HIERARCHICAL PARTITION-BASED REPRESENTATIONS FOR IMAGE SEQUENCES USING TRAJECTORY MERGING CRITERIA

Camilo C. Dorea, Montse Pardàs, Ferran Marqués

Technical University of Catalonia, Barcelona, Spain
{camilo, montse, ferran}@gps.tsc.upc.es

ABSTRACT

This paper describes a hierarchical analysis framework for image sequences. Region merging schemes traditionally used in the construction of partition hierarchies are extended to multiple frames using trajectory merging criteria. The merging criteria assess homogeneity among features throughout the entire sequence to recursively create partitions in the spatio-temporal domain. We propose similarity measures using *long-term affine* and *translational* motion features. Furthermore, the analysis of connectivity relations and the algorithm implementation over Trajectory Adjacency Graphs allow the generation of partition sets containing temporally consistent objects characterized by coherent motion. Lastly, we introduce the novel *Trajectory Tree* as a single, hierarchical representation of the partitions generated for the complete sequence. Experimental results are provided, illustrating the usefulness of the approach.

Index Terms— Sequence analysis, hierarchical video representation, region merging

1. INTRODUCTION

Several proposals in literature have been dedicated to the segmentation of image sequences. In such medium, the segmentation methods generally employ spatial and temporal features in defining image partitions. Starting from an over-segmented partition, works such as [1] and [2] use motion information to guide an independent region merging strategy for each frame. When dealing with image sequences, however, the temporal coherency between frames must be respected. In [3], watershed segments in the current frame are merged according to motion features and then projected onto the subsequent frame, establishing temporal links (tracking) and constraining future segmentation as well. Note that the temporal features used in the previous proposals are short-term in nature: only motion or partition projections between consecutive frames are taken into account. In order to fully exploit temporal information available in the entire image sequence, both [4] and [5] propose a two-level algorithm. First, an initial color-based partition is tracked across frames. Then, the spatio-temporal regions or volumes formed via tracking are merged according to their motion features over multiple frames.

In this paper we present a bottom-up, multi-scale segmentation scheme for image sequences along with an efficient hierarchical representation. The proposal has the following advantages:

- A complete bottom-up analysis containing temporally coherent partition hierarchies for every frame, focusing on the use of long-term motion and structural features.
- A single, hierarchical representation for the entire image sequence, allowing efficient storage and access to the various resolution levels and offering greater flexibility for segmentation, filtering and indexing applications.

Our approach is similar to [4] and [5] in their use of long-term temporal features. The proposed hierarchical representation is closely related in its creation and structure to the Binary Partition Tree introduced in [6]. The work presented herein extends proposals from [7] and introduces original long-term motion analysis and representations.

As depicted in the block diagram of Fig. 1, the approach consists of two main processing steps which define the set of regions contained in the hierarchical representation. Given a color image sequence and motion information in the form of dense optical flow [8], the *Partition Sequence Initialization* block is responsible for the formation of what is called a tracked partition sequence, meaning that regions are temporally linked or labeled across frames, thus forming *Trajectories*. This block is reviewed in Section 2. Next, connectivity constraints as well as long-term motion features are considered by the *Trajectory Merging Algorithm*. The algorithm recursively proposes a hierarchy of partition sequences and is discussed in Section 3. Lastly, given the initial tracked partition sequence and a merging order, a hierarchical representation referred to as the

*This work has been partly supported by the EU project NoE MUSCLE FP6-507752, by grant TEC2004-01914 of the Spanish Government and by CAPES - Brazilian Government.

Trajectory Tree is developed in Section 4. Experimental results and conclusions are presented in Sections 5 and 6, respectively.

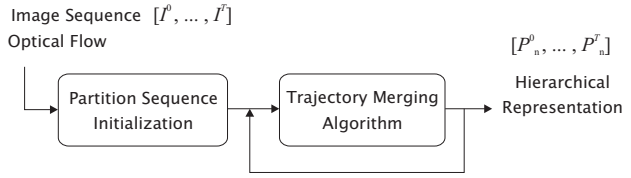

Fig. 1 Block diagram of the recursive segmentation technique used in generating a hierarchical representation for the image sequence.

