Video Segmentation by Tracing Discontinuities in a Trajectory Embedding

Katerina Fragkiadaki
University of Pennsylvania
3330 Walnut street
katef@seas.upenn.edu

Geng Zhang
Xi'an Jiaotong University
28 West Xianning Road
zhangtetsu@aiar.xjtu.edu.cn

Jianbo Shi
University of Pennsylvania
3330 Walnut street
jshi@cis.upenn.edu

Abstract

Our goal is to segment a video sequence into moving objects and the world scene. In recent work, spectral embedding of point trajectories based on 2D motion cues accumulated from their lifespans, has shown to outperform factorization and per frame segmentation methods for video segmentation. The scale and kinematic nature of the moving objects and the background scene determine how close or far apart trajectories are placed in the spectral embedding. Such density variations may confuse clustering algorithms, causing over-fragmentation of object interiors. Therefore, instead of clustering in the spectral embedding, we propose detecting discontinuities of embedding density between spatially neighboring trajectories. Detected discontinuities are strong indicators of object boundaries and thus valuable for video segmentation. We propose a novel embedding discretization process that recovers from over-fragmentations by merging clusters according to discontinuity evidence along inter-cluster boundaries. For segmenting articulated objects, we combine motion grouping cues with a center-surround saliency operation, resulting in "context-aware", spatially coherent, saliency maps. Figure-ground segmentation obtained from saliency thresholding, provides object connectedness constraints that alter motion based trajectory affinities, by keeping articulated parts together and separating disconnected in time objects. Finally, we introduce Gabriel graphs as effective per frame superpixel maps for converting trajectory clustering to dense image segmentation. Gabriel edges bridge large contour gaps via geometric reasoning without over-segmenting coherent image regions. We present experimental results of our method that outperform the state-of-the-art in challenging motion segmentation datasets.

1. Introduction

The goal of this work is to segment a video sequence into moving objects and the world scene. Motion, as the gestaltic principle of "common fate" suggests, is a strong perceptual cue for video segmentation [24]. In order to take advantage of motion information available in multiple frames, many recent video segmentation approaches use point trajectories. Multi-body factorization methods [5, 25, 15] cluster trajectories by reasoning about relationships between the corresponding trajectory motion subspaces. These works extend the low rank constraint on the trajectory matrix proposed in [22], under assumptions about 3D object deformation and camera projection. In contrast, works of [4, 7, 3, 8] cluster trajectories directly from similarities of their 2D motion profiles, without modelling the camera projection process. In recent work, trajectory spectral clustering computed from 2D motion information has shown to outperform factorization methods and per frame segmentation approaches [4]. The spectral embedding is obtained by the top K eigenvectors of a normalized affinity matrix, where pairwise affinities reflect motion similarity between the corresponding point trajectories.

Figure 1. *Segmentation by tracing discontinuities.* (a)A trajectory spectral embedding has varying density, depending on the scale and kinematic nature of the objects captured by the embedded trajectories. (b) Density discontinuities (shown in red) between spatially neighboring trajectories are strong indications of object boundaries. c) Video segmentation by discontinuity thresholding.

Determining the number of objects K automatically and computing a corresponding clustering (discretization) of the trajectory embedding has turned out to be a nuisance even under rigid body motions [4]. In this paper, we show that such difficulties stem from the scale variation of moving objects, that cause different corresponding densities in the embedding space. Articulated motion poses additional challenges to motion based trajectory clustering; articulated body parts may move distinctly while separate agents may move similarly, resulting in a difficult trade-off of body over-fragmentation versus cross-object leakage in

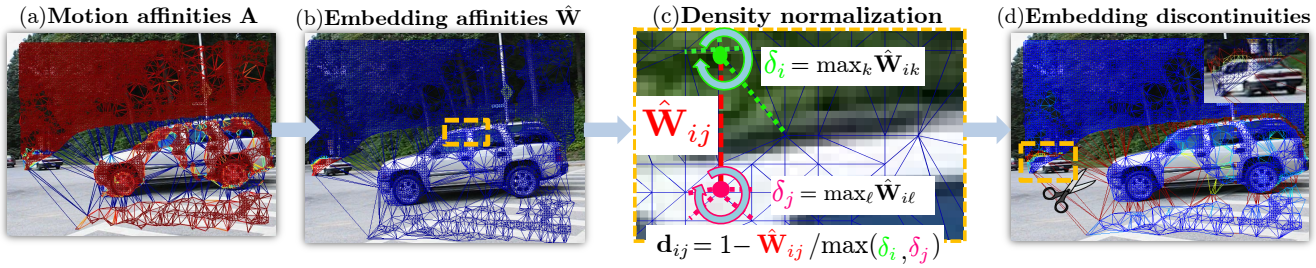

Figure 2. *Embedding discontinuity detector.* (a) Motion affinities $\mathbf{A}$ "break" inside the large car. (b) The corresponding embedding affinities $\hat{\mathbf{W}}_{ij}$ are smoother but vary according to the object scale and motion: we have high affinities $\hat{\mathbf{W}}_{ij}$ (red) on the small car interior and lower on the background or the larger car. (d) Our discontinuity detector adapts locally to the embedding density and outputs high discontinuity values across all object boundaries and low at object interiors.

video segmentation.

To deal with the limitations above, we propose an embedding discontinuity detector for localizing object boundaries in trajectory spectral embeddings. Instead of clustering, we detect sudden drops or peaks (discontinuities) of the embedding density, where density quantifies how close or far apart trajectories are placed in the embedding. We show that embedding discontinuities are strong indicators of object boundaries (see Figure 1). Detected discontinuities are incorporated in a novel embedding discretization process, that recovers from over-fragmentations by merging across inter-cluster boundaries that have weak discontinuity support. The proposed discretization is robust to the number of eigenvectors K, and has controlled over-segmentation error in contrast to previous approaches.

