Open access • Proceedings Article • DOI:10.1109/CVPR.2013.273

Hierarchical Video Representation with Trajectory Binary Partition Tree — Source link

Guillem Palou, Philippe Salembier

Published on: 23 Jun 2013 - Computer Vision and Pattern Recognition

Topics: Block-matching algorithm, Video processing, Cut, Representation (mathematics) and Trajectory

Hierarchical Video Representation with Trajectory Binary Partition Tree

Guillem Palou and Philippe Salembier
Technical University of Catalonia
guillem.palou@upc.edu, philippe.salembier@upc.edu

Abstract

As early stage of video processing, we introduce an iterative trajectory merging algorithm that produces a region-based and hierarchical representation of the video sequence, called the Trajectory Binary Partition Tree (BPT). From this representation, many analysis and graph cut techniques can be used to extract partitions or objects that are useful in the context of specific applications.

In order to define trajectories and to create a precise merging algorithm, color and motion cues have to be used. Both types of informations are very useful to characterize objects but present strong differences of behavior in the spatial and the temporal dimensions. On the one hand, scenes and objects are rich in their spatial color distributions, but these distributions are rather stable over time. Object motion, on the other hand, presents simple structures and low spatial variability but may change from frame to frame. The proposed algorithm takes into account this key difference and relies on different models and associated metrics to deal with color and motion information. We show that the proposed algorithm outperforms existing hierarchical video segmentation algorithms and provides more stable and precise regions.

1. Introduction

With the increase of CPU power and memory capacity, early stages of video analysis can tackle today issues that were traditionally considered as very challenging. One of these issues can be seen as making an initial abstraction step from the original pixel-based representation of the video sequence. The main goal of this abstraction step is to create a representation of the original data that relies on entities that are more meaningful than individual pixels, that is structured to ease the subsequent analysis tasks and that is multiscale to support a wide range of applications. One such representation, the Binary Partition Tree (BPT) [1], is based on regions that are hierarchically structured in a tree describing inclusion relationship. Once the tree is constructed, it can be processed in many different ways through graph cut to extract several partitions or through region analysis to extract meaningful objects in the scene [2]. The tree construction can be performed by keeping track of the merging steps of a hierarchical segmentation algorithm. In this paper, we focus on the definition of an efficient motion trajectory merging algorithm to construct the BPT and compare the algorithm to state of the art video segmentation algorithms.

Most video segmentation algorithms are extensions of image segmentation techniques. For example, a hierarchical version of the image segmentation approach [3] (GB) is proposed in [4] (GBH). A mean-shift algorithm [5] is also adapted for temporal sequences in [6] (Meanshift). The approaches [7] (Nyström) and [8] (SWA) proved to be scalable in complexity when the time dimension is added. In these algorithms, the extension to video is done by treating the temporal dimension as a third spatial dimension. As a result, 3D image segmentation [9] and video segmentation essentially become equivalent. However, the temporal axis on an image sequence introduces dynamic information, whereas spatial axes only provide static cues.

Motion information present in video sequences provides a very important cue for segmentation and early processing steps. State of the art motion and optical flow estimation algorithms [10, 11] are able to produce accurate and dense fields. With GPU parallel processing capabilities, many algorithms are able to almost run in real-time[12, 13]. By using optical flow, a set of sparse points where motion is reliable can be accurately tracked to form long term trajectories [14]. The notion of motion trajectory offers an alternative approach to feature-based tracking algorithms [15]. These tracked points can then be clustered into different classes [16, 17] to detect different motions which can then be related to objects in the scene. Motion trajectories are spatially sparse because the motion information is not reliable everywhere. Techniques such as [18] have been proposed to produce a dense coverage and to precisely detect object contours in a frame.

It is also possible to combine static segmentation cues with optical flow to produce dense segmentation on videos. The work [19] uses temporal information and the gPb contour detection algorithm [20] to compute voxel-based affini-

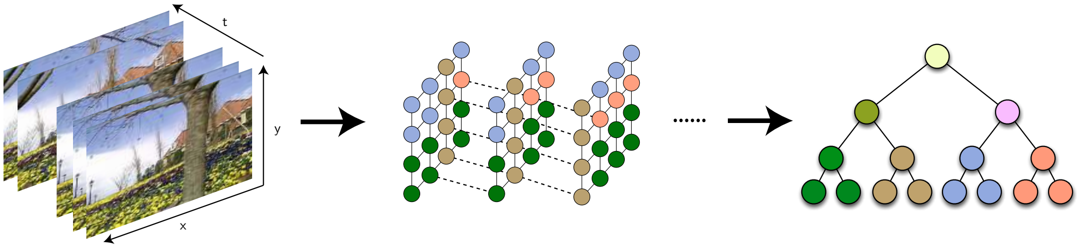

Figure 1. Outline of the proposed approach. Starting from the video frames, the first step identifies reliable trajectories between frames representing long term spatiotemporal coherent part of the scene (shown as dashed lines). Then, the algorithm constructs a Trajectory Binary Partition Tree by iteratively merging neighboring trajectories and builds a hierarchical representation of the entire sequence. Note: The node color approximately represents the mean color of the trajectory regions.

ties and relate pixels between frames. Affinities are then clustered using normalized cuts [21] and a segmentation is produced by means of ultra-metric contour maps [20] on the resulting segments. A part from its computational cost (5 minutes on a cluster of 34 GPUs), the algorithm does not take advantage of long term information introduced by trajectories. By contrast, the work [22] uses the tracked points [14] to propose a semi-supervised clustering and uses the obtained labels to produce a dense segmentation. However, the number of objects should be known in advance and, in practice, this information is most of the time unknown. There are other systems that propose contour detection and segmentation on single frames [23, 24], although they do not deal with full video sequences.

