Multi-View 3D Human Tracking in Crowded Scenes

Xiaobai Liu
Department of Computer Science, San Diego State University
GMCS Building, Campanile Drive
San Diego, CA 92182

Abstract

This paper presents a robust multi-view method for tracking people in crowded 3D scene. Our method distinguishes itself from previous works in two aspects. Firstly, we define a set of binary spatial relationships for individual subjects or pairs of subjects that appear at the same time, e.g. being left or right, being closer or further to the camera, etc. These binary relationships directly reflect relative positions of subjects in 3D scene and thus should be persisted during inference. Secondly, we introduce an unified probabilistic framework to exploit binary spatial constraints for simultaneous 3D localization and cross-view human tracking. We develop a cluster Markov Chain Monte Carlo method to search the optimal solution. We evaluate our method on both public video benchmarks and newly built multi-view video dataset. Results with comparisons showed that our method could achieve state-of-the-art tracking results and meter-level 3D localization on challenging videos.

Introduction

Tracking multiple people in 3D scene is crucial for high-level video understanding tasks, e.g. recognizing human activities, behaviours or social activities. Despite impressive results achieved Hofmann, Wolf, and Rigoll (2013) Fan, X.Shen, and Wu (2013), existing methods still suffer from occlusions, frequent intersections, missing detections of humans and other challenges, in particular for crowded scenes, e.g. parking-lot, office etc. In this work, we study a multi-view method for simultaneous tracking and localizing moving subjects in 3D scene map with sub-meter accuracy.

The major contribution of this work is to explore a set of binary spatial relationships between subjects in images, i.e. being left or right, being closer to the camera central, which are all expressed in binary forms and can be directly mined from 2D image observations, e.g. detection results. These relationships, once mined, are potentially helpful for two purposes: i) registering the detected boxes in other views or the 3D scene map; ii) tracking human over frames in different views.

On the one hand, in order to localize a detected human box in 3D or other views, existing multi-view 3D trackers Fleuret et al. Khan and Shah (2006) usually select the bottom-central point as footprints and apply the view-to-map or cross-view ground homograph. This is however infeasible for crowded scenes, e.g. office, because persons are usually occluded by furniture or other persons. Fortunately, human detection boxes, although being incomplete, usually reflect pair-wise 3D spatial relationship between subjects. Taking two boxes A and B detected in the same frame for instance, if A is larger than B, the person A is likely to stand closer to the camera than B; if the central point of box A locates on the left to box B, so do their footprints.

On the other hand, binary spatial relationships can be used to help track moving persons that have frequent intersections. The basic idea follows the Ising/Potts prior, that two moving people are likely to preserve their spatial relations, e.g. being left, over a certain time period. If person A locates on the left of person B at time t, the two people should have the same pair-wise spatial relationship at time $t+1$. In addition, these consistency constraints can be used to suppress noise detections while associating boxes to subjects .

We develop an unified probabilistic method to automatically mine binary constraints for various objectives, including localizing truncated human boxes in 3D scene, identifying detected boxes across views, recovering missing detections and tracking subjects over time. These objectives are mutually beneficial and should be solved jointly. In particular, we describe each trajectory as a continuous function of 3D locations w.r.t time and enforce high-order smoothness over time to facilitate trajectory estimation from incomplete observations.

For inference, we introduce an iterative cluster Markov Chain Monte Carlo (MCMC) method Tu and Zhu (2002) to search the optimal solution. Once initialized, we use a set of dynamics to drive the current solution to a new one to simulate a Markov Chain in the solution space. At each step, our method changes the status of a cluster of nodes, instead of a single node Khan, Balch, and Dellaert (2005), to speed up the sampling process Barbu and Zhu (2007). The new solution is accepted with a probability and the dynamics are paired with each other to make the search reversible that shall guarantee convergence to global optimal. We evaluate the proposed method on both public benchmarks and a newly built video dataset. Results with comparisons to other popular trackers show that our method could achieve state-of-the-art tracking results and localize moving subjects in

3D scene with meter-level accuracy in crowded scenes.

Related works

Our method is inspired by the pioneer works of Hoiem et al Hoiem, Efros, and Hebert (2006) which tried to utilize a property of perspective geometry, i.e., the perceived size of the objects scales inversely with the distance to the camera. Different from pruning object hypothesis Hoiem, Efros, and Hebert (2006) or recovering depth maps Ladicky, Shi, and Pollefeys (2014), our goal is to track objects in 3D scene.

Our work is also closely related to the following three research streams.

Multi-view object tracking is usually addressed as a data associating problem across cameras. Typical solutions include, homograph constrains Khan and Shah (2006), human-part constraints Gupta, Mittal, and Davis (2008), joint background reconstruction and foreground tracking Hofmann, Wolf, and Rigoll (2013), Marked Point Process Utasi and Benedek (2011),network flow optimization Leal-Taixe, Pons-Moll, and Rosenhahn (2013) Wu et al. (2009), multi-view SVM Zhang et al., shortest path method Berclaz et al. (2011) and probabilistic occupancy grid method Fleuret et al.. In contrast, we define a set of binary spatial constraints between subjects, and introduce an unified probabilistic method to automatically extract these constraints for 3D tracking and localization purposes .

