kiruthika.ee09@gmail.com

Abstract — This work describes an approach for object-oriented video segmentation based on motion coherence in a video sequence. Using a tracking process based on sampled point’s 2-D motion patterns are identified with an ensemble clustering approach. Particles are clustered to obtain a pixel-wise segmentation in space and time domains. The segmentation result is mapped to an image spatio-temporal feature space. Thus, the different constituent parts of the scene that move coherently along the video sequence are mapped to volumes in this spatio-temporal space. These volumes make the redundancy in the temporal sense more explicit, leading to potential gains in video coding applications. In order to illustrate the potential advantages of using the proposed motion segmentation approach in video coding applications, the PSNR of the temporal predictions and the entropies of prediction errors obtained.

Index terms — Ensemble clustering, motion segmentation, object-based video segmentation, point tracking.

I. INTRODUCTION

Motion segmentation is an important preprocessing step in many computer vision and video processing tasks, these applications motivated the development of several 2-D Motion segmentation techniques where each frame of a video sequence is split into regions that move coherently. By 2-D motion here we denote a motion of objects in a 3-D scene projected in the image. However 2-D motion segmentation leads to video over segmentation. The over segmented in several 2-D motion regions due to several reasons, such as depth discontinuities occlusions, and the perspective projection effect. Segmentation process can be approached by considering the objects as

Moving in 3-D space(x, y, z) and this approach has motivated several Works in 3-D motion segmentation [1]–[6]. Spatio-temporal motion segmentation [7]–[10] is a different approach, where different moving objects are segmented in volumes (called tunnels [10]) in the domain formed by the spatial dimensions (e.g. x, y) and the temporal dimension t, and these volumes are delimited by object motion boundaries (i.e., motion discontinuities). The definition of object in a video segmentation framework is related to the concept of Region homogeneity and different applications require different region homogeneity criteria. In video coding, Segmentation is frequently used to explore the data redundancy in time [11]. Thus, even if the object region moves along the temporal sequence, the region representation remains the same, i.e., redundant, within the object motion boundaries. Motion estimation and motion segmentation methods can be divided into two methods. Direct methods [5] and Feature based methods [6]. Direct methods recover the unknown parameters directly from measurable image quantities at each pixel in the image. Feature-based methods minimize an error measure that is based on distances between pixels in image.

Direct methods

Direct methods recover the unknown parameters directly from measurable image quantities at each pixel in the image, solving two problems simultaneously: 1) the motion of the camera and/or objects of the scene. 2) the correspondence of every pixel. It is important to observe that with direct methods the pixel correspondence/classification is performed directly with the measurable image quantities at each pixel. An important property of the direct methods is that they can successfully estimate global motion even in the presence of multiple motions and/or outlier.

Feature-based method

Feature-based methods for motion segmentation usually consist of two independent stages: 1) Feature selection and/or correspondence and 2) motion parameter estimation. The second stage often is
performed through factorization methods [2], although some simpler clustering strategy can be used. In factorization methods, motion and shape information are treated separately by applying constraints to the scene projection on the image formation plane, as well as on the object shape and motion.

The proposed approach generates a simple scene representation, adequate for object video coding, and also delivers a more redundant and temporally persistent partition of the scene than direct video segmentation methods and motion prediction strategies. Experimental results for synthetic and natural video sequences are used to illustrate the properties of the proposed method, and to show its potential in video coding applications.

II. METHOD OVERVIEW

The structure of the proposed coherent motion segmentation approach can be divided into three main parts:
1) Estimation of Particle Trajectories
2) Segmentation of Particle Trajectories
3) Dense Segmentation Extraction

The first part concerns the selection and tracking of a set of points of the scene (namely, particles). This stage takes as input the original video frames, and returns as output a set of particles and their respective trajectories. During the estimation of particle trajectories, the particles whose correspondent point locations in the scene suffer occlusion are eliminated, and new particles are created in regions that become newly visible along the video sequence.

The second part deals with the segmentation of particle trajectories, so that particles moving coherently are grouped together. This stage takes as input the particles trajectories computed in the first stage, and returns labels for all the particles as outputs, representing the motion segmentation of frame regions according to the particle trajectories. The segmentation of particle trajectories can be divided in four steps:
- Clustering of 2-frame motion vectors,
- Ensemble clustering of particles
- Meta-clustering validation
- Spatial filtering

Clustering of 2-frame motion vectors: In this step, clustering’s of particles are performed with displacement motion vectors taken from pairs of frames. Only neighboring frames are considered. For each pair of frames, the input to this step is the position of particles in each frame, and the output is a set of particle clustering’s and their labels, valid for each pair of frames considered.

