Unsupervised Motion Segmentation Using Metric Embedding of Features

Yusuf Osmanlıoğlu[1], Sven Dickinson[2], and Ali Shokoufandeh[1]

[1] Dept. of Computer Science, Drexel University, USA
{osmanlioglu,ashokouf}@cs.drexel.edu
[2] Dept. of Computer Science, University of Toronto, Canada
sven@cs.toronto.edu

Abstract. Motion segmentation is a well studied problem in computer vision. Most approaches assume a priori knowledge of the number of moving objects in the scene. In the absence of such information, motion segmentation is generally achieved through brute force search, e.g., searching over all possible priors or iterating over a search for the most prominent motion. In this paper, we propose an efficient method that achieves motion segmentation over a sequence of frames while estimating the number of moving segments; no prior assumption is made about the structure of scene. We utilize metric embedding to map a complex graph of image features and their relations into hierarchically well-separated tree, yielding a simplified topology over which the motions are segmented. Moreover, the method provides a hierarchical decomposition of motion for objects with moving parts.

Keywords: non-rigid motion segmentation, hierarchically well-separated trees, metric embedding

1 Introduction

Motion segmentation aims to identify moving objects in a video sequence by clustering features or regions over consecutive frames. There exist a wide variety of methods for motion segmentation. Image differencing [4, 18] is among the simplest methods available which consists of thresholding the intensity difference between consecutive frames. Another group of techniques used in segmentation is based on statistical models. Typically, the problem is formulated as a classification task in which each pixel is classified as either foreground or background. Maximum a posteriori (MAP) estimation [21], particle filters [20], and expectation maximization [22] are frameworks that are commonly exploited in statistical approaches. Wavelets [15], optical flow [25], layers [17], and factorization [9,11,23] form the basis of other common approaches to motion segmentation. One common drawback of many of these approaches is their reliance on a priori knowledge of the number of moving objects in the scene. In this paper, we overcome this drawback by approaching the motion segmentation problem from a graph theoretical perspective. From a complete graph over the set of image features, we use metric embedding techniques to yield a restricted tree topology, over which a quadratic optimization problem formulation yields an estimate of the number of motion clusters.

Due to its representational power, graphs are commonly used in many computer vision tasks. Features extracted from an image can be represented by an undirected complete graph with weighted edges. Since it is hard to solve problems over graphs in general, approximate solutions are a viable way to tackle such problems. *Metric embedding* is one of the fundamental techniques used to achieve this goal, and consists of mapping data from a source space to a "simpler" target space while preserving the distances. It is well known that approximate solutions to many NP-hard problems over graphs and general metric spaces can be achieved in polynomial time once the data is embedded into trees. However, such embeddings tend to introduce large distortion.

A common technique for overcoming such large distortion is the probabilistic approximation method of Karp [12]. Utilizing probabilistic embedding, Bartal [1] introduced the notion of *hierarchically well-separated trees* (HSTs), where edge weights on a path from the root to the leaves decrease by a constant factor in successive levels. Embedding graphs into HSTs is especially well-suited to segmentation problems in computer vision, since the internal nodes of the tree represent constellations of nodes of the original graph. Thus, HST structure captures the segment-level information at its internal nodes along with the individual features at its leaves. Following Bartal's seminal work, there have been several studies on HSTs which improved the upper bound of distortion and introduced deterministic embedding algorithms [2, 3, 16]. Finally, Fakcharoenphol et al. [8] devised a deterministic algorithm that achieved embedding of arbitrary metrics into HSTs with a tight distortion bound.

Given two consecutive frames of a video sequence along with a mapping between their features, our method first embeds the latter frame into an HST. Since internal nodes of the HST correspond to clusters of features in the image, our goal is to find a mapping between the features of the previous frame and the internal nodes of the HST. This goal is achieved by minimizing a quadratic cost function which maintains a balance between assigning similar features among frames and minimizing the number of segments identified in the latter frame. We also provide two extensions to our method. While our original formulation provides a single level of clustering for each feature, our first extension allows assigning a feature to more than one cluster. This translates into detection of non-rigid motion of objects such as motion of fingers in a moving hand. Our second extension is in applying the framework to an entire video sequence in an online fashion. We achieve this by keeping track of feature associations at each frame and calculate initial assignments of new frames by utilizing this information. In the rest of the paper, we explain the theoretical details of our method and provide its illustration over two consecutive frames of a video sequence as a proof of concept. We leave empirical evaluation of the method as a future work.