The existing methods address the challenges of **missing detections, false alarm, and occlusions** with different strategies. Mittal and Davis Mittal and Davis (2003), Fan et al. Fan, X.Shen, and Wu (2013) proposed to integrate high-level recognition and low-level object tracking which works well against occlusions or intersections. Lin et al. Lin et al. (2014) presented an inspring spatio-temporal method for extracting foregrounds that can significatlly reduce the effects of scene clutters. Yang et al. Yang et al. (2014) explicitly addressed occlusions in a probabilistic framework for multi-target tracking. Zhang et al. Zhang, Li, and Nevatia (2008), Henriques et al. Henriques, Caseiro, and Batista (2011) and Pirsiavash et al. Pirsiavash, Ramanan, and Fowlkes (2011) introduced global optimization frameworks to track objects over long-range, which are helpful to recovering trajectories from occlusions. Milan et al. Milan, Schindler, and Roth (2013) addressed multiple object tracking by defining bi-level exclusions. Wang et al. Wang et al. (2014) proposed to infer tracklets, i.e. short trajectories, and further solved data association problem. Possegger et al. Possegger et al. (2014) relied on geometric information to efficiently overcome detection failures when objects are significantly occluded. In contrast, this work presents a different way to address these challenges, i.e. mining binary spatial constraints that should be respected during inference, which is particularly effective for crowded scenes.

MCMC sampling has been used for tracking purpose in past literature. Khan et al. Khan and Shah (2006) integrated MCMC technique with particle filer tracking framework. Yu et al. Yu, Medioni, and Cohen (2007) utilized single site sampler for associating foreground blobs to trajectories. Liu et al. Liu, Lin, and Jin (2013) introduced a spatial-temporal graph to jointly solve region labeling and object tracking by Swendsen-Wang Cut method Barbu and Zhu (2007). In this work, we extend cluster sampling technique for 3D human tracking and design a set of reversible dynamics to efficiently search the optimal.

Probabilistic Formulation

We formulate multi-view 3D human tracking in a probabilistic framework. Let K denote the number of trajectories, V_i detected human boxes regardless of views, $c_i \in [0, K]$ the trajectory index. c_i=0 means the box i is a false alarm. Let $V = \{(V_i, c_i)\}$ denote the set of boxes, and $\tau = \{(\tau_k, t^s_k, t^e_k)\}$ pool all trajectories where t^s_k, t^e_k are staring frame and ending frame respectively. τ_0 collects all boxes not belonging to any trajectory. We aim to solve the following representation:

$$W = (K, V, \tau) \tag{1}$$

which formulates 3D localization and cross-view human tracking jointly.

We search for the optimal solution by maximizing a posterior:

$$p(W|I) \propto \exp\{-E(W, I)\} \tag{2}$$

where I is the input video sequences, $E(W, I)$ the energy function . We define $E(W, I)$ over the following aspects:

$$E(W, I) = E^{\text{pri}} + E^{\text{reg}} + \sum_v \{E^{\text{app}}_v + E^{\text{sloc}}_v + E^{\text{track}}_v\} + E^{\text{spline}} \tag{3}$$

where $E^{\text{pri}} = \exp\{-K\}$ is used to encourage the compactness.

In the rest of this section, we first introduce scene model used and then introduce other probabilistic terms.

Scene Model

We mainly consider crowded scenes in surveillance system, e.g. parking-lot, office etc., which include frequent intersections and heavy occlusions. For each scene, there are 5-6 cameras mounted on top of wall/building, sharing field of view (FOVs) with each other. All cameras are hold horizontally w.r.t. the ground. Each scene is provided with a top-view scene map, i.e. cropped Google Earth maps for outdoor scenes, or floorplan for indoor scenes.

We manually extract the following scene information. i) Ground-plane region in images; ii) cross-view homograph matrix from one view to another view, denoted as $\mathbf{H}^{u,v} \in R^{3\times3}$ where u, v index the cameras; iii) projection matrix that transforms a 3D coordinate into a view, denoted as $\mathbf{M}^u$; iv) view-to-map homograph matrix between each view (ground region) and the scene map, denoted as $\mathbf{H}^u$, being slightly misused to facilitate notations.

Objective: cross-view Registration

The term E^{reg} is defined over cross-view registration errors. Let $< (x^u_i, y^u_i), (x^v_i, y^v_i) >$ indicate the footprints of the same subject in the views u and v, respectively. We have.

$$E^{\text{reg}} = \sum_{u,v} \sum_i \|\mathbf{H}^{u,v} \circ (x^u_i, y^u_i) - (x^v_i, y^v_i)\| \tag{4}$$

where $\mathbf{H}^{u,v}\circ$ indicates an operator that applies a cross-view homograph matrix to register the input coordinate in a view into another view. Eq. (4) is used to minimize sum of cross-view registration errors.