In this paper, in order to compute a BPT of a video sequence by means of a hierarchical segmentation, we discuss a completely unsupervised way to introduce long term motion information. The main difference with the original BPT approach [1] concerns the elementary units that are iteratively merged. Instead of iteratively merging neighboring pixels, here neighboring trajectories are merged forming a Trajectory BPT. The approach is outlined in Figure 1. The system assumes that dense forward and backward optical flow information is available. To run the experiments, the Large Displacement Optical Flow (LDOF) estimation technique [11] is used, but other approaches could work as well. Prior to the Trajectory BPT computation, reliable trajectories are defined throughout the sequence using [14] and then spatially quantized to produce the initial partition used as starting point for the BPT algorithm. Unlike [22], trajectories are introduced in a fully unsupervised manner, without prior clustering into a predefined number of classes. The Trajectory BPT is then computed and, at each iteration, the two most similar trajectories are merged. The trajectory similarity is defined with color and motion information. The output of the system is a hierarchical representation of the whole video sequence which can be used to obtain multiple partitions, depending on the cut performed on it [25]. This kind of representation is useful since it allows subsequent systems to choose the desired partition granularity without having to re-run the segmentation algorithm. Alternatively, detection algorithms can analyze the tree structure to locate object of interest.

The main contributions of this work can be summarized as follows: First, we design a simple and efficient region merging approach for video representation and segmentation that addresses the main problem of video segmentation: temporal coherence of regions. The introduction of long term trajectories as initial partition greatly contributes to region consistency over time. Second, most of the state-of-the-art algorithms for video segmentation treat video strictly as a 3D volume and do not explicitly address motion and space separately for segmentation. To tackle this problem, we devise adequate color (spatial) and motion (temporal) models for regions resulting from the merging process. Third, we propose to represent the video as a binary tree of trajectory regions (which are set of spatially neighboring trajectories). Trees are an efficient and natural way to represent hierarchical structures. Moreover, the binary nature of the tree leads to efficient creation and processing of the video representation.

The rest of the paper is organized as follows. Section 2 introduces the initial trajectory estimation algorithm from [14] with the appropriate modifications to generate an initial partition for the proposed system. In section 3, the core of the Trajectory BPT algorithm is exposed. Finally, results are presented in Section 4, while section 5 reports the main conclusions and discuss possible future work.

2. Trajectory Estimation

Trajectories provide a way to reliably propagate long term motion information along the sequence. In contrast to descriptor-based tracking [15], optical flow tracking algorithms provide a denser coverage of tracked points while maintaining and even improving the tracking accuracy [17]. We propose to use a quantized version of the tracker [14] to provide the system with an initial partition to start the BPT

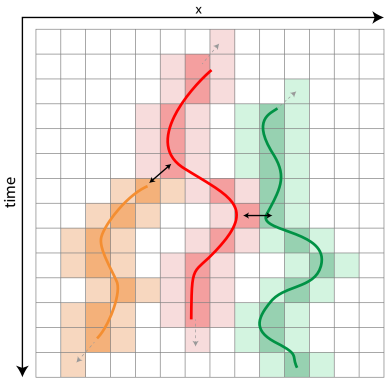

Figure 2. Horizontal cut of a video sequence. Estimated trajectories with sub-pixel accuracy are shown with red, green and orange curves. Quantized trajectories corresponds to voxels filled with dark colors whereas adjacent voxels are indicated in light colors. Trajectory adjacency relations are represented by two-way arrows.

computation.

In a nutshell, the tracking algorithm [14] finds reliable starting points for trajectories and tracks them from frame to frame using the estimated optical flow [11] until the flow reliability falls below a given threshold. Reliable optical flow estimates can be found at points fulfilling the following three conditions: 1) they have a visible spatiotemporal structure in their neighborhood 2) they do not become occluded 3) they are not on a motion boundary. Since the flow reliability is used to define the initial trajectories and also to measure the distance between trajectory regions during the tree creation process, we present the three key reliability notions [14] for a point $\boldsymbol{p} = (x, y, t)$ in the video.