To deal with the challenges of articulated motion, we complement motion cues with topological information. Recent work has shown that video figure-ground segmentation can provide semantic object connectedness constraints on point trajectories, for distinguishing object articulation versus object separation in video segmentation [8]. Our contribution lies in combining grouping information of a trajectory embedding with a standard center-surround filter for spatially and temporally coherent video saliency. We call this "context-aware" saliency. The center-surround context-aware filter essentially needs to label each (trajectory) group as salient or not salient, rather than discovering the precise extent of salient foreground, thus bypassing the hard scale selection problem [9]. Then, object connectedness constraints from thresholded saliency maps modify the motion based trajectory affinities by canceling attraction between trajectories that violate object connectedness.

Finally, we introduce constrained Gabriel graphs as effective per frame superpixel maps for converting trajectory clustering to dense pixel-wise segmentation. Gabriel graph construction converts a contour map to a set of closed regions by "bridging" contour gaps via geometric reasoning. In this way, region leakage is prevented without thresholding the image boundary map too low. Furthermore, resulting superpixels adapt to the complexity of the input contour map, i.e., they are larger in textureless areas and smaller in textured ones. We obtain a dense video segmentation by graph cuts on the Gabriel superpixels of all video frames.

We present quantitative and qualitative results of our method that outperform previous approaches on established segmentation datasets. Further, we systematically evaluate the various components of our system in isolation and demonstrate their individual contribution.

2. Embedding Discontinuity Detector

Work on perceptual organization in static images suggests *feature discontinuity* to be of equal importance as feature similarity for segmentation. In the video domain, motion boundary detectors seek motion discontinuities by detecting edges where motion cues aggregated from adjacent regions change abruptly [19, 21]. However, when the regions are too small for the computed cues to be reliable, spurious boundaries are detected. Also, body deformations may give rise to many interior boundaries, not corresponding to objects.

In this work, we propose an embedding discontinuity detector for localizing object boundaries by detecting density discontinuities in a trajectory spectral embedding. Acting on trajectories rather than pixels, our detector benefits from long range motion cues. Acting on the embedding rather than the initial motion space, it benefits from global propagation of motion information, avoiding spurious motion dissimilarities caused by body deformation. In Section 2.1 we present our trajectory spectral embedding, in Section 2.2 our embedding discontinuity detector and in Section 2.3 a discontinuity-aware discretization process, that recovers from over-fragmentations by exploiting detected discontinuities.

2.1. Trajectory Spectral Embedding

We define a point trajectory tr_i to be a sequence of points:

$$\mathrm{tr}_i = \{(x_i^k, y_i^k, t_i^k),\ \ k = 1 \cdots T_i\},\ \ i = 1 \cdots n,$$

where T_i the length of tr_i and n the number of trajectories. We obtain point trajectories by tracking densely using optical flow [20]. Between each pair of trajectories tr_i and tr_j we set affinities $\mathbf{A}_{ij}$ measuring their motion similarity by penalizing their maximum velocity difference, following [4]. We compute the spectral embedding given by the top K eigenvectors of the normalized affinity matrix $\mathbf{P} = \mathbf{D}^{-1}\mathbf{A}$,

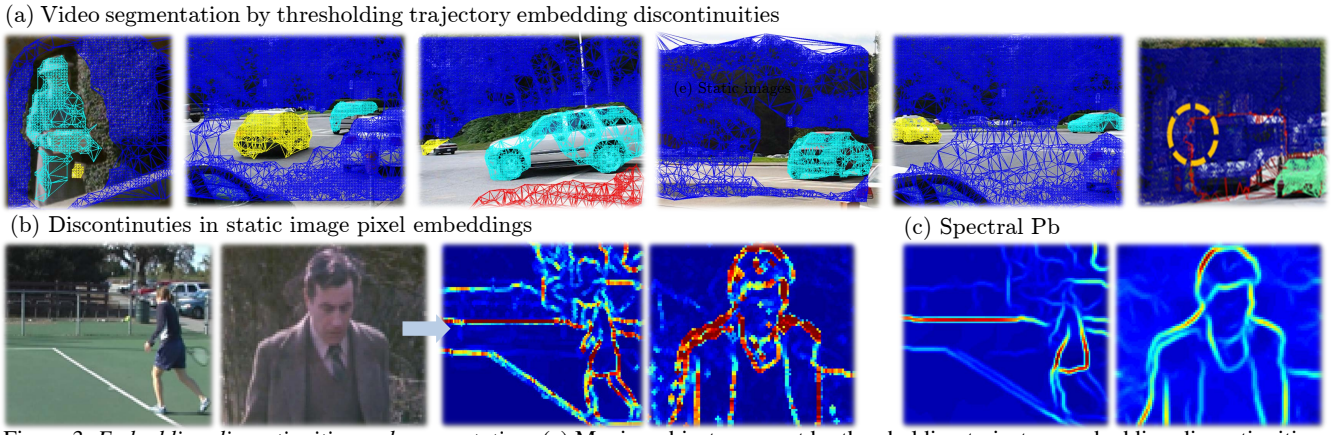

Figure 3. *Embedding discontinuities and segmentation.* (a) Moving objects pop out by thresholding trajectory embedding discontinuities. In last column, drifting trajectories in the yellow circle locally confuse the embedding. (b) Application of our discontinuity detection on spectral embedding computed from static pixel affinities (c) Comparison with spectral Pb.

where $\mathbf{D}$ is the degree diagonal matrix, $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{i,j}$ [18]. The embedding given by the top three non-trivial eigenvectors of $\mathbf{P}$ is visualized in Figure 4 (a). We define embedding affinities $\hat{\mathbf{W}} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$, with $\mathbf{V} \in \mathbb{R}^{n \times K}$ being the eigenvectors and $\mathbf{\Lambda}$ the diagonal matrix of the corresponding eigenvalues of $\mathbf{P}$. Embedding affinities $\hat{\mathbf{W}}$ are visualized in Figure 2 (b).