The rest of the paper is organized as follows: Section §2 gives an overview of notations and definitions. In Section §3, we state the optimization problem formulation, which is followed by its application to motion segmentation in Section §4. Finally, in Section §5, we draw conclusions and discuss future work.

2 Notations and Definitions

The term embedding refers to a mapping between two spaces. From a computational point of view, a major goal of embedding is to find approximate solutions to NP-hard problems. Another important use of embedding is to achieve performance gains in algorithms by decreasing the space or time complexity of a polynomial-time solvable problem. Given a set of points P, a mapping $d : P \times P \rightarrow R^+$ is called a distance function if $\forall\, p, q, r \in P$, the following four conditions are satisfied: $d(p, q) = 0$ iff $p = q$, $d(p, q) \geq 0$, $d(p, q) = d(q, p)$, and $d(p, q) + d(q, r) \geq d(p, r)$. The pair (P, d) is called a *metric space* or a *metric*. A finite metric space (P, d) can be represented as a weighted graph $G = (V, E)$ with shortest path as the distance measure, where points in P form the vertex set V and pairwise distances between points become the edge weights. However, the complexity of such graph-based problem formulations can be prohibitive, motivating approaches that reduce graph complexity. A commonly used approach for decreasing graph complexity is based on changing the structure of the graph by removing edges that change the distance metric of the graph, removing or adding vertices, or changing the weights of edges. This approach, however, introduces *distortion* on distances in the graph which is defined as the product of the maximum factors by which the distances in the graph are stretched and shrunk.

In general, it is hard to find an isometric embedding between two arbitrary metric spaces. Therefore, it is important to find an embedding in which the distances between vertices of the destination metric are as close as possible to their counterparts in the source metric space. In reducing the size of a graph by removing vertices and edges, we'd like the pruning process to culminate in a tree, since many problems can be solved much more efficiently on trees than on arbitrary graphs. Embedding of graphs into trees is a very challenging problem, even for the simple case of embedding an n-cycle into a tree. Karp [12] introduced the idea of *probabilistic embedding* for overcoming this difficulty, where given a metric d defined over a finite space P, the main idea is to find a set S of simpler metrics defined over P which dominates d and guarantees the expected distortion of any edge to be small.

Uniform metrics are among the simplest tessellation spaces where all distances are regularly distributed across cells. Such metrics are important from a computational point of view since one can easily apply a divide-and-conquer approach to problems under uniform metrics. Motivated by these observations, Bartal [1] defined the notion of *hierarchically well separated trees (HST)* for viewing finite metric spaces as a uniform metric. A k-HST is defined as a rooted weighted tree, where edge weights from a node to each of its children are the same and decrease by a factor of at least k along any root-to-leaf path. Assuming that the maximum distance between any pair of points (diameter) in the source space is Δ, the source space is separated into clusters (sub-metrics) of diameter

$\frac{\Delta}{k}$. The resulting clusters are then linked to the root as child nodes with edges of weight $\frac{\Delta}{2}$. The relation between parent and child nodes continues recursively until the child nodes consist of single data elements.

Bartal has shown the lower bound for distortion of embedding into HSTs to be $\Omega(\log n)$. He also provided a randomized embedding algorithm that utilizes probabilistic partitioning with a distortion rate of $O(\log^2 n)$. In subsequent work, both Bartal [2] and Charikar et al. [3] introduced deterministic algorithms with smaller distortion ($O(\log n \log\log n)$). Konjevod et al. [16] were the first to improve the upper bound on distortion to $O(\log n)$ for the case of planar graphs. Fakcharoenphol et al. [8] closed the gap for arbitrary graphs by introducing a deterministic algorithm with a tight distortion rate ($\Theta(\log n)$). The deterministic nature of their algorithm made this result of great practical value.

