FAST DETECTION OF INDEPENDENT MOTION IN CROWDS GUIDED BY SUPERVISED LEARNING

Yuan Li, Haizhou Ai

Computer Science and Technology Department, Tsinghua University, Beijing 100084, China
E-mail: ahz@mail.tsinghua.edu.cn

ABSTRACT

Different from appearance-based methods, clustering feature points only by their motion coherence is an emerging category of approach to detecting and tracking individuals among crowds. This paper reformalizes the problem and models a novel objective function for clustering with potential functions as in conditional random field approach. The merits include: 1) it integrates motion, spatial, temporal information; 2) the parameters are automatically obtained by supervised learning; 3) the objective function is based on feature-pair information, which enables effective learning on small amount of training data, as well as very fast online processing speed. Detection ROC curves are given on several datasets (including the CAVIAR set).

Index Terms— Motion detection, multi-object tracking, clustering

1. INTRODUCTION

Most recently there have been a few inspiring works on detecting individuals among crowds only by motion information [1][2]. Namely, based on the assumption that points that appears to move together is likely to be part of the same object, these algorithms cluster tracked feature points into detected objects by analyzing their motion coherence. The results are very encouraging and reveal the great potential of the motion cue alone.

This kind of approaches can be divided into two steps: tracking local features and clustering them. While the first step can be done by some well-established methods, the second step is the core of the approach. For the clustering step, [1] and [2] have some common elements: 1) they use spatial proximity in a form of distance tree or connectivity graph, which is built by thresholding on the max distance of feature pairs within a time window; 2) they define a measure of motion coherence and then use this measure to cluster neighboring features or clusters in the spatial tree or graph. For the measure of motion, [1] uses the variance of features' distance in history, while [2] groups trajectories which share an affine movement by RANSAC. These methods are both featured by their unsupervised manner.

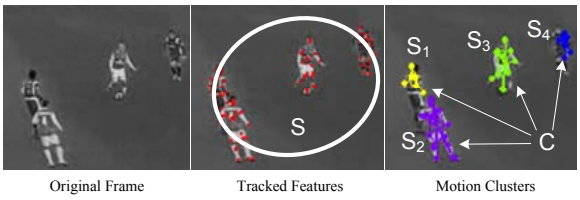


Fig. 1. Problem formalization.

While we focus on the same aim of detecting objects by motion analysis, we view this problem differently – as a *classification* or *labeling* problem. Motion characteristic or spatial relationship of feature points is essentially a type of evidence or observation (just like appearance), and the objective is to discriminate feature points which belongs to the same object from those which not. Therefore, we use supervised statistical learning to model the likelihood of whether a pair of features is on one object as the function of motion coherence or spatial relationship. These learned likelihoods are then integrated in one objective function to guide the hierarchical clustering. In fact, for different applications, such learning mechanism not only automatically adapts the algorithm to the specific scene, but also reveals the effectiveness of spatial-motion evidence by the separability of learned distributions of intra-object and extra-object classes.

2. PROBLEM PREPRESENTATION

Denote the trajectory of each local feature point by $X = \{(x_0, y_0), (x_1, y_1), \ldots, (x_T, y_T)\}$, and the set of all trajectories by $S = \{X_1, \ldots, X_n\}$. Our problem is to find a "optimal" set partition of S. Denote the partition as $C = \{S_1, S_2, \ldots, S_k\}$, which is a collection of disjoint subsets of S , and each S_i includes feature points which belong to the same object (Fig.1). Typically under the Bayesian framework, the "optimal" partition can be defined as

$$\hat{C} = \arg\max_{C} p(C|S), \tag{1}$$

where S can be viewed as the observation and C is the latent variable. Further, when dealing with sequential data, we have $\bar{C}_t = \{C_1, \ldots, C_t\}$ and $\bar{S}_t = \{S_1, \ldots, S_t\}$. By following Bayesian sequential estimation with Markov assumption, the

objective becomes

$$\hat{\bar{C}}_t = \arg\max_{\bar{C}_t} p(\bar{C}_t|\bar{S}_t), \tag{2}$$

$$p(C_t|\bar{S}_t) \propto p(S_t|C_t) \int p(C_t|C_{t-1}) p(C_{t-1}|\bar{S}_{t-1}) dC_{t-1}. \tag{3}$$

Although it is common practice to use sampling techniques [3] to calculate the integral part in (3) , it is not feasible here because the number of all possible C_{t-1} is combinatorial explosive. Therefore we instead estimate C_t by frame-wise greedy optimization based on the result from previous frame:

$$\hat{C}_t = \arg\max_{C_t} p(S_t|C_t) p(C_t|\hat{C}_{t-1}). \tag{4}$$

The objective function (4) and the optimization algorithm are two vital parts which determine the performance of the system. In the following sections, we show how supervised learning can be used to adapt both parts to specific application scenes.