Trajectories are embedded as lines rather than spherical clusters, as shown in Figure 4 (a). That is because optical flow measurements change smoothly along the object surface, rather than forming compact clusters. The scale and kinematic nature of the moving objects and the background scene determine the density with which corresponding trajectories are placed in the embedding space. Specifically, the smaller an object and the further it is from the camera, the more compactly embedded it is. In this case, all its point trajectories have similar rather than smoothly changing motion measurements (approximated by a translation rather than affine model) and thus very strong affinities between them. This is illustrated in Figure 2 (b): embedding affinities $\hat{\mathbf{W}}$ are high on the small car and much lower on the background or the larger car.

2.2. Embedding Discontinuities

Our main insight is that detecting motion discontinuities is easier than finding semantic motion clusters, since clustering in the embedding space may be confused by density variations. We define embedding discontinuities as sudden drops or peaks of the embedding affinities $\hat{\mathbf{W}}$. Mapping of trajectory discontinuities to dense pixel-wise region boundaries will be discussed in Section 4.

Spatially neighboring trajectory points in each frame are candidate places for motion embedding discontinuities. In each frame t, we capture neighborhood relations among trajectories with a Delaunay triangulation graph $\mathcal{D}^t$ built *on the trajectory points* of that frame (see Figure 2). By definition of Delaunay triangulation, three trajectory points are connected with triangulation edges if no other point is contained in the circumcircle of their triangle. Each $\mathcal{D}^t$ is a planar graph on trajectory points of frame t, with Delaunay edges e_{ij}^t spanning spatially neighboring trajectories tr_i and tr_j of that frame. For each trajectory tr_i, we define $\mathcal{N}_{xy}^i$ to be the set of neighboring trajectories in the Delaunay triangulation graph of any frame:

$$\mathcal{N}_{xy}^i = \{j, \text{ s.t. } \exists t, 1 \leq t \leq T, \ \ \mathrm{e}_{i,j}^t = 1\},$$

where T denotes the total number of frames. For each trajectory tr_i, we define density δ_i to be the maximum embedding affinity to its Delaunay neighbors:

$$\delta_i = \max_{j \in \mathcal{N}_{xy}^i} \hat{\mathbf{W}}_{ij}.$$

Trajectory densities quantify locally the density of the trajectory embedding. They are high when a trajectory is close in embedding distance to at least one of its spatial neighbors (e.g. interior of the small car in Figure 2 (b)) and low for loosely embedded trajectories (e.g. background scene or interior of the larger car in Figure 2 (b)).

For each pair of spatially neighboring trajectories tr_i, tr_j, we define the density discontinuity $\mathbf{d}_{ij}$ to be:

$$\mathbf{d}_{ij} = \begin{cases} 1 - \hat{\mathbf{W}}_{ij} \frac{1}{\max(\delta_i, \delta_j)}, & \text{if } j \in \mathcal{N}_{xy}^i \\ 0, & \text{otherwise} \end{cases}.$$

Density discontinuities capture sudden peaks or drops of embedding densities (it is a peak when traversing the edge in one direction and drop in the opposite direction). They provide a strong indication of object boundaries, as shown in Figure 3 (a): thresholding embedding discontinuities provides the desirable trajectory clustering results. Empirically we found 0.6 to be a suitable threshold.

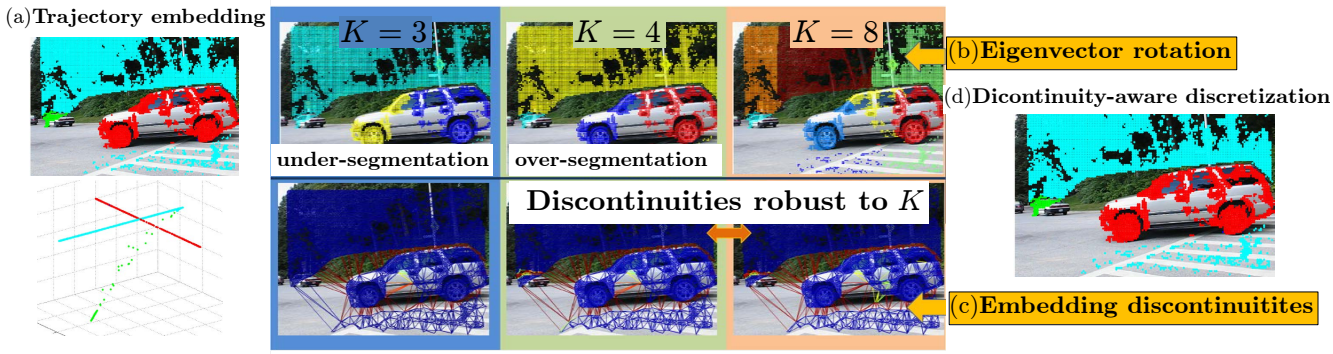

Figure 4. *Discontinuity-aware discretization.* Trajectory embedding discontinuities (shown in red in (c)) are robust to the number of eigenvectors K, in contrast to clustering based discretizations (b). Merging adjacent clusters with no local indication of discontinuity along their boundary recovers from artificial over-fragmentations (d). Notice that there is not a "right" eigenvector number K that would produce the same result: choosing $K = 3$ results in under-segmentation while $K = 4$ in over-segmentation.

