Tracking Using Motion Patterns for Very Crowded Scenes

Xuemei Zhao, Dian Gong, and Gérard Medioni

Institute for Robotics and Intelligent Systems,
University of Southern California Los Angeles, CA, 90089
{xuemeiz,diangong,medioni}@usc.edu

Abstract. This paper proposes Motion Structure Tracker (MST) to solve the problem of tracking in very crowded structured scenes. It combines visual tracking, motion pattern learning and multi-target tracking. Tracking in crowded scenes is very challenging due to hundreds of similar objects, cluttered background, small object size, and occlusions. However, structured crowded scenes exhibit clear motion pattern(s), which provides rich prior information. In MST, tracking and detection are performed *jointly*, and motion pattern information is integrated in both steps to enforce scene structure constraint. MST is initially used to track a single target, and further extended to solve a simplified version of the multi-target tracking problem. Experiments are performed on real-world challenging sequences, and MST gives promising results. Our method significantly outperforms several state-of-the-art methods both in terms of track ratio and accuracy.

Keywords: motion pattern, tracking, very crowded scenes.

1 Introduction

Object tracking has been of broad interest in several applications for decades, such as security and surveillance, human-computer interaction and traffic control. Specifically, tracking in crowded scenes gets more and more attention as it pushes the limit of traditional tracking algorithms.

Crowded scenes can be divided into two categories, structured and unstructured, depending on whether there are clear motion patterns in the scene. The definition is different from [1], in which the authors distinguish between the two based on whether each spatial location supports only one dominant crowd behavior or more. For instance, Fig. 1(d) shows a scene of Italian police riders putting on a motorbike display. Two groups of riders ride in the opposite directions. Due to occlusions, one location may have two opposite velocities which correspond to two groups of coordinated movements. The scene is structured because the two groups of movements form two clear motion patterns.

In this paper, we focus on the problem of single and multiple target tracking in structured crowded scenes (examples are shown in Fig. 1), and we want to track objects in general, instead of specific ones, such as pedestrians. The first

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part II, LNCS 7573, pp. 315–328, 2012.

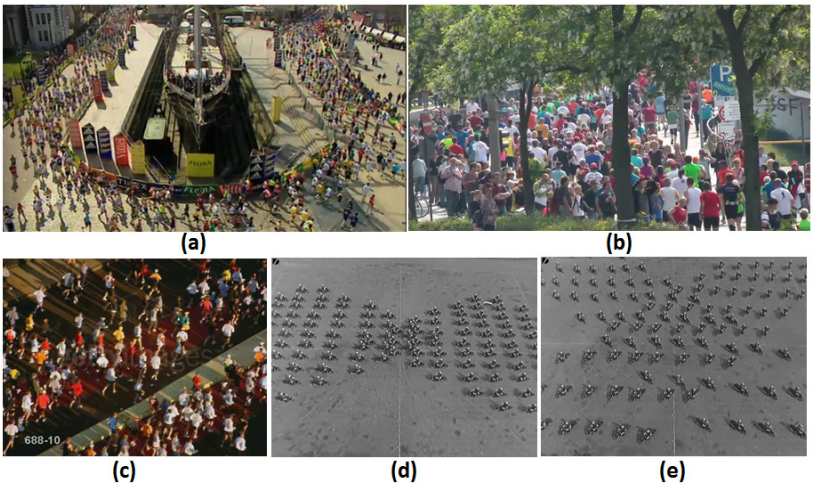

Fig. 1. Examples of structured crowded scenes. (a)(b)(c): Marathon sequences. (d)(e): Italian motorbike display sequences

issue is to solve the motion pattern problem. In very crowded scenes, tracking is difficult for many reasons: the size of a target is usually small; there is a large number of similar objects in the scene; partial and full object occlusions. Furthermore, the "detect and track" paradigm fails here. However, the most salient characteristic of structured scenes is, objects do not move randomly, but follow a pattern instead. Several efforts have been devoted to studying motion patterns, and some of them [2, 3] are successfully used in crowded scene tracking.

The second issue is single vs multi-target tracking. The fundamental difference between the two is whether targets are tracked individually or jointly. In single object tracking, a target is labeled in the first one or several frame(s), and matching is used for detection in the following frames. In multi-target tracking, targets are detected and associated in each frame. Two commonly used methods for detection are appearance based detector, and background modeling based motion blob detector, but neither of them works (as shown in Fig. 2) in very crowded scenes due to the small size, high density and the general object requirement. Thus, in both single and multiple object tracking, we require user labeling in the first frame as input. In multi-target tracking, supervised learning is used for detection. A supervised learning method requires a large number of samples from each kind of object for training. To save a user from tedious work, we only ask to label one example. After tracking the examplar for a few frames to train a detector, we go back to the first frame and use the learned detector to detect similar objects, then track. As an extension of previous work, our approach (Fig. 3) incorporates motion pattern learning results in both the detection and tracking stages, and extends the motion pattern based tracker from single target tracking to a simplified version of multiple target tracking.