A fundamental set of problems in computer science involves classifying a set of objects into clusters while minimizing a prescribed cost function. The main goal of the classification problem is to assign similar objects to the same cluster. Typical cost functions account for the cost of assigning an object to a cluster and the cost of assigning a pair of similar objects to two unrelated clusters (separation cost). The *multiway cut problem* of Dahlhaus et al. [5] is a simplified classification task that accounts only for the separation cost. Namely, for a given graph with nonnegative edge weights and a predefined set of terminal nodes, it builds an assignment of nonterminals to terminals that minimizes the sum of the edge weights between nodes assigned to distinct terminals:

Definition 1 *Given a graph $G = (V, E)$ with nonnegative edge weights $w : E \to \mathbb{R}$ and a subset $T \subseteq V$ of terminal nodes, find a mapping $f : V \to T$ that satisfies $f(t) = t$ for $t \in T$, and minimizes $\sum_{uv \in E, f(u) \neq f(v)} w(u, v)$.*

Karzanov [13] proposed a generalization of the multiway cut known as the 0-*extension problem*. In his formulation, the cost function accounts for distance between terminals when measuring the cut weight of nonterminal edges. Specifically, each term $w(u, v)$ with $\{uv \in E, f(u) \neq f(v)\}$ of the cost function in Definition 1 will be replaced by $w(u, v)\delta(f(u), f(v))$, where $\delta(f(u), f(v))$ is the distance between terminals to which u and v are assigned.

Finally, Kleinberg and Tardos [14] presented the most general form of the classification task known as *metric labeling problem*. Given a set of objects P and a set of labels L with pairwise relationships defined among the elements of both sets, metric labeling assigns a label to each object by minimizing a cost function involving both separation and assignment costs. Separation cost penalizes assigning loosely related labels to closely related objects while assignment cost penalizes labeling an object with an unrelated label. The cost function $Q(f)$ can be stated as follows:

$$Q(f) = \sum_{p \in P} c(p, f(p)) + \sum_{e=(p,q) \in E} w_e d(f(p), f(q)).$$

where, $c(p, l)$ represents the cost of labeling an object $p \in P$ with a label $l \in L$ and $d(\cdot, \cdot)$ is a distance measure on the set L of labels. Although, there has been ample studies on solving classification problems using labeling methods, their work was the first study that provided a polynomial-time approximation algorithm with a nontrivial performance guarantee.

Metric labeling is closely related to one of the well-studied combinatorial optimization problems called *quadratic assignment*. Given n activities and n locations in a metric space, the goal of quadratic assignment is to place each activity at a different location by minimizing the cost. Similar to the metric labeling, there are two terms affecting the cost of assignments. Placing an activity i at a location l introduces an operating cost of $c(i, l)$. Moreover, popular activities should be located close to each other to minimize the overall cost which leads the cost function to penalize the separation of closely related activities. Assuming that a value w_{ij} measures the interaction between activities i and j, and a distance function $d(l_1, l_2)$ measures the distance between labels l_1 and l_2, the quadratic assignment problem seeks to minimize $\sum_i c(i, f(i)) + \sum_{i,j} w_{ij} d(f(i), f(j))$ over all bijections f.

3 Optimizing Number of Segments

Motivated by the quadratic assignment and metric labeling problems, we pose the following optimization problem:

Definition 2 *Given an object graph $G_{\mathcal{O}}$ and a label graph $G_{\mathcal{L}}$ both equipped with shortest path metric, and a similarity function defined between nodes of the two graphs, find a mapping f from nodes of $G_{\mathcal{O}}$ to clusters of nodes of $G_{\mathcal{L}}$, in which similar nodes of the two graphs are matched and a minimum number of clusters of $G_{\mathcal{L}}$ is used in the mapping.*

This problem differs from the quadratic assignment problem in that the sizes of the two graphs can be different and the nodes of the first graph match to clusters of nodes in the second graph. The method that we propose to solve this problem involves HSTs which makes it closely related to Kleinberg and Tardos' approach on metric labeling. Our method differs from [14] in that we utilize HSTs to optimize the number of active labels whereas they use it in obtaining a linear programming formulation for the problem. In the next section, we will show how the solution to this problem can be applied to motion segmentation while overcoming the requirement of a priori knowledge of the number of clusters.