The proposed discontinuity detector is not limited to trajectory embeddings. In Figure 3 (b) we show its application to an image pixel embedding, where input pixel affinities $\mathbf{A}_{ij}$ are computed from static image cues ([10]). For each pixel i, the neighborhood $\mathcal{N}^i_{xy}$ is the set of four pixels around it. Image boundaries are captured by embedding density discontinuities, as shown in Figure 2 (g). In contrast to Spectral Pb [10], our discontinuity detector does not involve any feature weight learning. Instead, it relies on the intrinsic variations of embedding density. Furthermore, in our approach, the embedded elements are not required to reside on a regular grid in the input space. Instead, neighborhood relationships are captured via triangulation.

2.3. Discontinuity-Aware Discretization

In previous work, there are two popular methods for discretizing a spectral embedding: K-means clustering (K-**means**) with embedding distances [18] and eigenvector rotation (**rot**) [26]. As the number of eigenvectors K varies, both methods may break large coherent regions into chunks [1]. More interestingly, as shown in Figure 4, there may not be an ideal K: for $K = 3$ the large car is over-fragmented before the small car is delineated from its surroundings.

We propose a discontinuity-aware discretization that merges clusters whose inter-cluster boundary is not supported by embedding discontinuity evidence. We use eigenvector rotation (**rot**) to obtain an initial trajectory over-segmentation. For each pair of spatially neighboring trajectory clusters $\mathrm{C}_p, \mathrm{C}_q$, we define their inter-cluster discontinuity $\mathbf{d}^{\mathrm{C}}_{pq}$ to be:

$$\mathbf{d}^{\mathrm{C}}_{pq} = \frac{\sum_{\mathrm{tr}_i \in \mathrm{C}_p,\, \mathrm{tr}_j \in \mathrm{C}_q} \mathbf{d}_{ij}}{|\{(i,j),\ \mathrm{tr}_i \in \mathrm{C}_p,\ \mathrm{tr}_j \in \mathrm{C}_q,\ j \in \mathcal{N}^i_{xy}\}|}.$$

To recover from artificial fragmentations, we merge clusters whose inter-cluster discontinuities $\mathbf{d}^{\mathrm{C}}$ are below ρ. We found empirically $\rho = 0.4$ to be a suitable threshold.

3. Context-Aware Trajectory Saliency

Motion information alone is often insufficient for segmenting articulated bodies since motion discontinuities may exist both across distinctly moving articulated parts of the same object as well as across objects. This is illustrated in Figure 7 (b) where the human body is over-fragmented (in torso and legs) while at the same time segmentation leaks across similarly moving agents. Recently, authors of [8] complemented motion trajectory affinities in $\mathbf{A}$ by setting repulsive weights between trajectories violating object connectedness constraints. Two trajectories violate object connectedness if at any point during their time overlap, they belong to two different connected components of the video foreground. In this way, figure-ground video segmentation provides semantic information that is valuable for untangling the articulated agents.

Center-surround filtering on per frame flow magnitude has been used by numerous works for spatio-temporal figure-ground segmentation [9]. Our contribution lies in coupling the center-surround saliency computation with the trajectory embedding. In each video frame t, we compute a pixel-wise center-surround saliency map $\mathbf{S}^t$ using the publicly available code of [14] (Figure 5 (b)). For each trajectory tr_i, we compute trajectory saliency $\mathbf{s}_i$ as the *maximum* of the saliencies of its points: $\mathbf{s}_i = \max_{1 \le k \le T_i} \mathbf{S}^{t^k_i}(x^k_i, y^k_i)$. This propagates saliency in time and assigns an object as salient even at frames it is stationary [8]. For each trajectory tr_i, we define context-aware trajectory saliency $\bar{\mathbf{s}}_i$ as the average of trajectory saliencies $\mathbf{s}_j$ in its *embedding* neighborhood $\mathcal{N}^i_{\mathbf{s}}$:

$$\bar{\mathbf{s}}_i = \tfrac{1}{|\mathcal{N}^i_{\mathbf{s}}|} \textstyle\sum_{j \in \mathcal{N}^i_{\mathbf{s}}} \mathbf{s}_j,$$

where $\mathcal{N}^i_{\mathbf{s}} = \{j, \text{s.t. } \max(\frac{\hat{\mathbf{W}}_{ij}}{\delta_i}, \frac{\hat{\mathbf{W}}_{ij}}{\delta_j}) < \ell\}$ and ℓ is a threshold controlling the neighborhood size.

The above operation smooths saliency information across closely embedded trajectories. As a result, context-aware trajectory saliency is space and time coherent; it re-