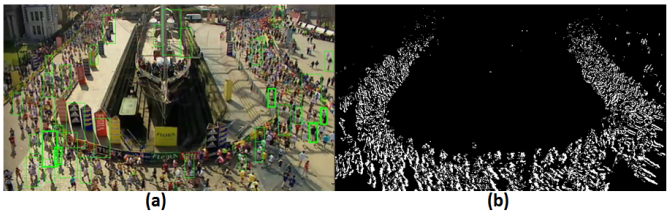

Fig. 2. The commonly used detection methods fail in very crowded scenes. (a) Pedestrian detection results by [4]. (b) Foreground extraction results by MoG

In summary, MST has several advantages compared to existing methods:

−**Better tag and track:** to track a single object in very crowded scenes, MST combines tracking and detection, in both of which motion pattern information is used as prior knowledge, and outperforms state-of-the-art trackers.
−**Improved multi-target tracking:** although it is almost impossible to track all objects in very crowded scenes, we partially address this by proposing a simplified version and solving it by MST.
−**Online:** the proposed algorithm can sequentially process both temporally stationary and non-stationary scenes, infer motion patterns, and use them in both single and multiple object tracking.

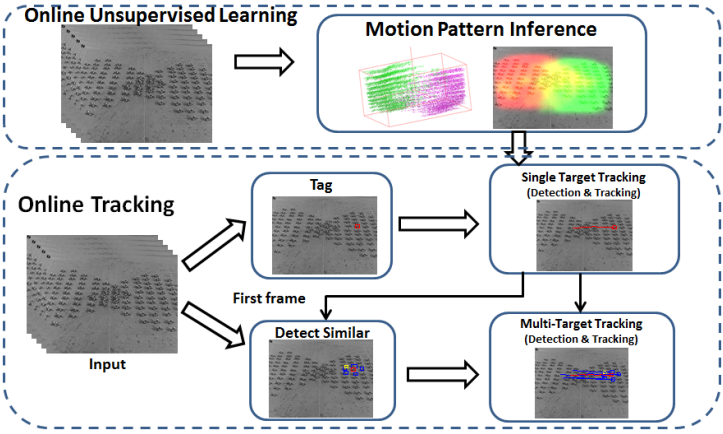

Fig. 3. An overview of Motion Structure Tracker

2 Related Work

Tracking has been a major focus of research in computer vision. Interested readers are referred to a survey [5] for a review. In this section, we give a brief review from several aspects, (1) visual tracking, (2) motion pattern, (3) tracking in crowded scenes, and (4) multiple target tracking.

Visual tracking addresses the problem of tracking a specific object labeled in the first frame or first few frames by a user, and it faces several challenges, such as abrupt motion. IVT Tracker [6] presents a method that incrementally learns a representation, efficiently adapting online to changes in the appearance of the target. P-N Tracker [7] addresses these problems by exploring the structure of unlabeled data, which are positive and negative structures. Single object tracking in a structured crowded scene is similar to visual tracking, but some differences exist. 1) Due to the high density constraint, objects in crowded scene move smoothly, and abrupt motion is rare. 2) The size of a target in crowded scenes is small, thus advanced appearance model based matching is not so helpful. 3) The scale of targets is relatively stable in crowded scenes. 4) Many objects look similar to the target. 5) Although crowded scenes may have cluttered background, it rarely suffers from big change. Fortunately, some of these characteristics contribute to solve the difficult problem of tracking in crowded scenes. For example, due to the smooth movement constraint, a search area can be assumed as in [8, 9]. Thus, we solve single object tracking in crowded scene based on the techniques in state-of-the-art visual tracker, with special constraints from the application.
Motion patterns are the most salient feature in structured crowded scenes. They convey rich information such as how the objects move, and how they interact with each other. From the point of view of what input to learn motion patterns, [10–12] use optical flow, [13] uses object tracking results, and [14] uses keypoints tracking results. These methods have their pros and cons. In the application of crowded scenes, optical flow is too noisy, and object tracking is impossible to get and use as input (and it's just the problem we want to solve). Therefore, keypoints tracking is chosen. From the application point of view, motion patterns have been used to understand scenes [11, 14], improve tracking [13, 15], detect anomalous events [16, 17], and learn traffic rules [18].
Tracking in crowded scenes catches a lot of attention in recent years, and most related works directly or indirectly use motion pattern information. [1–3] all propose novel algorithms and get promising results. Specifically, [1] focuses on unstructured scenes. [2, 3] use motion patterns to assist tracking. The seminal work [2] proposes static floor field, dynamic floor field and boundary floor field to determine the probability of moving from one location to another. [3] presents an elegant framework by training a Hidden Markov model to capture spatio-temporal motion patterns to describe pedestrian movement at each space-time location. [19] proposes a ground breaking idea to first learn a set of crowd behavior priors off-line, then match crowd patches in testing to the database to get priors. Compared to the previous work, we combine tracking and detection, and integrate motion pattern knowledge into both stages in an online fashion.
Multiple target tracking(MTT) is a well studied problem that has received considerable attention [20–23]. Besides the same issues encountered in classic tracking, it deals with additional challenges, e.g., unknown number of targets and the interactions among them. To handle these, jointly optimization of data association is the key, and progress has been made. However, MTT in crowded scenes is seldom successful. Detecting objects in each frame is a challenge, let

alone associating them. [24] addresses the problem of person detection and tracking in crowded scenes, by exploring constraints imposed by crowd density to localize individual people. That's a major improvement of MTT in crowded scenes, but it's not for general objects (human heads specifically), and it requires a large training dataset. In this paper, we look into the problem from another viewpoint, and attempt to solve a simpler problem of MTT: once a user labels a target in the first frame, we try to find similar objects and track all of them.