Structure reliability This reliability measures the presence of visible structures around the point to be tracked. When no structure is present, the motion estimation and the tracking cannot be precisely done. The structure reliability is defined by means of the second eigenvalue λ_2 of the structure tensor: $J_s = K_s * (\nabla I \nabla I^\top)$ for each point in the video. $\nabla I = [I_x, I_y, I_t]^T$ denotes a spatio-temporal gradient, K_s is a Gaussian kernel of standard deviation $\sigma = 1$ and the operator $*$ denotes the convolution. The reliability is expressed as:

$$\rho_s(\boldsymbol{p}) = 1 - \exp\left(-\lambda_2(\boldsymbol{p})/\widehat{\lambda}_2(t)\right) \tag{1}$$

where $\widehat{\lambda}_2(t)$ is the average second eigenvalue of the current frame. Candidate points with $\rho_s \approx 1$ appear in corners and edges, similar to a Harris detector [26].

Occlusion reliability Assume that $\boldsymbol{w}(\boldsymbol{p}) = (u(\boldsymbol{p}), v(\boldsymbol{p}))$ is the forward motion field. The backward flow field corresponding to $\boldsymbol{p}$ is $\widetilde{\boldsymbol{w}}(\widetilde{\boldsymbol{p}})$ where $\widetilde{\boldsymbol{p}} = (x+u(\boldsymbol{p}), y+v(\boldsymbol{p}), t+1)$.

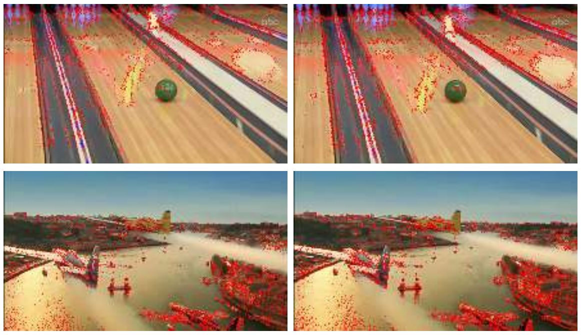

Figure 3. Example of points belonging to trajectories in two consecutive frames of two sequences. Tracked points are marked in red. Most tracked points belong to edges or corners. In homogeneous regions, almost no pixel is tracked.

The flow reliability according to the forward-backward consistency is defined as:

$$\rho_o(\boldsymbol{p}) = \exp\left(-\frac{|\boldsymbol{w}(\boldsymbol{p}) + \widetilde{\boldsymbol{w}}(\widetilde{\boldsymbol{p}})|^2}{0.01(|\boldsymbol{w}(\boldsymbol{p})|^2 + |\widetilde{\boldsymbol{w}}(\widetilde{\boldsymbol{p}})|^2) + 0.5}\right) \tag{2}$$

In the case of non occlusion, $\rho_o \approx 1$, as the forward and the backward flows compensate ($\boldsymbol{w}(\boldsymbol{p}) \approx -\widetilde{\boldsymbol{w}}(\widetilde{\boldsymbol{p}})$). The case where $\rho_o \approx 0$ indicates that $\boldsymbol{p}$ is being occluded and thus the tracking should be stopped.

Motion boundary reliability At motion boundaries, the estimated optical flow is unreliable and the trajectory should not be continued. The flow reliability can be assessed by computing the flow gradient in the horizontal and vertical directions:

$$\rho_{mb}(\boldsymbol{p}) = \exp\left(-\frac{|\nabla u(\boldsymbol{p})|^2 + |\nabla v(\boldsymbol{p})|^2}{0.01|\boldsymbol{w}(\boldsymbol{p})|^2 + 0.002}\right) \tag{3}$$

If any of ρ_s, ρ_o or ρ_{mb} falls below a given threshold, the trajectory stops. The used threshold values are the same as in [14]. The motion estimation algorithm [11] provides sub-pixel accuracy on flow values. Therefore, bilinear interpolation is used to track pixels with sub-pixel accuracy resulting in a precise definition of the trajectory location. The trajectory can be expressed as a sequence of points $P = \{(x_t, y_t, t), \ldots, (x_{t+l-1}, y_{t+l-1}, t+l-1)\}$. Once the complete trajectory is computed with sub-pixel accuracy, each point location is quantized to the closest spatial integer position for each frame: $P_Q = \text{round}(P)$, see Figure 2 for examples of quantization. We found very important to perform the whole tracking process with sub-pixel accuracy prior to quantization, specially in scenes with small displacements. In average, around 10% of voxels belongs to a trajectory of length higher than 2. Examples of points belonging to trajectories can be seen in Figure 3.

3. Trajectory Binary Partition Tree

Once the initial trajectories have been defined as described in Section 2, they are used to form the initial partition for the creation of the Trajectory BPT. The regions forming the initial partition are the trajectories as well as the non-tracked points which are considered trajectories of length 1 in the sequel. Then, a weighted adjacency graph is constructed where nodes represent regions of the initial partition, i.e. trajectories, and edges describe the adjacency relations. Spatial adjacency is defined as 4-connectivity for trajectory points of the same frame. Temporal adjacency is created by connecting each trajectory endpoint to its forward or backward motion-compensated neighbors, see Figure 2 for a few examples.

The Trajectory BPT is then constructed by iteratively merging the two most similar adjacent trajectories. As adjacent trajectories are grouped together, they form what can be called *trajectory regions*. The important characteristics of trajectory regions, in particular their color and motion composition, are captured in a model. At each merging step, the trajectory region models are used to identify the pair of most similar neighboring regions. This pair is merged forming a new trajectory region. The model of this new region is computed and the similarity with neighboring regions is evaluated.