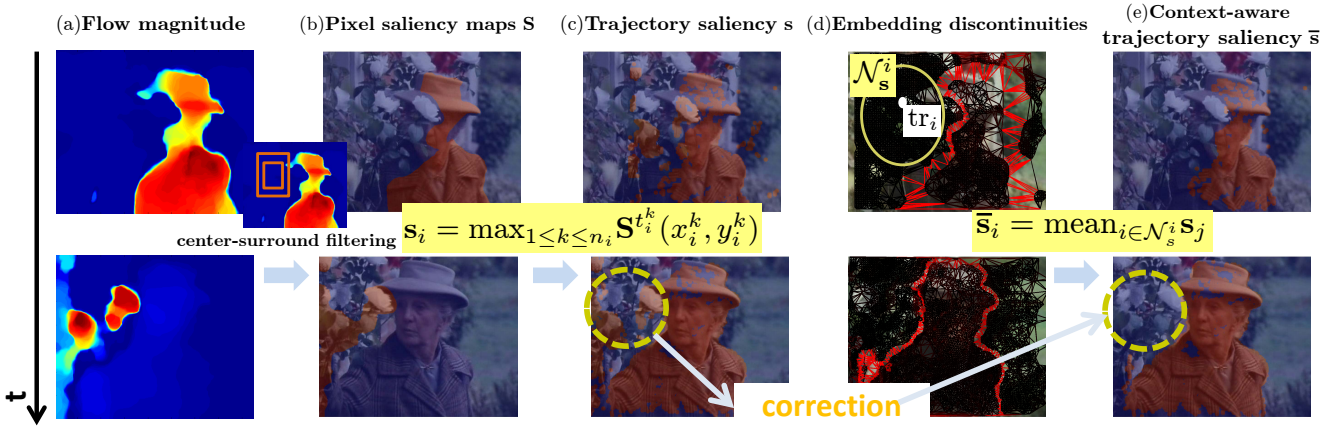

Figure 5. *Context-aware trajectory saliency.* Thresholding of trajectory saliencies $\mathbf{s}$ erroneously isolates the flowers from the bush in (c). In contrast, context-aware trajectory saliency $\bar{\mathbf{s}}$ in (e) has less noise and better spatial coherence thanks to the employed grouping cues. For ease of visualization, we show thresholding of the saliency maps at 0.5 rather than their initial values.

covers from the noise of center-surround filtering thanks to long range grouping constraints, as shown in Figure 5 (e).

By thresholding saliencies $\bar{\mathbf{s}}$ at 0.5, trajectories are classified as foreground and background. Foreground trajectories are shown in red in Figure 7 (a) top row. We then segment using object connectedness by setting to zero affinities between trajectories belonging to distinct connected components of the foreground, as shown in second row of Figure 7. The final trajectory clustering is obtained by discretizing the motion and topology embedding with the method of Section 2.3.

4. Trajectory Clustering to Pixel Boundaries

To obtain a dense video segmentation we convert trajectory clusters to image regions. Recently, authors of [13] used a superpixel hierarchy in a variational framework for trajectory to superpixel region mapping. In this paper, we propose constrained Gabriel graphs as per frame superpixel region maps and compute a dense video segmentation by graph cuts on Gabriel superpixels. We describe constrained Gabriel graph construction in Section 4.1 and Gabriel superpixel labelling in Section 4.2.

4.1. Contours to Regions via Gabriel Graphs

We introduce constrained Gabriel graphs, a novel to the vision community representation, for converting locally detected contours to a set of closed regions in an image. We define a constrained Gabriel graph as the subset of the corresponding constrained Delaunay triangulation (CDT) after deleting edges violating the Gabriel property [11], i.e., edges whose circumcircle encloses other input points. Given a set of line segments, fitted to the image thresholded Pb, a CDT is a variant of the Delaunay triangulation for which the input line segments are constrained to lie in the triangulation. CDT has been used in computer vision for contour completion [17, 16]; however, CDT can contain arbitrarily thin triangles. As such, it has not been popular as a superpixel graph. As a result of edge deletion, Gabriel superpixels are no longer necessarily triangles; they are rounder, since edges of "thin" triangles of CDT are likely to violate the Gabriel property. A CDT and the corresponding constrained Gabriel graph are shown in Figure 8.

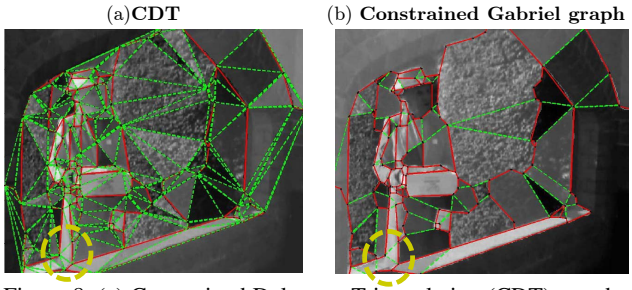

Figure 8. (a) Constrained Delaunay Triangulation (CDT) on edge line segments. Constraining line segments are shown in red and added Delaunay edges in green. (b) Constrained Gabriel graph. Gabriel superpixels are rounder and larger than Delaunay triangles, while Gabriel edges still bridge large contour gaps of missing or faint contours (yellow circle).

A constrained Gabriel graph has a number of desirable properties as a superpixel region graph. First, Gabriel edges bridge faint or missing contours based on *geometric reasoning* rather than image intensity. In this way, region leakage is prevented without thresholding the image boundary map (e.g. Pb) too low. In contrast, most approaches on superpixel segmentation cannot bridge large faint contour gaps without resorting to an overwhelming over-segmentation of the image [12, 6]. Second, Gabriel superpixels adapt to the complexity of the input contour map, i.e., they are larger in textureless areas and smaller in textured ones. Third, constrained Gabriel graph construction is efficient, it can be computed in linear time given the corresponding CDT.