3. MODELING THE OBJECTIVE FUNCTION

The objective function is modeled as the product of probabilities associated with feature point pairs:

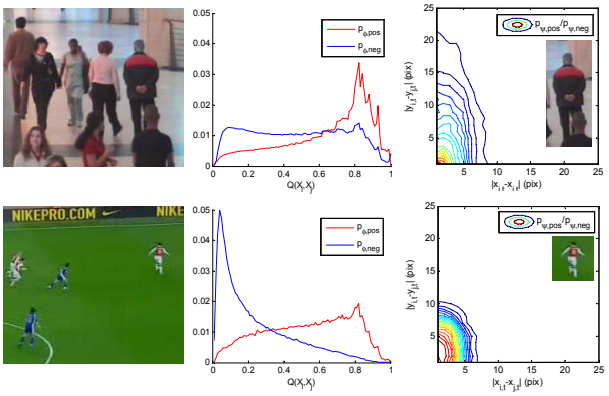
$$\begin{aligned} f(C_t) &= p(S_t|C_t) p(C_t|\hat{C}_{t-1}) \\ &= \prod_{X_i, X_j \in S_t} \Big(p(X_i, X_j|c_{i,t}, c_{j,t}) p(c_{i,t}, c_{j,t}|\hat{c}_{i,t-1}, \hat{c}_{j,t-1}) \Big), \end{aligned} \tag{5}$$

where $c_{i,t}$ is the label of the subset which X_i belongs to, in the partition C_t. We further represent it in the form of the sum of different potential functions:

$$\begin{aligned} \log\left(f(C_t)\right) = \sum_{X_i, X_j \in S_t} \Big(& \overbrace{\phi(X_i, X_j, c_{i,t}, c_{j,t})}^{\text{motion coherence}} \\ & + \overbrace{\psi(X_i, X_j, c_{i,t}, c_{j,t})}^{\text{spatial}} + \overbrace{\lambda(c_{i,t}, c_{j,t}, \hat{c}_{i,t-1}, \hat{c}_{j,t-1})}^{\text{temporal inertia}} \Big) \\ & - \log Z. \end{aligned} \tag{6}$$

Each potential function models a type of useful information, and their parameters should be learned from data. This form is quite like that of a conditional random field [5][6], for which the parameters are learned by gradient ascent to maximize the conditional probability of the true C_t given the labeled training sequence. However, it is difficult to calculate or approximate the normalization (partition function) Z in our problem settings. Hence we choose to learn each potential function independently as follows.

Motion potential. Similar to [1], we assume that two feature points are more likely to be on the same object if the variance of distance is small. The variance $Var(X_i, X_j)$ is calculated in X_i and X_j's overlap frames, and mapped to $(0, 1]$ by $Q(X_i, X_j) = 1/\left(1 + Var(X_i, X_j)\right)$. However, unlike in [1] where $Q(\cdot)$ is directly used as the motion likelihood, here the likelihood is learned as piece-wise functions of $Q(\cdot)$



(a) Oringal scenes. (b) Motion likelihood: $p_{\phi,pos}$ and $p_{\phi,neg}$. (c) Spatial likelihood: $p_{\psi,pos}/p_{\psi,neg}$.

Fig. 2. Motion and spatial likelihoods learned from different scenes (CAVIAR[4] and SOCCER).