We tackle the problem in two steps which consist of: 1) embedding $G_{\mathcal{L}}$ into an HST; followed by 2) solving a quadratic optimization problem. We assume that a mapping of object nodes $p \in G_{\mathcal{O}}$ to label nodes $a \in G_{\mathcal{L}}$ is initially given. Our goal is to update this mapping by minimizing the number of so called *active* labels, *i.e.*, labels that have objects assigned to them.

Embedding $G_{\mathcal{L}}$ into the HST $\mathcal{H}$ results in a natural clustering of features of the label graph. The leaf nodes of $\mathcal{H}$ will correspond to the label nodes $G_{\mathcal{L}}$, whereas internal nodes of $\mathcal{H}$ will represent clusters of labels in the $G_{\mathcal{L}}$. The initial assignment between objects and labels can be visualized as assigning object nodes to the leaves of the resulting label tree, as shown in Fig. 1, where only

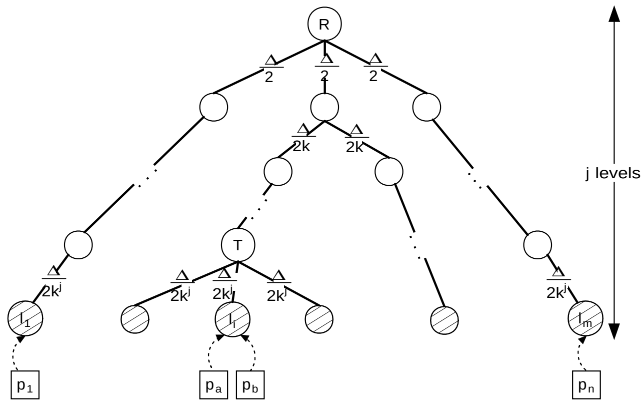

Fig. 1. Embedding of the object graph $G_{\mathcal{O}}$ into the HST representation $\mathcal{H}$ of the label graph $G_{\mathcal{L}}$. Object nodes $p_i \in G_{\mathcal{O}}$ that are assigned to labels $l_i \in G_{\mathcal{L}}$ are shown connected to the leaves of the HST with $\mathcal{H}$ unweighted edges. T is an internal node that represents a cluster of labels, which is the root of the subtree emanating from it. R represents the root of H.

the leaf level nodes are active. We utilize the hierarchical structure of $\mathcal{H}$ in order to update the mapping such that the object nodes get assigned to the internal nodes of $\mathcal{H}$ as labels instead of to its leaves. The following quadratic optimization problem provides an update mechanism to the initial mapping.

$$\min \quad \sum_{p \in G_{\mathcal{O}}} \sum_{T \in \mathcal{H}} d(p,T) w_{p,T} x_T + \sum_{p,q \in G_{\mathcal{O}}} \sum_{T,T' \in \mathcal{H}} d(T,T') w_{p,T} x_T w_{q,T'} x_{T'} \tag{1}$$

$$\text{s.t.} \quad \sum_{T \in \mathcal{H}} w_{p,T} x_T = 1, \ \forall p \in G_{\mathcal{O}} \tag{2}$$

$$x_T \in \{0,1\}$$

where $w_{p,T} = 1$ if the leaf l_i that the object p is assigned to is a descendant of internal node $T \in \mathcal{H}$ and $w_{p,T} = 0$ otherwise, and $d(p,T)$ is the distance between l_i and T measured on HST $\mathcal{H}$. Note that value of $w_{p,T}$ is known a priori based on the initial assignment of objects to labels for all $p \in G_{\mathcal{O}}$ and $T \in \mathcal{H}$. The first term in the above objective function will be minimized if all the objects in $G_{\mathcal{O}}$ are assigned to labels at the leaf level. The second term of the objective function will reduce the number of active labels by enabling nodes that are closer to each other in the tree. Constraint (2) ensures that only one of the labels on the path from $p_i \in G_{\mathcal{O}}$ to the root R will become activated. In Fig. 2, as T and T' are chosen closer to the root R, the contribution of the second term to the cost will be reduced. Also note that the contribution of the second term will be zero for the two nodes p and q if one of their common ancestors becomes activated.

