A General Framework for Tracking Multiple People from a Moving Camera

Wongun Choi, Caroline Pantofaru, Silvio Savarese

Abstract—In this paper, we present a general framework for tracking multiple, possibly interacting, people from a mobile vision platform. In order to determine all of the trajectories robustly and in a 3D coordinate system, we estimate both the camera's ego-motion and the people's paths within a single coherent framework. The tracking problem is framed as finding the MAP solution of a posterior probability, and is solved using the Reversible Jump Markov Chain Monte Carlo Particle Filtering method. We evaluate our system on challenging datasets taken from moving cameras, including an outdoor street scene video dataset, as well as an indoor RGB-D dataset collected in an office. Experimental evidence shows that the proposed method can robustly estimate a camera's motion from dynamic scenes and stably track people who are moving independently or interacting.

Index Terms—multi-target tracking, person detection, people tracking, RJ-MCMC Particle Filtering

1 Introduction

Understanding how people move through the world is a key problem in computer vision. There is currently a wealth of video data of people available from the Internet, from indoor mobile robot platforms, and from car-mounted sensors, to name a few sources. Accurately detecting and tracking people in video can facilitate action understanding for better video retrieval. Tracking in real time from a mobile robot can form the basis for human-robot interaction and more efficient robot performance in human environments. Detecting and tracking pedestrians from a car can help people drive safely, and it will keep people safe in the presence of autonomous cars. In this paper, we tackle the problem of detecting and tracking multiple people as seen from a moving camera. Our goal is to design an algorithm that can be adapted for the wide array of applications in which person tracking is needed.

In practice, unfortunately, person tracking is extremely difficult. Examples of the data we wish to tackle are displayed in Fig. 1. The first challenge evident in these images is that people's appearances vary widely, and people change their appearance in different environments, which complicates person detection. Despite excellent advances in detection [11], [15], it is still far from trivial to detect people in a variety of poses, wearing a variety of clothing, and in cluttered environments full of occlusions. A tight field-of-view that truncates people, as well as high contrast illumination, are additional difficulties often encountered in indoor environments. To improve the odds of finding people, our system combines multiple different detection cues, and allows additional cues to be added or removed as needed.

Another challenge is the complexity of the motion patterns of multiple people in the same scene. Tracking a single person is sufficiently difficult as they move willfully and unpredictably. Tracking multiple people, however, is complicated by their interactions; assuming independence between targets' motions is insufficient. As in Fig. 1, people stay out of each other's personal space and never occupy exactly the same space. On the other hand, people may choose to move together as a group for awhile. To model these interactions, we propose placing constraints between the targets' motions, partially removing the independence assumption.

In the scenarios we wish to address, the camera is moving as well. Estimating camera motion and reconstructing a scene is a well-studied problem when the scene is stationary [21], however the scenes described herein contain multiple large dynamic elements. Background subtraction is also likely to fail in these scenes. To tackle this issue, our tracker is capable of separating stationary and dynamic features in the scene, allowing it to estimate the camera motion and separate it from the motion of the targets.

Given that the mobile platform on which the camera is mounted needs to react to people's positions online, for example to plan to drive around them, our tracking method is capable of near real-time performance at 5-10 frames per second.

To address the issues discussed above, we propose a principled method for tracking multiple people and estimating a camera's motion simultaneously. Our contributions are as follows. First, we propose a novel model which can naturally explain the process of video generation from a single moving camera. Second, we propose a

- W. Choi is with the Department of Electrical and Computer Engineering, the University of Michigan, Ann Arbor, MI, 48105.
E-mail: wgchoi@umich.edu
- C. Pantofaru is with Willow Garage, Inc., Menlo Park, CA, 94025
E-mail: pantofaru@willowgarage.com
- S. Saverese is with the Department of Electrical and Computer Engineering, the University of Michigan, Ann Arbor, MI, 48105.
E-mail: silvio@eecs.umich.edu

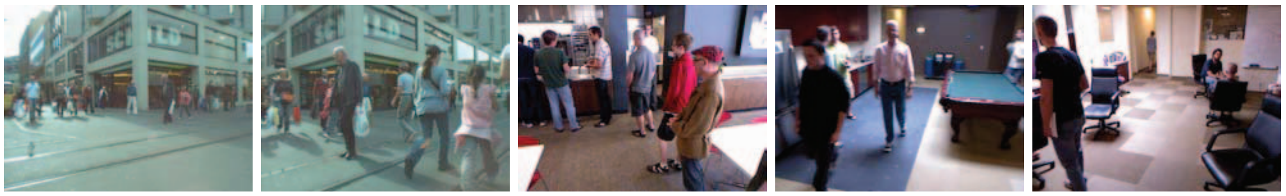

Fig. 1: Typical examples of outdoor and indoor tracking scenarios. Correspondences between video frames are difficult to compute due to camera motion and multiple dynamic subjects. People are difficult to detect due to occlusions and limited field of view. Indoor environments are especially challenging as people tend to adopt various types of poses (standing, leaning, sitting, etc). We aim at providing a general framework for tracking multiple people in a wide variety of difficult situations.

motion model that can capture the interactions between targets. Third, we introduce a principled method for fusing multiple person detection methods to build a more robust and adaptable tracker. Finally, our system is flexible enough to operate on video data alone, and to integrate depth data when it is available. This unified tracking framework is made efficient through the use of Reversible Jump - Markov Chain Monte Carlo (RJ-MCMC) particle filtering.

We demonstrate our method using the challenging ETH tracking dataset from [12] which contains video data taken from onboard a car driving through a city, as seen in the left two images in Fig. 1. In addition, we contribute a dataset of color and depth image (RGB-D) data taken from onboard a robot moving in an indoor environment, as seen in the right three images in Fig. 1.

Throughout the paper, we discuss our tracking framework as it applies to the task of tracking people. However, the framework is general and could be applied to other tracking tasks by replacing the detection components and changing the target motion interaction model.

2 Related Work

The method introduced in this paper is designed to track multiple people in the world from a moving camera. To solve this problem a number of challenges must be overcome, including coping with the varying appearance of people as they deform over time, occlusions among people and between people and the environment, possibly missing detections, and the difficulties of estimating a moving camera's position. In this section, we discuss the related work designed to overcome one or more of these challenges.