This strategy provides a more precise way to construct a hierarchy of partitions than the one proposed in [4], where the granularity of each hierarchical level relies on a predefined threshold. The Trajectory BPT algorithm iterates the merging steps until one region representing the whole video is left. During the BPT creation process, many partitions are obtained following the merging sequence. But note that, once the tree is constructed, many more partitions can be extracted by applying different graph cut strategies.

As can be seen, the Trajectory BPT algorithm relies mainly on the model describing the trajectory regions and the similarity between two trajectory regions. These two issues are addressed in the following sections.

3.1. Trajectory Region Model

Whether a given segmentation algorithm works with trajectories or other kind of regions, there are many ways to model the partitions elements and to define distance between these elements [27, 28]. Motion segmentation algorithms dealing with trajectories such as [17, 16] only use motion information to define similarity between elements, while other systems such as [5] (implementation by [29]) rely only on region color characteristics. We adopt here an hybrid approach as in [4, 22], noting that color is the most discriminative cue for segmentation and motion allows to introduce dynamic information to the segmentation process.

Trajectories produced by [17] can be as long as the entire video sequence if no occlusion occurs. This provides a very stable starting point for the merging process. However, prior to define the color and motion model, it is important to differentiate between spatial and temporal diversity:

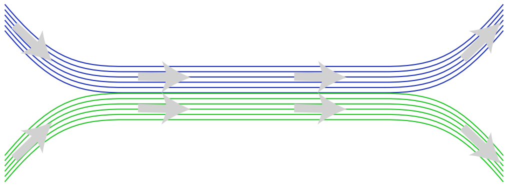

Figure 4. Importance of modeling the temporal evolution of trajectories. If the model only describes the statistical distribution of the motion (for example with motion histogram), the two trajectories will have the same motion representation. However, they clearly belong to different objects because they involve different motions at the same time instant.

- Objects tend to involve rich color distributions that are stable over time.
- Object motion tends to be spatially simple (uniform translation, rotation or zoom for example), but changing over time.

Therefore, color presents high spatial but low temporal diversity, while motion characteristics are the opposite. This encourages the use of different models for color and motion.

Color Model Color stability over time is, in fact, an assumption made by the optical flow estimation algorithm. Therefore, it is reasonable to assume that an image region can be represented with few colors regardless of its temporal span. Therefore we consider the trajectory region color model to be an adaptive histogram (signature) described by at most $n = 8$ dominant colors in the Lab color space [2]. The signature of a region R is a set of pairs $s_R = \left\{(p_1^R, \boldsymbol{c}_1^R), \ldots, (p_i^R, \boldsymbol{c}_i^R)\right\}, i \leq n$, where $\boldsymbol{c}_i^R$ is a representative color and $0 < p_i^R \leq 1$ its corresponding percentage of occurrence. This kind of representation is advantageous because it provides a rather accurate representation of the region color without having to deal with the complete 3D color histogram.

For the initial trajectories, signatures are estimated by clustering colors using k-means, while a greedy algorithm is used for trajectory regions created during the BPT merging process for computational reasons. When a region is created from its two children, the resulting histogram is the union of the two underlying histograms. If the resulting number of dominant colors is above n, the two most similar colors are merged according to the distance (4) weighted by the percentage $(p_i + p_j)c_{ij}$ and replaced by their average. The distance c_{ij} between different colors, $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$, is defined as:

$$c_{ij} = 1 - \exp\left(-\frac{|\boldsymbol{c}_i - \boldsymbol{c}_j|}{\gamma_c}\right) \tag{4}$$

with $\gamma_c = 14$, defining a soft threshold on color difference.

Motion Model Object motion can be easily described between two consecutive frames. Typically, motion between frames is composed of piecewise-smooth regions. However, in spite of this spatial simplicity, object motion can change over time (unlike color). Therefore, the most important role of the motion model is to capture the different motions across frames and to preserve the order in which they appear. Figure 4 illustrates the importance of modeling the temporal evolution of motion and therefore why models based on motion histogram should be avoided.

Therefore, the motion of each trajectory region R is represented by a set of motion vectors $m_R = \left\{\widehat{\boldsymbol{u}}_t^R, \widehat{\boldsymbol{u}}_{t+1}^R, \ldots, \widehat{\boldsymbol{u}}_{t+l-1}^R\right\}$ where $\widehat{\boldsymbol{u}}_t^R$ is the mean motion vector of the trajectory region at a given time instant t.

3.2. Trajectory Region Distance

The merging sequence creating the BPT is defined by a similarity measure between neighboring regions. This similarity is based on several distance notions.

