Multiple Target Tracking in World Coordinate with Single, Minimally Calibrated Camera

Wongun Choi and Silvio Savarese

Department of Electrical and Computer Engineering
University of Michigan, Ann Arbor, USA
{wgchoi,silvio}@umich.edu

Abstract. Tracking multiple objects is important in many application domains. We propose a novel algorithm for multi-object tracking that is capable of working under very challenging conditions such as minimal hardware equipment, uncalibrated monocular camera, occlusions and severe background clutter. To address this problem we propose a new method that jointly estimates object tracks, estimates corresponding 2D/3D temporal trajectories in the camera reference system as well as estimates the model parameters (pose, focal length, etc) within a coherent probabilistic formulation. Since our goal is to estimate stable and robust tracks that can be univocally associated to the object IDs, we propose to include in our formulation an interaction (attraction and repulsion) model that is able to model multiple 2D/3D trajectories in space-time and handle situations where objects occlude each other. We use a MCMC particle filtering algorithm for parameter inference and propose a solution that enables accurate and efficient tracking and camera model estimation. Qualitative and quantitative experimental results obtained using our own dataset and the publicly available ETH dataset shows very promising tracking and camera estimation results.

1 Introduction

Designing algorithms for tracking objects is critical in many applications such as surveillance, autonomous vehicle and robotics. In many of these applications it is desirable to detect moving humans or other targets as well as identify their spatial-temporal trajectories. Such information can enable the design of activity recognition systems for interpreting complex behaviors of individuals and their interaction with the environment. This can also provide crucial information to help an autonomous system to explore and interact with complex environments.

Among the key desiderata of an ideal tracking system researchers have identified the ability to: i) estimate stable and accurate tracks and uniquely associate them to a specific object; ii) associate tracks to 2D/3D-temporal trajectories in the 3D scene; iii) work with the minimal hardware equipment (e.g., single camera-vs-stereo cameras; no laser data); iv) work with a moving camera. Meeting all these desiderata is extremely challenging. For instance estimating stable tracks is difficult as objects are often subject to occlusions (they cross each other

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part IV, LNCS 6314, pp. 553–567, 2010.

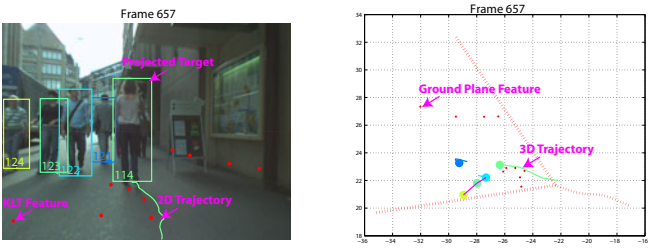

Fig. 1. Example result on ETH Seq.#2. Left: targets(colored bounding boxes) and the trajectories automatically produced by our algorithm in the image plane. Right: estimated location of targets (filled circles) on the 3D coordinate. Our algorithm not only tracks targets in the 3D coordinate system, but also estimates ego-motion of the camera by using ground plane features (red dots) and automatically discovers group of people in the scene (magenta link). Our algorithm is capable to estimate the 3D trajectories from a monocular moving camera.

in the image plane), illumination conditions can change in time, the camera motion can disturb the tracking procedure. Estimating tracks (trajectories) in the 3D world (or camera) reference system is also very hard as estimating 3D world-2D image mapping is intrinsically ambiguous if only one camera is available and camera parameters are unknown. Structure from motion (SFM) techniques are often inadequate to estimate motion parameters in that: i) the reconstruction is noisy and unreliable if small base-line is considered, ii) cluttered dynamic scene elements violate the SFM assumption of static background, iii) the procedure is computationally expensive and can be hardly implemented in real time.

Inspired by the work of [1] wherein a method for integrating multiple cues (such as odometry, depth estimation, and object detection) into a cognitive feedback loop was proposed, we present a new framework for tackling most of the issues introduced above in a coherent probabilistic framework. Specifically, our goals are to: i) solve the multi-object tracking problem by using a single uncalibrated moving camera; ii) handle complex scenes where multiple pedestrians are moving at the same time and occluding each other; iii) estimate the 2D/3D-temporal trajectories within the camera reference system.

The key contribution of our work relies on the fact that we simultaneously estimate the camera parameters (such as focal length and camera pose) and track objects (such as pedestrians) as they move in the scene (Fig.1). Tracks provide cues for estimating camera parameters by using their scale and velocity in the image plane; at the same time, camera parameters can help track objects more robustly as critical prior information becomes available. This, in turn, allows us to estimate object 3D trajectories in the camera reference system. Inspired by [2], we utilize a simplified camera model that allows to find a compact (but powerful) relationship between the variables (targets and camera parameters) via camera projection constraints. The identification of a handful of feature tracks associated with the static background allows us to add additional constraints to the camera model. Eventually, we frame our problem as a maximum-posterior

problem in the joint variable space. In order to reduce the (otherwise extremely) large search space caused by the high dimensionality of the representation, we incorporate MCMC particle filtering algorithm which finds the best explanation in sequential fashion. Notice that, unlike previous methods using MCMC, our method is the first that uses MCMC for efficiently solving the joint camera estimation and multi-target problem.