Tracking by Online Learning: To track an object whose appearance is changing over time requires an adaptable object model. A number of related works seek to address this problem through online learning, learning the appearance model of a specific target and applying the model to track that target [10], [5], [34], [7], [26]. For example, Comaniciu and Meer [10] used color histograms created from user-initialized bounding boxes, and tracked those models with the mean-shift algorithm. Avidan [5] showed promising results on tracking a single target using a boosting-based learning framework. A common issue for these methods is tracker drift. In addition, they all require that a target's initial position be provided manually.

Human Detection: One solution for improving tracker drift and enabling automatic track initialization is the use of person detectors. Over the last decade, algorithms for detecting humans have improved a great deal [41], [27], [11], [39], [44], [16], [15]. Modern human detection methods are quite reliable when applied to large pedestrians in simple scenes that include minimal crowding, occlusion and clutter. Methods by Ferrari *et al.* [16] and Felzenszwalb *et al.* [15] are also able to detect humans in non-pedestrian poses with reasonable accuracy. However, in real-world environments which are crowded, include large amounts of occlusion and clutter, as well as wide pose variation, none of these methods is satisfactory. For this reason, we combine a number of person detection cues into our system.

Tracking-by-detection: Thanks to the improvement in human detection methods, the tracking problem can be reformulated as a tracking-by-detection problem such as in [44], [8], [24], [3], [43], [9]. This approach can generate reliable tracking results if the camera is kept stationary. Multi-target tracking problems can either be formulated to estimate target locations online, such as in the works of Wu and Nevatia [44] and Breitenstein *et al.* [8], or to find the globally optimal association among detections at different time stamps, such as is done by Pirsiavash *et al.* [32], Zhang *et al.* [46] and Shitrit *et al.* [37] using a linear programming framework. Methods which perform global optimization may produce more consistent targets with fewer identity switches, but they require an entire video sequence as input and so cannot be used for real-time planning. In this paper, we instead focus on designing an online algorithm that is applicable to autonomous driving and robot navigation, with a focus on higher detection accuracy. More consistent trajectories could be obtained by applying a tracklet association method [45] on top of our results. Also, most of the methods which do not explicitly consider camera motion are prone to failure when a camera moves since the camera motion and target motions become intertwined.

Tracking with a Moving Camera: To address the challenges of tracking from a moving platform, several approaches [43], [42], [12], [13], [9] have recently been proposed. Wojek *et al.* [43], [42] proposed a probabilistic framework to detect multiple people in a busy scene by combining multiple detectors [43] and explicitly reasoning about occlusions among people [42]. However, they did not associate detections between frames, so no tracking was performed. In addition, they relied on odometry readings from the car on which the camera

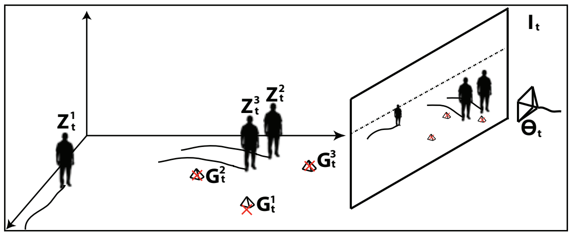

Fig. 2: Given sensor inputs (i.e. images) I_t over time, the goals of this work are 1) to track targets to obtain their trajectories, $\{Z_t^i\}$, in a world coordinate system, and 2) to estimate the camera's motion, Θ_t. Stationary geometric features from the scene, $\{G_t^j\}$, are used to guide the camera parameter estimation.

was mounted to obtain the camera position. Our work performs data association to track people and is capable of estimating the camera motion. Our work is most similar in spirit to the work by Ess *et al.* [12], [13], which combines multiple detectors to estimate camera odometry and track multiple people at once. Unlike [12], [13], we track targets and estimate camera motion in a unified framework and do not require stereo information.

3 System Overview

A pictorial overview of the problem is presented in Fig. 2. Given a stream of sensor inputs (i.e. images) $\{I_t\}$, the high-level goal of our system is to determine people's trajectories, $\{Z_t^i\}$, in a world coordinate system, while simultaneously estimating the camera's motion, Θ_t. To stabilize the camera's parameter estimation, a number of features which are hypothesized to be stationary, $\{G_t^j\}$, are extracted from the scene.

A system diagram is presented in Fig. 3. The core of the system is the RJ-MCMC particle filter tracker, which generates proposals for subjects' track states and the camera state, and evaluates proposals given both observations from the scene and a motion model.

There are three key ingredients in making such a system perform well for person tracking. The first is the observation model and cues that are used which must account for the large variation in both people's appearance and scene statistics. The second is the motion model which must account both for people's unexpected motions as well as interactions between people. The third is the sampling procedure for the RJ-MCMC tracker, which must efficiently sample the space of possible trajectories while also accounting for people's erratic movements.

In the following sections, we will first describe the mathematical model for our system, and then describe each of the system components in detail.

4 Model Representation

We model the tracking problem using a sequential Bayesian framework, which seamlessly integrates both the estimation of camera motion and multiple target tracking. The camera parameters Θ_t, a set of targets' states Z_t and a set of geometric features' states G_t at each time frame are modeled as random variables and the relationships among them are encoded by a joint posterior probability. With this model, the tracking

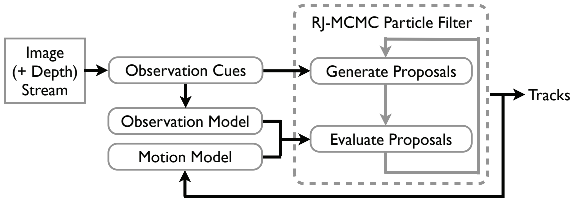

Fig. 3: System overview. Given an input sequence, our system outputs trajectories for both targets and the camera's motion in 3D world coordinates. By employing a diverse set of observation cues to generate detection hypotheses and an observation model, our system adapts to various scenarios in a principled manner. RJ-MCMC Particle Filtering efficiently samples the set of possible trajectories, allowing for online use. Note that the system is versatile enough to cope with either monocular camera data or RGB-D sensor data.

and camera estimation problem is formulated as finding the maximum-a-posteri (MAP) solution of the joint probability. In this section, we explain our probabilistic formulation using the notation summarized in Table 1. Here, we keep the mathematical formulation general and explain the details in subsequent sections.