2. PARTITION SEQUENCE INITIALIZATION

The initial partition sequence is formed with the tracking algorithm described in [7] and is briefly reviewed in this section. The algorithm is responsible for tracking a set of regions homogeneous in color and coherent in terms of affine motion models between consecutive frames (short-term coherence). This set of regions will form the finest level within the final hierarchical representation.

Partition tracking is a recursive algorithm and relies on the projection of a previous partition P^{t-1} onto the current frame. Given a set of dense motion vector estimates [8], an affine model is used to forward motion compensate each region $R_i^{t-1} \in P^{t-1}$. Compensated regions are accommodated to a fine color-based partition of the current frame via fitting procedure. The resulting markers inherit the region labels from frame (t-1) while areas not covered by compensation or fitting are labeled as uncertainty. An example of this mechanism for Foreman frames #0-1 is shown in Fig. 2.

A color-based region merging procedure is used to grow the projected markers over the uncertainty areas. Region merging proceeds until a termination PSNR is reached for the partition. Any remaining regions are assigned new labels. Regions of the resulting partition whose affine models present large residuals are subject to motion-based splitting. This mechanism allows the introduction of motion boundaries across regions homogeneous in color. The algorithm is initiated with a color-based partition and iterated for all consecutive pairs of frames in the sequence.

(a) (b) (c)

Fig. 2 (a) sample partition P^0, (b) markers for frame #1 after compensation and fitting, (c) tracked partition P^1.

3. TRAJECTORY MERGING ALGORITHM

The temporal links established via partition tracking allow segmentation algorithms to access long-term spatio-temporal features. In order to efficiently process the sequence, regions under a common label i are grouped into *trajectories* (T_i) such that:

$$T_i = \left\{ R_i^t \mid R_i^t \neq \varnothing, \forall t \right\}. \quad (1)$$

Trajectories are assumed to be temporally connected and form the primitives used by the merging algorithm and in the final hierarchical representation.

3.1. Trajectory Merging and Splitting

The trajectories of a partition sequence are represented within a Trajectory Adjacency Graph (TAG). This data structure was initially introduced in [7] and here it is extended and further justified. Two trajectories T_i and T_j are said to be *adjacent* if R_i^t and R_j^t co-exist in at least one frame and are neighbors in every frame they co-exist in. The TAG contains a node for each trajectory and a weighted link connecting adjacent trajectories. The trajectory merging algorithm proceeds by removing the link of the TAG with the largest similarity (Section 3.2), merging R_i^t and R_j^t in every frame they co-exist in, and updating the TAG accordingly.

Merging operations propose new trajectories based on an assessment of inter-trajectory homogeneity. Complementary *splitting* operations are introduced to act upon temporal inconsistencies within each trajectory. Trajectories presenting a large relative area difference between regions in consecutive frames (typically 100%) are temporally split at the instant of maximum difference. Judicious splitting can improve homogeneity assessments and correct connectivity flaws as illustrated in the example of Fig. 3.

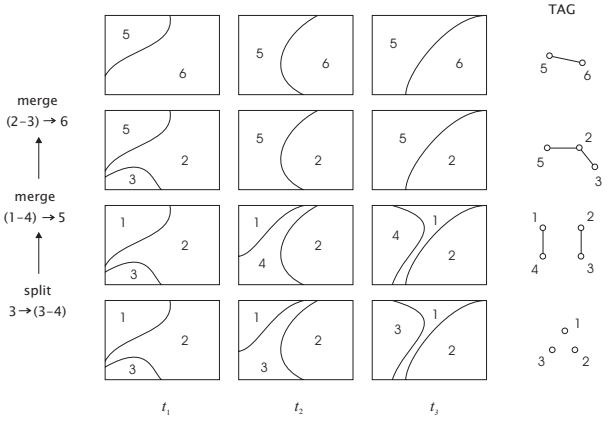

Fig. 3 Initial partition sequence with disconnected TAG (bottom row) and subsequent temporal splitting and merging operations.