Color Distance The color model relies on an adaptive histogram representing at most $n = 8$ dominant colors. The similarity between two models can be computed with the Earth Mover's Distance (EMD) [30] which states the histogram comparison as a transportation problem:

$$d_c(s_1, s_2) = EMD(s_1, s_2) = \min \sum_{i \le n_1, j \le n_2} c_{ij} f_{ij} \tag{5}$$

$$s.t \quad f_{ij} \ge 0, \quad \sum_{i \le n_1} f_{ij} = p_j^2, \quad \sum_{j \le n_2} f_{ij} = p_i^1 \tag{6}$$

The goal is to find a set of flows f_{ij} that *transports* the probability masses from the histogram s_1 to the histogram s_2 and that minimizes the cost function (5). Each histogram s_1, s_2 has n_1 and n_2 representative colors respectively, and it is possible that $n_1 \neq n_2$. The elementary cost c_{ij} is defined by (4).

Motion Distance As stated in [17], even if two objects share the same motion during a long period of time, as soon as they move differently, they can be assigned to two different entities. Therefore, two trajectories are as different as their maximum motion difference at a given time instant:

$$d_m(m_1, m_2) = \max_{t \in T} \quad 1 - \exp\left(-\frac{\rho_t \|\widehat{\boldsymbol{u}}_t^1 - \widehat{\boldsymbol{u}}_t^2\|}{\gamma_m}\right) \tag{7}$$

where T is the common period of time of both trajectories. The coefficient $\gamma_m = 4$ acts similarly to γ_c in (4), defining a soft threshold. An important factor is ρ_t which measures the intra-frame flow reliability. It is defined as:

$$\rho_t = \min_{\substack{i=1,2 \\ q=s,v,mb}} \widehat{\rho}_q^i(t) \tag{8}$$

Basically, for each frame, ρ_t is set to the minimum of the three reliabilities (structure, occlusion and motion boundary) of the two trajectories $i = 1, 2$ at each frame. At the last merging steps of the BPT, trajectory regions may be composed of many pixels of the same frame. Therefore, for each trajectory, the mean value of the structure $\widehat{\rho}_s^i(t)$, occlusion $\widehat{\rho}_o^i(t)$, and motion boundary $\widehat{\rho}_{mb}^i(t)$ reliability is computed.

Final trajectory region distance Although color and motion are two key characteristics for video segmentation, other factors can help to improve results. In this work, we use a size factor $d_v(v_1, v_2)$ that encourages the merging of regions of small size:

$$d_v(v_1, v_2) = \log(1 + \frac{\min(v_1, v_2)}{\gamma_v}) \tag{9}$$

where v_1 and v_2 are the volumes of the two trajectory regions in voxels. γ_v acts similarly as γ_c, γ_m and it is set to 5% of the video volume. Introducing this factor prevents smaller regions to be considered of equal importance as the bigger ones. The final region distance is:

$$d = (1 - (1 - d_c)(1 - d_m))\, d_v \tag{10}$$

where notation has been simplified for clarity purposes. d is close to zero when both color and motion are very similar, while it is close to d_v if either d_c or d_m are close to one. Other combinations of region model characteristics have been proposed in the literature [27], but (10) proved to give good results.

4. Results

Multiple segmentations of the same input space can be extracted from the hierarchical trajectory BPT. Depending on the application, several graph cut strategies can be applied on the tree to capture regions representing semantic notions. However, in this paper, we focus on the tree construction and the quality of the merging sequence. As a result, we will restrict ourselves to the evaluation of the partitions obtained through the merging sequence [25]. We do not consider any particular application and analyze the quality of partitions involving between 900 and 100 regions. Note that following the merging sequence, we have an exact control on the desired number of regions unlike methods like GBH or Meanshift.