A configuration of all the variables at time t is represented by $\Omega_t = \{\Theta_t, Z_t, G_t\}$. To find the most probable configuration, we estimate the MAP solution of $P(\Omega_t|I_{0,\ldots,t})$, which can be factored as in [4]:

$$P(\Omega_t|I_{0,\ldots,t}) \propto \underbrace{P(I_t|\Omega_t)}_{(a)} \int \underbrace{P(\Omega_t|\Omega_{t-1})}_{(b)} \underbrace{P(\Omega_{t-1}|I_{0,\ldots,t-1})}_{(c)} d\Omega_{t-1} \quad (1)$$

The first term (Eq. 1(a)) represents the **observation likelihood** of the model configuration at time t, Ω_t, given the sensor input at time t, I_t. This measures the compatibility of a hypothetical configuration with the sensor input. The second term (Eq. 1(b)) is the **motion model**, which captures both smoothness of the trajectory over time, as well as the target interactions. The third term (Eq.1(c)) is the posterior probability at time $t-1$.

By assuming that the posterior probability at the initial time is available, the posterior probability at an arbitrary time t can be calculated from the posterior probabilities from time 1 to $t-1$ sequentially. The best model configuration Ω_t is then the MAP solution.

One important characteristic of this model is that it allows the number of targets and features to vary. Tracks can be initiated when people enter the scene, and terminated automatically when people leave. The entrance or exit of a person i either introduces or removes a variable into the target set Z_t. Similarly, a static geometric feature j can go out of the scene when the camera moves or if it is occluded by a dynamic element. This also enables the model to decide which target or feature hypothesis is actually valid as the introduction of a false detection hypothesis will result in a lower joint probability. Despite this changing dimensionality of Ω_t, we can estimate the posterior using Reversible Jump Markov Chain Monte Carlo (RJ-MCMC) particle filtering [24].

5 Observation Likelihood

The observation likelihood is a measure for evaluating which configuration Ω_t best matches the input data I_t.

Given a hypothesis for the configuration, $\hat{\Omega}_t$, evaluation is broken into two steps: 1) project each hypothesized target and geometric feature into the input space, and then 2) evaluate the observation likelihood given the input data as in Eq. 2. Our input data is an image, so step 1 is equivalent to using the camera projection function f_Θ with the estimated camera parameters, $\hat{\Theta}_t$. This method can be generalized to other input modalities (e.g. lidar).

$$P(I_t|\Omega_t) = \prod_i P(I_t|Z_t^i,\Theta_t) \prod_j P(I_t|G_t^j,\Theta_t) \tag{2}$$

$$\underbrace{P(I_t|Z_t^i,\Theta_t) = P(I_t|f_{\Theta_t}(Z_t^i))}_{target\ observation}, \underbrace{P(I_t|G_t^j,\Theta_t) = P(I_t|f_{\Theta_t}(G_t^j))}_{feature\ observation} \tag{3}$$

5.1 Camera model

We consider two different types of camera projection functions: a simplified camera projection function [20], [9] and a general pinhole camera projection function [19].

Simplified camera model: The simplified camera model [20] assumes that all objects of interest rest on the ground plane. Given the image location of the horizon and the camera height, the model estimates objects' 3D locations from the top and bottom of their bounding boxes in the image (see [20] for details).

The camera Θ is parameterized with the following variables: focal length f, image center u_c, horizon line v_h, yaw angle ϕ, velocity μ, camera height h_Θ and 3D location (x_Θ, z_Θ). For parameters Θ and object location Z, the projection function f_Θ is defined as:

$$Z_0 = \begin{bmatrix} R(\phi) & 0 \\ 0 & 1 \end{bmatrix} Z + \begin{bmatrix} x_\Theta \\ z_\Theta \\ 0 \end{bmatrix}, X = f_\Theta(Z_0) = \begin{bmatrix} \frac{f x_Z}{z_Z} + u_c \\ \frac{f h_\Theta}{z_Z} + v_h \\ \frac{f h_Z}{z_Z} \end{bmatrix} \tag{4}$$

where Z_0 represents the location of a target in the current camera coordinates, and $X = (x, y, h)$ is the corresponding bounding box in the image plane with a fixed aspect ratio. The projection function for geometric features is defined similarly (and is identical to the projected location for a target's feet.)

Pinhole camera model: If additional 3D input is available (i.e. a depth image), we employ a pinhole camera model to obtain a more accurate camera projection. Following the general pinhole camera model, the camera parameterization includes the focal length f, the 3D location (x, y, z) and the orientation angles $(roll, pitch, yaw)$. See [19] for details.

Θ_t	camera parameters at time t
Z_t^i	a target's state at time t (location and velocity in 3D).
G_t^j	a geometric feature's state at time t (location in 3D).
Z_t	$= \{Z_t^0, Z_t^1, ..., Z_t^N\}$, the set of all targets' states at time t
G_t	$= \{G_t^0, G_t^1, ..., G_t^N\}$, the set of all geometric features' states at time t
Ω_t	$= \{\Theta_t, Z_t, G_t\}$, the set of all random variables at time t
$I_{0...t}$	all sensor inputs upto time t
f_Θ	the camera projection function parameterized by Θ

TABLE 1: Notation definitions

5.2 Target Observation Likelihood

Given the projection of a target's hypothesized location into the image, the observation likelihood measures both the validity of the target, as well as the accuracy of the location. The localization is modeled directly via the observation likelihood $P(I_t|f_{\Theta_t}(Z_t^i))$.

It is more difficult for the validity measure to adjust to the possibility that the target does not actually exist at all. The measure we would like to use is the ratio of the likelihoods $P(I_t|f_{\Theta_t}(Z_t^i))/P(I_t|f_{\Theta_t}(\emptyset))$, which allows the dimensionality of the target states variable Z_t^i to vary. However, since the likelihood of the empty set is ambiguous, we instead model the ratio by taking a soft max $g(\circ)$ of the hypothesis likelihood, as in Eq. 6. The soft max makes the measure robust to sporadic noise.