Objective: Appearance

The term E^{app} is defined to encourage that appearance similarities of detected human boxes belonging to the same trajectory and appearance discrepancies between trajectories should be both maximized. Formally, for a view v, let i index the detected boxes, c_i index the trajectories, f_i appearance feature (e.g. color, gradient). We have the following appearance energy term,

$$E_v^{\text{app}} = -\sum_{i,j} \log \frac{P(c_i = c_j, \|f_i - f_j\|)}{P(c_i \neq c_j, \|f_i - f_j\|)} \tag{5}$$

where $\|f_i - f_j\|$ indicates the norm of feature distance between f_i and f_j, $P(c_i = c_j, \|f_i - f_j\|)$ denotes the probability of $c_i = c_j$ given the feature measurement. We adopt the nearest neighbor method Hoiem, Efros, and Hebert (2005) to estimate the likelihood ratio from the current tracking results. Note that we extract appearance from individual views, instead of cross-view, because the same patterns (e,g. t-shirt) could have arbitrarily different appearances while being observed from different cameras. We rely on 3D locations, rather than appearance, to register human boxes as introduced later.

Objective: 3D Localization

The term E_v^{sloc} is defined over 3D localization of human boxes, following three observations. *Firstly*, the size of a head box, once detected Rothrock and Zhu (2013), usually linearly reflects the relative distance between a subject to the camera. *Secondly*, binary spatial relationship between two detected human boxes, e.g. being left or right in images, is not sensitive to occlusions to some extent. *Thirdly*, if we can project a 3D cubic of standard human size (e.g., 1.8 meter tall, 0.6 meter width/length) into a view, the simulated box should at least overlap with the detected box at the same position.

We localize all the detected boxes jointly that appears in same image. Let v indicate the camera, (x_i, y_i) denote the desired footprint of the box i. Let (X_i, Y_i) denote the 3D location of box i. We have $(X_i, Y_i) = \mathbf{H}^{v^{-1}} \circ (x_i, y_i)$. We define another operator: $\mathrm{box_i} = \mathbf{M}^{\mathrm{v}} \star (\mathrm{X_i}, \mathrm{Y_i})$ that applies the projection matrix $\mathbf{M}^v$ to project a 3D cubic of average person size standing at (X_i, Y_i) into the view v. Let $\mathrm{b\bar{o}x}_i$ denote the detected box with center position $(\bar{x}_i, \bar{y}_i)$ and head-size $\bar{s}_i$.

For each image, we measure the detected boxes and collect two sets of pairings of boxes:

- $U^{\text{left}} = \{(i,j)\}$ where the detected box i locates on the left of the box j and the x-components of their footprints should satisfy: $x_i < x_j$;
- $U^{\text{size}} = \{(i,j)\}$ where the box i is smaller than the box j, i.e., $\bar{s}_i < \bar{s}_j$ and thus the y-components of their footprints should satisfy: $y_i < y_j$.

Thus, we formulate 3D localization as minimizing the following objective:

$$E_{t,v}^{\text{sloc}} = \sum_i \|(x_i, y_i) - \mathcal{R}(\bar{x}_i, \bar{s}_i))\|^2 + \lambda^l \sum_{<i,j>\in U^{\text{left}}} [x_i < x_j]$$
$$+\lambda^s \sum_{<i,j>\in U^{\text{size}}} [y_i < y_j] + \lambda^o \sum_i [\pi(\mathrm{box}_i, \mathrm{b\bar{o}x}_i) = 0] \tag{6}$$

where the operator $[\cdot]$ returns 1 if the inside expression is true. $\pi(box_i, b\bar{o}x_i)$ returns 0 if two boxes overlap. The linear function $\mathcal{R}(\bar{x}_i, \bar{s}_i) \rightarrow (x_i, y_i)$ returns the estimated footprint. To obtain $\mathcal{R}()$ for each view, we apply the operators $\circ$ and $\star$ at every point on the ground region to generate the population of human boxes.

Objective: Spline Fitting

We describe each moving subject as a continuous spline function of 3D locations w.r.t. time t. Let $\vec{X}_i^k = (X_i^k, Y_i^k)$ denote the 3D location of the subject k at the time t_i, $B_l(t)$ denote the quadratic basis function Eilers and Marx (1996), t_k^s and t_k^e index the frame when a person appears or disappears, respectively. Our goal is to solve a function $\tau : t_i \rightarrow \vec{X}_i^k$ that can be written as a linear combination of B_l:

$$\tau_k(t) = \sum_l \alpha_l^k B_l(t), \quad s.t., \tau_k''(t_k^s) = \tau_k''(t_k^e) = 0 \tag{7}$$

where $\tau_k''(t)$ is the second derivation, and the constraints enforce that curvature at time t_k^s and t_k^e equals zero. Following Eq. (7), the objective of spline fitting can be formulated as follows:

$$E^{\text{spline}} = \sum_k \sum_i |\vec{X}_i^k - \sum_l \alpha_l^k B_l^k(t_i)|^2 \tag{8}$$

Eq. (8) is used to learn a parametric representation for each subject from a set of noisy or incomplete 3D points. This is possible with the enforcement of high-order smoothness.