We note that representing (1) as a positive semidefinite program will simplify the quadratic terms and help us prove the performance bound of the method

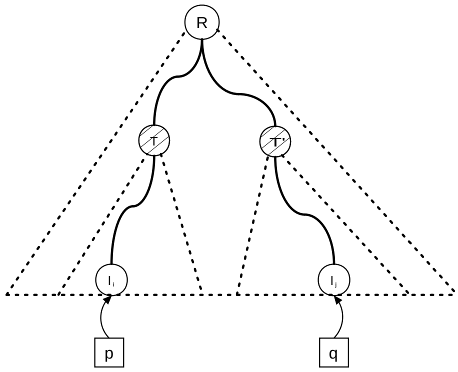

Fig. 2. Two objects being assigned to separate non-leaf labels.

after relaxation. Since $x_{T_i} \in \{0, 1\}$, we have:

$$\min \sum_{p \in G_{\mathcal{O}}} \sum_{T \in \mathcal{H}} d(p, T) w_{p,T} \cdot x_T = \min \sum_{p \in G_{\mathcal{O}}} \sum_{T \in \mathcal{H}} d(p, T) w_{p,T} \cdot x_T^2.$$

Let $T_1, \cdots, T_c$ be the subtrees in $\mathcal{H}$ and $\mathcal{X}$ be a matrix such that $\mathcal{X} = \left[x_{T_i} x_{T_j}\right]_{i,j=1..c}$. Since $\mathcal{X} = x \cdot x^T$, where $x = [x_{T_1}, \cdots, x_{T_c}]$, $\mathcal{X}$ is clearly a PSD matrix. Thus, (1) can be reformulated as follows:

$$\min \quad \sum_{p \in G_{\mathcal{O}}} \sum_{i=1..c} d(p, T_i) w_{p,T_i} \mathcal{X}_{i,i} + \sum_{p,q \in G_{\mathcal{O}}} \sum_{i,j=1..c} d(T_i, T_j) w_{p,T_i} w_{q,T_j} \mathcal{X}_{i,j} \quad (3)$$

$$\text{s.t.} \quad \sum_{i=1..c} w_{p,T_i} \mathcal{X}_{i,i} = 1, \ \forall p \in G_{\mathcal{O}} \quad (4)$$

$$\mathcal{X} \text{ is PSD}$$

$$\mathcal{X}_{i,j} \in \{0, 1\}.$$

Since solving (3) is NP-hard, finding approximate results is desirable. One can obtain a fractional solution to (3) in polynomial time by relaxing the integrality constraint. Then, an approximation can be achieved by using a proper rounding technique [24]. We leave finding a proper rounding algorithm and making performance bound proofs as a future work.

4 Application to Motion Segmentation

Formulation (3) can be applied to the motion segmentation problem where graphs $G_{\mathcal{O}}$ and $G_{\mathcal{L}}$ correspond to features of two consecutive frames. Here the goal is to segment objects in a video sequence according to the relative movement of features across the frames. In this section, we illustrate the application of our method to motion segmentation as a proof of concept and then suggest improvements to the formulation.

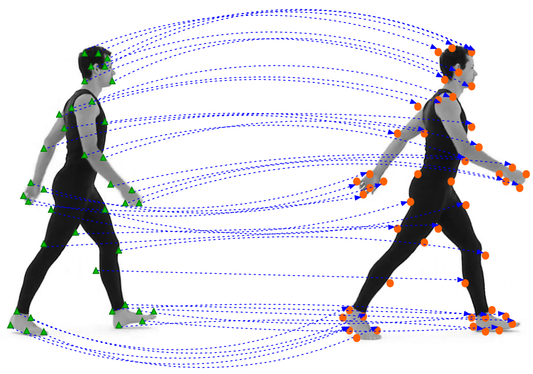

Fig. 3. Initial mapping between individual features of the two frames. Images taken from the video in [7].

Using the proposed method, we first obtain clusters of features by embedding $G_{\mathcal{L}}$ into the HST and then minimize the number of segments by solving the optimization problem based on the relative motion of the features. In general, motion segmentation methods fall short of segmenting sequences without a priori knowledge of the number of independently moving objects. Our formulation overcomes this problem by optimizing the number of segments.