In order to accommodate the wide array of data inputs and tracking scenarios we wish to address, our system combines a number of different detectors to evaluate the observation likelihood. This is one of the key ingredients in our approach. Each single detector has its strengths and weaknesses. For example a face detector is extremely reliable when a frontal face is presented, but uninformative if a person shows his back to the camera. We propose to combine the ensemble of detectors by using a weighted combination of detection responses as in Eqs. 5 and 6. Our experimental analysis shows that this helps make our system more robust and reliable (Sec.8). For simplicity, we adopt the log likelihood l_j instead of the likelihood P_j for each detector j with weight w_j.

$$P\left(I_t|f_{\Theta_t}(Z_t^i)\right) \propto exp\left(\sum_j w_j l_j\left(I_t|f_{\Theta_t}(Z_t^i)\right)\right) \tag{5}$$

$$\frac{P\left(I_t|f_{\Theta_t}(Z_t^i)\right)}{P\left(I_t|f_{\Theta_t}(\emptyset)\right)} = exp\left(\sum_j g\left(w_j l_j(I_t|f_{\Theta_t}(Z_t^i))\right)\right) \tag{6}$$

We combine seven detectors to generate the observation likelihood: 1) a pedestrian detector, 2) an upper body detector, 3) a target-specific detector based on appearance model, 4) a detector based on upper-body shape from depth, 5) a face detector, 6) a skin detector, and 7) a motion detector. The model is flexible enough to allow the addition of other observation modules as necessary for other applications. A description of each observation measurement follows.

Pedestrian and Upper Body Detectors

The first two observation cues are based on the distribution of gradients in the image, encoded by the Histogram of Oriented Gradient detector (HOG) [11]. We incorporate two HOG detection models, an upper body detector and a full body detector as trained in [11] and [16], respectively. Using both models allows us to cope with lower body occlusions, different pose configurations, as well as different resolutions of people in images.

To obtain a detection response, the HOG detector performs a dot product between the model parameter w and the HOG feature h, and thresholds the value

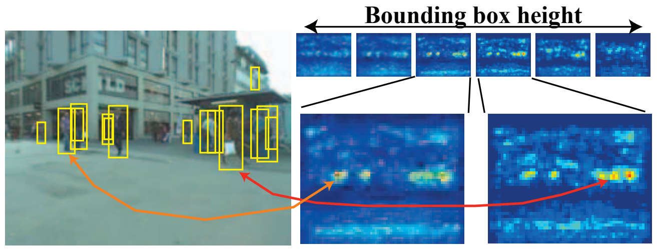

Fig. 4: **Left:** HOG [11] positive detections. **Right:** the corresponding confidence maps for each candidate window size (measured by the bounding box heights). Each target projected into the image plane corresponds to one $(x, y, scale)$ position in the corresponding confidence map.

(above zero). Our previous work [9] used a Gaussian model centered on positive detections to obtain the observation model. However, this is a brittle approach in the case of missed detections or false positives. Instead, in this work both the positive detections and confidence values are used to model the observation likelihood from the HOG detector, as inspired by [8] (see Fig.4). The positive detector outputs and confidence value terms in the observation likelihood are as follows:

$$l_{Det^+}(I_t|f_{\Theta_t}(Z_t^i)) = \begin{cases} N(d_t^i; f_{\Theta_t}(Z_t^i), \Sigma_d) & \text{if } d_t^i \text{ exists} \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

$$l_{Det^c}(I_t|f_{\Theta_t}(Z_t^i)) = w \cdot h(f_{\Theta_t}(Z_t^i)) \quad (8)$$

where $N(\circ; \mu, \Sigma)$ is a multivariate normal distribution with mean μ and covariance Σ, d_t^i is the positive detector output corresponding to Z_t^i, $h(\circ)$ represents the HOG feature from the region $\circ$ and w is the linear detector model parameter vector. This detector can be easily replaced by a more sophisticated approach as demonstrated in our experiments (we use the Deformable Parts Model [15] for the experiments on the ETH dataset).

Face Detector

The ability to detect frontal faces [40] has proven to be useful for tracking. In our system, we employ the Viola-Jones face detector [40] as implemented in OpenCV [28], [1]. This method detects faces reliably given a face size of greater than 24 pixels and minimal blur. The face detector likelihood is calculated as the maximum overlap ratio between all the face detection outputs X_t^k and the projection of target state Z_t^i into the image:

$$l_{Face} = \max_k OR(X_t^k, T_f(f_{\Theta_t}(Z_t^i))) \quad (9)$$

where T_f is the face portion of the image projection and $OR(\cdot, \cdot)$ is the overlap ratio (intersection over union) between two rectangles.

Skin Color Detector

The next cue used is skin color. If a person exists in a location Z_t^i, then pixels corresponding to the face region are likely to be observed even if the face is observed from the side view (face profile). To detect pixels with skin color appearance, we threshold each pixel in HSV color space and apply a median filter on the skin image I_{Skin}, an image of binary pixels that indicate skin region. Although simple, we have found this approach to be

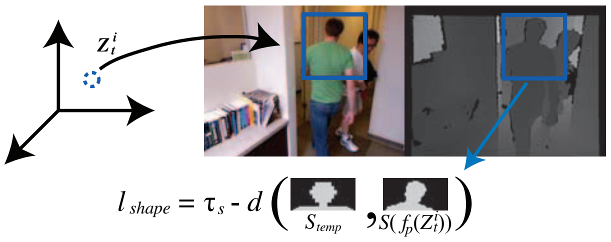
Z_t^i
$l_{shape} = \tau_s - d\left(S_{temp}, S(f_p(Z_t^i)) \right)$

Fig. 5: The shape vector is computed from the top half of the depth image bounding box. A binary vector of the person's head and shoulder area is compared to a template using the Hamming distance.

sufficient. A more sophisticated approach (such as [22]) could be adopted to learn a statistical model for skin. The observation likelihood is obtained by computing the percentage of skin pixels lying in the predicted face region of a hypothesis:

$$l_{Skin} = \frac{1}{|T_f(f_{\Theta_t}(Z_t^i))|} \sum_{(x,y)\in T_f(f_{\Theta_t}(Z_t^i))} I_{Skin}(x, y) \quad (10)$$

where $|\cdot|$ represents the area of a bounding box and I_{Skin} is the filtered binary skin image.