The second key contribution is that we obtain robust and stable tracking results (i.e. uniquely associate object identities to each track) by incorporating interaction models. Interaction between targets have been largely ignored in the object tracking literature, due to the high complexity in modeling moving targets and the consequential computational complexity. The independent assumption is reasonable when the scene is sparse (only few objects exists in the scene). In a crowded scene, however, the independent motion model often fails to account for the target's deviation from the prediction, e.g. if a collision is expected, targets will change their velocity and direction rapidly so as to avoid a collision. Thus, modeling interactions allows us to disambiguate occlusions between targets and better associate object labels to underlying trajectories. This capability is further enhanced by the fact that our trajectories are estimated in 3D rather than in the image plane. Our interaction models are coherently integrated in the graphical model introduced above.

We validate our theoretical results in a number of experiments using our own dataset [3] and the publicly available ETH dataset [1]. Our dataset contains several sequences of multiple humans observed under challenging conditions (moving camera with shakes, occlusions, etc). Our algorithm shows very promising results (all the results were superior than the detector baseline). Also, we evaluate our system functionalities and report detection rates by turning on and off interaction/repulsion models and camera estimation capabilities. Such results confirm our intuition that both camera models and interaction models play a critical role into the construction of stable tracks. Moreover, experiments with the ETH dataset show that our method outperforms (in terms of tracks detection accuracy) the state-of-the-art results [1]. Anecdotal examples on both datasets demonstrate that our algorithm is capable to estimate the 3D trajectories of multiple targets in the camera reference system as well as estimate the (moving) camera trajectory in a given world reference system.

2 Related Work

Multi-target tracking has received a large amount of interest among computer vision researchers. Tracking algorithms based on appearance infromation [4,5,6] are often able to track targets very well when the scene is uncluttered and the camera is static. However, as the complexity of the scene increases (complex background, crowded scene, etc), these algorithms suffer from the well-known tracker drift problem [7]. Recent improvement in object detection [8,9] makes it possible to apply detection algorithms which can effectively reduce the amount of error accumulated during tracking [10,1,11,12]. However, nearly none of these

algorithms [10,12,11] take advantage of the scene geometry or the interplay between the scene and the camera model to improve the tracking capabilities, especially for pruning out unlikely target trajectories, such as a floating human. Furthermore, methods relying on detection results [8,9] are still prone to high degree of false alarms which result in false trajectory initializations. Not only does this make the system unreliable, but also increases the complexity of the correspondence problem – a critical component of the multi-target tracking algorithm. Recently, [1] proposed a mobile tracking system that can simultaneously carry out detection and tracking by incorporating various sources of information (depth map, visual odometry). While this method demonstrates that putting together cues into a cognitive loop greatly helps reduce the false alarm rate, it leverages on the usage of stereo cameras and other specific hardware components. Multi-target tracking can be also aided by considering interaction between targets [13,14,15]. The usage of such interaction model, however, is mainly limited to the repulsion models which cannot explain the targets moving as a group. Moreover, such interaction models have never been incorporated into framework for simultaneous camera and scene estimation, and object tracking.

3 Multi-target Tracking Model

3.1 Overall Method

Given a video sequence, our goal is to jointly track multiple moving or static targets (e.g. cars, pedestrians), identify their trajectories in 3D with respect to the camera reference system, and estimate camera parameters (focal length, viewing angle, etc). We model each target as a hidden variable Z_i in 3D space whose trajectory in time must be estimated and separated from all other trajectories. We argue that estimating trajectories in 3D is more robust than estimating trajectories in the image plane because we can impose a number of priors in actual 3D space as we shall see next.

Such trajectories in 3D are estimated by measuring their projections onto 2D image plane which represent our observation variables X_i (Fig.2). Given the observations, tracks Z_i in 3D are estimated by jointly searching the most plausible explanation for both camera and all the existing targets' states using the projection characterized by the camera model (Sec.3.3). Clearly the projection from Z_i to observation X_i is a function of camera parameters. Thus, we introduce a simplified camera model (Sec.3.3) which allows us to reduce the number of parameters that are required to be estimated. We assume rough initial camera parameters are given and can be better estimated by the detected targets in the image plane. All camera parameters at time t are collected in the variable Θ_t. Moreover, as an important contribution of our work, we do not assume that targets are moving independently but their motion may be interrelated. Thus we introduce an interaction model which allows us to better estimate the states of all target. Our interaction model is composed of repulsion and attraction model (Sec.3.6). We assume i) targets cannot take the same location in the 3D coordinate and cannot collide with each other (*repulsion model*), and ii) targets that

have moved in a coherent fashion (as a group) up to time t are likely to move as a group after time t as well (*attraction model*).

In our work, we assume that the following information can be extracted from the video sequence: i) Visible targets' location and bounding box can be detected at each frame with some number of false alarms. Target detections are used to initiate tracks and gather evidence for existing targets. We use the state-of-the-art object detector [9] for detecting targets (sec.3.2). ii) Rough trajectories in the image plane are available. This additional piece of information is used as a complementary cue to better locate targets in the image plane and it is useful when the target is not properly detected by the detector. We use mean-shift algorithm [4] to obtain them (sec.3.2). iii) Feature points from static background can also be identified and tracked. Background features help the algorithm estimate the camera parameters' variations in time. For this task, KLT tracker [16] is incorporated in our algorithm. Given above cues, the targets are automatically identified/tracked/terminated using our coherent multi-target model.