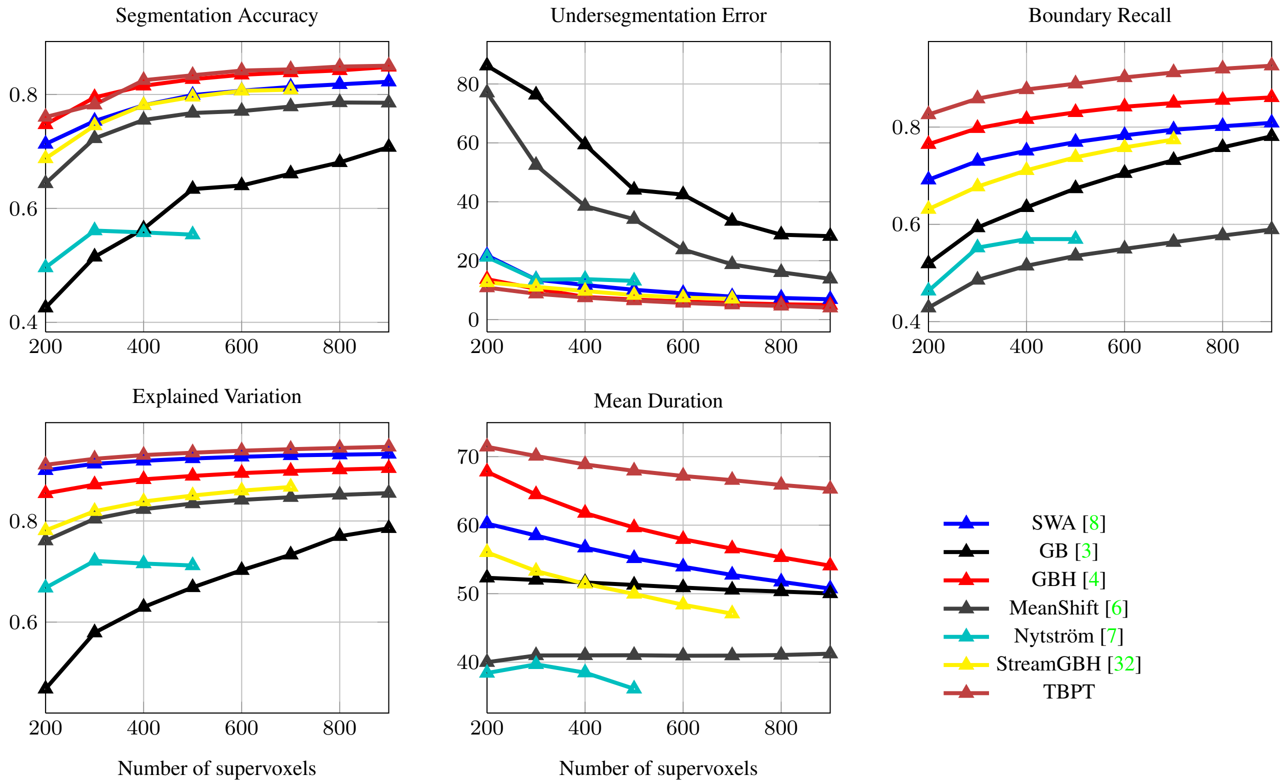

Figure 5. Results on the dataset of [31]. From left to right and top to bottom: Segmentation Accuracy (SA), Undersegmentation Error (UE), Boundary Recall (BR) and Mean Duration versus the region number. The proposed system is among the best ones in terms of SA and UE and the best in BR. The Trajectory BPT creates regions spanning longer temporal intervals than other state of the art methods.

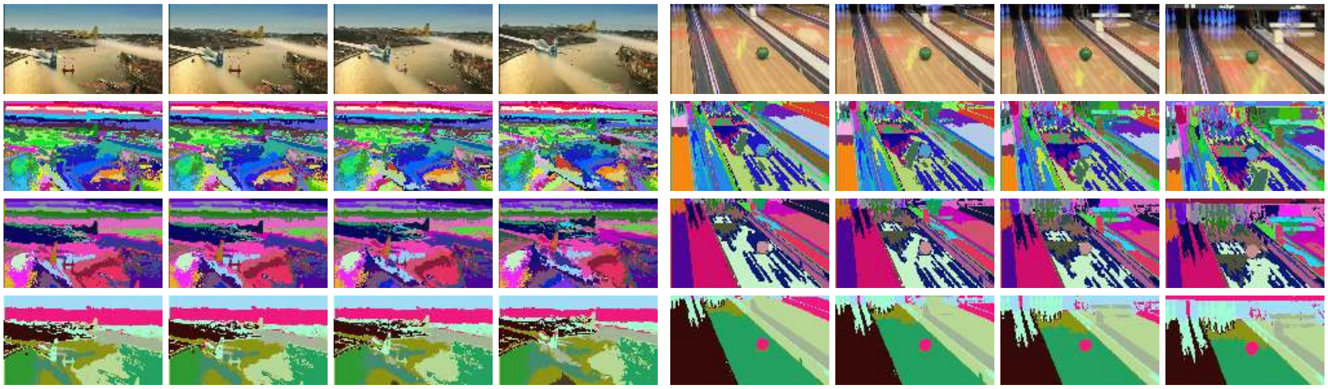

Figure 6. Two examples of the segmentation hierarchy. For each of the two examples, the first row contains frames 1, 5,10 and 15. The second, third and fourth rows show segmentations with 100, 40 and 10 segments respectively. A segment is uniquely colored across frames.

4.1. Metrics and datasets

We use the evaluation method proposed in [29] with the dataset from *xiph.org* used in [31] composed of 8 sequences of approximately 80 frames each. Each frame has a semantic ground-truth segmentation leading to a total of 639 annotated frames. The evaluation metrics are the ones discussed in [29]: the Undersegmentation Error (UE) measures what fraction of voxels exceeds the volume boundary of the ground-truth region; the Boundary Recall (BR) assesses the quality of the spatiotemporal boundary detection; Segmentation Accuracy (SA) quantifies what fraction of ground-truth segments is correctly classified and Explained Variation (EV) is a human independent measure assessing spatio-temporal uniformity. A formal and detailed definition of these measures can be found in [29]. Additionally, since in

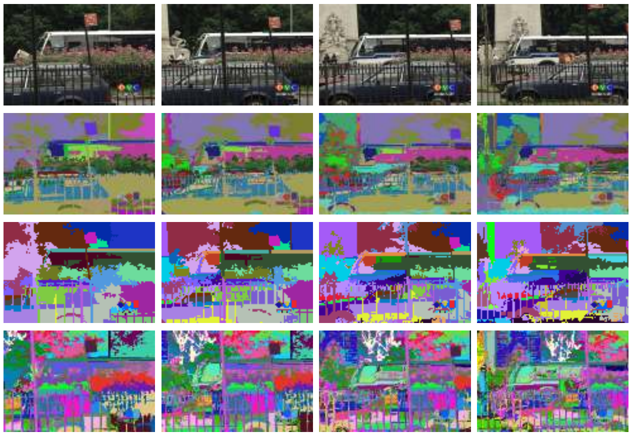

Figure 7. Frames 1,11,21,31 from the bus sequence in the dataset [31]. Comparison of the partitions obtained with the SWA method (second row), the GBH algorithm (third row) and the proposed Trajectory BPT (fourth row). Each region is colored with a unique color that is consistent over time. Each partition involves roughly 100 regions.

video segmentation it is important to have a stable segmentation over time, the mean duration of trajectory regions is also presented.