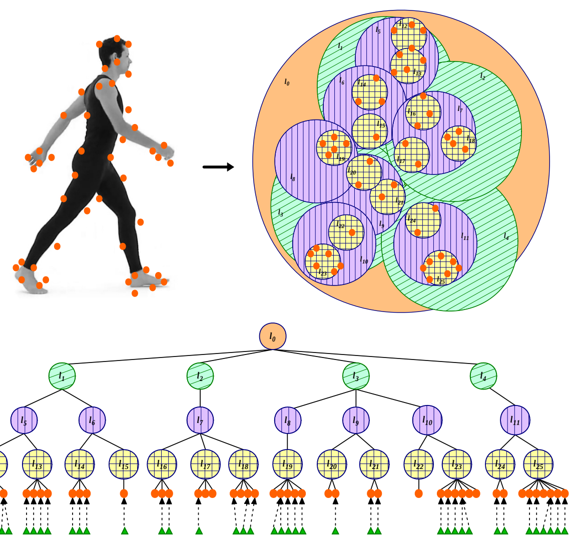

Fig. 4. Embedding features of the second frame into an HST. In the tree, features of the second frame are located at leaf level of the tree and represented as circles. The features of the first frame are represented as triangles and shown as mapping to corresponding features of second frame.

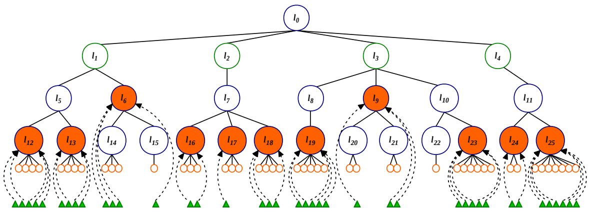

Fig. 5. Darkly shaded internal nodes of the HST are used as labels in the assignment. This implicitly enforces segmentation in the second frame, which is obtained based on the motion.

We begin by establishing a mapping between the features of two consecutive frames by using a graph matching method such as [6, 19] (see Fig. 3). Our method then embeds the feature graph of the second frame into an HST using the deterministic algorithm in [8]. As illustrated in Fig. 4, the embedding procedure recursively groups features into hierarchical clusters. In the resulting HST, actual features are located in the leaf level whereas internal nodes of the tree correspond to clusters of features of the original image. The initial mapping then can be visualized as an assignment of the features of the first frame into the leaves of the resulting HST. By solving the optimization problem (3), the initial mapping is updated such that features of the first frame are mapped to the internal nodes of the HST which represent segments of features in the second frame. Some of the segments will become active as a result of the optimization, shown in darker color in Fig. 5. Thus, features of the first frame enforce a clustering of the second frame based on the similarity between features and the choice of internal nodes of HST to become active. Fig. 6 illustrates the result of the segmentation. In the rest of this section, we will propose further improvements to our method.

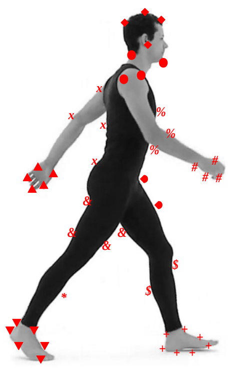

Fig. 6. Image features are segmented into clusters representing independent motions; features with the same symbol belong to the same cluster.

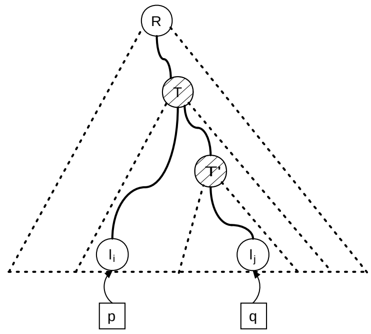

Fig. 7. An object having two active labels in its path to the root.

4.1 Adjusting Rigidity in Segmentation

Due to (4) being an equality, solving (3) will activate exactly one label in a path from each leaf to the root R, as shown in Fig. 2. By relaxing this constraint, we can obtain a partial hierarchical labeling where a node will be assigned to several labels that are hierarchically related to each other, demonstrated in Fig. 7. This can be achieved by replacing (2) with the following:

$$\sum_{T \in \mathcal{H}} w_{p,T} \cdot x_T \geq 1, \ \forall p \in \mathcal{O} \tag{5}$$

This relaxation can be used for relaxing the rigidity assumption in motion segmentation. For example, label T' might represent the features corresponding to the fingers of a hand, while label T represents the features of the entire hand, including the fingers, which move relative to each other.