Objective: Temporal Consistency

We exploit binary spatial constraints for tracking purpose. The basic idea is to encourage consistency of pair-wise relationships between human boxes over time. Formally, let $\tau_i(t)$ and $\tau_j(t)$ denote the 3D positions of the subjects i and j at the time t, respectively. Let $\tau_j^v(t) = (\mathbf{H}^v)^{-1} \circ \tau_j(t)$ denote the footprint in the view v. For each view v, we pool all the pairs of boxes $< i, j >$ at the time t if box i locates on the left of box j, denoted as $< v, t, i, j > \in V^{\text{left}}$. Similarly, we collect the box pairs$< i, j >$ for head box i being larger than head box j, denoted as $< v, t, i, j > \in V^{\text{size}}$. Thus, we have the objective of temporal consistency defined as follows:

$$E_v^{\text{track}} = \sum_{<v,t,i,j>\in V^{\text{left}}} \mathbf{1}^{\text{left}}(< v, t+1, i, j >) \tag{9}$$
$$+\lambda^{\text{width}} \sum_{<v,t,i,j>\in V^{\text{size}}} \mathbf{1}^{\text{size}}(< v, t+1, i, j >) \tag{10}$$

where $\mathbf{1}^{\text{left}}(< v, t+1, i, j >)$ returns 0 if the box i locating on the left of the box j at the time $t+1$, and 1 otherwise. Similarly, $\mathbf{1}^{\text{size}}(< v, t+1, i, j >)$ returns 0 if the head box i is larger than the head box j at the time $t+1$, and 1 otherwise.

Algorithm 1 Algorithm for multi-view 3D human tracking

```
1: Input: multi-view image sequences;
2: Initialization: scene calibration (Section 5.1); detect human
   boxes and head boxes;
3: Build an adjacent graph and
  3.1 Initialization: 3D localization;
  3.2 Initialization: cross-view box identification;
  3.3 Initialize 3D tracking Einicke and White (1999);
4: Iterate until convergence,
  4.1 Randomly choose the Dynamic I or II to get the new solution W';
  4.2 Top-down: 3D trajectory re-projection;
  4.3 Bottom-to-up: robust spline fitting by the Perturb-and-
      Fitting method;
  4.4 Accept $W'$ with a probability $\alpha(W \to W')$;
```

Inference

We develop an iterative procedure based on cluster sampling technique Tu and Zhu (2002) to maximize a posterior probability $p(\mathbf{I}|W)$.

Our method starts with an initial solution and then simulates a Markov Chain in solution space by a set of dynamics. We first build an adjacent graph that takes human detection boxes as graph nodes. We link two nodes in the same frame or consecutive frames if they are spatially close Liu, Lin, and Jin (2013). There are two types of dynamics: I) sampling the number of trajectories K from a Gaussian distribution learnt from the training data; II) assigning each graph node to one of K trajectories or false alarm. The new solution is accepted with a probability. Let W denote the current solution status, and W' the new status, the acceptance probability is defined following the Metropolis-Hasting strategy Tu and Zhu (2002):

$$\alpha(W \to W') = \frac{q(W \to W')p(W'|I)}{q(W' \to W)p(W|I)} \tag{11}$$

where $q(W \to W')$ is the proposal probability. For the Dynamic I, $q(W \to W')$ is fixed to be a constant.

Algorithm 1 summarizes the sketch of our method. In step 3.3, we utilize the extended Kalman Filtering Einicke and White (1999) to get the initial 3D tracking results. In the rest of this section, we introduce the details of the initialization, dynamics and bottom-to-up/ top-down computations.

Initialization: Scene Calibration, 3D localization, Cross-view Identification

We *calibrate* a scene with multiple cameras as follows.

- Utilize the Deep Neural Network Wang et al. (2014) method to segment video frame into semantic regions (e.g. ground, tree).
- Extract key points from the ground region, and match to other views to obtain cross-view correspondences. Then, we solve the objective function of Eq. (4) to minimal to get the cross-view homographs. Eq. (4) is a nonlinear least square problem, which can be efficiently solved by the Levenberg-Marquardt algorithm Pujol (2007).

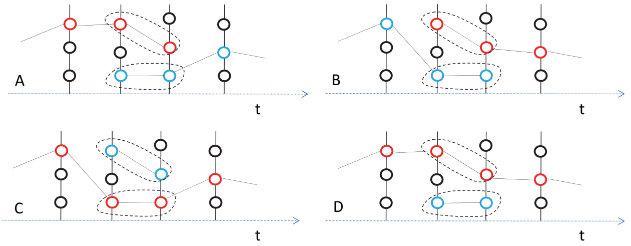

Figure 1: Four atomic status for graph coloring. Each blob represents a 3D location.

- Manually annotate view-to-map homograph for one of the cameras, from which we could obtain view-to-map homographs for all other cameras (by multiplying the cross-view homographs).
- For each view, we calculate the projection matrix, denoted as $\mathbf{M}^v$ by using $\mathbf{H}^v$ and a set of detected human boxes which are assumed to be the projections of a human of 1.8 meter tall.