3 Motion Pattern Inference

As observed in [13], tracklet points form manifold structures which correspond to motion patterns in (x, y, θ) space, where (x, y) is spatial position and θ stands for velocity direction. However, in [13], tracklets are obtained by extracting motion blobs and associating them, making velocity magnitude unreliable. In very crowded scenes, we use KLT keypoint tracker [25] to get short tracklets, making velocity magnitude more reliable. Since magnitude conveys important information of how objects move, we embed tracklet points into (x, y, v_x, v_y) space. After normalizing (v_x, v_y) to the same scale as (x, y), manifold structures emerge. Two examples of the projection in (x, y, v_x) space are shown in Fig. 4. To explore manifold property, Tensor Voting is performed to learn the local geometric structure. Then outliers are filtered out, again using Tensor Voting. As a result, a motion pattern is represented by a set of points $\boldsymbol{q_l} = (x_l, y_l, v_{xl}, v_{yl}),\ l = 1, 2, ..., n$.

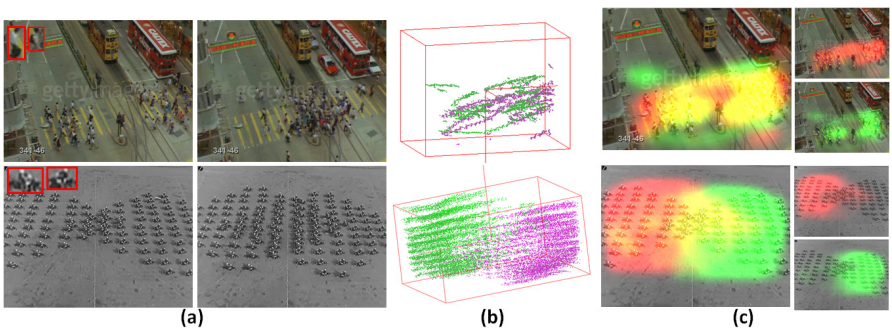

Fig. 4. Temporally non-stationary scenes. First row: Hongkong. Second row: Motorbike. (a) Input sequences and examples of targets. (b) The visualization of motion patterns learning results projected in (x, y, v_x) space. (c) The visualization of motion patterns learning results in image space

3.1 *N*D Tensor Voting

Tensor Voting [26] is a perceptual organization method to analyze the local structures at input points in ND space. Instead of a global estimation for the entire input, local learning results enable us to learn geometric structure and

estimate dimensionality, and furthermore to measure geodesic distance and perform nonlinear interpolation for input with outliers and intersections.

The geometric information at each ND point is represented by a second order, symmetric, and non-negative tensor $\boldsymbol{T}$, corresponding to an $N \times N$ matrix and an ellipsoid in ND space. $\boldsymbol{T}$ represents the manifold structure going through the point by encoding the manifold's normals and tengents as eigenvectors corresponding to $\boldsymbol{T}$'s non-zero and zero eigenvalues respectively. Any such tensor is decomposed as $\boldsymbol{T} = \sum_{d=1}^{N} \lambda_d \boldsymbol{e_d} \boldsymbol{e_d}^T$, where $\{\lambda_d\}$ are the eigenvalues in descending order of magnitude and $\{\boldsymbol{e_d}\}$ are the corresponding eigenvectors. By analyzing the eigen-system of $\boldsymbol{T}$, normals and tangents can be calculated. Also, the confidence, or saliency, of the structure which has d normals is encoded as $\lambda_d - \lambda_{d+1}$, or λ_N for the ball tensor.Therefore, the local structure that the point is assumed to belong to is corresponding to the one with the largest saliency.

Inputs are encoded with tensors, propagate their information to their neighbors' tensors, and collect information from them, in a voting process. The local geometric information of each point can be obtained by examining its tensor.

3.2 Outlier Filtering

Tracklet extraction results are inaccurate due to many reasons, such as low resolution of image, occlusions, illumination change, etc. Therefore, velocities calculated from tracklets are noisy and bring in outliers in (x, y, v_x, v_y) space, making outlier filtering a necessary step. In practice, we use λ_1 as inlier degree measure. Intuitively, λ_1 can be viewed as the sum of the probabilities for all manifolds (a point's local geometric structures) with different intrinsic dimensionalities. By ranking all points according to their λ_1, outliers can be filtered out.

This filtering process is similar to diffusion process [27] in spirit. Conceptually, ranking all the data points according to λ_1 is similar to ranking according to their stationary probabilities, which are calculated from the random walk model built on the data graph [28]. The difference is, in the diffusion process, the weights between points are calculated from a pre-defined kernel, e.g., Gaussian kernel, but in our method, they are calculated from the Tensor Voting process, which is more robust to outliers.

4 Motion Structure Tracker

To track object(s) in crowded scenes, we combine visual tracker, learning-detection [7], and motion pattern learning together, and propose Motion Structure Tracker(MST), which utilizes motion pattern information in both detection and tracking stages.