4.2 Aggregating Motion Over Frames

So far we proposed a framework for establishing motion segmentation of two consecutive frames where we initially assumed in (1) that prior assignment information $w_{p,T}$ for all $p \in G_{\mathcal{O}}$ is given. Considering the application of the framework to a video sequence, we would be interested in utilizing the assignment information of the previous pair of frames in the calculation of the initial mappings of the succeeding frame pair. This will allow us to aggregate motion information over the video sequence in an online fashion as new frames appear. Assume without loss of generality that there exist k labels at a certain level of the tree, one can set the assignment probabilities as $w_{p,T} = 1/k$ for all internal nodes T at that level. This uniform association scheme is suitable for the features that appear for the first time in the sequence. For the features that appear in more than one frame, we propose a voting scheme over the image sequence which keeps track of associations between features and clusters. Specifically, if a feature p appears in close proximity of another feature q in several prior frames but it never appears close to feature r, then probability of assigning p and q to the same cluster will be higher than that of p and r at this level. Thus, newly appearing features will get assigned based only on feature similarities whereas assignment of reappearing features will be biased towards highly correlated recurring features.

5 Discussion and Future Work

In this paper, we presented a novel technique for motion segmentation which, unlike many existing techniques, does not require a priori knowledge of the number of moving objects. Our method overcomes this constraint by embedding image features into hierarchically well-separated trees and then solving a quadratic optimization problem over the tree. We demonstrate the use of our method over two consecutive frames of a walking athlete. We also provided two extensions to our initial formulation. First, we relax the constraint of assigning one label to each feature, enabling us to allow for nonrigidity in motion segmentation, such as detecting the motion of a hand versus the motions of fingers within a moving hand. Second, we propose using the footprints of a feature over previous frames to define an assignment probability for the features in the current frame. In future work, we will apply this strategy to a video sequence and compare the results with the state of the art.

Our method has a limitation arising from the way HST embedding is performed. The embedding algorithm of [8] clusters the features based on their spatial distribution. Thus, for example, in Fig. 6, the features located at the back of the walking athlete and the upper part of the arm are segmented together. However, we would expect the features at the arm to be clustered together with the features of the hand. One of the reasons for this artifact is the low density of the features that we used for demonstrating the method. As the number of features extracted in a frame increases, this misclassification will be less prominent.

Another direction for future improvement lies in the calculation of initial matchings for consecutive pairs of frames. Our method assumes that we are given an initial assignment of features between the former and the latter frames which is then updated to obtain the optimal number of active labels. As we noted earlier, this assumption is viable since existing methods can be efficiently utilized to obtain such a mapping. However, it would be interesting to update an existing initial mapping between the previous two frames to obtain an initial mapping for the next frame. This, in turn, translates to making dynamic updates in an existing matching as the underlying topology changes. Developing an algorithm along the lines of Goemans and Williamson's primal-dual method [10] for obtaining dynamic matching is a promising direction for future study. Proposed method provides an online segmentation in that it makes use of the motion information obtained so far in a video sequence to conclude about the clustering of features in a new frame. Investigating the possibility of an optimization formulation that calculates the segmentation over the entire video sequence is another promising direction for future study. We did not address how to handle the noise and occlusion of object in this study which requires further investigation.