We *localize* a detected human box in 3D scene map by optimizing Eq. (6) which aims to infer both footprints $\{(x_i, y_i)\}$ and 3D locations for each box input. This leads to a least square problem with three bound constraints (i.e. the last three terms), which can be efficiently solved by the Trust-region based method proposed by Coleman et al. Coleman and Li (1996).

Once human boxes localized in 3D scene, we *identify* two boxes in different views as the same subject if their 3D distance is less then a threshold (e.g. 0.5 meter). We simply prune the isolated boxes that are not matched to any boxes in other views. This step may lead to miss some detections. We shall deal with this issue in other bottom-to-up computations.

Dynamic-II: Graph based Data Association

The dynamic-II is used to reconfigure the current result as follows.

Firstly, we link every two consecutive nodes belonging to the same subject, and compute a local probability, denoted as q_e, for how likely two nodes (i.e. human boxes) belong to the same subject or same color. We set the edge probability as the feature likelihood, i.e. $q_e \propto \sum_v \log P(c_i = c_j, \|f_i^v - f_j^v\|)$) where f_i^v is the feature descriptor extracted from the region of subject i in the view v.

Secondly, we turn off edges at random according to their associated probability to form a set of connected components (CCPs).

Last, we choose one of the CCPs and assign the selected nodes to a new color(or new subject) to change the solution status as follows: if there is no lifespan overlapping between the selected CCP and the existing nodes of the new color, we simply add edges to link them together [1]; otherwise, we

[1] For a trajectory, if there is no box assigned at a time, we simply add a virtual node.

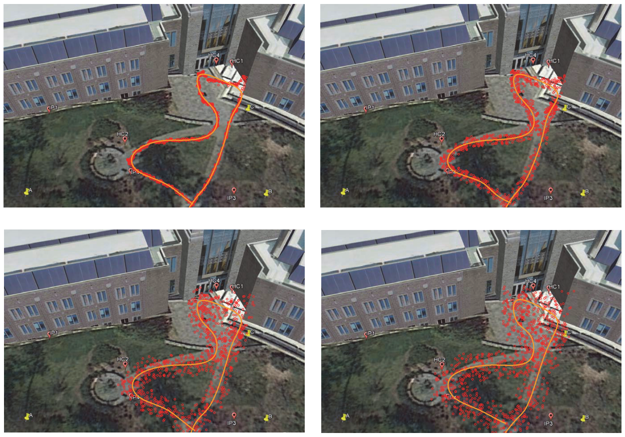

Figure 2: Spline fitting under different levels of noises (i.e. $2\sigma = 0.6, 1.2, 2, 3$ meters).

will randomly choose one of the following moves, as Fig. 1 illustrates: $A \leftrightarrow B$ or $C \leftrightarrow D$, *switch*; $A|B \rightarrow C$ or $A|B \rightarrow D$, *merge*; $C \rightarrow A|B$ or $D \rightarrow A|B$, *split*. Herein, $A|B$ means A or B.

The proposal probability for the Dynamic II is calculated as $q(W \rightarrow W') = \prod_{e \in \mathrm{C}} q_e$, where C denotes the edges turned off around the selected CCP.

Bottom-to-up: Robust Spline Fitting

We introduce a robust Perturb-and-Fitting algorithm to interpolate 3D points that belong to the same subject. Our method is motivated by the RANSAC method and the work by Papandreou and Yuille Papandreou and A.Yuille (2010). It starts with randomly selecting part of the input 3D points as inliers and alternates the following steps.

- Inject random noises into each point $(X_i, Y_i) = (X_i, Y_i) + \sigma * (rand() - 1)$ where $rand()$ returns a random value between 0 and 1 and 2σ is the maximal noise allowed (e.g. 2 meters);
- Solve the optimal spline by minimizing Eq. (8) w.r.t α_l^k, which has closed form solution;
- Identify in-liner points according to the fitting errors by the newly solved spline model.

We alternate these three steps until convergence. Once the optimal spline model solved, we can apply the operator $\mathbf{M}^v \star$ to estimate the 2D box sequence.

Fig. 2 shows several exemplar results of the Perturb-and-Fitting algorithm under different levels of noises . We use the groundtruth 3D locations of a moving person and add four different levels of noises, i.e. $2\sigma = 0.6, 1.2, 2, 3$ meters. We then apply the Perturb-and-Fitting method to interpolate these noisy 3D points. These results show that our fitting algorithm can work robustly against noises.

Top-down: 3D Trajectory Re-projection

We project the learnt 3D trajectory splines into individual views to get the refined footprints or recover the missing detections. This involves the operators: $\circ$ and $\star$. In this work, we assume all persons are standing and thus share the roughly same height in 3D space. Note that we might use more cubic models to handle other gestures, e.g. sitting, crawling, etc.

Experiments

We apply the proposed 3D tracking method over crowded scenes and compare to other popular methods.