4.1 Exploiting Motion Pattern in Detection

Objects in structured crowded scenes move smoothly by following some patterns. To establish an object correspondence in the next frame, we search in a window

within which the largest corresponding velocity is no higher than twice the largest velocity found in motion pattern prior. The search space is thus greatly reduced.

The detection probability $Pr_{\boldsymbol{f},\boldsymbol{m}} = Pr(y = 1|\boldsymbol{f}, \boldsymbol{m})$ in MST is calculated as,

$$Pr(y = 1|\boldsymbol{f}, \boldsymbol{m}) \propto Pr(y = 1|\boldsymbol{f}) \times Pr(y = 1|\boldsymbol{m}) \tag{1}$$

where $y = 1$ denotes a detection is positive, $\boldsymbol{f}$ denotes the appearance feature vector used to judge a detection, and $\boldsymbol{m}$ denotes the motion structure information in (x, y, v_x, v_y) space used to judge a detection. In particular, $Pr(y = 1|\boldsymbol{f})$ indicates the appearance based detection probability and $Pr(y = 1|\boldsymbol{m})$ indicates motion detection probability. Eq. 1 approximately holds based on the assumption that the two marginal probabilities with $\boldsymbol{f}$ and $\boldsymbol{m}$ are independent.

Appearance Detection Probability. Random ferns [29] are proposed to speed up random forest. Random fern classifiers have proven to be efficient and effective in tasks such as tracking [7], image classification [30], and action recognition [31]. Thus, we use a random fern classifier for detection, and as in [7], explore the structure of unlabeled data, i.e. the positive and negative structures. Given a video, in the first frame, image patches close to the target are used as positive training examples, and those far from it are used as negative training examples. In the following frames, once the target is validated, corresponding examples (positive and negative) are extracted and used to update the detector.

The detector contains a set of ferns. Once a patch is given, each fern evaluates it independently, by taking a set of measurements in feature vector $\boldsymbol{f}$. At the leaf node that $\boldsymbol{f}$ points to, the posterior probability of whether the patch is positive $(y = 1)$ is calculated based on how many positive (s^+) and negative (s^-) samples are already recorded by that leaf, $s^+/(s^+ + s^-)$. And $Pr(y = 1|\boldsymbol{f})$ is an average of the posterior probabilities from all the ferns.

Motion Detection Probability. Moreover, each candidate patch is examined to test whether it is consistent with motion pattern prior knowledge. For a target centered at (x_i, y_i) in frame t, possible correspondences with displacements $(v_{xij}, v_{yij}), (j = 1, ..., m)$ in frame $t + 1$ produces a set of points $\boldsymbol{p_{ij}} = (x_i, y_i, v_{xij}, v_{yij})$. Intuitively, we want to check whether these points get support from motion pattern prior. Thus, each point $\boldsymbol{q_l} = (x_l, y_l, v_{xl}, v_{yl}), (l = 1, 2, ..., n)$ from motion pattern learning votes for $\boldsymbol{p_{ij}}$, and the sum of all the votes is,

$$vote_{ij} = \sum_{l=1}^{n} e^{-||\boldsymbol{q_l} - \boldsymbol{p_{ij}}||^2/\sigma_1^2} \tag{2}$$

We normalize the votes as follows:

$$Pr(y = 1|\boldsymbol{m}) = \frac{vote_{ij}}{\sum_{j=1}^{m} vote_{ij}} \tag{3}$$

By combining $Pr(y = 1|\boldsymbol{f})$ and $Pr(y = 1|\boldsymbol{m})$, the final MST detection probability not only captures the object appearance feature but also incorporates the motion structure, which is important to detect objects in crowded scenes.

4.2 Exploiting Motion Pattern in Tracking

In addition to detecting correspondence of a target, we also track the target directly by selecting keypoints on it, and find their correspondences in the next frame. However, crowded scenes have low resolution and cluttered background, making optical flow results unreliable. To solve the problem, we propose a motion structure based optical flow, i.e., Structure Flow (SF), by generalized Tikhonov regularization. Structure flow is a Bayesian extension of optical flow, since motion pattern information gives some prior knowledge about the movement.

Formally, by using the prior knowledge as regularization term, the structure flow $\boldsymbol{v} = (v_x, v_y)$ is optimized by minimizing the following loss function,

$$\mathcal{L}_{SF}(\boldsymbol{v}) = \mathcal{L}_{AF}(\boldsymbol{v}) + \lambda \mathcal{L}_{MP}(\boldsymbol{v}) \tag{4}$$

where $\mathcal{L}_{AF}(\boldsymbol{v})$ stands for the loss based on the appearance features and $\mathcal{L}_{MP}(\boldsymbol{v})$ stands for the loss based on motion pattern prior. λ is the regularization parameter to control the impact of priors. These two items are described as follows.

