Motion Segmentation by Consensus

Roberto Fraile, David C. Hogg and Anthony G. Cohn
School of Computing
The University of Leeds
United Kingdom
Email: rf@comp.leeds.ac.uk

***Abstract*—We present a method for merging multiple partitions into a single partition, by minimising the ratio of pairwise agreements and contradictions between the equivalence relations corresponding to the partitions. The number of equivalence classes is determined automatically. This method is advantageous when merging segmentations obtained independently. We propose using this consensus approach to merge segmentations of features tracked on video. Each segmentation is obtained by clustering on the basis of mean velocity during a particular time interval.**

I. Introduction

Motion is an important cue for object segmentation and finding regions of similar motion is an important step for computer vision applications such as object detection and tracking.

Pixels corresponding to objects in motion can be grouped on the basis of their appearance and relative motion [1]. Appearance models of dense features have been used for two-layer segmentation [2] (where each layer is a set of features that follow the same motion in the image plane) and multiple-layer segmentation [3]. Such global methods minimise an energy functional constructed from local appearance, imposing a constraint on the motion. For example, in [4], the constraint is given by a Hidden Markov Model of actions, alternating between segmentation and model fitting. Another constraint is that a higher derivative of the position of each feature is bounded, effectively requiring smooth motion. But dense optical flow is a highly redundant representation of motion. Feature trackers [5]–[8] effectively reduce the amount of motion to process to a few features chosen to maximise the stability of their appearance along time. Common fate can be used to group features tracked into objects.

Relative motion models have been used by [9], depending only on the position of tracked features, and not on their appearance. Groups of features were split and merged using an on-line update which assumed features from the same object undergo an affine transformation between frames, at the cost of sensitivity to tracking errors and density of features. Each feature group had an associated reference frame, and the affine transform was defined with respect to that frame. Features were segmented differently during initialisation and update. Such approaches will fail when there is a lack of support for affine grouping, for example when the geometric transformations involved are degenerate, or the features are short lived and there is not sufficient overlap in their support intervals.

In [10] a consensus of segmentations of an image's pixels was achieved by constructing a matrix that counted the number of times two elements were grouped together, effectively building a histogram of instances of an equivalence relation. Their approach included a statistical measure of consensus, in order to choose the number of clusters that achieved the best consensus. This approach for merging multiple segmentations into one can in principle be applied to any other kind of segmentation, including the segmentation of sparse features.

We propose an algorithm that takes feature trajectories and uses the mean velocity over different time intervals to produce a family of independent segmentations that capture motion at a variety of granularities. This is in contrast with [9], where groups of features were formed and destroyed on-line, in order to allow the detection of groups of objects moving at different velocities. Each of our segmentations corresponds to motion saliency at different time scales, and, because features are relatively short lived, segmentations are defined on overlapping sets of features.

The reason why time intervals of multiple lengths are considered is that sets of features which move together over short timescales may separate over larger timescales due to objects, parts of objects or groups of objects moving apart. In contrast, objects, parts or groups which remain together over a long timescale may exhibit different short-term motion.

A binary partition is then obtained for each time interval by applying the k-means algorithm, using Euclidean distance, to an estimated mean velocity. Multiple partial segmentations are then merged into a single segmentation by a consensus algorithm. Each segmentation is considered as an equivalence relation defined on a subset of features, over a particular time interval. Features which should fall in the same class but do not appear in the same segmentations, are linked through transitivity. For example, if features a and b appear in the same class in frame 100, and features b and c appear in the same class in frame 110, then features a and c should belong to the same class, unless there is a segmentation which separates them. The consensus algorithm approximates the simplest segmentation which does not group together features which were separated in any of the partitions. Although each partition has only two classes, the consensus algorithm generates as many classes as needed.

Section II describes a method to segment features on the basis of their mean velocity at a particular time interval. Section III presents the consensus method, which can be

applied to any kind of partial set partitions. The implied motion model is that an object is a maximal set of features that have a distinct velocity at different time scales, requiring no other geometric spatio-temporal constraints such as Euclidean or affine similarity. Experiments on videos of people walking in an underground station are used to illustrate the algorithm.

II. Segmentation of a time interval

Feature tracks are grouped together on the basis of their estimated mean velocity during a time interval. In this section we explain how a segmentation is produced from a number of such tracks, $\{\mathbf{p}_i\}$, given an interval $(t-\epsilon, t+\epsilon)$.