Depth-based Shape Detector

Observations can also be extracted from the depth image. Each pixel in a depth image specifies the distance of the pixel from the camera in the world coordinate system. In a depth image, the head-and-shoulders outline of a person is clearly distinguishable as shown in Fig.5. This can be converted into the observation likelihood l_{Shape} by taking the Hamming distance between a binary template of the head-and-shoulder region, with a thresholded version of the depth image projection region of Z_t^i, as in Fig.5. Then the likelihood term becomes:

$$l_{Shape}(I_t|Z_t^i) = \tau_s - d(S_{temp}, S(f_{\Theta_t}(Z_t^i); I_t)) \quad (11)$$

where τ_s is a threshold, S_{temp} is the template, $S(f_{\Theta_t}(Z_t^i); I_t)$ is the shape vector of Z_t^i, and $d(\cdot, \cdot)$ is the distance between template and shape vector of Z_t^i.

Motion Detector

The presence of motion in a scene is a strong indicator of the presence of a person, especially indoors. Given depth information, motion can be efficiently identified by using a change detector in 3D. In our implementation, we use the octree-based change detection algorithm between the point clouds in consecutive frames, as described in [23]. A binary motion image is obtained by projecting the moving points into the image plane and thresholding. The likelihood is then computed as the ratio of moving pixels lying in the body region of a hypothesis.

$$l_{Motion} = \frac{1}{|f_{\Theta_t}(Z_t^i)|} \sum_{(x,y)\in f_{\Theta_t}(Z_t^i)} I_{Motion}(x, y) \quad (12)$$

where I_{Motion} is the binary motion image.

Target Specific Appearance-based tracker

A detector often fails to detect the target even when it is present (false negatives). Appearance-based tracking [10], [5], [7] can be used to help link consecutive detections. By limiting the use of appearance-based tracking to a small number of consecutive frames, issues due to tracker drift can be minimized. We employ a color-based tracking algorithm [10] to provide target-specific tracking information at each time frame. Denote the output for the tracker for target Z^i at time t as Y_t^i. Then the color-tracker observation likelihood term is:

$$l_{Tr}(I_t|f_{\Theta_t}(Z_t^i)) = N(Y_t^i; f_{\Theta_t}(Z_t^i), \Sigma_{tr}) \tag{13}$$

Note that for many of these cues, such as face detection, skin color detection and motion detection, a positive observation increases the likelihood that a person is present, but the lack of observation does not decrease the likelihood that a person is present.

Geometric Feature Observation Likelihood

In addition to detecting and localizing targets, we also want to compute the camera's location and orientation in the world. As in previous tracking work (i.e. the KLT tracker [38]), this is accomplished by detecting stationary features in the world which we call *geometric features*.

Observing geometric features can be interpreted as a generative process in which features in the world are projected onto the image plane and then detected by an interest point detector. The detection process is noisy, so the observation likelihood is modeled as a normal distribution centered on the projection of the feature, $f_{\Theta_t}(G_t^j)$. Since some of the hypothesized features may become occluded between frames, or may in fact be non-stationary features, we introduce a uniform background model for invalid features.

Let the interest point corresponding to a geometric feature G_t^j be τ_t^i. Then the likelihood can be written as:

$$P(I_t|f_{\Theta_t}(G_t^j)) = \begin{cases} N(\tau_t^i; f_{\Theta_t}(G_t^j), \Sigma_G) & \text{if } G_t^j \text{ is valid} \\ K_B & \text{if } G_t^j \text{ is invalid} \end{cases} \tag{14}$$

Through the combination of the Gaussian component for valid features and the uniform component for invalid features, the inference process rejects outliers and estimates camera motion more robustly.

6 Motion Prior

We now discuss the motion prior term $P(\Omega_t|\Omega_{t-1})$ in Eq. 1. The motion model encodes smooth transitions between configurations through time via three components: 1) a camera motion prior, 2) a target motion prior and 3) a geometric feature motion prior, as follows:

$$P(\Omega_t|\Omega_{t-1}) = \underbrace{P(\Theta_t|\Theta_{t-1})}_{camera}\underbrace{P(Z_t|Z_{t-1})}_{targets}\underbrace{P(G_t|G_{t-1})}_{geom.features} \tag{15}$$

These three motion priors are discussed in detail below.

6.1 Camera Motion Prior

The motion of a camera over a short period of time can be assumed to be smooth both in position and rotation, and so can be modeled using a linear dynamic model with constant velocity.

For scenarios in which the simplified camera model is used, a constant perturbation model is employed for the horizon, camera height, velocity, and yaw angle, i.e. $\phi_{t+1} = \phi_t + \epsilon$ (where ϵ is an error term that accounts for uncertainty.) The location update is:

$$x_{t+1} = x_t + v_t cos(\phi_t) + \epsilon, \; z_{t+1} = z_t + v_t sin(\phi_t) + \epsilon \tag{16}$$

The constant perturbation model is used with the pinhole camera model for all location-related camera parameters $(x, y, z, roll, pitch, yaw)$. The internal camera parameters (focal length, skewness, optical center, etc.) are assumed to be provided for both parameterizations.

6.2 Target Motion Prior

The motion model for the moving targets includes two factors: the existence of a target at time t, P_e, and the smoothness of its motion, P_m. The former encodes the probability of the person's presence at adjacent time stamps; a person is more likely to exist at time stamp t if they existed at time stamp $t-1$, and vice versa. Then full target motion model is:

$$P(Z_t|Z_{t-1}) = P_{Ex}(Z_t|Z_{t-1})P_{Motion}(Z_t|Z_{t-1}) \tag{17}$$

In this work, we consider two possible ways to model the targets' motions: i) independent motion and ii) interactions between people affect their motion. The independence assumption has been traditionally used to simplify model inference. However, recent studies [36], [31], [24] including our own [9] suggest that modeling the interaction between targets enhances tracking accuracy significantly. We now describe the two terms in Eq.17.