$\mathcal{L}_{AF}(\boldsymbol{v})$: Recall the Lucas-Kanade [32] method. To calculate the velocity (v_x, v_y) of a point $q = (x, y)$, we consider K points $(\{p_i = (x_i, y_i)\}, i = 1, 2, ..., K)$ in q's neighborhood. Let $\boldsymbol{I_x}(p_i)$, $\boldsymbol{I_y}(p_i)$ and $\boldsymbol{I_t}(p_i)$ represent the partial derivatives of the image $\boldsymbol{I}$ with respect to position x, y and time t, evaluated at the point p_i. Based on LK assumptions, (v_x, v_y) must satisfy

$$\boldsymbol{I_x}(p_i)v_x + \boldsymbol{I_y}(p_i)v_y + \boldsymbol{I_t}(p_i) = 0 \tag{5}$$

Let

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{I_x}(p_1) & \boldsymbol{I_y}(p_1) \\ \boldsymbol{I_x}(p_2) & \boldsymbol{I_y}(p_2) \\ ... & ... \\ \boldsymbol{I_x}(p_K) & \boldsymbol{I_y}(p_K) \end{pmatrix}, \boldsymbol{v} = \begin{pmatrix} v_x \\ v_y \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} \boldsymbol{I_t}(p_1) \\ \boldsymbol{I_t}(p_2) \\ ... \\ \boldsymbol{I_t}(p_K) \end{pmatrix}$$

Then the K points satisfy $\boldsymbol{Av} = \boldsymbol{b}$ ideally. However, usually the points do not move in the same way, so this does not hold in practice. Thus, we can have the following loss function,

$$\mathcal{L}_{AF}(\boldsymbol{v}) = ||\boldsymbol{Av} - \boldsymbol{b}||^2 \tag{6}$$

where eq. 6 itself can also be viewed as an over-determined system if $K > 2$.

$\mathcal{L}_{MP}(\boldsymbol{v})$: We use motion pattern information in (x, y, v_x, v_y) space as prior. For a point $\boldsymbol{x} = (x, y)$, consider a $W \times W$ area around it. It collects the information from each point $\boldsymbol{x_i} = (x_i, y_i), (i = 1, 2, ..., L)$ in the area with a correspondence $(x_i, y_i, v_{ix}, v_{iy})$ in the motion pattern prior. We weigh the impact as $w_i = e^{-||\boldsymbol{x}-\boldsymbol{x_i}||^2/\sigma_2^2}$, and normalize it as $\overline{w_i} = \frac{w_i}{\sum_{j=1}^{L} w_i}$. Then the weighted sum of velocity is used as an estimation of the expected velocity $\boldsymbol{v_0} = (v_{0x}, v_{0y})^T$, and the covariance matrix Σ_v can also be derived. Based on the estimations of mean and co-variance matrix, we design the following loss function,

$$\mathcal{L}_{MP}(\boldsymbol{v}) = (\boldsymbol{v} - \boldsymbol{v_0})^T \Sigma_v^{-1} (\boldsymbol{v} - \boldsymbol{v_0}) \tag{7}$$

Essentially, eq. 7 can be viewed as a multivariate Gaussian probabilistic framework to model the prior of structure flow.

$\mathcal{L}_{SF}(\boldsymbol{v})$**:** By replacing the two proposed loss functions into eq. 4, structure flow $\boldsymbol{v}$ is estimated by minimizing

$$\mathcal{L}_{SF}(\boldsymbol{v}) = ||\boldsymbol{A}\boldsymbol{v} - \boldsymbol{b}||^2 + \lambda||\boldsymbol{v} - \boldsymbol{v_0}||^2_{\Sigma_v^{-1}} \tag{8}$$

where $||\boldsymbol{v} - \boldsymbol{v_0}||^2_{\Sigma_v^{-1}}$ stands for the Mahalanobis distance $(\boldsymbol{v} - \boldsymbol{v_0})^T \Sigma_v^{-1} (\boldsymbol{v} - \boldsymbol{v_0})$.

According to the generalized Tikhonov regularization, the closed-form solution is

$$\boldsymbol{v} = \boldsymbol{v_0} + (\boldsymbol{A}^T\boldsymbol{A} + \lambda(\Sigma_v)^{-1})^{-1}\boldsymbol{A}^T(b - \boldsymbol{A}\boldsymbol{v_0}) \tag{9}$$

Thus, for each keypoint on the target, a velocity estimation $\boldsymbol{v}$ incorporating motion structure information is generated. The target position can be estimated.

It is worth noting that, as $\lambda \to 0$, eq. 9 degenerates to $(\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{b}$, which is the standard Lucas-Kanade optical flow. This is because no regularization is used in $\mathcal{L}_{SF}(\boldsymbol{v})$. On the other hand, as $\lambda \to \infty$, eq. 9 degenerates to $\boldsymbol{v}_0$, which means we fully trust the motion pattern priors.

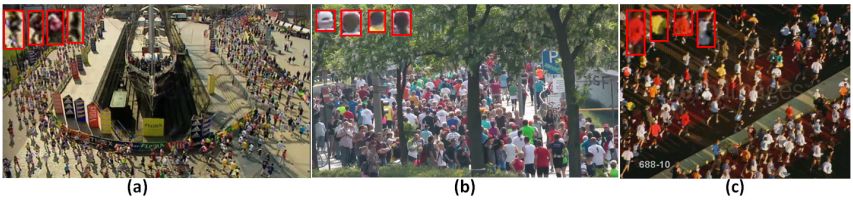

Fig. 5. Temporally stationary scenes and examples of targets. (a) Marathon-1. (b) Marathon-2. (c) Marathon-3

4.3 Simplified Multiple Target Tracking in Structured Crowded Scenes

In crowded scenes, tracking many similar objects is extremely challenging. One of the difficulties is how to detect multiple targets. Due to the small target size and large intra class variance caused by viewpoint and occlusion of objects, an object detector does not provide satisfying results (an example of pedestrian detection results by state-of-the-art detector [4] is given in Fig. 2). Therefore, we step back to solve a simpler problem: once a user labels a target in the first frame, find similar objects and track all of them.