Each track consists of temporally and spatially contiguous coordinates. In practice, this is the position at time t, $\mathbf{p}(t)$, for a time interval. The mean velocity $\mathbf{v}_i(t,\epsilon)$ of each track $\mathbf{p}_i$ in the interval $(t-\epsilon, t+\epsilon)$ is computed by finite differences on the features' positions in frames $t-\epsilon$ and $t+\epsilon$.

$$\mathbf{v}_i(t,\epsilon) \approx \frac{\mathbf{p}_i(t+\epsilon) - \mathbf{p}_i(t-\epsilon)}{2\epsilon}$$

The velocity estimates $\{\mathbf{v}_i\}$ are partitioned into two sets using k-means on the Euclidean distance. In order to favour segmentations with the smallest number of groups, the parameter k in the k-means algorithm was set to 2, the number of groups will typically be greater than 2 after consensus.

Time intervals $(t-\epsilon, t+\epsilon)$ are varied in centre and width. Different interval centres t are chosen in order to contain overlapping sets of features, and different interval radii ϵ are chosen to measure the mean velocity of features over different scales. The choice of intervals will therefore depend on the lifespan of the trajectories produced by the feature tracker.

Each of those time intervals will yield a partial segmentation P_j which is not defined on the entire set of features but on a subset S_j. Features which belong to the same object in motion but do not appear in the same frame, should be grouped together by transitivity on their relation with common features. This is done by the consensus algorithm.

III. Consensus of segmentations

In this section we describe how the family of segmentations $\{P_j\}$ is merged into a single segmentation. In [10] the sum of adjacency matrices for all P_j was computed. We normalise this sum over the number of support sets S_j in which the two features in a pair are present. We will also simplify the choice of number of segments by introducing a threshold over the resulting matrix, and then computing an approximation of its transitivity closure [11].

Each segmentation P_j corresponds to an equivalence relation $\sim_j$ defined over the features S_j for which the segmentation is defined. Two elements a and b in S_j are related, and we denote it as $P_j(a) = P_j(b)$ if they belong to the same class in P_j. The purpose is to define a single equivalence relation $\sim$ over all the features $\cup_j S_j$, which is maximally consistent, in some sense, with the relations $\sim_j$.

A family of equivalence relations $\{\sim_j\}$ is consistent over two elements a and b when the ratio of number of relations i such that $a \sim_j b$ over the number of relations such that $a \not\sim_j b$ is above a threshold λ.

The criterion for relation consensus is to maximise the consistency of the merged relation with each of the contributing relations, and also maximise the number of relation instances in the merged relation (which corresponds to minimising the number of equivalence classes or segments).

a) Consensus algorithm: Given a set Ω, subsets $\{S_k\}_{k=1}^n$, and a partition P_k of each S_k, $P_k : S_k \to \{1, 2, \ldots\}$, produce a partition P of Ω that approximates the transitivity closure of thresholded ratios of pair-wise agreements over contradictions with each partition P_k:

1) define the matrix A as
$$a_{ij} = \mid \{k : i, j \in S_k, P_k(i) = P_k(j)\} \mid$$
2) define the matrix B as
$$b_{ij} = \mid \{k : i, j \in S_k, P_k(i) \neq P_k(j)\} \mid$$
3) for $0 < \lambda \leq 1$, define element-wise $C_\lambda = (\frac{A}{B} \geq \lambda)$ for a scalar threshold λ.
4) Compute possibly overlapping family of sets D, formed by the equivalence classes X_k defined in the following way: let $G_0 = \Omega$, by iterating for increasing k until G_k is empty:
   a) choose an element g_k from G_k.
   b) compute X_k as the equivalence class of g_k in Ω.
   c) define $G_{k+1} = G_k - X_k$
5) Compute P as the partition given by the the maximal non-overlapping subsets $E_k = G_k - \cup_{l \neq k} G_l$.

Note that this algorithm is sensitive to the choice of representatives g_k in step 4.a.

The partitions being defined over non-overlapping sets S_k, and therefore features which are not directly related by any particular segmentation might end up in the same equivalence class, this is achieved by computing the transitivity closure of the final relation (Step 4) which is enforced by grouping into sets X_k. But, in order to avoid one single misclassified feature to join two equivalence classes together, features which appear in two different classes are removed (Step 5). The threshold λ can be chosen, for example, to minimise the number of features dropped in this way.

IV. Experiments

A sequence from the i-LIDS dataset, made available for research by the UK Home Office [12], were processed using an implementation of the KLT feature tracker [5], [7], [8] with default settings, requesting 700 features. Affine checks were switched off in order to maximise the length of feature tracks.