3.2. Long-term Motion Similarity

Two distinct motion-based trajectory similarity measures are defined in this paper. The first measure (S_{T_aff}) is aimed at identifying groups of trajectories undergoing similar affine motion. The second measure (S_{T_trans}) focuses on translational motion. The measures may be applied

sequentially in a merging algorithm thus forming groups of affine components which present a common translational motion. For instance, merging of the wheels and body of a car.

Affine trajectory similarity (S_{T_aff}) is a temporal average of motion affinity between adjacent regions in the Displaced Frame Difference (DFD) space. Let $\mathbf{A}_R$ be the 6-parameter affine model for a region R determined via least squares fitting over the optical flow estimates and $\mathbf{d}(\mathbf{A}_R, p)$ the modeled motion vector for each pixel $p \in R$. (Superscripts of R indicating time are omitted for conciseness.) The average incremental modeling error committed by adopting a new motion model is given by:

$$E(R,\mathbf{A}^{new}) = (DFD(R,\mathbf{A}^{new}) - DFD(R,\mathbf{A}_R))/|R| \quad (2)$$

where $DFD(R,\mathbf{A}_R) = \sum_{p\in R} \left| I^t(p) - I^{t+1}(p - \mathbf{d}_k(p,\mathbf{A}_R)) \right|$

and $|R|$ is the area. The affine similarity measure between R_i and R_j tests whether $R = R_i \cup R_j$ is best modeled by $\mathbf{A}_{R_i}$, $\mathbf{A}_{R_j}$ or a new model $\mathbf{A}_R$ derived from the entire support:

$$S_{aff}(R_i,R_j,t) = \min\{E(R_i,\mathbf{A}_{R_j}), E(R_j,\mathbf{A}_{R_i}), \max(E(R_i,\mathbf{A}_R), E(R_j,\mathbf{A}_R))\}. \quad (3)$$

The evaluation of $S_{aff}(R_i,R_j,t)$ also indicates which of the models should be adopted when updating R. The range of choices offers greater resilience to motion estimation errors in particular for small regions along motion boundaries. Finally, the trajectory similarity is the maximum value using a sliding window $\varphi(\cdot)$:

$$S_{T\_aff}(T_i,T_j) = \max\{\varphi(S_{aff}(R_i,R_j,t))\} \quad \forall t \mid R_i,R_j \neq \varnothing . \quad (4)$$

The maximum operator detects motion dissimilarities and enforces them throughout the entire analysis interval. This, for example, allows objects that have presented movement in other frames but are currently static to be properly segmented. Sliding window size is usually 1/5 of sequence duration.

Translational trajectory similarity (S_{T_trans}) emphasizes robustness rather than accuracy. It is evaluated directly in the motion parameter space. Let $\mathbf{B}_R$ be the translational model comprised of the median x and y component values of the estimated optical flow for region R. The region similarity measure, defined with the Euclidean distance, and the trajectory similarity are, respectively:

$$S_{trans}(R_i,R_j,t) = \left\| \mathbf{B}_{R_i} - \mathbf{B}_{R_j} \right\| \quad (5)$$

$$S_{T\_trans}(T_i,T_j) = \max\{\varphi(S_{trans}(R_i,R_j,t))\} \quad \forall t \mid R_i,R_j \neq \varnothing . \quad (6)$$

4. HIERARCHICAL REPRESENTATION

The *Trajectory Tree* is proposed as the hierarchical representation for the set of partition sequences obtained with the Trajectory Merging Algorithm. Each node of the tree represents a trajectory and the links connect merged trajectories with their results. Due to the inclusion of splitting operations, the leaves of the tree are formed by relabeling the initial partition sequence in terms of the finest among all trajectories. Consider, for example, the merging procedure initiated in Fig. 3 and culminating with a single trajectory for the entire sequence support (root node). The associated Trajectory Tree and partition sequence of leaf nodes are shown in Fig. 4.

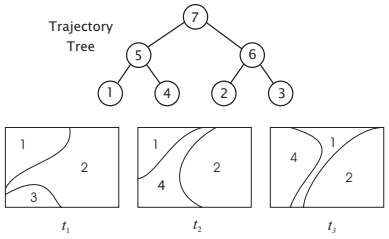

Fig 4. Partition sequence of leaf nodes and the Trajectory Tree.