3.2 Track Initiation, Termination and Correspondence

Target initiation and termination. As detection results are given by the detector, our multi-target tracking algorithm automatically initiates targets. If there exists a detection that is not matching any track, the algorithm initiates a target hypothesis. If enough matching detections for the hypothesis are found in N_i consecutive frames, the algorithm will recognize the hypothesis as a valid track and begins tracking the target. Conversely, if no enough detections are found for the same target within N_t consecutive frames, the track is automatically terminated.

Correspondence. Target correspondence is a very challenging problem by itself. For simplicity, we use the Hungarian algorithm [17] which is based on the overlap ratio between existing targets and detections. We employ two independent sources of information to solve the correspondence problem: affinity matrices of prediction and appearance tracking. The first one is constructed using the image plane prediction of i^{th} target $\hat{X}_{it} = E[X_{it}|Z_{i(t-1)}, \Theta_{t-1}]$ in time t, where t indicates the time dependency at instant (time stamp) t. By computing the negative log of pairwise overlap ratio between the predictions $\hat{X}_{it}$ and detections X_{jt}, $A(X_{it}, X_{jt}) = -log(\frac{Intersection(X_{it}, X_{jt})}{Union(X_{it}, X_{jt})})$, we construct a pairwise affinity matrix between detections and predictions. The second one leverages on the mean-shift tracker [4] (this cue is also used in the estimation). When a new target hypothesis is created, an individual mean-shift tracker is assigned to each target and applied to each frame until the target tracking is terminated. The appearance model (color histogram) is updated only when there is a supporting (matching) detection to avoid tracker-drift. Similarly to the prediction-detection affinity matrix, we compute another affinity matrix between mean-shift output Y_{it} and detections. Given the two affinity matrices, we sum the two matrices to calculate the final matrix which will be the input of Hungarian algorithm. In following sections, we assume the correspondence is given by this algorithm, so Z_{it} and its observation X_{it} are assumed to be matched.

3.3 Camera Model and KLT Features

Camera Model. Due to the inherent uncertainty in the camera projection matrix, it is very challenging to infer the exact location of an object in 3D given image plane location and camera parameters. To mitigate this problem, we set a number of assumptions on the underlying geometry and camera configuration similarly to [2]. We additionally assume that the camera follows forward motion only. With these assumptions, the camera parameters can be represented by following variables: focal length f_θ, height h_θ, horizontal center point u_θ, horizon position v_θ, panning angle ϕ_θ, absolute velocity r_θ, and 3D location (x_θ, z_θ) with respect to the reference system associated to the initial frame. Thus, the projection function f_P can be defined

$$X = f_P(\hat{Z};\Theta) = \begin{bmatrix} \frac{f_\theta x_z}{z_z} + u_\theta \\ \frac{f_\theta h_\theta}{z_z} + v_\theta \\ \frac{f_\theta h_z}{z_z} \end{bmatrix}, \quad \hat{Z} = f_P^{-1}(X;\Theta) = \begin{bmatrix} \frac{h_\theta(u_x - u_\theta)}{v_x - v_\theta} \\ \frac{f_\theta h_\theta}{v_x - v_\theta} \\ \frac{h_x h_\theta}{v_x - v_\theta} \end{bmatrix}, \quad Z = \begin{bmatrix} R(\phi_\theta) & 0 \\ 0 & 1 \end{bmatrix} \hat{Z} + \begin{bmatrix} x_\theta \\ z_\theta \\ 0 \end{bmatrix} \tag{1}$$

where $X = [u_X, v_X, h_X]^T$ and $Z = [x_Z, z_Z, h_Z]^T$; u_X, v_X, and h_X are the (bottom) center point location and height of an observation in the image plane respectively; x_Z, z_Z, and h_Z are the location and the height of the object in world coordinate. Here $\hat{Z}$ denotes the location of the target in current camera coordinate system, and Z denotes the state of the target in global reference system.

KLT for Camera Motion. In order to track multiple targets reliably, it is crucial to get a good estimate of the camera's extrinsic parameters (panning, location, and velocity). At that end, we use KLT features [16] as additional observations. Suppose we can extract feature points τ_t which are lying on the ground plane. Then by applying the inverse projection f_P^{-1} and forward projection f_P on τ_{t-1} with camera parameters in each time frame Θ_{t-1}, Θ_t, we can obtain the expected location of $\hat{\tau}_t$. By comparing the difference between τ_t and $\hat{\tau}_t$, we can infer the amount of camera's motion in the time. As we will show in the experimental section, this feature improves the tracking performance significantly. This is inspired by SLAM procedures such as [18,19].

3.4 Target Class Model

We use state-of-the-art object detector [9] for detecting targets. Despite its excellent performance, [9] still yields false detections in the challenging experimental setting we work with. In order to differentiate such false detections, we introduce one more multinomial hidden variable c in the target states Z_t, which indicates the object class of the target being tracked. If one target's c variable is set to be 0, then the target is not a valid object and thus can be removed. To guide the algorithm estimate the category of the class, we assign a height prior to each variable describing an object class. This follows a normal distribution with a particular mean and variance. Non-object class are described by a uniform distribution.