The video was filmed using a public transport surveillance camera. It shows pedestrians walking on an underground platform and trains arriving and departing. Which regions were segmented out depended on the radius ϵ of the sampling interval $(t-\epsilon, t+\epsilon)$ around frame t, as shown on Figure 1. Applying the consensus algorithm to segmentations obtained with fixed t and increasing ϵ, produces a segmentation which

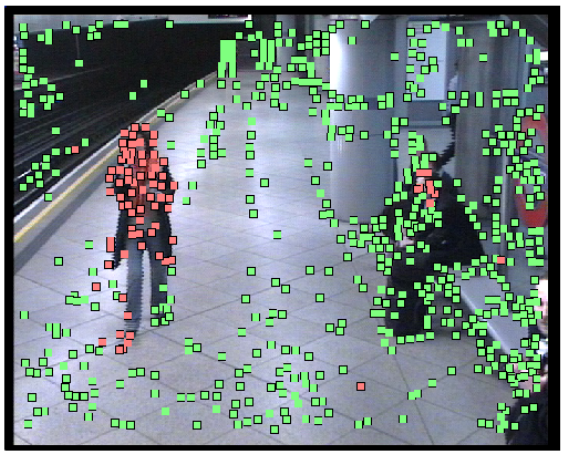

Fig. 1. Segmentation of the mean velocity computed over the interval centred on frame 50 with $\epsilon = 2$

.

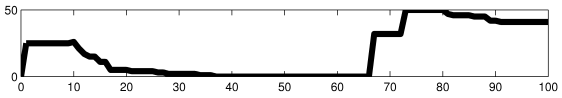

Fig. 2. Number of features dropped for each choice of λ(scaled $\times$ 100).

separates different objects in motion. Trains were segmented out for low values of ϵ, typically between 2 and 6 frames. Pedestrians were segmented out for higher values of ϵ, typically around 12 frames. The number of features segmented decreases exponentially for increasing ϵ, and features on faster-moving objects exist during a narrower interval. Pedestrians are segmented out for values of ϵ for which there are no train features.

The number of features for a particular segmentation can be increased by applying the consensus algorithm for various values of t. For a fixed ϵ, consensus of segmentations with varying t produces binary segmentations showing a particular objects in motion: a train or a group of pedestrians. Then, consensus is applied again to produce a segmentation into multiple classes that separates the multiple objects in motion.

A rational choice of threshold λ in the consensus algorithm would be a value that minimises the number of features that are discarded. In the experiments it was observed that such a choice corresponded to good segmentations. Figure 2 shows the number of features dropped as a function of λ when performing consensus of the 11 partitions with $\epsilon = 24$, $t = 1000$ and t offset by $-5, -4, \ldots, 4, 5$. The value $\lambda = 0.5$ was used in the consensus algorithm when computing the bottom-right segmentation of Figure 3.

V. Conclusion

We have presented an algorithm for segmentation of tracked features which does not rely on rigidity assumptions and uses the output of an off-the-shelf feature tracker and for which the features in each segmented region have similar mean velocities over different scales. It first produces a family of partial partitions of the sets of features, each of them defined over a time interval. It then selects automatically the number of objects in motion. It uses an algorithm for consensus of multiple segmentations, which effectively obtains new relation instances between unrelated features by applying transitivity.

The underlying motion model only requires that features belonging to the same object have a distinct mean velocity at different time intervals, avoiding rigidity constraints. This consensus method is advantageous when exploiting the information from data partitions obtained independently and the number of clusters is unknown.

Future work will be directed towards making a well-founded choice of sampling intervals.

References

[1] R. Tron and R. Vidal, "A benchmark for the comparison of 3-D motion segmentation algorithms," in *CVPR*, 2007.

[2] A. Criminisi, G. Cross, A. Blake, and V. Kolmogorov, "Bilayer segmentation of live video," in *IEEE Computer Vision and Pattern Recognition or CVPR*, 2006, pp. I: 53–60. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2006.69

[3] N. Jojic and B. J. Frey, "Learning flexible sprites in video layers," in *CVPR*. IEEE Computer Society, 2001, pp. 199–206. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/CVPR.2001.990476

[4] L. Gui, J.-P. Thiran, and N. Paragios, "Joint object segmentation and behavior classification in image sequences," in *CVPR*. IEEE Computer Society, 2007. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2007.383234

[5] J. Shi and C. Tomasi, "Good features to track," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, Seattle, Jun. 1994. [Online]. Available: citeseer.nj.nec.com/shi94good.html

[6] M. Grabner, H. Grabner, and H. Bischof, "Learning features for tracking," in *IEEE Computer Vision and Pattern Recognition or CVPR*, 2007, pp. 1–8. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2007.382995

[7] J.-Y. Bouguet, "Pyramidal implementation of the lucas kanade feature tracker: Description of the algorithm," Jean-YvesBouguet, 2002.

[8] S. Birchfield, "KLT: An implementation of the Kanade-Lucas-Tomasi feature tracker," http://www.ces.clemson.edu/~stb/klt/, 2007.