Existence Prior ($P_{Ex}(Z_t|Z_{t-1})$)

The existence prior is modeled by two binomial probabilities, the first of which is parameterized by the probability of a target staying in the scene from one time to another (p_s^t). The second is parameterized by the probability of a new target entering the scene (p_e^t).

$$P_{Ex}(Z_t|Z_{t-1}) = \prod_i P_{Ex}(Z_t^i|Z_{t-1}^i) \tag{18}$$

$$P_{Ex}(Z_t^i|Z_{t-1}^i) = \begin{cases} p_s^t & \text{if } i \text{ exists at } t-1 \text{ and } t \\ 1-p_s^t & \text{if } i \text{ exists at } t-1 \text{ but not } t \\ p_e^t & \text{if } i \text{ exists at } t \text{ but not } t-1 \\ 1-p_e^t & \text{if } i \text{ does not exist at either time} \end{cases} \tag{19}$$

Independent Targets ($P_{Motion}(Z_t|Z_{t-1})$)

The motion prior based on independent targets can be expressed as:

$$P_{Motion}(Z_t|Z_{t-1}) = \prod_i P_{Motion}(Z_t^i|Z_{t-1}^i) \tag{20}$$

The motion prior for a particular target, $P_{Motion}(Z_t^i|Z_{t-1}^i)$, can be modeled by a constant velocity model, giving the update rule:

$$Z_t^i = Z_{t-1}^i + \dot{Z}_{t-1}^i dt, \; \dot{Z}_t^i = \dot{Z}_{t-1}^i + \epsilon_Z \tag{21}$$

where ϵ_Z is a process noise for individual target's motion that is drawn from a normal distribution.

Interacting Targets

In real world crowded scenes, targets rarely move independently. Often, targets stay out of each other's personal space and they never occupy the same space. At

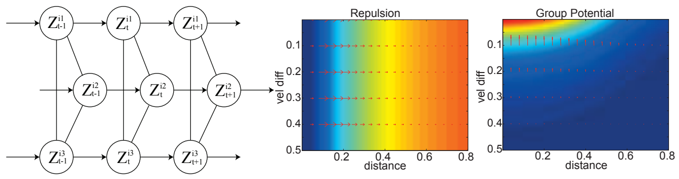

Fig. 6: **Left:** the interactions between people are modeled by pairwise potentials, linking their positions and velocities in the motion model. **Middle and Right:** the potential functions for repulsion and group interaction, respectively, over different distances (x-axis) and velocities (y-axis) in the jet color map. Function gradients are shown as red arrows. The repulsion model pushes people apart, while the group motion enforces similar velocity. The potentials were generated with $c_r = 2, s_g = 3, t_g = 1, c_g = 10$.

other times, some targets may choose to move together as a group for awhile. One of the contributions of our work is to introduce such interactions into the motion model through a *repulsion model* and a *group model*.

We model target interactions by using an MRF as in Fig.6. In particular, interactions are captured by pairwise potentials between the current targets' states. Since two targets cannot both repel and form a group at the same time, a hidden mode variable $\beta_t^{i1,i2}$ selects between the interaction modes. The model then becomes:

$$P(Z_t|Z_{t-1}) = \prod_{i1<i2} \psi(Z_t^{i1}, Z_t^{i2}; \beta_t^{i1,i2}) \prod_{i1<i2} P(\beta_t^{i1,i2}|\beta_{t-1}^{i1,i2}) \prod_{i1=1}^{N} P(Z_t^{i1}|Z_{t-1}^{i1}) \tag{22}$$

where $\psi(Z_t^{i1}, Z_t^{i1}; \beta_t^{i1,i2})$ is the pairwise potential.

Mode variable: The mode variable selects the interaction type for a given pair of targets. The transition probability $P(\beta_t^{i1,i2}|\beta_{t-1}^{i1,i2})$ is modeled as p_β if $\beta_t^{i1,i2} = \beta_{t-1}^{i1,i2}$ or $1-p_\beta$ otherwise. The mode variable selects the interaction type such that:

$$\psi(Z_t^{i1}, Z_t^{i2}; \beta_t^{i1,i2}) = \begin{cases} \psi_g(Z_t^{i1}, Z_t^{i2}) & \text{if } \beta_t^{i1,i2} = 1 \\ \psi_r(Z_t^{i1}, Z_t^{i2}) & \text{otherwise} \end{cases} \tag{23}$$

Repulsion: The repulsion potential pushes apart targets that are too close together. Let $r_{i1,i2}$ be the distance between two targets in 3D, and let c_r control the repulsion force. Then the repulsion potential is:

$$\psi_r(Z_t^{i1}, Z_t^{i2}) = e^{-\frac{1}{c_r r_{i1,i2}}} \tag{24}$$

The repulsion between two targets is higher as they get close to each other, and approaches 0 when they are far apart. Two targets will push each other away unless they have a group relationship and $\beta_t^{ij} = 1$.

Group Motion: Targets can also interact as a group, remaining at the same small distance from each other and moving in the same direction, $\dot{Z}_t^{i1} \approx \dot{Z}_t^{i2}$. By modeling the proximity between targets using a sigmoid function of their distance, we obtain:

$$\psi_g(Z_t^{i1}, Z_t^{i2}) = \frac{1}{1 + e^{s_g(r_{i1,i2} - t_g)}} e^{-c_g\|\dot{Z}_t^{i1} - \dot{Z}_t^{i2}\|} \tag{25}$$

where c_g controls the velocity similarity, s_g controls the sigmoid slope, and t_g controls the distance.

The proposed interaction model improves localization when the reliability of the detection results is affected by noise. For example, detectors have trouble distinguishing between two people in close proximity whose bounding boxes overlap. In this case, the repulsion model will keep the hypotheses separate. On the other hand, the group interaction model provides constraints on the location of neighboring targets if at least one of the targets is confidently detected. Our model can be naturally extended to incorporate other interaction types, such as people approaching for a handshake.