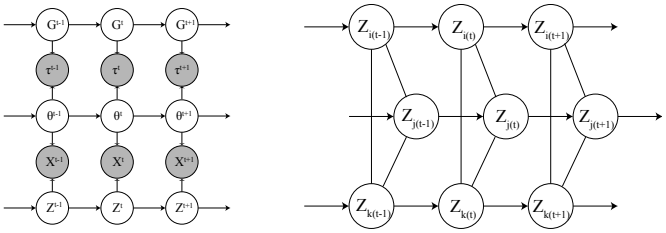

Fig. 2. Graphical model describing underlying model for multi-object tracking. Shaded nodes represent observable variables and empty nodes shows the hidden variables. Panel (a) shows overall relationship between observations X, Y, τ, camera parameters Θ, ground features' state G and targets' states Z. Here, we dropped the mean-shift observation Y on the graph to avoid clutter. Panel (b) shows the interaction between targets (i, j, k). The interaction is modelled by the undirected edges between targets.

Thus, if any observation yields a very unlikely large or small height for a certain target class (such as 1 meter for humans), than the algorithm will automatically reject out this observation. Not only does this help the algorithm to reduce the number of false alarms, but it also helps the camera parameter estimation to be more robust since it essentially rejects out outliers in the estimation process.

3.5 Sequential Tracking Model with Independent Assumption

In this section, we discuss in details the probabilistic relationship between the hidden states $\Omega_{t-1} = [Z_{t-1}, \Theta_{t-1}, G_{t-1}]$ and all the observations $\chi_t = [X_t, Y_t, \tau_t]$. Given the evidences χ_t and the estimates Ω_{t-1} at a previous time stamp, we can compute the posterior distribution $P(\Omega_t|\chi^t)$. Here, we use superscripts for denoting all the history up to time t and subscripts for current time t variables. Notice that Z_t, G_t, X_t, Y_t, and τ_t collects variables for each individual target state, ground feature state, target observation, and feature observation respectively. Following the basic Bayesian sequential model, the posterior distribution $P(\Omega_t|\chi^t)$ can be factorized as follows :

$$P(\Omega_t|\chi^t) \propto P(\Omega_t, \chi_t|\chi^{t-1}) = P(\chi_t|\Omega_t) \int P(\Omega_t|\Omega_{t-1}) P(\Omega_{t-1}|\chi^{t-1}) d\Omega_{t-1} \quad (2)$$

Here, the first term $P(\chi_t|\Omega_t)$ represents the *observation model* and the term $P(\Omega_t|\Omega_{t-1})$ explains the *motion model*. Based on the conditional independence assumption represented in Fig.2, each term can be further factorized into :

$$P(\chi_t|\Omega_t) = P(X_t, Y_t|Z_t, \Theta_t) P(\tau_t|G_t, \Theta_t) \quad (3)$$

$$P(\Omega_t|\Omega_{t-1}) = P(Z_t|Z_{t-1}) P(\Theta_t|\Theta_{t-1}) P(G_t|G_{t-1}) \quad (4)$$

Target Model. The targets' state $Z_t = \{Z_{it}\}_{i=1}^N$ is composed of 6 variables, (x, z) 3D location, (v^x, v^z) velocity, h height, and c class indicator. Given this

parameterization, we formulate the motion model for targets as $P(Z_t|Z_{t-1}) = \prod_{i=1}^{N} P(Z_{it}|Z_{i(t-1)})$ where

$$P(Z_{it}|Z_{i(t-1)}) \propto P(m_{it}|m_{i(t-1)})P(h_{it}|h_{i(t-1)})P(c_{it}|c_{i(t-1)})P(h_{it}|c_{it}) \quad (5)$$

Here, the variables x, z, v^x, v^z were substituted with m to make the notation more compact. The first term $P(m_{it}|m_{i(t-1)})$ is modeled as a simple first order linear dynamic motion model with an additive gaussian noise. The second term $P(h_{it}|h_{i(t-1)})$ is modeled as $P(h_{it}|h_{i(t-1)}) \sim N(h_{i(t-1)}, \sigma_h)$ to allow some degree of variation in height. $P(c_{it}|c_{i(t-1)})$ is modeled as a indicator function $I(c_{it} = c_{i(t-1)})$, since we do not allow the target's class to be changing in time. The targets' height prior, $P(h_{it}|c_{it})$, is represented either as a normal distribution with mean and standard deviation (h_{c_k}, σ_{c_k}) when $c = k$ or as a uniform distribution p_{c_0} when $c = 0$ (no object). In our MCMC particle filter implementation, this uniform-gaussian mixture formulation plays an "outliers-rejection" role similar to RANSAC, since it will "push out" targets from class k which are not consistent with the "consensus" to maximize the posterior distribution. Observations are modeled using the forward projection f_P: $X_{it} = f_P(Z_{it}, \Theta_t) + W$, where W is gaussian noise. Similarly, we assume that mean shift tracker can be modelled as $Y_{it} = f_P(Z_{it}, \Theta_t) + V$, again V is gaussian noise.

Ground feature Model. As stated in Sec.3.3, we use KLT tracker to track stationary features on the ground so as to get a robust estimate of the camera motion. This can be achieved by introducing the hidden state G_{it} which captures the true location of a ground feature in 3D. Let τ_{it} be the ground feature tracked in the image plane at time t, and $\hat{\tau}_{it}$ the projection of G_{it} into the image plane at t. This indicates the expected location of the feature G_{it} at t. G_{it} is composed of three variables x, z, α (its 3D location and a binary indicator variable, such variable encodes whether the feature is static and lies on the ground or not) and τ_{it} have two variables u, v (its location in the image plane). Assuming the ground plane features are static, the motion model of G_{it} will have a simple form of indicator function, $P(G_{it}|G_{i(t-1)}) = I(G_{it} = G_{i(t-1)})$.