We track the user labeled object for a few frames, and in parallel train our detector. Then we go back to the first frame, detect the top n_1 (user input) similar objects by the detector. Motion structure tracker tracks multiple targets in a way similar to single target tracking. Specifically, we treat each of the $n_1 + 1$ objects as a single object to track. If the tracking results in frame t are good

(measured by confidence scores from detector and structure flow tracker), we move to frame $t+1$. If not, we consider a window of size L, from frame t to frame $t+L-1$, and locate candidates by detector and structure flow tracker based on frame t. Then association is formulated as inference in a set of Bayesian networks [22]. Confidence score is the driving force behind finding the MAP data association estimate. Then we move forward to frame $t+1$ and repeat the process.

5 Experimental Validation

We apply Motion Structure Tracker(MST) in four sets of experiments. The video sequences we use can be divided into two groups: *temporally stationary* and *temporally non-stationary* scenes, depending on whether the motion patterns change with time. Fig. 5 shows three examples of Marathon sequences (from [2, 15] and YouTube), in which motion patterns are the same from the beginning to the end. For such sequences, we only need to learn motion patterns in the first few frames, and then use them for the whole sequence. On the contrary, the HongKong sequence [15] and the Italian motorbike sequence (from Youtube) have changing motion patterns, making it necessary to online update motion pattern learning results. Besides, the task performed in each sequence can also be divided into two groups: single target tracking and multiple target tracking. Therefore, four sets of experiments are designed for the four combinations. In all the experiments, we use 10 ferns and 13 features per fern, and we fix $\sigma_1 = 10$, $\sigma_2 = 5$, $\lambda = 2$. Results are robust to a certain range of these parameters. In outlier filtering step, the points whose λ_1 is smaller than 0.02 of the median of the λ_1 of all the points are filtered out. The appearance similarity between the target and candidate is calculated by normalized cross correlation(NCC) at the gray-level. Our single target tracking results are compared with IVT Tracker [6], P-N Tracker [7] and CTM [1].

1. Single target tracking results in temporally stationary scenes.

The three Marathon sequences (Fig. 5) capture athletes from static overhead cameras. They are challenging real-world scenes due to the existence of hundreds of similar small-size objects, and occlusions from time to time. The three have 343, 249 and 143 frames respectively, and resolutions are 720×404, 1280×720 and 480×360 respectively. In each experiment, we manually select a rectangular region around a target in the first frame. In each sequence, the first 50 frames are used to learn motion patterns, and 10 targets (whole body or upper body in Marathon-1 and Marathon-3, and heads in Marathon-2) are randomly selected to test, with an average size of target as 15×21, 21×26 and 17×29 pixels respectively. In the Matlab application provided by [1], bounding box size is fixed, and only target center is the input. Since target size varies in different sequences, we resize the input images of every sequence to the most proper size.

Ground truth is manually labeled for each target in each frame. Two criteria are used to compare the tracking results between different trackers. (1) Average Center Location Error (ACLE), which measures the pixel difference between

tracked object center and the ground truth. (2) Average Track Ratio (ATR), which is calculated as successfully tracked frames divided by total number of frames in which a target is in FOV. The results are presented in Table 1. It can be seen that our tracker outperforms other state-of-the-art, and we get low ACLE because even if our tracker shifts to a wrong object, it is still following motion pattern, thus close to the target. The best results are on Marathon-3 sequence, since it contains relatively large-size targets with clear appearance. In Marathon-1, runner size is small, and there is viewpoint change as runners run through the U-shape street. In Marathon-2, we only track heads of runners, thus the discrimination power is weak. Also, trees in Marathon-2 cause occlusion, which is a major source of errors. An example of tracking results is shown in the first row of Fig. 6.

Table 1. Tracking evaluation results of single target in temporally stationary scenes

Method	Marathon-1		Marathon-2		Marathon-3	
	ATR	ACLE	ATR	ACLE	ATR	ACLE
IVT Tracker [6]	35.21%	62.8	33.47%	86.5	40.03%	64.1
P-N Tracker [7]	56.16%	35.1	68.60%	56.4	69.16%	33.9
CTM [1]	52.35%	38.8	65.72%	62.8	71.69%	30.5
Ours	**81.40%**	**6.7**	**73.12%**	**28.5**	**91.08%**	**4.8**