4.2. Gabriel Cut

Trajectory clustering induces a labelling on the set of Gabriel superpixels in the video sequence. Let $\mathcal{R}$ denote

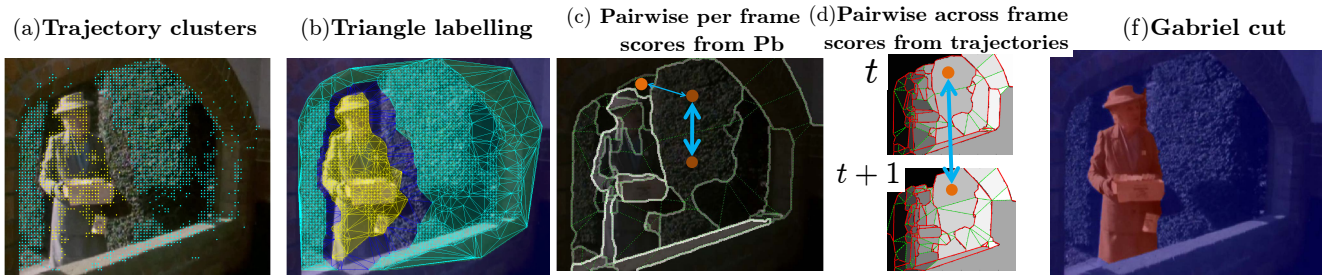

Figure 6. *From trajectories to regions.* (a)Dragging effect. Notice the yellow trajectories residing on the background, above woman's shoulder. In (f), Gabriel cut correctly labels such pixels on oversmoothed foreground boundaries. At the same time, it propagates information to untextured image regions that are sparsely populated with trajectories. In (c-d) the weight of each arrow indicates smoothness cost between the corresponding Gabriel superpixels: the larger the weight the higher the penalty for label disagreement.

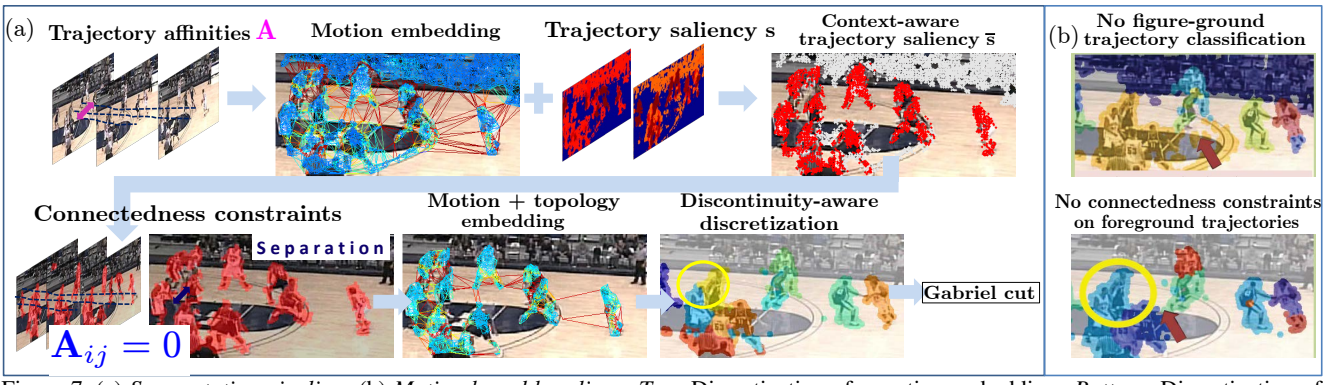

Figure 7. (a) *Segmentation pipeline.* (b) *Motion-based baselines. Top:* Discretization of a motion embedding. *Bottom:* Discretization of a motion embedding of foreground only trajectories without connectedness constraints. Under articulation, motion segmentation leaks to the background or across agents. After cancellations of affinities $\mathbf{A}$ between trajectories violating object connectedness, the two agents in the yellow circle are correctly separated while the player holding the ball is no longer over-fragmented.

the superpixel set and L the number of possible labels. We consider a pairwise MRF on $\mathcal{R}$. For each superpixel r, unary costs are set according to the normalized histogram of trajectory labels that intersect its interior, denoted by $\mathrm{h}_r \in [0,1]^{L\times 1}$. For each pair of spatially adjacent superpixels r_p, r_q, pairwise costs are set according to mean Pb along their common boundary, denoted by $\overline{\mathrm{pb}}_{p,q}$. Finally, for each pair of temporally adjacent superpixels r_p, r_q, pairwise costs are set according to the fraction of their common trajectories divided by the maximum number of trajectories intersecting either one of them and denoted by $\overline{\mathrm{fl}}_{p,q} \in [0,1]$. We compute a labelling f that minimizes the energy:

$$\text{min. } E(f) = \sum_{r\in\mathcal{R}} \underbrace{\lambda_r \cdot (1 - \mathrm{h}_r(f_r))}_{unary} + \underbrace{\sum_{r_p \sim r_q} \mathbf{1}_{f_p \neq f_q} \overline{\mathrm{pb}}_{p,q} + \sum_{r_p \backsim r_q} \mathbf{1}_{f_p \neq f_q} \overline{\mathrm{fl}}_{p,q}}_{pairwise},$$

where $\sim$ denotes spatial adjacency, $\backsim$ denotes temporal adjacency, $\mathbf{1}$ is the delta function and λ_r is a weight on the unary term of each superpixel r.

Unary costs computed from superpixels at object interiors are more reliable than those computed from superpixels close to object boundaries due to the "dragging effect" of optical flow, visualized in Figure 6. We identify unreliable superpixels by converting trajectory labels to triangle labels in the Delaunay graphs $\mathcal{D}^t, t = 1 \cdots T$, built on per frame trajectory points. In Figure 6 (b) we show in yellow and light blue, Delaunay triangles whose vertices share the same trajectory cluster label. Such triangles are likely to capture object interiors. In the same Figure, we show in blue, triangles whose vertices do not agree on their trajectory labels. They are likely to capture inter-object space and be susceptible to dragging. In practice, for superpixels with more than 30% intersection with blue (ambiguous) area we set the corresponding λ_r weights to zero, encouraging the smoothing pairwise costs to dominate their labelling. The Gabriel superpixel labelling is computed via graph cuts [2] and is visualized in Figure 6 (f).