The relationship between the state and observation $(P(\tau_t|G_t, \Theta_t))$ can be modelled using the camera projection function f_P if the feature is truely static and lying on the ground plane ($\alpha = 1$). However, if either the feature is moving or the feature is not on the ground plane ($\alpha = 0$), the projection function f_P does not model the correct relationship between τ_{it} and G_{it}. Thus, the observation process is modeled as $P(\tau_t|G_t, \Theta_t) \sim N(f_P(G_{it}, \Theta_t), \Sigma_G)$ if α_i is 1, otherwise $P(\tau_t|G_t, \Theta_t) \sim unif(p_G)$. Similar to the class variable in target model, those features that are not consistent with the majority of other features will be automatically filtered out.

Camera Model. In order to deal with camera motion, we also model camera motion parameters. Note that the camera parameters are coupled with the target and feature observations and cannot be directly observed. The temporal relationship between camera parameters is simply represented as a linear dynamic model $x_{t\theta} = x_{(t-1)\theta} - r_{(t-1)\theta} * sin(\phi_{(t-1)\theta}) * dt$ and $z_{t\theta} = z_{(t-1)\theta} + r_{(t-1)\theta} * cos(\phi_{(t-1)\theta}) * dt$

(we defined the positive value of ϕ for the left direction so there appears minus sign on x_t). We inject uncertainty in the velocity parameter by adding gaussian noise. The uncertainty of the other camera parameters ($f_\theta, h_\theta, u_\theta, v_\theta$ and ϕ_θ) are just modeled as additive gaussian noise.

3.6 From Independent to Joint Target Model

In real world crowded scenes, targets rarely move independently from each other. Targets rarely occupy the same physical space (*repulsion model*). Moreover, once human targets form a group, they typically tend to move together in subsequent time frames (*group model*). In this work, we employ two interaction models between targets (repulsion and group model) to aid the tracking alogrithm. However, since these two interactions cannot occur at the same time, we introduce a hidden variable β_{ijt} that lets us select the appropriate interaction model (mode variable).

The interaction models are modeled as pairwise potentials between current targets' states, thus forming a Markov Random Field as shown on fig.2. Thus the targets' motion model $P(Z_t|Z_{t-1}) = \prod_{i=1}^{N} P(Z_{it}|Z_{i(t-1)})$ can be substituted by: $\prod_{i<j} \psi(Z_{it}, Z_{jt}; \beta_{ijt}) \prod_{i<j} P(\beta_{ijt}|\beta_{ij(t-1)}) \prod_{i=1}^{N} P(Z_{it}|Z_{i(t-1)})$ where $\psi(Z_{it}, Z_{jt}; \beta_{ijt})$ is the pairwise potential.

Mode variable. In order to model transitions between interactions, we describe the transition probability $P(\beta_{ijt}|\beta ij(t-1))$ as p_β if $\beta_{ijt} = \beta ij(t-1)$, and as $1-p_\beta$, otherwise. In our implementation, p_β is set to be 0.9. Again, this variable is automatically estimated given observations. Thus,

$$\psi(Z_{it}, Z_{jt}; \beta_{ijt}) = \begin{cases} \psi_g(Z_{it}, Z_{jt}), \ if \ \beta_{ijt} = 1 \\ \psi_r(Z_{it}, Z_{jt}), \ otherwise \end{cases} \tag{6}$$

Repulsion model. In order to push away targets that are too close, we model the repulsion potential as $\psi_r(Z_{it}, Z_{jt}) = e^{\frac{-1}{c_r r_{ij}}}$ where r_{ij} denotes the distance between two targets in the 3D space and c_r is a parameter controlling the repulsion force between those. This pairwise potential has larger values as two targets are located far away, and has a value closer to 0 when two targets are nearby.

Group Motion Model. The assumption here is that, if two targets are moving together while keeping the same distance (group movement), they will tend to keep the same relative location in consecutive time frames as well. This can be modelled as $p_{it} - p_{jt} \approx p_{i(t-1)} - p_{j(t-1)}$, which is in turn equivalent to $v_{it} \approx v_{jt}$, where p_{it} is the target's location in 3D and v_{it} is the velocity component of Z_{it}. Thus, we model the group motion potential as a $\psi_g(Z_{it}, Z_{jt}) = e^{-c_g * \|v_{it} - v_{jt}\|}$, where c_g is a parameter controlling the similarity of velocities. Since groups of targets are also defiend by the distance among each others, we enforce that the distance r_{ij} between two targets should be close enough in order to be considered as a group. This can be modeled by multiplying ψ_g with a soft step function and obtain $\psi_g(Z_{it}, Z_{jt}) = \frac{1}{1+e^{s_g(r_{ij}-t_g)}} \dot{e}^{-c_g * \|v_{it} - v_{jt}\|}$, where s_g is a parameter regulating the slope of soft step function and t_g is a distance threshold.