(see Fig.2(b), $p_{\phi,pos}$ for pairs which belong to the same object and $p_{\phi,neg}$ for those which not), from feature pairs in training sequence. Hence the motion potential can be calculated as

$$\begin{aligned} &\phi(X_i, X_j, c_{i,t}, c_{j,t}) = \\ &\begin{cases} \log p_{\phi,pos}(Q(X_i, X_j)), & \text{if} \quad c_{i,t} = c_{j,t}; \\ \log p_{\phi,neg}(Q(X_i, X_j)), & \text{else.} \end{cases} \end{aligned} \tag{7}$$

Spatial potential. Relative position of two feature points can be informative for clustering, since in many applications the rough shape or size of target objects are quite stable. We use the horizontal and vertical distance of two features as the basic spatial evidence. Again piece-wise functions are learned for feature pairs which belong to the same object or not respectively (see Fig.2(c)).

$$\begin{aligned} &\psi(X_i, X_j, c_{i,t}, c_{j,t}) = \\ &\begin{cases} \log p_{\psi,pos}(|x_{i,t} - x_{j,t}|, |y_{i,t} - y_{j,t}|), & \text{if} \quad c_{i,t} = c_{j,t}; \\ \log p_{\psi,neg}(|x_{i,t} - x_{j,t}|, |y_{i,t} - y_{j,t}|), & \text{else.} \end{cases} \end{aligned} \tag{8}$$

Temporal potential. For sequential data, temporal smoothness is a natural assumption. Temporal potential helps reduce the "jumpy" assignment of feature points in frame-independent motion detection. From the training sequence it is easy to learn the probability p_λ of two features remaining the state of whether or not belonging to the same object in two consecutive frames. p_λ is usually a value quite close to 1.

$$\begin{aligned} &\lambda(c_{i,t}, c_{j,t}, \hat{c}_{i,t-1}, \hat{c}_{j,t-1}) = \\ &\begin{cases} \log p_\lambda, & \text{if} \quad (c_{i,t} = c_{j,t} \quad \text{and} \quad \hat{c}_{i,t-1} = \hat{c}_{j,t-1}), \\ & \text{or} \quad (c_{i,t} \neq c_{j,t} \quad \text{and} \quad \hat{c}_{i,t-1} \neq \hat{c}_{j,t-1}); \\ \log\left(1 - p_\lambda\right), & \text{else.} \end{cases} \end{aligned} \tag{9}$$

The above feature pair based objective function has two major advantages: 1) it does not require large amount of training data, because m frames of n feature points will provide $mn(n-1)/2$ feature pairs for learning the parameters; 2) it is especially efficient to compute for the clustering process, as will be introduced in the next subsection.

(a) $\Gamma = 2.5$. (b) $\Gamma = 2$. (c) $\Gamma = 1$.

Fig. 3. Different stages of hierarchical clustering.

4. HIERARCHICAL CLUSTERING

Since enumerating all possible partitions of S_t to find the optimal C_t is obviously intractable, we use hierarchical clustering. It is a greedy strategy similar to [1], in the sense that we merge two subsets if the resulting objective function value is larger than leaving them split.

If two subsets S_u and S_v ($S_u, S_v \subset S_t$) are merged, the change of objective function value is

$$\Delta_{u,v} = \sum_{X_i \in S_u, X_j \in S_v} (\Delta^{motion}_{u,v} + \Delta^{spatial}_{u,v} + \Delta^{temporal}_{u,v}) \quad (10)$$

$$= \sum_{X_i \in S_u, X_j \in S_v} \Big(\phi(X_i, X_j, u, u) - \phi(X_i, X_j, u, v) + \psi(X_i, X_j, u, u) - \psi(X_i, X_j, u, v) + \lambda(u, u, \hat{c}_{i,t-1}, \hat{c}_{j,t-1}) - \lambda(u, v, \hat{c}_{i,t-1}, \hat{c}_{j,t-1}) \Big) \quad (11)$$

This $\Delta_{u,v}$ is calculated for every (S_u, S_v) pair, during each round of merging. For fast computation, we keep the Δ values for all the pairs, and update them whenever a merge occurs. Fortunately, the update process is very efficient. According to (11), if S_u and S_v are merged to become $S_{u'}$, for any other subset S_w, we will have

$$\Delta_{u',w} = \Delta_{u,w} + \Delta_{v,w} \quad (12)$$

In other words, whenever two subsets are merged, only n times of addition is needed to update all Δ values (where n is the current number of subsets). Note that (12) also applies to Δ^{motion}, $\Delta^{spatial}$ and $\Delta^{temporal}$ individually.