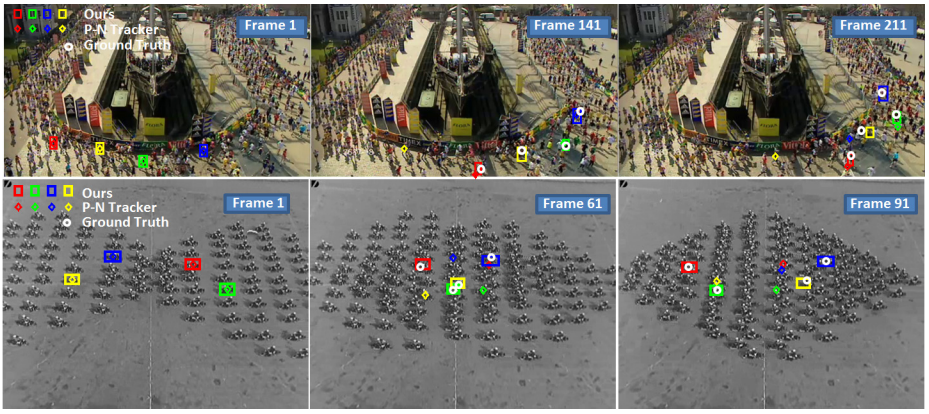

Fig. 6. Examples of tracking results comparison. First row: temporally stationary scenes. Second row: temporally non-stationary scenes.

2. Single target tracking results in temporally non-stationary scenes. To capture changing motion patterns, an online motion pattern learning and tracking framework is built. Each time, we consider a fixed-length window of size 40 in time, extract motion pattern information in the window and utilize it to assist tracking. Then the window shifts 40 frames to deal with the next 40 frames (or less in the last window), and so on.

The Hongkong sequence (first row in Fig. 4) and Motorbike sequence (second row in Fig. 4) have 248 frames and 100 frames respectively. In each sequence, 10 targets are randomly selected, with an average size of 15×24 and 30×22 pixels respectively. Since each of the two sequences contains two motion patterns, we divide motion pattern points in (x, y, v_x, v_y) space into two groups by k-means. To calculate $\mathcal{L}_{MP}(\boldsymbol{v})$, we first decide which motion pattern the point (or the object it's from) belongs to by the vote from the object's past trajectory.

The motion pattern learning results for the two whole sequences are shown in Fig. 4. Fig. 4 (b) shows the visualization of projection in (x, y, v_x) space, and Fig. 4 (c) shows the projection on images. Tracking results comparison are presented in Table 2. An example is shown in the second row of Fig. 6. Our tracker still significantly outperforms the others. In Motorbike sequence, a large number of similar objects exist, and motion pattern prior effectively reduces drift. The Hongkong sequence is the most challenging one, since motion patterns are not as clear as others.

Table 2. Tracking evaluation of single target in temporally non-stationary scenes

Method	Hongkong		Motorbike	
	ATR	ACLE	ATR	ACLE
IVT Tracker [6]	27.63%	58.9	31.56%	69.7
P-N Tracker [7]	39.58%	42.3	47.22%	55.4
CTM [1]	52.17%	35.2	42.35%	58.3
Ours	**62.31%**	**28.5**	**88.75%**	**5.6**

3. Multi-target tracking results in temporally stationary scenes.
In each experiment, we manually select a rectangular region around a target in the first frame. We track the target for 10 frames to train our detector. Going back to the first frame, we use the detector to detect $n_1 = 6$ similar objects. Window size L is fixed as 8. If the tracking result for each target (by detector and structure flow tracker) has confidence ([0,1]) larger than 0.8, we move to the next frame. Otherwise use the $L = 8$ window to jointly optimize. The detection and tracking results are shown in Fig. 7(a). A red rectangle denotes the user labeled target, a blue rectangle denotes true positive detection of similar objects, and a yellow rectangle denotes false positive detection in the first frame. As we increase n_1, more false positives are brought in.

4. Multi-target tracking results in temporally non-stationary scenes.
Settings are the same as before. Online motion pattern learning is performed in the same setting as in Section 5.2. Detection and tracking results are shown in Fig. 7(b). It shows that tracking helps us remove some false alarms, and correct some others. For example, the false positive on the right in Fig. 7(b) detects two people in the first frame, but the tracker gradually moves to one target, so the detection is kept in tracking results.

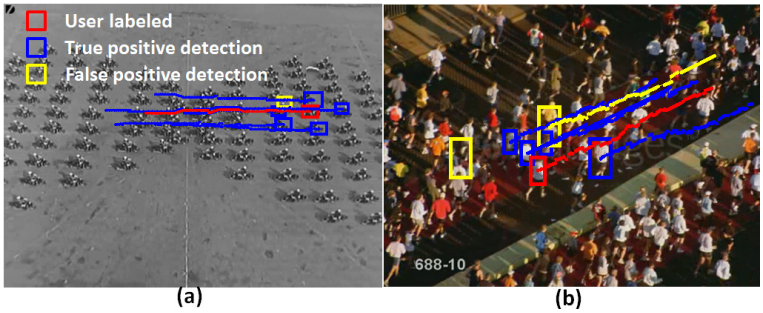

Fig. 7. Simplified multi-target detection and tracking results in temporally (a) stationary and (b) non-stationary scenes respectively. Red rectangle denotes the user labeled target. Blue rectangles denote the similar objects detected by the learned detector

6 Conclusion

This paper addresses the problem of tracking single and multiple targets in structured crowded scenes by Motion Structure Tracker, which combines several research topics: visual tracking, motion pattern learning, and multiple target tracking. Although each topic has been intensively studied, they are not jointly considered before. The experimental results on several challenging sequences and the comparison with state-of-the-art methods demonstrate the effectiveness of motion structure tracker. In the future, we will generalize our tracker to tracking in unstructured crowded scenes, and go further in multiple-target tracking.