Ensemble clustering of particles: Here, all the clustering’s computed in the previous step are processed simultaneously to produce a unique division of the full set of particles in sub-sets of particles in motion, called meta-clusters; several sets of clustering labels are taken as input to this step, and a unique set of segmentation labels are returned as output.

Meta-clustering validation: In this step, particles that were segmented in the previous step are compared to meta-cluster prototypes in terms of motion to detect incorrectly labeled particles and, when this occurs, particles are re-labeled. A set of segmentation labels is taken as input, and a corrected set of segmentation labels is returned as output.

Spatial filtering: In this step, outliers are eliminated and groups of adjacent particles that are not significant. This step takes as input a set of particle labels, and returns as output a filtered set of particle labels.

III. ESTIMATION OF PARTICLE TRAJECTORIES

In this section, we show how the particles are selected in each video frame, and how they are located in the video frames of a video sequence. Some important properties of this trajectory estimation approach are outlined here, first, the estimation of particle trajectories this means robust to abrupt camera motion, and objects motion discontinuities. Second, the particle sampling density is adaptive, in the sense that regions with more details are sampled with more particles, while homogeneous regions are sampled with fewer particles. Third, motion information can be inferred from neighboring particles, reducing the effect of the aperture problem in homogeneous regions. The method for estimating particle trajectories used in this work. The next section shows the particle pruning method employed in our work. This method introduces a modification in the method so the number of false occlusions can be reduced.

Particle pruning: In the work proposes the same objective function that is minimized to optimize the location of particles is employed as a measure of
particle location reliability in the particle pruning stage. This objective function is composed by three terms as namely Ep, Ef, and Ed as described below. Grouping these three terms, we have the complete objective function

\[ E(p, t) = \sum_p E_p(p, t) + \alpha_E E_f(p, t) + \alpha_D E_d(p, q, t) \]

The first term ‘Ep’ represents the projection error of the particle in the frame at time with respect to the particle first appearance in a frame of the sequence. The second term ‘Ef’ represents the particle in the difference between the actual displacement frame at time. The third term ‘Ed’ measures the relative motion between linked particles and in the frame at time.

Where \( \alpha_E \) and \( \alpha_D \) are constants that provide a compromise among the terms, and were set to 5 and 10 in the particle location optimization step. The performance of the trajectories estimation is not very sensitive to changes in these constants. So, the values fixed here for \( \alpha_E \) and \( \alpha_D \) may work for a large variety of sequences.

IV. SEGMENTATION PARTICLE TRAJECTORIES

This describes the approach adopted in this work for grouping particles in coherent motion. A clustering algorithm is applied to each set of particles that coexist in the video sequence (i.e., have lifetime intersection), and then these particle sub-set clustering’s combined in larger particle clusters. Thus, motion analysis can be performed by combining information of particle subsets of data sub-sets is similar to the combination of data partitions in ensemble clustering. The majority of ensemble clustering methods require the computation of pair wise similarities for all objects in the collection.

A. Ensemble Clustering of Particles

In order to group particles that are in coherent motion along the video sequence, we first identify the subsets of particles that present similar motion in neighboring frames. Let \( P = P_1, P_2, \ldots, P_n_p \) be the whole set of \( n_p \) particles in the video sequence, \( E_p(t+1) \) be the displacement vector of particle \( p \) between frames at time \( t, t+1 \).

For each frame at time \( t \), three clustering’s are computed with the particles at present state having \( n_p \) particles.

The mean-shift method is employed in this work to obtain the clusters \( H^{(j)} \) in \( H^{(j)} \) because of the ability of the mean shift method to identify clusters with arbitrary shapes in feature space. The bandwidth of the mean-shift kernel was set to 3.1, for all the frames \( (t, t+1) \) and \( l=1,2,3 \). The bandwidth increases with in order to reduce clustering diversity and, consequently, to produce meaningful cluster correspondence, because low diversity within clusters is required by the ensemble clustering algorithm employed in this work. The bandwidth of the mean-shift kernel is related to the expected range of relative motions between objects (i.e., particle clusters). The bandwidth can be tuned according to the kind of object motion (slow, fast) found in the sequence.