The Trajectory Tree is binary in nature, encoding all the merging operations executed over an initial partition sequence. Its nodes contain video objects defined through the applied homogeneity criteria. Partitions of various resolutions can be easily accessed by moving up or down through the various node levels. Sophisticated processing techniques such as prunning or propagation [6] may also be applied to the Trajectory Tree.

5. RESULTS

Trajectory Trees and their most significant trajectories are presented for Foreman QCIF [0,30] and Table Tennis CIF [0,22] sequences. The sequence of leaf nodes (initial partition sequence, see Section 2) for Foreman is shown in Fig. 5. The partition of Fig. 2(a) containing 40 regions was used to initiate the tracking procedure [7] responsible for generating a total of 79 initial trajectories (leaves). Merging order between trajectory pairs was established with S_{T_aff} until the 40th merging operation after which S_{T_trans} was used. Video objects of interest presenting the largest motion dissimilarities can be easily identified in the upper nodes of Fig. 5. The *head* for frames [0,30] is one of the last two nodes along with a node containing *background*, *shoulders* (shown in Fig. 5) and *Siemens sign* (not shown). Note that the shoulder was not completely separated from the background in the initial partition. The additional use of translational-based similarity allows trajectories which may not be adequately modeled by affine parameters to be grouped together. Figs. 7(a) and (b) present nodes of a Trajectory Tree generated with affine similarity only. Note that the helmet is merged to the background and the most significant head node is that of Fig. 7(b). The corresponding instance of nodes from Fig. 5 are repeated in Fig. 7(d) and (e) for comparison purposes.

A total of 53 leaves are contained in the initial partition sequence for Table Tennis in Fig. 6. The rigid motions allowed the Trajectory Tree to be generated exclusively with S_{T_aff}. The top three nodes correspond to the *ball*, *background* and *paddle+arm* for frames [0,22]. The latter node may be decomposed into segments of greater affine similarity: *paddle*, *upper arm* and *lower arm* (see Fig. 6). Note that the upper arm is almost static in several frames. In this case, affine-based merging without the use of long-term information merges the arm to the background before merging it to the paddle as in Fig. 7(c). The complete partition sequences and other examples are available at http://gps-tsc.upc.es/imatge/_Camilo/icassp07.

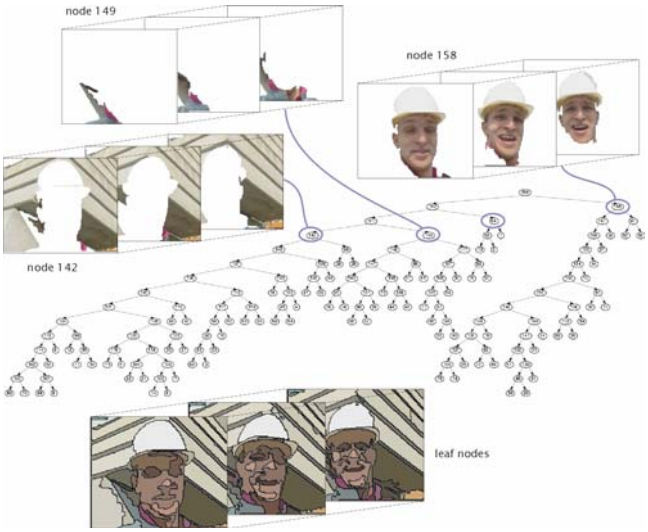

Fig. 5 Trajectory Tree for Foreman [0, 30] and selected node instances at frames #0,15,30.