4 Tracking Multi-target by MCMC Particle Filter

Considering the complexity of the given probabilistic formulation, it is extremely challenging to design an analytical inference method for estimating the Maximum-a-Posteri solution. This challenge is due to the presence of: 1) the high nonlinearity of projection function(EQ.1); 2) the MRF induced by pairwise potential; 3) the non-gaussian nature of the posterior and prior distribution. Instead of relying on an analytical solution, we employ a sampling based sequential filtering algorithm (the Monte-Carlo Markov-Chain (MCMC) Particle Filter [13]). Inspired by [13], we employ MCMC sampling scheme to propagate the posterior distribution in the particle filtering framework. In each frame, we keep the number of samples without weights and thus approximate the prior distribution with N dirac samples $P(\Omega_{t-1}|\chi^{t-1}) \approx \{\Omega_{t-1}^r\}_{r=1}^N$. Subsequently the final posterior distribution in time t can be approximated by following equation

$$P(\Omega_t|\chi^t) \approx cP(X_t|Z_t,\Theta_t)P(\tau_t|G_t,\Theta_t)\sum_{r=1}^{N} P(Z_t|Z_{t-1}^{(r)})P(G_t|G_{t-1}^{(r)})P(\Theta_t|\Theta_{t-1}^{(r)}) \quad (7)$$

As a condition for the construction of an MCMC method, we need to design a Markov chain over the joint space of Ω. This has the same stationary distribution as the posterior distribution $P(\Omega_t|\chi^t)$. First, we define the *proposal distribution* as a combination of 1) weighted sampling from all existing targets, features, and camera and 2) appropriate random perturbation to the chosen node (additive gaussian or switching state). The proposal density can be represented as follows: 1) sample one hidden state(camera, target, feature) with probability $p_i = \frac{w_i}{\sum_{k=0}^{M} w_k}$. 2) if the camera Θ is selected, sample from a multinomial normal distribution to get new sample $\Theta_t' = \Theta_t^{(s)} + \mu$, where μ is the gaussian sample. 3) If a target Z_i is chosen: i) sample from a multinomial normal distribution and add it to x, z, v_x, v_z, h; ii) switch the class variable c_i by p_c^f; iii) switch interaction mode β_{ijt} for all j by p_β^f. 4) If a feature G_i is selected: i) sample from a multinomial normal distribution and add it to x, z; ii) switch the indicator variable α_i by p_α^f. Here we assign higher weight w_i onto the camera's state since camera parameters are coupled with all states and so this estimation process requires a larger number of trials. Since this proposal distribution is symmetric (the probability to move from Ω_t' to $\Omega_t^{(s)}$ and from $\Omega_t^{(s)}$ to Ω_t' is the same), we can drop the proposal distribution term from the acceptance ratio a. Thus a can be written as :

$$a = \begin{cases} \frac{\prod_{i=1}^{n} P(X_{it}|Z_{it}',\Theta_t')\prod_{i=1}^{m} P(\tau_{it}|G_{it}',\Theta_t')P(\Omega_t'|\chi^{t-1})}{\prod_{i=1}^{n} P(X_{it}|Z_{it}^{(s)},\Theta_t^{(s)})\prod_{i=1}^{m} P(\tau_{it}|G_{it}^{(s)},\Theta_t^{(s)})P(\Omega_t^{(s)}|\chi^{t-1})} & ,when\ the\ camera\ is\ chosen \\ \frac{P(X_{kt}|Z_{kt}',\Theta_t')P(\Omega_t'|\chi^{t-1})}{P(X_{kt}|Z_{kt}^{(s)},\Theta_t^{(s)})P(\Omega_t^{(s)}|\chi^{t-1})} & ,when\ a\ target\ is\ chosen \\ \frac{\prod_{i=1}^{n} P(\tau_{it}|G_{it}',\Theta_t')P(\Omega_t'|\chi^{t-1})}{\prod_{i=1}^{n} P(\tau_{it}|G_{it}^{(s)},\Theta_t^{(s)})P(\Omega_t^{(s)}|\chi^{t-1})} & ,when\ a\ feature\ is\ chosen \end{cases} \quad (8)$$

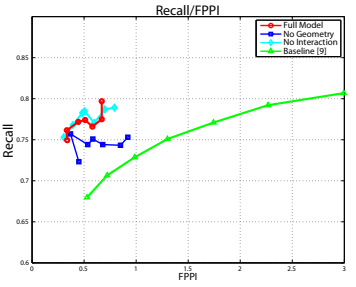

Fig. 3. Our full model obtains the best Recall rates when compared to the baseline detector [9]. Note that the effect of τ is quite significant. To obtain plots, we run the algorithm with different threshold values for the detector.

where $P(\Omega_t|\chi^{t-1}) \approx \sum_{r=1}^{N} P(Z_t|Z_{t-1}^{(r)})P(G_t|G_{t-1}^{(r)})P(\Theta_t|\Theta_{t-1}^{(r)})$. Note that the computation of other observation likelihood is not necessary in the case of sampling a target or ground feature. Even though the computation of prediction term $P(\Omega_t|\chi^{t-1})$ involves several multiplication and summation operations. The majority of target's state remains unchanged during a sampling iteration. This enables an efficient implementation by caching unchanged priors.

5 Experimental Results and Implementation Details

In order to evaluate the performance of our tracking algorithm, we applied it to two datasets: our own dataset and ETH moving vehicle dataset [1]. During all experiments, we assumed rough initial camera configuration is given (focal length, camera height, horizon). Since cameras are not calibrated in our dataset, camera configuration is initialized to some reasonable value. For ETH dataset, we use the calibration information provided by the authors to initiate the algorithm.