We have three clustering’s \( \{ H^{(j)} \} \) for each frame. Only one data partition is obtained for the whole set of particles in the video sequence, by combining all clustering’s (three for each frame) in a single clustering \( H \). To perform this task, we use the meta-clustering algorithm (MCLA) [14].

Each particle \( P_i = (i, 1, \ldots, n_p) \) is assigned to one of the clusters \( h^{(j)}_i \) in each clustering \( H^{(j)} \) based on its motion patterns in different time spans \((l=1,2,3)\). Therefore, each cluster \( h^{(j)}_i \) in clustering \( H^{(j)} \) can be represented by a binary label vector, where a particle
Pi is labeled as “1” if it belongs to , or “0” otherwise (see Table 1).

<table>
<thead>
<tr>
<th></th>
<th>$H^{(1)}$</th>
<th>$H^{(2)}$</th>
<th>$H^{(3)}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>1 0 0</td>
<td>1 0 0</td>
<td>1 0</td>
</tr>
<tr>
<td>P2</td>
<td>1 0 0</td>
<td>0 1 0</td>
<td>0 0</td>
</tr>
<tr>
<td>P3</td>
<td>0 1 0</td>
<td>0 1 0</td>
<td>0 1</td>
</tr>
<tr>
<td>P4</td>
<td>0 1 0</td>
<td>0 1 0</td>
<td>0 1</td>
</tr>
<tr>
<td>P5</td>
<td>0 0 1</td>
<td>0 0 1</td>
<td>0 0</td>
</tr>
<tr>
<td>P6</td>
<td>0 0 1</td>
<td>0 0 0</td>
<td>0 0</td>
</tr>
</tbody>
</table>

Table: 1 Hyper- graph with 8 hyper-edges

Initially, according to the MCLA approach, the $n_h$ clusters $h_i^{()}$, represented by binary label vectors, are transformed in hyper-graph. A hyper-graph is constituted by nodes and hyper-edges, and it is a generalization of a graph in the sense that a hyper-edge can connect any set of nodes. In our hyper-graph representation, nodes represent particles and the hyper-edges represent clusters of particles that jointly occur in 1, 2, and 3 consecutive frames.

The Table represents the particles ($p=1, 2...6$), the clustering’s are represented by the binary matrices $H^{(1, 2, 3)}$, and the individual clusters are represented by the hyper-edges $h_1, 2, 3^{(0)}$, $h_1, 2, 3^{(0)}$. And. Note that, since we have three frames in this example, three clustering’s are formed using: 1) motion vectors between frames 1 and 2 ($H^{(0)}$); 2) motion vectors between frames 2 and 3 ($H^{(0)}$). Since a particle can belong to only one cluster $h_i^{(0)}$ in each clustering $H^{(0)}$ the lines in the corresponding binary matrix sum up to “1” when the particle cluster is known. Corresponding to particles that were not assigned to any cluster sum up to “0”.

The ideal number K of meta-clusters is selected as the number of clusters that is more stable in the hierarchical clustering dendrogram. We observe how the number of clusters vary as the threshold values increases from “0” to “1”, when the dendrogram is built; K is the number of detected clusters in the longest range of consecutive threshold increases when the dendrogram is built, for which the number of clusters $K=3$ does not change. This is illustrated in Fig. 1, where the range in the dendrogram defines the ideal number of meta-clusters (since this is the longest sequence of threshold value changes). After the meta-clusters is identified, all the hyper-edges belonging to the same meta-cluster are grouped together in a single meta-hyper-edge.

**B. Particle Meta-Clustering Validation**

As mentioned before, particle tracking errors may occur, resulting in incorrect particle-to-meta-cluster assignments. To detect these inconsistencies after the ensemble clustering stage cluster validation step is performed by analyzing trajectories particles grouped together (same meta-cluster). This is particularly important when particles are assigned to one object but migrate to another object during the tracking process, because of occlusion detection imperfections.