References

1. Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *Proceedings of the* 37^{th} *Annual Symposium on Foundations of Computer Science*, FOCS '96, pages 184–, Washington, DC, USA, 1996. IEEE Computer Society.
2. Yair Bartal. On approximating arbitrary metrices by tree metrics. In *Proceedings of the* 30^{th} *Annual ACM Symposium on Theory of Computing*, STOC '98, pages 161–168, New York, NY, USA, 1998. ACM.
3. Moses Charikar, Chandra Chekuri, Ashish Goel, Sudipto Guha, and Serge Plotkin. Approximating a finite metric by a small number of tree metrics. In *Proceedings of the* 39^{th} *Annual Symposium on Foundations of Computer Science*, FOCS '98, pages 379–388, Washington, DC, USA, 1998. IEEE Computer Society.
4. Andrea Colombari, Andrea Fusiello, and Vittorio Murino. Segmentation and tracking of multiple video objects. *Pattern Recognition*, 40(4):1307–1317, 2007.
5. E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiway cuts (extended abstract). In *Proceedings of the* 24^{th} *Annual ACM Symposium on Theory of Computing*, STOC '92, pages 241–251, New York, NY, USA, 1992. ACM.
6. M. Fatih Demirci, Yusuf Osmanlioglu, Ali Shokoufandeh, and Sven Dickinson. Efficient many-to-many feature matching under the l1 norm. *Comput. Vis. Image Underst.*, 115(7):976–983, July 2011.
7. endlessreference. Animation reference - athletic male standard walk - realtime. https://www.youtube.com/watch?v=GBkJY86tZRE. Accessed:2015-01-12.
8. Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the* 35^{th} *Annual ACM Symposium on Theory of Computing*, STOC '03, pages 448–455, New York, NY, USA, 2003. ACM.
9. Fernando Flores-Mangas and Allan D Jepson. Fast rigid motion segmentation via incrementally-complex local models. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2259–2266. IEEE, 2013.
10. Michel X. Goemans and David P. Williamson. The primal-dual method for approximation algorithms and its application to network design problems. In Dorit S. Hochbaum, editor, *Approximation Algorithms for NP-hard Problems*, pages 144–191. PWS Publishing Co., Boston, MA, USA, 1997.
11. Carme Julia, Angel Sappa, Felipe Lumbreras, Joan Serrat, and Antonio López. Motion segmentation from feature trajectories with missing data. In *Pattern Recognition and Image Analysis*, pages 483–490. Springer, 2007.
12. Richard M Karp. A 2k-competitive algorithm for the circle. *Manuscript, August*, 5, 1989.
13. Alexander V. Karzanov. Minimum 0-extensions of graph metrics. *Eur. J. Comb.*, 19(1):71–101, January 1998.
14. Jon Kleinberg and Éva Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. *J. ACM*, 49(5):616–639, September 2002.
15. Mingqi Kong, J-P Leduc, Bijoy K Ghosh, and Victor M Wickerhauser. Spatiotemporal continuous wavelet transforms for motion-based segmentation in real image sequences. In *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, volume 2, pages 662–666. IEEE, 1998.

16. Goran Konjevod, R Ravi, and F Sibel Salman. On approximating planar metrics by tree metrics. *Information Processing Letters*, 80(4):213–219, 2001.
17. M Pawan Kumar, Philip HS Torr, and Andrew Zisserman. Learning layered motion segmentations of video. *International Journal of Computer Vision*, 76(3):301–319, 2008.
18. Renjie Li, Songyu Yu, and Xiaokang Yang. Efficient spatio-temporal segmentation for extracting moving objects in video sequences. *Consumer Electronics, IEEE Transactions on*, 53(3):1161–1167, 2007.
19. Yusuf Osmanlıoğlu and Ali Shokoufandeh. Multi-layer tree matching using hsts. In *Proceedings of the* 3^{rd} *IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition*, GbR '15, 2015.
20. Yogesh Rathi, Namrata Vaswani, Allen Tannenbaum, and Anthony Yezzi. Tracking deforming objects using particle filtering for geometric active contours. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(8):1470–1475, 2007.
21. Huanfeng Shen, Liangpei Zhang, Bo Huang, and Pingxiang Li. A map approach for joint motion estimation, segmentation, and super resolution. *Image Processing, IEEE Transactions on*, 16(2):479–490, 2007.
22. Rustam Stolkin, Alistair Greig, Mark Hodgetts, and John Gilby. An em/e-mrf algorithm for adaptive model based tracking in extremely poor visibility. *Image and Vision Computing*, 26(4):480–495, 2008.
23. Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.
24. Vijay V Vazirani. *Approximation algorithms.* Springer Science & Business Media, 2013.
25. Jing Zhang, Fanhuai Shi, Jianhua Wang, and Yuncai Liu. 3d motion segmentation from straight-line optical flow. In *Multimedia Content Analysis and Mining*, pages 85–94. Springer, 2007.