Detector. Human targets are detected using the part-based detector [9] which is trained on the VOC 2006 dataset. As a benchmark, we report the recall/FPPI (False Positive Per Image) measure of the detector [9] along with our result in Fig.3.

Mean-shift. Appearance-based tracks are obtained using the mean-shift tracker with a similarity threshold of 0.94 in order to avoid false correspondences.

Feature selection. We extract 1300 KLT features to cover the visible area for every frame. After extracting KLT features, we select candidate ground plane features by rejecting out those lying on a target's prediction area or above the horizon. Among those candidates, a maximum number of 10 features were used in MCMC to reduce the computational burden. In practice these were sufficient to obtain robust estimation of the camera parameters.

MCMC Implementation. In the actual implementation of MCMC particle filter, we incorporated a burn-in and thinning scheme. We ignored the inital

Table 1. The recall/FPPI of ETH[1] algorithm is measured at the point having similar FPPI value to our algorithm. The performance of our algorithm was comparable or superior to ETH algorithm for Seq.#2. Notice in Seq.#3, as the number of false positives increases drastically, our algorithm was not able to correctly differentiate between true and false positives.

Recall/FPPI on ETH dataset			
Method		Seq.#2	Seq.#3
Our Algorithm	Recall	0.556 0.541 0.519	0.339 0.421 0.497
	FPPI	0.792 0.442 0.267	2.792 1.608 0.647
ETH [1]	Recall	0.498 0.404 0.338	0.673 0.616 0.484
	FPPI	0.781 0.431 0.262	2.772 1.593 0.638

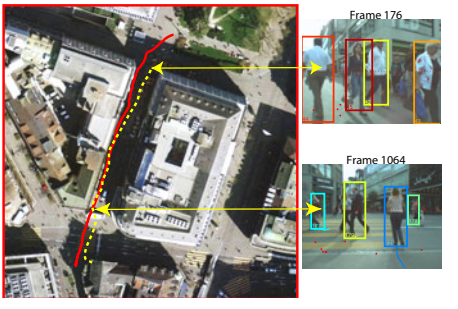

Fig. 4. The camera's trajectory estimated by our algorithm (yellow) for Seq.#2 and by [1] (red). The trajectories are overlaid onto the satellite image by rotating and rescaling with the same factor in (x,z) direction. Notice our trajectory was obtained without using stereo cameras or SFM.

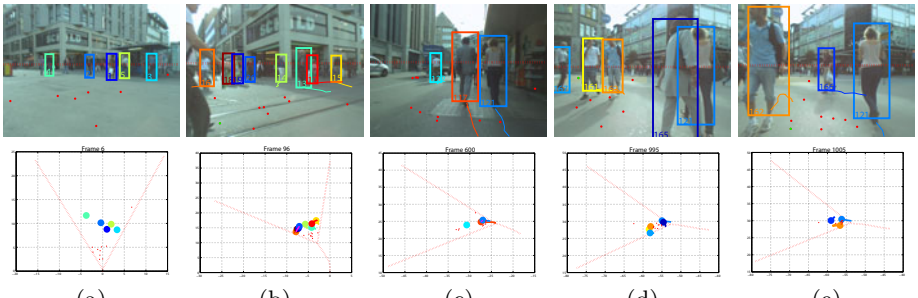

Fig. 5. Examples of tracking subsequences obtained by our algorithm in Seq.#2. Top: trajectories in the image plane; bottom: trjectory estimates in 3D space along with the camera's location and viewing angle estimates.

number of samples to avoid wrong estimation. In each experiment, we set 2000 of burn-in samples and selected one out of 100 samples.

Semi-static Camera. Firstly, we show the performance of our algorithm using our own dataset. Our dataset is ideal to test sequences with semi-static camera mation [3]. It is composed of nine short video sequences recorded at 30FPS by a hand-held video camera. These contain random shakes, sudden camera panning, and multiple number of pedestrians. The dataset contains 4749 frames and 3685 pedestrian annotations in total (every 10th frame is manually annotated). In our experiment, we ignored every other frame, so the tracking algorithm is applied at 15 fps. We report the quantitative measure of the performance by the recall/FPPI rate. Fig.3 shows the overall recall/FPPI curve obtained by our algorithm and the baseline detector [9]. In order to evaluate the effect of the tracked ground features and interaction model to the final performance, we show recall/FPPI curves when no features τ were tracked (blue) and no interaction models were used (cyan).