Results of segmentation measures are shown in Figure 5. It can be observed that the Trajectory BPT approach, while maintaining a competitive UE and UA, clearly outperforms the other methods in BR and EV. This means that 1) 3D boundaries are very well preserved, achieving recalls above 0.8 and 2) produced voxels are more uniform in color statistics according to EV. This is specially difficult in complex scenes involving a lot of details and small regions. We believe that this difference in BR is mainly due to the introduction of the flow reliability into the region similarity. The average duration of the resulting trajectory regions can also be seen in Figure 5 for different number of regions. The introduction of trajectories into the segmentation process has allowed the creation of temporally stable regions spanning throughout longer time intervals than other methods.

For subjective evaluation, we show the partitions for three methods in Figure 7. The sequence is particularly challenging as it involves many small details and severe occlusions. State of the art algorithms have difficulties, but the Trajectory BPT algorithm is able to preserve important boundaries such as the front fence. Although horizontal motion is dominant in the sequence, the proposed algorithm is also able to track thin vertical structures such as the front and back posts.

To see how the algorithm behaves as the hierarchy progresses, Figure 6 shows results on two sequences from the dataset used in [23]. The airplane sequence is specially challenging because many areas have similar and homogeneous colors. As can be noticed, at finer levels of the hierarchy, boundaries are still well preserved. At coarser levels, regions with different semantic may be merged. Nevertheless, instead of simply following the merging sequence, graph cut techniques can be applied on the tree to recover useful objects for a given application [2]. In the bowling sequence, the color contrast is higher, but difficult challenges arise because of big displacements, appearing objects and specular reflections. The Trajectory BPT is able to track most of the objects of the scene and, even at coarser levels, the produced regions have semantic homogeneity. A unix binary (64bit) is provided in the supplemental material to generate the results shown in the paper. Videos showing segmentations for the dataset [31] are also available.

Computational cost The CPU time is governed by the complexity of the Trajectory BPT priority queue used to handle the distance values. Its complexity is $O(E \log E)$ where E is the number of edges between regions. Consumed memory is dominated by the storage of color and motion models for each region. Since region adjacency is sparse, the number of edges E can be considered proportional to the number of regions N. Therefore, the overall algorithm complexity is $O(N \log N)$ in time and $O(N)$ in memory. Overall, the algorithm is able to process video sequences of 3 million voxels in around 1000 seconds using no more than 20GB of memory in a single threaded CPU.

5. Conclusions and future work

In this work, we have proposed an algorithm to construct a hierarchical video representation by merging trajectories. The resulting representation is called Trajectory BPT. The algorithm works in a fully unsupervised manner without making any assumption on the kind of scene nor the type of objects it contains. The proposed algorithm has been compared with state of the art systems and it was shown that the Trajectory BPT improves boundary recall, explained variation, and temporal stability while maintaining undersegmentation and accuracy to very competitive levels. With these results, our claim is that video segmentation algorithm must assess spatial and temporal coherency separately. Since color and motion cues exhibit different statistics both in the spatial and temporal domain, its interpretation and processing should be done accordingly.

The principal limitation of most video segmentation algorithms is the amount of data they have to process. Standard video data rates can go from tenths to hundreds of MB/s, making the processing of medium or even small videos unfeasible. Moreover, there are situations where the entire video is not available and batch processing is needed. We leave this task as future work noting that the Trajectory BPT algorithm can easily be adapted to a streaming scheme as the one discussed in [32], processing chunks of frames in a causal order to deal with video sequences of arbitrary length. Moreover, one of our future objectives is to develop efficient graph cut techniques to be used on the Trajectory BPT. As an initial application, we will target the recovery of depth planes.

References

[1] P. Salembier and L. Garrido. “Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval”. In: *IEEE Transactions on Image Processing* 9.4 (2000), pp. 561–576.

[2] G. Palou and P. Salembier. “2.1 Depth Estimation of Frames in Image Sequences Using Motion Occlusions.” In: *ECCV Workshops*. Vol. 7585. 2012, pp. 516–525.

[3] P. F. Felzenszwalb and D. P. Huttenlocher. “Efficient Graph-Based Image Segmentation”. In: *IJCV* 59.2 (2004), pp. 167–181.

[4] M. Grundmann, V. Kwatra, M. Han, and I. A. Essa. “Efficient hierarchical graph-based video segmentation.” In: *CVPR*. 2010, pp. 2141–2148.