Acknowledgments. This work was supported in part by grant DE-FG52-08NA28775 from the U.S. Department of Energy and NIH Grant EY016093.

References

1. Rodriguez, M., Ali, S., Kanade, T.: Tracking in unstructured crowded scenes. In: ICCV, pp. 1389–1396 (2009)
2. Ali, S., Shah, M.: Floor Fields for Tracking in High Density Crowd Scenes. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 1–14. Springer, Heidelberg (2008)
3. Kratz, L., Nishino, K.: Tracking with local spatio-temporal motion patterns in extremely crowded scenes. In: CVPR, pp. 693–700 (2010)
4. Huang, C., Nevatia, R.: High performance object detection by collaborative learning of joint ranking of granules features. In: CVPR, pp. 41–48 (2010)
5. Yilmax, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Journal of Computing Surveys 38 (2006)
6. Ross, D., Lim, J., Lin, R., Yang, M.: Incremental learning for robust visual tracking. In: IJCV, pp. 125–141 (2008)
7. Kalal, Z., Matas, J., Mikolajczyk, K.: P-n learning: Bootstrapping binary classifiers by structural constraints. In: CVPR, pp. 49–56 (2010)

8. Babenko, B., Yang, M.H., Belongie, S.: Visual tracking with online multiple instance learning. In: CVPR, pp. 983–990 (2009)
9. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised On-Line Boosting for Robust Tracking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 234–247. Springer, Heidelberg (2008)
10. Mehran, R., Moore, B., Shah, M.: A streakline representation of flow in crowded scenes. In: ECCV, pp. 439–452 (2010)
11. Saleemi, I., Hartung, L., Shah, M.: Scene understanding by statistical modeling of motion patterns. In: CVPR, pp. 2069–2076 (2010)
12. Hu, M., Ali, S., Shah, M.: Learning motion patterns in crowded scenes using motion flow field. In: ICPR, pp. 1–5 (2011)
13. Zhao, X., Medioni, G.: Robust unsupervised motion pattern inference from video and applications. In: ICCV, pp. 715–722 (2011)
14. Zhou, B., Wang, X., Tang, X.: Random field topic model for semantic region analysis in crowded scenes from tracklets. In: CVPR, pp. 3441–3448 (2011)
15. Ali, S., Shah, M.: A lagrangian particle dynamics approach for crowd flow segmentation and stability analysis. In: CVPR, pp. 1–6 (2007)
16. Hu, W., Xiao, X., Fu, Z., Xie, D., Tan, T., Maybank, S.: A system for learning statistical motion patterns. In: PAMI, pp. 1450–1464 (2006)
17. Wang, X., Ma, X., Grimson, E.: Unsupervised activity perceptionby hierarchical bayesian models. In: CVPR, pp. 1–8 (2007)
18. Kuettel, D., Breitenstein, M.D., Gool, L.V., Ferrari, V.: What's going on? discovering spatio-temporal dependencies in dynamic scenes. In: CVPR, pp. 1951–1958 (2010)
19. Rodriguez, M., Sivic, J., Laptev, I., Audibert, J.: Data-driven crowd analysis in videos. In: ICCV, pp. 1235–1242 (2011)
20. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: CVPR, pp. 1–8 (2008)
21. Huang, C., Wu, B., Nevatia, R.: Robust Object Tracking by Hierarchical Association of Detection Responses. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 788–801. Springer, Heidelberg (2008)
22. Prokaj, J., Medioni, G.: Inferring tracklets for multi-object tracking. In: Workshop of Aerial Video Processing Joint with IEEE CVPR, pp. 37–44 (2011)
23. Zhang, L., Li, Y., Nevatia, R.: Global data association for multi-object tracking using network flows. In: CVPR, pp. 1–8 (2008)
24. Rodriguez, M., Sivic, J., Laptev, I., Audibert, J.: Density-aware person detection and tracking in crowds. In: ICCV, pp. 2423–2430 (2011)
25. Tomasi, C., Kanade, T.: Detection and tracking of point features. In: IJCV (1991)
26. Mordohai, P., Medioni, G.: Dimensionality estimation, manifold learning and function approximation using tensor voting. JMLR 11, 411–450 (2010)
27. Coifman, R.R., Lafon, S., Lee, A., Maggioni, M., Nadler, B., Warner, F., Zucker, S.: Geometric diffusion as a tool for harmonic analysis and structure definition of data, part i: Diffusion maps. The National Academy of Sciences (2005)
28. Zhou, D., Weston, J., Gretton, A., Bousquet, O., Schölkopf, B.: Ranking on data manifolds. In: NIPS (2004)
29. Ozuysal, M., Calonder, M., Lepetit, V., Fua, P.: Fast keypoint recognition using random ferns. In: PAMI, pp. 448–461 (2010)
30. Bosch, A., Zisserman, A., Munoz, X.: Image classification using random forests and ferns. In: ICCV, pp. 1–8 (2007)
31. Oshin, O., Gilbert, A., Illingworth, J., Bowden, R.: Action recognition using randomised ferns. In: ICCV, pp. 530–537 (2009)
32. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. IJCAI, 674–679 (1981)