6.3 Geometric Feature Motion Prior

The geometric features' motion prior captures whether the features are valid and whether their positions are consistent with those in previous times. To estimate the camera motion robustly, the inference must separate the stationary background features from dynamic ones. Let P_{Val} be the validity prior and P_{Cons} be the consistency prior. Then:

$$P(G_t|G_{t-1}) = P_{Val}(G_t|G_{t-1}) P_{Cons}(G_t|G_{t-1}) \tag{26}$$

Similar to the target existence prior, the validity prior is modeled by two binomial probabilities which are parameterized by the probability of staying in the scene, p_s^g, and the probability of entering the scene, p_e^g. This encodes the intuition that a valid (stationary) geometric feature will likely remain valid in the next time stamp. The validity prior becomes:

$$P_{Val}(G_t|G_{t-1}) = \prod_j P_{Val}(G_t^j|G_{t-1}^j) \tag{27}$$

$$P_{Val}(G_t^j|G_{t-1}^j) = \begin{cases} p_s^g & \text{if j is valid at } t-1 \text{ and } t \\ 1-p_s^g & \text{if j is valid at } t-1 \text{ but not } t \\ p_e^g & \text{if j is valid at } t \text{ but not } t-1 \\ 1-p_e^g & \text{if j is not valid at } t \text{ and } t-1 \end{cases} \tag{28}$$

Since the features are defined as stationary 3D world points, a single feature's consistency prior $P_{Cons}(G_t|G_{t-1})$ can be modeled by an indicator function I to enforce the stationary assumption:

$$P_{Cons}(G_t|G_{t-1}) = \prod_j I(G_t^j = G_{t-1}^j) \tag{29}$$

Overall, the target and geometric features motion priors ensure that all of the configuration variables change smoothly, but can also appear, disappear and interact. This makes our model both robust and flexible.

7 Tracking with RJ-MCMC

We have thus far discussed how to evaluate proposed tracking states through the observation likelihood and the motion model, terms (a) and (b) of Eq. 1, and the left half of the system diagram in Fig. 3. We now need to explore the space of these hypotheses to find the MAP solution to the posterior distribution $P(\Omega_t|I_{1,\dots,t})$. Unfortunately, the structure of the posterior is extremely complex because: i) both targets and geometric features may change their cardinality in time which, in turn, changes the dimensionality of Ω_t, ii) Ω_t has high dimensionality and iii) the interaction model couples states together. As a result, traditional methods for obtaining MAP solutions are difficult to apply.

To efficiently explore the configuration space and obtain the MAP solution, we use the Reversible Jump Markov Chain Monte Carlo Particle filtering method (RJ-MCMC) introduced by Khan *et al.* [24] (see the right half of the diagram in Fig. 3). The RJ-MCMC algorithm enables the addition and removal of targets via random jump proposal moves between dimensions . The well-known *"sample impoverishment"* problem is avoided by re-sampling via MCMC sampling at each time frame, thus reducing the correlation among samples [30]. Unlike Khan et al., however, our goal is to estimate the camera motion and identify target interaction as well as track multiple moving targets, so we need to explore the combined configuration state space. To this end, an important contribution of this work is the introduction of additional jump proposal moves to the RJ-MCMC algorithm.

7.1 RJ-MCMC sampling

As in Eq. 1, the goal of tracking is to find the state that maximizes the posterior configuration:

$$\hat{\Omega}_t = \underset{\Omega_t}{\operatorname{argmax}}\, P(\Omega_t|I_{1,\ldots,t}) \tag{30}$$

We apply RJ-MCMC to obtain the posterior $P(\Omega_t|I_{1,\ldots,t})$. At each timestep, we approximate the posterior by a number of samples:

$$P(\Omega_t|I_{1,\ldots,t}) \approx \{\Omega_t^{(r)}\}_{r=1}^{N} \tag{31}$$

where N is the number samples and $\Omega_t^{(r)}$ is the r^{th} sample. These samples can be obtained by performing RJ-MCMC sampling on the posteriors from 1 to t. Given the set of samples at time $t-1$, the posterior distribution at t can be approximated as:

$$P(\Omega_t|I_{1,\ldots,t}) \propto P(I_t|\Omega_t)\sum_r P(\Omega_t|\Omega_{t-1}^{(r)}) \tag{32}$$

In this section, we explain the details of our proposal distribution and sampling. Section 7.3 explains the acceptance ratio for the Metropolis-Hastings algorithm. In the remainder of this section, we assume that a weak detection hypothesis X_t and the correspondences between targets and detections are available to guide the sampling. The detections are necessary to help initiate targets and bias sampling. Notice our algorithm is capable of accommodating missing detections and false positives as well.

7.2 Proposal Moves

As explained in Sec. 4, the configuration variable is composed of three components, $\Omega_t = \{Z_t, G_t, \Theta_t\}$. Sampling from the whole configuration variable's space results in very slow convergence to the steady state distribution due to high dimensionality. Thus, instead, we randomly choose one variable to sample at a time. More specifically, one of targets, geometric features or camera parameters is randomly chosen and its state is randomly perturbed to propose a new sample. Following the Metropolis-Hasting rule, the proposed sample is accepted or rejected to construct the Markov Chain, $\{\Omega_t^{(0)}, \Omega_t^{(1)}, \ldots \Omega_t^{(N)}\}$.

Let the proposal distribution be $Q(\Omega'_t, \Omega_t)$. Also, let Q_Z be the target proposal that is perturbed with probability q_z, Q_G be the geometric feature proposal that is perturbed with probability q_g and Q_Θ the camera proposal which is perturbed with probability q_Θ. Then:

$$Q(\Omega'_t, \Omega_t) = q_Z Q_Z(\Omega'_t, \Omega_t) + q_G Q_G(\Omega'_t, \Omega_t) + q_\Theta Q_\Theta(\Omega'_t, \Omega_t) \tag{33}$$

For example, assume that the geometric proposal is randomly chosen. Then, upon perturbation, the new configuration will be $\Omega_t^{(r+1)} = \{Z_t^{(r)}, G_t^{(r+1)}, \Theta_t^{(r)}\}$. Only a single geometric feature's state will be changed in $G_t^{(r+1)}$, and the remaining terms will remain unchanged.