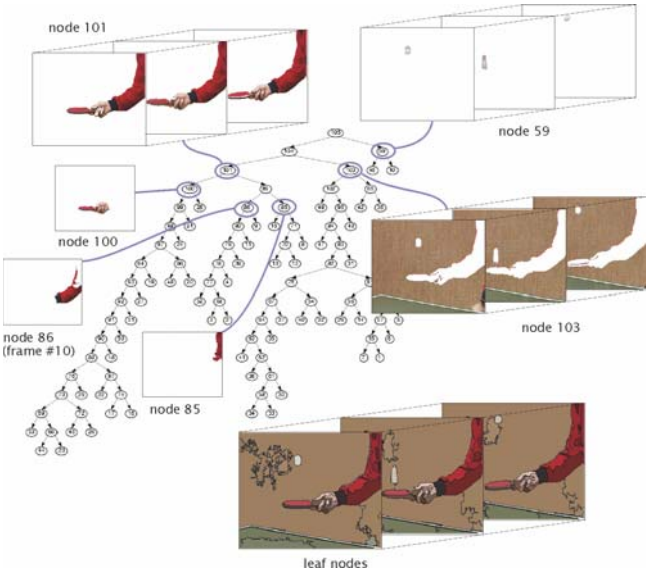

Fig. 6 Trajectory Tree for Table Tennis [0, 22] and selected node instances at frames #0,10,22.

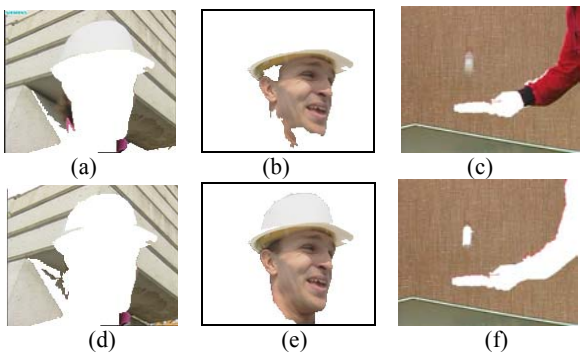

Fig. 7 (a), (b) Selected trajectory intances (frame #10) for Foreman [0,30] generated only with S_{T_aff} and (d),(e) generated with S_{T_aff} + S_{T_trans} . (c) Trajectory instance (frame #9) for Table Tennis when analyzing frame [8,9] only and (f) when analyzing [0,22].

6. CONCLUSIONS

In this work we have proposed the use of trajectories and long-term motion features for the bottom-up analysis of image sequences. The Trajectory Tree has been introduced as an efficient hierarchical representation of the partition sequences obtained with our merging algorithm. Currently we are investigating the use of MPEG-7 and other trajectory descriptors for sequence indexing and filtering applications on the Trajectory Tree.

7. REFERENCES

[1] Y. Altunbasak, P. Eren and A. Tekalp, "Region-based parametric motion segmentation using color information", *Graphical Models and Image Proc.,* vol. 66, pp. 13-23, 1998.

[2] F. Moscheni, S. Bhattacharjee and M. Kunt, "Spatial-temporal segmentation based on region merging", *IEEE Trans. Patt. Anal. Mach. Intell*, vol. 20, pp. 897-915, 1998.

[3] D. Wang, "Unsupervised video segmentation based on watersheds and temporal tracking", *IEEE Trans. Circuits Sys. Video Tech.*, vol. 8, pp. 539-546, Sept. 1998.

[4] V. Mezaris, I. Kompatsiaris and M. Strintzis, "Video object segmentation using Bayes-based temporal tracking and trajectory-based region merging", *IEEE Trans. Circuits Sys. Video Tech.*, vol. 14, pp. 782-795, June 2004.

[5] Y. Tsai, C. Lai, Y. Hung and Z. Shih, "A Bayesian approach to video object segmentation via merging 3-D watershed volumes", *IEEE Ttrans. Circuits Syst. Video Tech*., vol. 15, pp. 175-180, Jan. 2005.

[6] P. Salembier and L. Garrido, "Binary Partition Tree as an efficient representation for image processing, segmentation and Information Retrieval", *IEEE Trans. Image Processing*, vol. 9, pp. 561-576, April 2000.

[7] C. Dorea, M. Pardàs and F. Marqués, "Generation of long-term color and motion coherent partitions", in Proc. *ICIP'06*, Atlanta, USA, Oct. 2006.

[8] M.J. Black and P. Anandan, "The robust estimation of multiple motions: parametric and piece-wise smooth flow fields", *Comp. Vision and Image Understanding*, vol. 63, pp. 75-104, 1996.