Now that we have Δ as the discriminant function to decide whether two subsets should be merged or not, a simple strategy is to iteratively select the subset pair with the maximum Δ and merge them, until all Δ's are below zero. However, in experiments the result is not always satisfactory, because: 1) the objective function is learned by assuming independency between each potential function, hence is not very accurate; 2) there is no backtracking during clustering, therefore the order in which subsets are merged may also affect the final result.

Therefore some heuristic and adjustment are made to make the algorithm more flexible: 1) a threshold Γ is added so that clustering continues as long as there is some $\Delta > \Gamma$; 2) subsets with large $\Delta^{spatial}$ are considered to be merged first, to achieve better spatial integrity of the final partition. The algorithm is shown in Table 1.

For each frame, do:

- Track features and cluster them spatially into tiny subsets (to reduce problem size).
- Calculate Δ values for each subset pair by (11).
- While there still exists subset pair which satisfy the merge condition: $\Delta > \Gamma$, do:
  - Find $\max\{\Delta^{spatial}\}$ among all subset pairs which satisfy the merge condition: $\Delta > \Gamma$.
  - Find the subset pair (S_u, S_v) with the maximum $\Delta_{u,v}$, while satisfying $\Delta^{spatial}_{u,v} > \max\{\Delta^{spatial}\} - \beta$.
  - Merge S_u and S_v.
  - Update Δ values for each existing subset pair by (12).

Table 1. The clustering algorithm

Algorithm / Dataset	Processing time per frame
Algorithm of [1] / –	10 sec. to 3 min. (clustering only, reported in [1])
Ours / CAVIAR (384x288)	0.08 sec. (overall)
Ours / SOCCER (320x240)	0.07 sec. (overall)

Table 2. Processing speed (tested on P4 2.8GHz CPU).

The algorithm parameters such as Γ and β are selected by running the algorithm multiple rounds with different settings, and evaluating its performance on the training data. This results in ROC curves shown in Fig.5, from which suitable parameter set can be chosen for specific application.

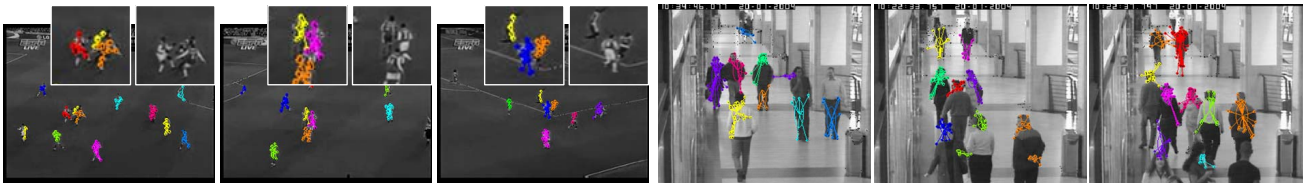
5. EXPERIMENTS

Before clustering, feature points are selected by FAST feature detector [7][8] and tracked by [9]'s implementation of Kanade-Lucase-Tomasi tracker. Our algorithm is implemented in C++. See Table 2 for a comparison of speed.

Detection results. Two testing sets are evaluated: CAVIAR [4] (33932 frames for testing, all with groundtruth; 1000 frames for training) and SOCCER (collected from TV programs, 2364 frames for testing, manually labeled 1 frame out of every 12 frames; 200 frames for training). By changing the parameters Γ and β, ROC curves are obtained for each dataset (Fig.5), and some results are shown in Fig.4. The ROC curve of SOCCER is significantly better than CAVIAR, which can be explained by the learned motion likelihood in Fig.2: the separability of intra-object and extra-object feature pairs of SOCCER is clearly better than that of CAVIAR. This is because in CAVIAR, motion of different entities are less salient than in SOCCER, and the motion of body-parts also undermines the rigid body assumption.

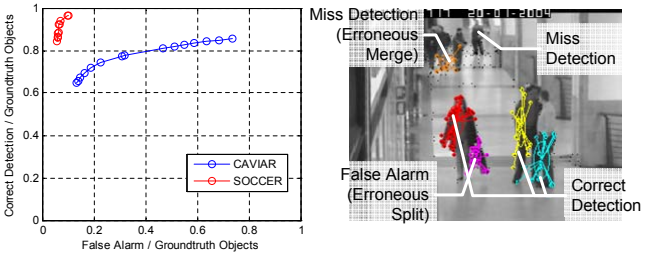
When motion information is not very discriminative, spatial model can improve performance, Fig.6 shows an example.