Dataset We use two datasets. *Dataset-1* is collected by ourself, including three scenes: parking lot, garden, and office areas. There are 8, 6, and 10 cameras mounted on top of building or wall, respectively. For each scene, there are two groups of cameras and each group shares the same FOV. For each camera, there is one video sequence of 8-10 minutes long. we annotate human boxes using the toolkit developed by Carl et al. Vondrick, Patterson, and Ramanan (2012). There is a total of 254 moving subjects in these videos. t*Dataset-2* is collected by Berclaz et al. Berclaz et al. (2011), known as EPFL dataset, which includes five scenes. For each scene, there are 3-5 cameras and each video is about 3-5 minutes long.

We implement Algorithm 1 as follows. We assign two boxes in different views to the same subject if they locate within 0.5 meter in 3D space. To apply the operator $\star$, we approximate a human body with a 3D cubic with 0.6 meter length/width and 1.8 meters tall. For each view, we train a linear regression model $\mathcal{R}$ from the simulated bounding boxes. We set the tuning parameters (i.e. λs) empirically for each scene and fix them throughout the evaluations. For spline fitting, we fix the number of knot points to be 15, and the random noises level be $2\sigma = 0.6$ meters. To handle the streaming videos, we run Alg. 1 over a window of 200 frames and slide it forward at the step of 20 frames. To estimate the appearance likelihood or the q_e, we utilize the same features as Hoiem, Efros, and Hebert (2005), including color, texture and gradients. For each window, we utilize the results from the previous window as initial solution. Alg. 1 usually converges within 1000 iterations. On an DELL workstation (with 64GB memory, i7 CPU @2.80GHz, and NVIDIA Tesla K40 GPU), our algorithm can process on average 15 frames per second.

Baseline We compare our method to two recent multi-view trackers: i) the K-shortest Path (KSP) method by Fleuret et al. Berclaz et al. (2011); ii) the multi-view SVM method (mvSVM) by Zhang et al. Zhang et al.. We also implemented several single-view based human trackers for comparisons, including: iii) The local sensitive histogram based tracker (LSH) He et al. (2013). iv) the discrete-continuous tracking (DCT) method proposed by Andriyenko et al. Andriyenko and Schindler (2011); v) the occlusion geodesic (Geodesic) based tracker Possegger et al. (2014). We use the default parameter configuration in their source codes. We also report the initial 3D tracking results of our method, i.e. that by the step 3.2 in Alg. 1, denoted as *mvKL*.

Metric We evaluate the proposed method from two aspects, tracking and 3D human localization. For *tracking*, we project the obtained 3D trajectories into each view and

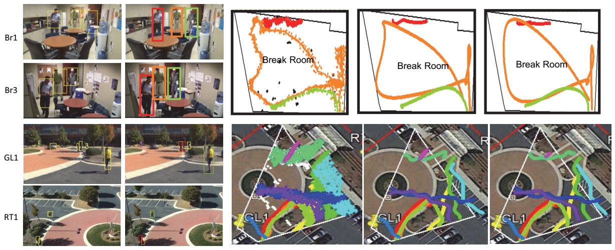

Figure 3: Results of 3D human tracking. Column 1: initial detections; Column 2: refined boxes; Columns 3-5: scene maps overlaid with the estimated 3D points, recovered 3D trajectories, and groundtruth 3D trajectories, respectively.

Table 1: Quantitative tracking results on the newly built Dataset-1.

Metrics	R	P	FAF↓	MT	ML↓	MOTP	IDS↓
DCT Andriyenko and Schindler (2011)	52.4%	54.3%	2.1	69.4%	18.8%	63.5%	85
AVT Yang, Yuan, and Wu (2007)	63.5%	64.1%	1.5	78.8%	17.2%	72.1%	71
LSHT He et al. (2013)	62.1%	60.7%	1.3	70.6%	15.3%	71.8%	79
GeodesicPossegger et al. (2014)	64.2%	66.1%	1.2	74.2%	14.5%	70.4%	73
KSP Berclaz et al. (2011)	71.6 %	73.4%	1.1	74.3%	14.1%	71.6%	59
mvSVM Zhang et al.	68.5 %	71.8%	1.3	72.7%	15.9%	76.8%	82
mvKL	72.5 %	78.3%	0.8	76.7%	13.1%	82.6%	67
Ours	**81.4**%	**85.0**%	**0.3**	**81.2**%	**10.2**%	**87.1**%	**21**

Table 2: Quantitative tracking results on the Dataset-2: EPFL Berclaz et al. (2011).

Metrics	R	P	FAF↓	MT	IML↓	MOTP	IDS↓
DCT Andriyenko and Schindler (2011)	62.4%	69.6%	1.4	68.1%	16.2%	67.8%	61
AVT Yang, Yuan, and Wu (2007)	73.3%	72.8%	1.6	70.4%	14.1%	72.6%	53
LSHT He et al. (2013)	69.4%	67.2%	2.1	68.3%	15.4%	71.3%	48
GeodesicPossegger et al. (2014)	73.2%	72.1%	1.1	69.2%	15.2%	70.1%	41
KSP Berclaz et al. (2011)	78.6%	76.1%	1.2	75.3%	14.3%	72.1%	25
mvSVM Zhang et al.	79.5%	75.3%	1.1	76.3%	12.9%	78.8%	34
mvKL	80.2%	82.3%	0.7	78.3%	12.3%	85.2%	25
Ours	**84.7**%	**87.2**%	**0.4**	**84.5**%	**8.3**%	**88.4**%	**16**

Table 3: Average 3D localization errors (meter) on Dataset-1.