5. Experiments

We test our method on Moseg and Figment segmentation datasets. Moseg (*Mo*tion *seg*mentation) [4] is a publicly available dataset which contains objects of various scales under mostly rigid motions. We use the trajectories and the evaluation software delivered with the dataset. We discard trajectories shorter than seven frames. We test on the first 50 frames in each sequence (when the sequence has less than 50 frames we use the whole sequence). First, we

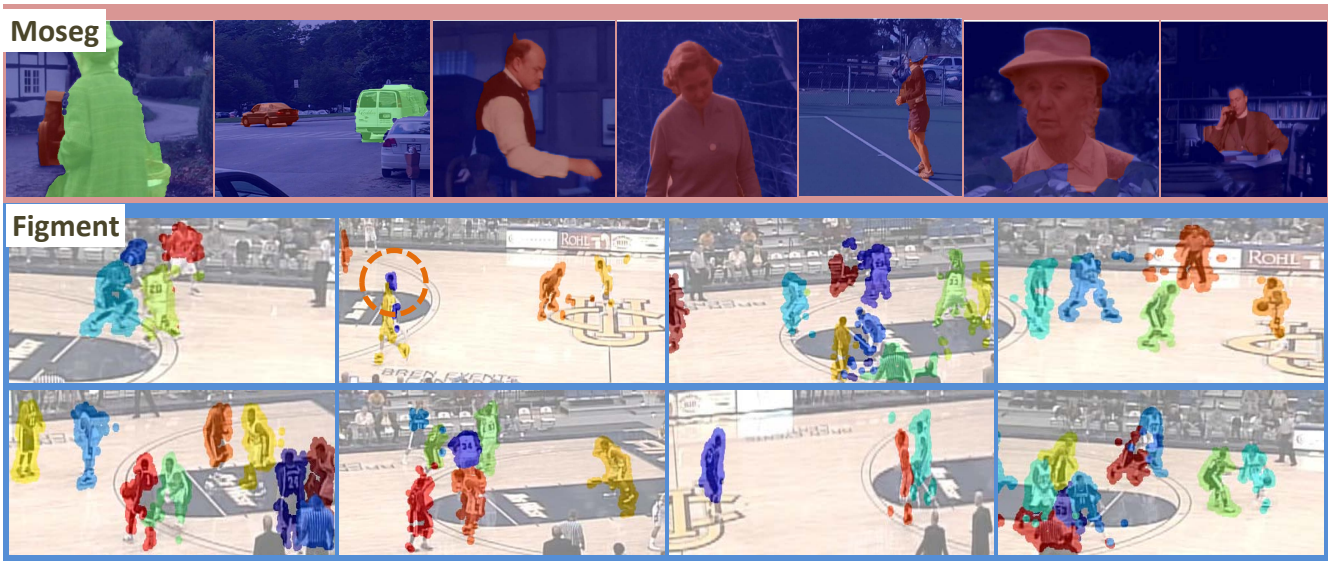

Figure 9. *Top row:* Experiments on Moseg dataset. We correctly segment objects of various scales. *Bottom row:* Experiments on Figment dataset. Due to the low resolution of Figment dataset, we only show dilated trajectory points rather than pixel segmentation. Notice the two players inside the orange circle: our model can find the right spatial support of objects under persistent partial occlusions.

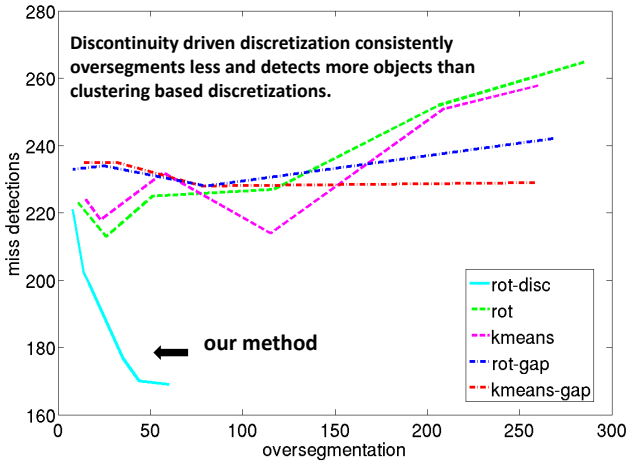

Figure 10. *Discretization evaluation.*

evaluate our discontinuity-aware discretization (**rot-disc**) in isolation, and compare with four other discretization algorithms: K**-means** and **rot** with K selected by thresholding eigenvalues, K**-means-gap** and **rot-gap** with K selected by thresholding consecutive eigenvalue gap. In Figure 10 we plot the average across sequences over-segmentation error (i.e. the number of interior fragmentations not corresponding to object boundaries) against the average miss detection error (i.e. the number of groundtruth objects or world scene that were not matched to a cluster with intersection over union score above 70%), as we vary the thresholds of the various algorithms. Our method outperforms standard discretizations, it has considerably smaller over-segmentation error for the same miss-detection error.

Second, we evaluate our segmentation pipeline, with connectedness constraints (*our method*) as well as without (*our method w/o FG*). We evaluate both trajectory clustering as well as dense pixel segmentation. We use trim mean to average results across sequences where we reject the top and bottom 10 $\%$ of the measurements. Results are shown in Table 2 and in Figure 9. Our approach, both with and w/o connectedness constraints, outperforms previous approaches. Pixel segmentation has increased error in comparison to trajectory clustering. This is due to possible erroneous segment labels in absence of trajectories. Increasing the minimum allowed trajectory length would provide better trajectory to regions mapping results but may cause errors due to accidental similarities of short trajectories.