Tracking and counting passed objects. By connecting largely overlapping clusters in consecutive frames, objects can be tracked temporally. We can both estimate the number of objects present in each frame, and count the objects



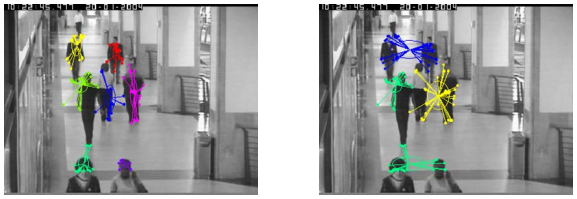
(a) Samples from SOCCER dataset. (b) Samples from CAVIAR dataset.

Fig. 4. Results of object detection on SOCCER and CAVIAR dataset.



(a) ROC curves for each dataset. (b) Criterion for evaluation.

Fig. 5. Detection ROC curves.



(a) With spatial potential. (b) Without spatial potential.

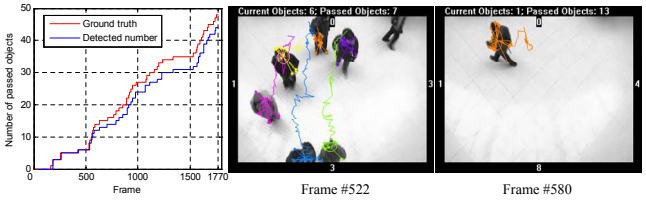
Fig. 6. The effect of spatial potential: without spatial shape knowledge, people walking side by side are clustered together in (b), since they exhibit coherent motion.

that have passed through the scene. We captured a PASSAGE sequence with 1770 frames and tested on it (Fig.7).

6. CONCLUSION AND FUTURE WORK

This paper proposes a new objective function to guide the hierarchical clustering of motion-independent features into distinct objects. This objective function provides a neat framework for integrating different information (motion, spatial, temporal), which are learned from short training sequences to increase algorithm adaptivity to specific application scenes. Detection performance are evaluated on several datasets of typical multi-target visual tasks, including CAVIAR, which is one of the most challenging public dataset for human detection / tracking. The algorithm efficiency is also greatly increased compared with previous similar work.

We have also observed the drawbacks of such motion-based detection method, such as in the case of lack of trackable features (e.g., silhouette-like targets), motion ambiguity of features on boundaries of objects and non-rigid motion. Therefore, complementing motion-based methods with



Frame #522 Frame #580

Fig. 7. Results of object detection and tracking on PASSAGE dataset. The count of passed objects are shown on each edge of the frame. Notice that when five people in #522 exit from the bottom in #580, the count changed from 3 to 8.

appearance-based ones is a promising direction. Within our framework, this could be achieved by adding appearance terms in our objective function. Also, the features should be changed into stronger descriptors.

7. ACKNOWLEDGEMENT

This work is supported in part by National Science Foundation of China under grant No.60332010, No.60673107.

8. REFERENCES

[1] Gabriel J. Brostow and Roberto Cipolla, "Unsupervised bayesian detection of independent motion in crowds," in *CVPR*, 2006.

[2] Vincent Rabaud and Serge Belongie, "Counting crowded moving objects," in *CVPR*, 2006.

[3] M. Isard and A. Blake, "Condensation – conditional density propagation for visual tracking," *IJCV*, vol. 28(1), pp. 5–28, 1998.

[4] "Caviar dataset," http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.

[5] Andrew McCallum John Lafferty and Fernando Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, 2001.

[6] Carsten Rother Jamie Shotton, John Winn and Antonio Criminisi, "Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," in *ECCV*, 2006.

[7] Edward Rosten and Tom Drummond, "Fusing points and lines for high performance tracking.," in *ICCV*, 2005.

[8] Edward Rosten and Tom Drummond, "Machine learning for high-speed corner detection," in *ECCV*, 2006.

[9] S. Birchfield, "Source code of the klt feature tracker," http://www.ces.clemson.edu/~stb/klt/, 2006.