	Parking-lot	Garden	Office
DCT Andriyenko and Schindler (2011)	2.13	2.54	N/A
AVT Yang, Yuan, and Wu (2007)	2.12	1.89	N/A
LSHT He et al. (2013)	2.11	2.13	N/A
Geodesic Possegger et al. (2014)	2.40	1.45	N/A
KSP Berclaz et al. (2011)	1.42	1.16	N/A
mvSVM Zhang et al.	1.21	1.22	N/A
mvKL	0.83	0.72	0.64
Ours	**0.61**	**0.52**	**0.33**

compare with ground-truth trajectories. We adopt the metrics in Yang and Nevatia (2012), including: **R** , recall rate, number of correctly matched detections over total number of ground-truth detections; **P**, precision rate, number of correctly matched detections over total number of output detections; **FAF**$^{\downarrow}$, average false alarms per image; **MT**, *mostly tracked*, percentage of ground truth trajectories which are covered by tracker output for more than 80% in length; **ML**$^{\downarrow}$, *mostly lost*, percentage of ground-truth trajectories which are covered by tracker output for less than 20% in length; **IDS**$^{\downarrow}$, *ID Switch*, the number of times that an object trajectory changes its matched id. **MOTP**, *multi object tracking precision*, the average ratio of the spatial intersection divided by the union of an estimated object bounding box and the ground-truth bounding box. $^{\downarrow}$ indicates that a metric is better if smaller. For *3D localization*, we report the box-wise localization errors (in meters).

Fig. 3 visualizes a few results by Algorithm 1. We can observe the following. i) Although human detectors usually generate truncated boxes or even false alarm detections, our algorithm can successfully predict 3D locations and other attributes (e.g. 2D box) for the detected human box. ii) The proposed perturb-and-fitting method is capable of sampling points from the true trajectories, as shown in Fig. 3 (c). iii) The recovered 3D trajectories are fairly close to the groundtruth trajectories. Note that the white or black points in Column 3 belong to τ_0, i.e. false alarm 3D points.

Tables 1 and 2 report numeric results of all methods on two datasets, respectively. Among these baselines, mvSVM Zhang et al. and KSP Berclaz et al. (2011) are two widely used multi-view tracking methods, while other four trackers are working on individual cameras. From the results, we have following two observations. i) The proposed method clearly outperforms these baselines. In particular, our method generated much less false alarms than other methods: our method achieves FAF of 0.3 in the Dataset-1, while the best score among the baselines is 1.1. ii) mvKL can achieve reasonably good results with the help of multi-view setting and 3D localization module, and the iterative procedure in Alg. 1 can further boost performance.

Table 3 reports the numerical 3D Localization results over the Dataset-1. For the baselines, we takes the bottom-central points of human boxes as footprints and smooth them by averaging over consecutive frames. Then we apply the operator $\circ$ to get the 3D locations. Our method can achieve sub-meter level 3D Localization for both outdoor and indoor scenarios. In particular, for parking-lot and garden, the average 3D localization errors are 0.61 meter and 0.52 meter respectively, which are much lower than the best baselines (1.21m and 1.16m, respectively). For the indoor offices, all baseline methods can not work because most of the subjects are partially occluded by furniture. In contrast, the proposed method can exploit binary constraints to predict footprints. For the office areas, the average localization error of our method is 0.33 meter . These high-quality localization results are the key to the success of our 3D human tracker.

Conclusions

This paper presents a probabilistic method for accurate 3D localization and cross-view tracking in crowded scenes. We introduce a set of simply yet effective binary constraints for localizing truncated human boxes in 3D scene map and separating moving subjects. These constraints are directly exploited in the proposed Bayesian formula in order to address occlusions, frequent intersections, missing detections and other challenges that might fail 3D tracking. We evaluated our method on both public benchmarks and a newly built video dataset. Results with comparisons to other popular methods showed that our method can achieve state-of-the-art 3D tracking performance.

Acknowledgements

This work was supported by the SDSU Faculty Start-up funds. The data used in this work was provided by the Center for Vision, Cognition, Learning and Autonomy (VCLA) at the University of California at Los Angeles (UCLA).

References

Andriyenko, A., and Schindler, K. 2011. Multi-target tracking by continuous energy minimization. In *CVPR*.

Barbu, A., and Zhu, S. 2007. Generalizing swendsen-wang to sampling arbitrary posterior probabilities. *TPAMI*.

Berclaz, J.; Fleuret, F.; Turetken, E.; and Fua, P. 2011. multiple object tracking using k-shortest paths optimization. *TPAMI*.