[5] S. Paris and F. Durand. “A Topological Approach to Hierarchical Segmentation using Mean Shift”. In: *CVPR*. 2007, pp. 1–8.

[6] S. Paris. “Edge-Preserving Smoothing and Mean-Shift Segmentation of Video Streams”. In: *ECCV*. Vol. 5303. 2008, pp. 460–473.

[7] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. “Spectral grouping using the Nystrom method”. In: *IEEE TPAMI* 26.2 (2004), pp. 214–225.

[8] J. Corso et al. “Efficient Multilevel Brain Tumor Segmentation With Integrated Bayesian Model Classification”. In: *IEEE Transactions on Medical Imaging* 27.5 (2008), pp. 629–640.

[9] O. Wirjadi. *Survey of 3D image segmentation methods*. Tech. rep. 123. Fraunhofer Institut fur Techno und Wirtschaftsmathematik, 2007.

[10] L. Xu, J. Jia, and Y. Matsushita. “Motion detail preserving optical flow estimation”. In: *CVPR*. 2010, pp. 1293–1300.

[11] T. Brox and J. Malik. “Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation”. In: *IEEE TPAMI* 33.3 (2011), pp. 500–513.

[12] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert. “Highly accurate optic flow computation with theoretically justified warping”. In: *IJCV* 67.2 (2006), pp. 141–158.

[13] A. Wedel, T. Pock, C. Zach, H. Bischof, and D. Cremers. “Statistical and Geometrical Approaches to Visual Motion Analysis”. In: Berlin, Heidelberg, 2009. Chap. An Improved Algorithm for TV-L1 Optical Flow, pp. 23–45.

[14] N. Sundaram, T. Brox, and K. Keutzer. “Dense point trajectories by GPU-accelerated large displacement optical flow”. In: *ECCV*. 2010.

[15] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. “MonoSLAM: Real-Time Single Camera SLAM”. In: *IEEE TPAMI* 29.6 (2007), pp. 1052–1067.

[16] S. Rao, R. Tron, R. Vidal, and Y. Ma. “Motion Segmentation in the Presence of Outlying, Incomplete, or Corrupted Trajectories”. In: *IEEE TPAMI* 32.10 (2010), pp. 1832–1845.

[17] T. Brox and J. Malik. “Object segmentation by long term analysis of point trajectories”. In: *ECCV*. Heraklion, Crete, Greece, 2010, pp. 282–295.

[18] P. Ochs and T. Brox. “Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions”. In: *ICCV*. Washington, DC, USA, 2011, pp. 1583–1590.

[19] N. Sundaram and K. Keutzer. “Long term video segmentation through pixel level spectral clustering on GPUs.” In: *ICCV Workshops*. 2011, pp. 475–482.

[20] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. “Contour Detection and Hierarchical Image Segmentation”. In: *IEEE TPAMI* 33 (2011), pp. 898–916.

[21] J. Shi and J. Malik. “Normalized Cuts and Image Segmentation”. In: *IEEE TPAMI* 22.8 (2000), pp. 888–905.

[22] J. Lezama, K. Alahari, J. Sivic, and I. Laptev. “Track to the Future: Spatio-temporal Video Segmentation with Long-range Motion Cues”. In: *CVPR*. 2011.

[23] P. Sundberg, T. Brox, M. Maire, P. Arbelaez, and J. Malik. “Occlusion boundary detection and figure/ground assignment from optical flow”. In: *CVPR*. Washington, DC, USA, 2011, pp. 2233–2240.

[24] A. N. Stein and M. Hebert. “Occlusion Boundaries from Motion: Low-Level Detection and Mid-Level Reasoning”. In: *IJCV* 82.3 (2009), pp. 325–357.

[25] J. Pont-Tuset and F. Marqués. “Supervised Assessment of Segmentation Hierarchies”. In: *ECCV*. 2012.

[26] C. Harris and M. Stephens. “A Combined Corner and Edge Detector”. In: *Alvey Vision Conference*. 1988, pp. 147–151.

[27] V. Vilaplana, F. Marques, and P. Salembier. “Binary Partition Trees for Object Detection”. In: *IEEE Transactions on Image Processing* 17.11 (2008), pp. 2201–2216.

[28] F. Calderero and F. Marques. “Region Merging Techniques Using Information Theory Statistical Measures”. In: *IEEE Transactions on Image Processing* 19.6 (2010), pp. 1567–1586.

[29] C. Xu and J. Corso. “Evaluation of super-voxel methods for early video processing”. In: *CVPR*. 2012, pp. 1202–1209.

[30] Y. Rubner, C. Tomasi, and L. Guibas. “A metric for distributions with applications to image databases”. In: *ICCV*. 1998, pp. 59–66.

[31] A. Chen and J. Corso. “Propagating multi-class pixel labels throughout video frames”. In: *Image Processing Workshop (WNYIPW)*. 2010, pp. 14–17.

[32] C. Xu, C. Xiong, and J. J. Corso. “Streaming Hierarchical Video Segmentation”. In: *ECCV*. Vol. 7577. Heidelberg, 2012, pp. 626–639.