Let $p=(xp(t),yp(t))$ be the $p$th particle, represented by its spatial coordinates in frame at time $t$. The context of this particle is defined by a window of size $2w_v+1$ given by $(xp(t)+Δv_x,yp(t)+Δv_y)$ where $-W_v < Δv_x, Δv_y < W_v$ $Δv_x, Δv_y ≪ ε$

In all our experiments, a context of size was use $W=2$ used. The projection error of a point in a window at $[-W_v, W_v]$ sub-pixel level, that specifies the context of particle, given the motion of the particle between frame at time $t$ and frame at time $t-ρ_v$ can be computed. In order to detect context changes, We compute the mean of the $w$-best matches (i.e., the mean of the $w$ smallest motion estimation errors). Based on experiments, we verified that $w=15$ and $ρ_{v=3}$ offer a good compromise between false positives and false negatives. The choice for $ρ_v$ is strongly related to the bandwidth of the mean-shift kernel. The slower the motion is, the larger is $ρ_v$ (and smaller the bandwidth kernel is); however, the faster is the motion, the smaller is $ρ_v$ (and larger the bandwidth kernel is).
The association between adjacent particles is represented by assigning binary weights to the edges that is, an edge receives “1” if it connects two particles belonging to the same meta-cluster, or it receives “0” if connects particles belonging to different meta-clusters.

V. DENSE SEGMENTATION EXTRACTION

In many computer vision and image processing tasks, and in many video coding problems, it is necessary to extract a dense representation of motion segmentation. It means that we must determine which pixels are assigned to each moving object.

Let \( P=p_1, p_2, \ldots, p_{n_p} \) be the whole set of particles of \( n_p \) The video and \( \lambda_P \in \{1, \ldots, k\} \) the set of labels of indicates for each particle ,to which meta-cluster \( b=\{1, \ldots, k\} \) the Particle is associated.so each particle \( p \) n a frame at time t. represented by its spatial coordinates \( (x^p, y^p, t^1) \) and \( (x^p, y^p, t^2) \) represent the displacement of particle in frame at time , and \( \Sigma \) is the covariance matrix given by

\[
\Sigma = \begin{bmatrix}
\sigma_s & 0 & 0 & 0 \\
0 & \sigma_s & 0 & 0 \\
0 & 0 & \sigma_m & 0 \\
0 & 0 & 0 & \sigma_m
\end{bmatrix}
\]

where \( \sigma_s \) and \( \sigma_m \) the spatial and motion standard deviations. These coefficients are chosen based on a compromise between precision and smoothness of the objects boundaries, and can be modified according to the application. The smaller the value of \( \sigma \) the more precise the boundaries will be. The larger the value \( \sigma_m \), the smoother the boundaries will be. On the other hand, controls the weight of motion information used in pixel classification.

Gaussian kernels at the corresponding pixel position

\[
\text{N Max } \Sigma_{\{p\in P\}} G_p(t)(x,y,t), v(x,y,t), t)
\]
we use the optical flow components $u(x,y,t)$ and $v(x,y,t)$ as the motion information at pixel level, and compare $(u_p(t), v_p(t))$ them with the motion of particles using the implicit functions given by the Gaussian kernels.

VI. EXPERIMENTAL RESULTS

![ Particle segmentation for the frame 5 (first row) and frame 45 (second row) of the coastguard sequence: (a), (d) particle tracking results; (b), (e) particle meta-clustering results; and (c), (f) final particle segmentation. ]

The proposed motion segmentation methods were calculated PSNR parametric measure by compare final segmentation with the original.

VII. CONCLUSION

A method for unsupervised identification of coherent motion in adaptively sampled videos was proposed in this work. This technique provides a new way of linking low-level information in videos to high-level concepts that can be employed directly in video coding. This approach can be useful in many other image processing and computer vision tasks, including object tracking, information retrieval and video analysis. The proposed method allows us to identify temporally discontinuous motion patterns, and is robust to abrupt camera motion. The proposed particle segmentation method uses ensemble clustering to combine particle clusters obtained for adjacent frames, allowing the identification of long-range motion patterns, which we represent as spatial-temporal volumes called tunnels. The identification of long-range motion patterns is crucial to take full advantage of temporal redundancy in segmentation-based video coding. The mean-shift algorithm [15] is employed to obtain the particle clusters associated to adjacent Frames, and has the important property of identifying clusters with arbitrary shapes in feature space. PSNR values for temporal predictions along segmented tunnels, and the entropies of the prediction errors. That the proposed segmentation approach potentially can be used in video coding with advantages in terms of bit-rate, PSNR and flexibility to handle moving objects.

VIII. ACKNOWLEDGMENT

The authors thank to D. Comaniciu and P. Meer for providing the mean shift algorithm code [15].

IX. REFERENCES