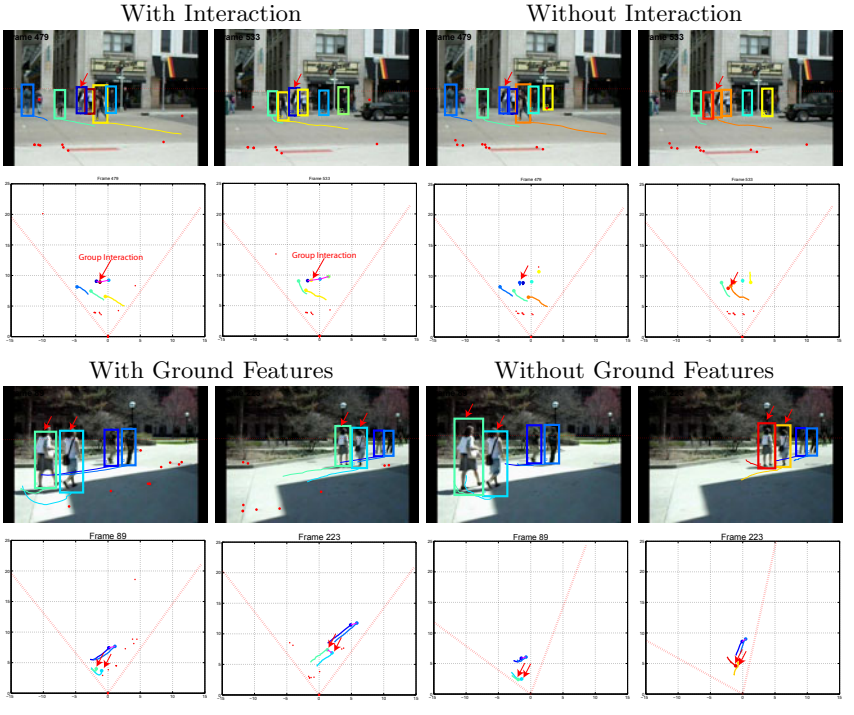

Fig. 6. Tracking comparison. Upper two rows: example of tracking results with and without the interaction model. Note that the group interaction (magenta link) prevents possible ID switch (red arrows) between two similar targets after occlusion. Bottom two rows: tracking results with and without ground features. Ground features not only helps the algorithm estimate the camera motion robustly but also generates better trajectories. Note that the ID of two targets (red arrows) are not maintained (if ground features are not used) due to poor camera motion estimation.

ETH dataset. To show the versatility of our algorithm, we also applied our algorithm on the ETH dataset [1]. ETH dataset is taken by a stereo pair of cameras mounted on a small cart which navigates through busy downtown environment. We evaluated our algorithm using only left camera images for tracking. Among five sequences listed in [1], we applied our algorithm on the "Seq#2" and "Seq#3". Both sequences contain large number of pedestrians walking around a downtown area. In both sequences, our algorithm was working better or as well as [1]. Unlike [1], we used only the single (left) camera sequence throughout the experiments so the performance of tracking algorithm was solely relied on better estimation capability of our algorithm. Quantitative results are reported in Table.1. Following the evaluation criteria of [11,1], we report the number of pedestrians, the number of trajectories, the number of mostly hit trajectories, mostly missed trajectories, the number of false alarm, and the number of ID switch for the Seq.#2 as following: 33, 47, 28, 8, 3, 2. Since [1] did not report exact frame numbers, we choose 350 to 800 frames. Following [1], we also counted as a new trajectory if a person is occluded for more than 10 frames. Overall, about 60% of the trajectories were covered and most of the missed trajectories belonged to small people. Qualitative results are reported in Fig.4, 5 and 6. For additional results, please visit our project's webpage [3].

6 Conclusion

In this paper, we presented a fully automatic multi-target tracking algorithm. Different sources of information were integrated into one coherent probabilistic framework and all the variable was estimated in a joint fashion. Our framework has very flexible structure, so other additional cues can be incorporated for further stabilization. Combining other geometric cues is our plan for investigation.

Acknowledgment. This work is supported through a grant from Ford Motor Company via the Ford-U of M Innovation Alliance (Award #N011537). We also would like to thank Jeffrey Remillard for his valuable feedbacks throughout this project.

References

1. Ess, A., Leibe, B., Schindler, K., van Gool, L.: A mobile vision system for robust multi-person tracking. In: CVPR (2008)
2. Hoiem, D., Efros, A., Hebert, M.: Putting objects in perspective. In: CVPR (2006)
3. Project-webpage (2010), `http://www.eecs.umich.edu/vision/mttproject.html`
4. Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. PAMI (2002)
5. Avidan, S.: Ensemble tracking. PAMI (2007)
6. Yin, Z., Collins, R.: On-the-fly object modeling while tracking. In: CVPR (2007)
7. Matthews, I., Ishikawa, T., Baker, S.: The template update problem. PAMI 26, 810–815 (2004)

8. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
9. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. In: PAMI (2009)
10. Okuma, K., Taleghani, A., Freitas, N.D., Freitas, O.D., Little, J.J., Lowe, D.G.: A boosted particle filter: Multitarget detection and tracking. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 28–39. Springer, Heidelberg (2004)
11. Wu, B., Nevatia, R.: Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors (2007)
12. Breitenstein, M.D., Reichlin, F., Leibe, B., Koller-Meier, E., Gool, L.V.: Robust tracking-by-detection using a detector confidence particle filter. In: ICCV (2009)
13. Khan, Z., Balch, T., Dellaert, F.: Mcmc-based particle filtering for tracking a variable number of interacting targets (2005)
14. Pellegrini, S., Ess, A., Schindler, K., van Gool, L.: You'll never walk alone: Modeling social behavior for multi-target tracking. In: ICCV (2009)
15. Scovanner, P., Tappen, M.: Learning pedestrian dynamics from the real world. In: ICCV (2009)
16. Tomasi, C., Kanade, T.: Detection and tracking of point features. In: Carnegie Mellon University Technical Report (1991)
17. Kuhn, H.W.: The hungarian method for the assignment problem. In: Naval Research Logistics Quarterly (1955)
18. Davison, A.J., Reid, I.D., Molton, N.D., Stasse, O.: Monoslam: Real-time single camera slam. PAMI 29, 1052–1067 (2007)
19. Smith, P., Reid, I., Davison, A.: Real-time monocular slam with straight lines. In: BMVC (2006)