Figment (*Fig*ure untangle*ment*) dataset contains 18 video sequences of 50-80 frames each, with scenes from a basketball game [23]. For each sequence, all players and the background scene are labelled every seven frames. For evaluation, each trajectory cluster is optimally assigned to one groundtruth object based on maximum intersection. Given this assignment, *clustering error* measures for each sequence the percentage of wrong pixels, i.e., pixels overlapping with a trajectory cluster not assigned to their labelled object). *Per region clustering error* measures percentage of wrong pixels per groundtruth object. Please refer to [8] for explanation of the rest of the metrics. We show results in Table 1 and in Figure 9. In contrast to the Moseg dataset case, where the gain from the use of foreground topological information is small, under articulation and object deformation, connectedness constraints improve performance by a large margin. Additional results, videos and code are available at: `http://www.seas.upenn.edu/~katef/videoseg`.

6. Conclusion

We presented a novel density discontinuity detector applied on trajectory embedding affinities for detecting motion boundaries from long range motion cues. The pro-

Figment	density	clustering error	per region clustering error	over-segmentation	recall	leakage	tracking time
our method	**7.05**%	7.90%	**18.47**%	**1.5**	**33.28**%	19.55%	**82.29**%
our method w/o FG	4.90%	17.49%	41.06%	3.21	19.19%	44.96%	48.49%
Fragkiadaki et al [8]	5.21%	**4.73**%	20.32%	1.57	31.07%	**16.52**%	75.13%

Table 1. *Results in Figment.* Our method has lower *per region clustering error*, which is the essential metric that does not take into account the background, as *clustering error* does. Slightly higher object leakage is attributed to the increased density of our approach.

Moseg	density	clustering error	per region clustering error	over-segmentation	extracted objects
our method (trajectory clustering)	3.07%	**2.29**%	20.93%	0.29	**29**
our method w/o FG (traject. clustering)	3.15%	2.55%	**20.63**%	0.48	28
our method (pixel segmentation)	**93.72**%	3.95%	26.14%	**0.25**	26
Fragkiadaki et al. [8]	3.22%	3.76%	22.06%	1.15	25
Brox et al. [4]	3.32%	3.43%	27.06%	0.4	26

Table 2. Results in Moseg.

posed discontinuity-driven embedding discretization is robust to the number of eigenvectors chosen and recovers from over-fragmentations that occur in typical, clustering based, discretization algorithms. Further, we presented context-aware trajectory saliency for space and time coherent figure-ground video segmentation. It provides object connectedness constraints that modify the motion affinity graph for effectively segmenting articulated moving objects. Finally, we presented constrained Gabriel graphs as flexible per frame superpixel maps for converting trajectory clustering to dense pixel segmentation. We showed quantitative and qualitative results of our method, outperforming the state-of-the-art.

Acknowledgments The authors would like to thank Kosta Derpanis, Elena Bernardis, Weiyu Zhang and Ben Sapp for useful discussions on the writing of this paper.

References

[1] P. Arbelaez, M. Maire, C. C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *CVPR*, 2009.

[2] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *TPAMI*, 23, 2001.

[3] G. J. Brostow and R. Cipolla. Unsupervised Bayesian detection of independent motion in crowds. In *CVPR*, 2006.

[4] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*. 2010.

[5] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. *ICCV*, 1995.

[6] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59, 2004.

[7] M. Fradet, P. Robert, and P. Pérez. Clustering point trajectories with various life-spans. In *CVMP*, 2009.

[8] K. Fragkiadaki and J. Shi. Exploiting motion and topology for segmenting and tracking under entanglement. In *CVPR*, 2011.

[9] D. Gao, V. Mahadevan, and N. Vasconcelos. On the plausibility of the discriminant center-surround hypothesis for visual saliency. *Journal of vision*, 8, 2008.

[10] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. Using contours to detect and localize junctions in natural images. In *CVPR*, 2008.

[11] D. W. Matula and R. R. Sokal. Properties of Gabriel graphs relevant to geographic variation research and the clustering of points in the plane. *Geographical Analysis*, 12, 1980.

[12] G. Mori, X. Ren, A. A. Efros, and J. Malik. Recovering human body configurations: combining segmentation and recognition. In *CVPR*, 2004.

[13] P.Ochs and T.Brox. Object segmentation in video: a hierarchical variational approach for turning point trajectories into dense regions. In *ICCV*, 2011.

[14] E. Rahtu, J. Kannala, M. Salo, and J. Heikkil. Segmenting salient objects from images and videos. In *ECCV*. 2010.

[15] S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories. In *CVPR*, 2008.

[16] X. Ren, C. C. Fowlkes, and J. Malik. Mid-level cues improve boundary detection. Technical report, UC Berkeley, 2005.

[17] X. Ren, C. C. Fowlkes, and J. Malik. Scale-invariant contour completion using conditional random fields. In *ICCV*, 2005.

[18] J. Shi and J. Malik. Normalized cuts and image segmentation. *TPAMI*, 2000.

[19] A. Stein, D. Hoiem, and M. Hebert. Learning to find object boundaries using motion cues. In *ICCV*, 2007.

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

[21] P. Sundberg, T. Brox, M. Maire, P. Arbelaez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *CVPR*, 2011.

[22] C. Tomasi and T. Kanade. shape and motion from image streams: a factorization method. Technical report, IJCV, 1991.

[23] C. Vondrick, D. Ramanan, and D. Patterson. Efficiently scaling up video annotation with crowdsourced marketplaces. In *ECCV*, 2010.

[24] M. Wertheimer. Laws of organization in perceptual forms. *A Sourcebook of Gestalt Psycycholgy (Partial translation)*, 1938.

[25] J. Yan and M. Pollefeys. A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate. In *ECCV*, 2006.

[26] S. Yu and J. Shi. Multiclass spectral clustering. In *ICCV*, 2003.