Coleman, T., and Li, Y. 1996. An interior, trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization* 6:418–445.

Eilers, P., and Marx, B. 1996. Flexible smoothing with b-splines and penalties. *Statistical Science* 11(2):89–121.

Einicke, G., and White, L. 1999. Robust extended kalman filtering. *TSP* 47(9):2596–2599.

Fan, J.; X.Shen; and Wu, Y. 2013. What are we tracking: A unified approach of tracking and recognition. *TIP* 22(2):549–560.

Fleuret, F.; Berclaz, J.; Lengagne, R.; and Fua, P. Multi-camera people tracking with a probabilistic occupany map. *TPAMI*.

Gupta, A.; Mittal, A.; and Davis, L. 2008. Constraint integration for efficient multiview pose estimation with self-occlusions. *TPAMI* 30(3):493–506.

He, S.; Yang, Q.; Lau, R.; Wang, J.; and Yang, M. 2013. Visual tracking via locality sensitive histograms. In *CVPR*, 2427–2434.

Henriques, J.; Caseiro, R.; and Batista, J. 2011. Globally optimal solution to multi-object tracking with merged measurements. In *CVPR*.

Hofmann, M.; Wolf, D.; and Rigoll, G. 2013. Hypergraphs for joint multi-view reconstruction and multi-object tracking. In *CVPR*.

Hoiem, D.; Efros, A.; and Hebert, M. 2005. Geometric context from a single image. *ICCV*.

Hoiem, D.; Efros, A.; and Hebert, M. 2006. Putting objects in perspective. In *CVPR*.

Khan, S., and Shah, M. 2006. A multiview approach to tracking people in crowded scenes using a planar homography constraint. In *ECCV*.

Khan, Z.; Balch, T.; and Dellaert, F. 2005. Mcmc-based particle filtering for tracking a variable number of interacting targets. *TPAMI* 27(11):1805–1819.

Ladicky, L.; Shi, J.; and Pollefeys, M. 2014. Pulling things out of perspective. In *CVPR*.

Leal-Taixe, L.; Pons-Moll, G.; and Rosenhahn, B. 2013. Branch-and-price global optimization for multi-view multi-object tracking. In *CVPR*.

Lin, L.; Xu, Y.; Liang, X.; and Lai, J. 2014. Complex background subtraction by pursuing dynamic spatio-temporal models. *TIP* 23(7).

Liu, X.; Lin, L.; and Jin, H. 2013. Contextualized trajectory parsing via spatio-temporal graph. *TPAMI*.

Milan, A.; Schindler, K.; and Roth, S. 2013. Detection- and trajectory-level exclusion in multiple object tracking. In *CVPR*.

Mittal, A., and Davis, L. 2003. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *IJCV*.

Papandreou, G., and A.Yuille. 2010. Gaussian sampling by local perturbations. In *NIPS*, 1858–1866.

Pirsiavash, H.; Ramanan, D.; and Fowlkes, C. 2011. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 1201–1208.

Possegger, H.; Mauthner, T.; Roth, P.; and Bischof, H. 2014. Occlusion geodesics for online multi-object tracking. In *Proc. CVPR*.

Pujol, J. 2007. The solution of nonlinear inverse problems and the levenberg-marquardt method. *Geophysics* 72.

Rothrock, B., and Zhu, S. 2013. Integrating grammar and segmentation for human pose estimation. In *CVPR*.

Tu, Z., and Zhu, S. 2002. Image segmentation by data-driven markov chain monte carlo. *TPAMI* 24(5):657–673.

Utasi, A., and Benedek, C. 2011. A 3-d marked point process model for multi-view people detection. In *CVPR*.

Vondrick, C.; Patterson, D.; and Ramanan, D. 2012. Efficiently scaling up crowdsourced video annotation. *IJCV*.

Wang, B.; Wang, G.; Chan, K.; and Wang, L. 2014. Tracklet association with online target-specific metric learning. In *CVPR*.

Wu, Z.; Hristov, N.; Hedrick, T.; Kunz, T.; and Betke, M. 2009. Tracking a large number of objects from multiple views. In *ICCV*.

Yang, B., and Nevatia, R. 2012. An online learned crf model for multi-target tracking. In *CVPR*.

Yang, M.; Liu, Y.; Wen, L.; You, Z.; and Li, S. 2014. A probabilistic framework for multitarget tracking with mutual occlusions. In *CVPR*.

Yang, M.; Yuan, J.; and Wu, Y. 2007. Spatial selection for attentional visualtracking. In *CVPR*, volume 1, 1–8.

Yu, Q.; Medioni, G.; and Cohen, I. 2007. Multiple target tracking using spatio-temporal markov chain monte carlo data association. In *CVPR*, 1–8.

Zhang, S.; Yu, X.; Sui, Y.; Zhao, S.; and Zhang, L. object tracking with multi-view support vector machines. *TMM*.

Zhang, L.; Li, Y.; and Nevatia, R. 2008. Global data association for multiobject tracking using network flows. In *CVPR*.