[9] S. Pundlik and S. Birchfield, "Motion segmentation at any speed," in *Proceedings of BMVC*, vol. I, 2006, p. 427. [Online]. Available: http://www.bmva.ac.uk/bmvc/2006/papers/226.pdf

[10] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, "Consensus clustering. A resampling based method for class discovery and visualization of gene expression microarray data," *Machine Learning*, vol. 52, no. 1-2, pp. 91–118, 2003.

[11] S. S. Skiena, *The Algorithm Design Manual*. pub-SV:adr: Springer-Verlag, 1998.

[12] Home Office Scientific Development Branch, "Imagery library for intelligent detection systems (i-LIDS)," http://homeoffice.gov.uk, 2007.

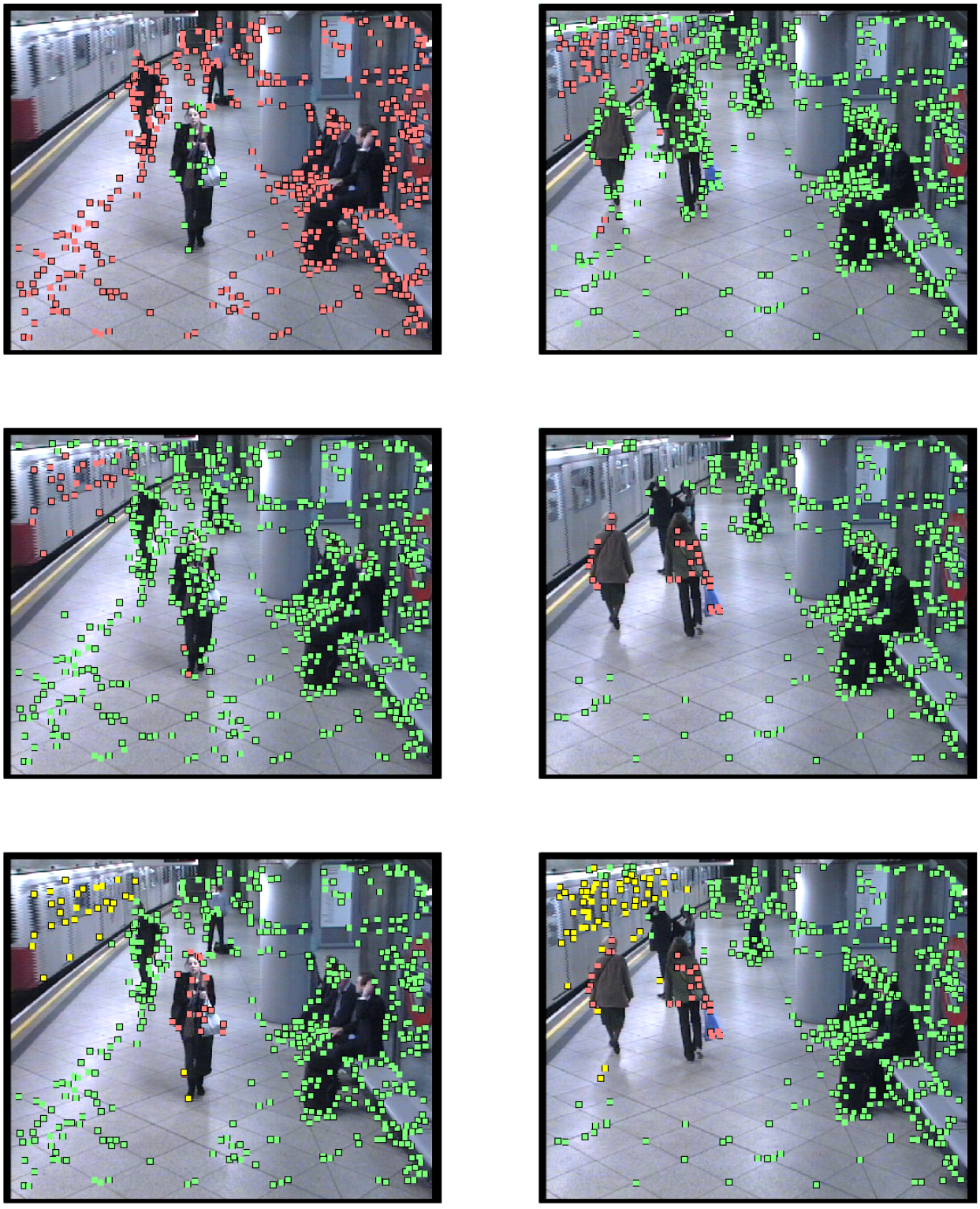

Fig. 3. Segmentation (top and middle) and consensus (bottom) for frames 1000 (left) and 1250 (right).