Target Proposal Q_Z

The target proposal Q_Z generates a new sample $Z_t^{(r+1)}$ from the current sample $Z_t^{(r)}$. The information contained in $Z_t^{(r)}$ includes the status of each target's presence and state, which has variable dimensionality depending on the number of targets present. Thus, the proposal distribution must allow efficient exploration of a space with varying dimensionality. This efficient exploration is accomplished through the use of *jump moves*.

We define a set of six reversible jump moves: *Stay*, *Leave*, *Add*, *Delete*, *Update* and *Interaction Flip*. Each move is designed to act as a reversible counterpart of another move in the set (this guarantees that the Markov Chain satisfies the *detailed balance* condition). For example *Stay* and *Leave* counteract each other. During exploration, one of the six moves is chosen randomly with probabilities of q_S, q_L, q_A, q_D, q_U, and q_I, respectively. Below, we describe each jump type.

Stay: Let $S_t^{(r)}$ be the set of targets that existed in Z_{t-1} but are not in sample $Z_t^{(r)}$. The *stay* move inserts one of these targets, i, into sample $Z_t^{(r+1)}$. The specific target to insert is chosen with uniform probability. Unlike [24] (which samples from only the previous posterior $P(Z_t^i|Z_{t-1}^i)$), we sample the new target location from a mixture distribution of $P(Z_t^i|X_t^i)$ and $P(Z_t^i|Z_{t-1}^i)$, where X_t^i is a corresponding detection. This makes the sampling process more robust to accommodate moving targets. If no detection is available for target i, the new proposal is sampled from the previous posterior distribution. The proposal is then:

$$Q_S(Z_t^{(r+1)}; Z_t^{(r)}) = \begin{cases} \frac{1}{|S_t^{(r)}|} Q_i(Z_t^{i(r+1)}) & \text{if } i \text{ in } S_t^{(r)} \\ 0 & \text{otherwise} \end{cases} \tag{34}$$

where $Q_i(Z_t^{i(r+1)})$ is equal to $P(Z_t^i|Z_{t-1}^i)$ when there is no corresponding detection, and $\frac{1}{2}[P(Z_t^i|Z_{t-1}^i) + P(Z_t^i|X_t^i)]$, otherwise.

Leave: If a target *Stays* in sample $Z_t^{(r)}$, the *Leave* move proposes to remove the target from the new sample $Z_t^{(r+1)}$. This is the reverse of *Stay*. Let $L_t^{(r)}$ be the set of targets that exist in $Z_t^{(r)}$ and existed in Z_{t-1}. From this set, a target i is selected with uniform probability and removed. The proposal is then:

$$Q_L(Z_t^{(r+1)}; Z_t^{(r)}) = \begin{cases} \frac{1}{|L_t^{(r)}|} & \text{if } i \text{ in } L_t^{(r)} \\ 0 & \text{otherwise} \end{cases} \tag{35}$$

Add: This proposal initiates a new target from the new detections, X_t^{new}, which do not correspond to any existing targets. Let $A_t^{(r)} = X_t^{new} \backslash Z_t^{(r)}$ be the new detections that are not in the current target set. From this set, one target i is randomly selected with a uniform probability. The new location of target i, $Z_t^{i(r)}$, is proposed from the distribution $P(Z_t^{i(r)}|X_t^i)$. The corresponding proposal distribution can be written as

$$Q_A(Z_t^{(r+1)}; Z_t^{(r)}) = \begin{cases} \frac{1}{|A_t^{(r)}|} P(Z_t^{i(r)}|X_t^i) & \text{if } i \text{ in } A_t^{(r)} \\ 0 & \text{otherwise} \end{cases} \tag{36}$$

Delete: *Delete* is the reverse jump move of *Add*. Among new detections in the previous sample, $D_t^{(r)} = X_t^{new} \cap Z_t^{(r)}$, one target i is randomly drawn with uniform probability and removed.

$$Q_D(Z_t^{(r+1)}; Z_t^{(r)}) = \begin{cases} \frac{1}{|D_t^{(r)}|} & \text{if } i \text{ in } D_t^{(r)} \\ 0 & \text{otherwise} \end{cases} \tag{37}$$

Update: *Update* proposes a new location for a target. From the targets in sample $Z_t^{(r)}$, a target i is randomly selected and a new location is proposed from the distribution $Q(Z_t^{i(r+1)}; Z_t^{i(r)}) \sim \mathcal{N}(Z_t^{i(r)}, \Sigma_U)$. Note that one *Update* move can be "reversed" by another *Update* move. The proposal can be expressed as

$$K_t^{(r)} = L_t^{(r)} \cup D_t^{(r)}$$
$$Q_U(Z_t^{(r+1)}; Z_t^{(r)}) = \begin{cases} \frac{1}{|K_t^{(r)}|} Q(Z_t^{i(r+1)}; Z_t^{i(r)}) & \text{if } i \text{ in } K_t^{(r)} \\ 0 & \text{otherwise} \end{cases} \tag{38}$$

note that $K_t^{(r)}$ is the set of target indices that exist in the current target set $Z_t^{(r)}$.

Interaction Flip: The final target proposal considers pairs of targets and their interactions. *Interaction Flip* proposes an alternative interaction mode for a selected pair of targets, $\beta_t^{i1,i2}$. Among all possible pairs of targets in a sample $Z_t^{(r)}$, a pair of targets $(i1, i2)$ is randomly selected and the mode of interaction is flipped between repulsion and group interaction, or vice versa, with probably p_f.

$$Q_I(Z_t^{(r+1)}; Z_t^{(r)}) = \begin{cases} \frac{2Q(\beta_t^{i1,i2,(r+1)}; \beta_t^{i1,i2,(r)})}{|K_t^{(r)}|(|K_t^{(r)}|-1)} & \text{if } i1, i2 \text{ in } K_t^{(r)},\ i1 \neq i2 \\ 0 & \text{otherwise} \end{cases} \tag{39}$$

Geometric Feature Proposal Q_G

Similarly to the target states in Z_t, the geometric features' states stored in G_t also need to be updated. G_t is also a high dimensional vector with a variable dimensionality. Thus, we use the same scheme as that used for targets to update it. To update the geometric feature states, we use the proposal moves: *Stay*, *Leave* and *Update*. As for the target proposals, one of the proposals is randomly chosen with probability of q_S, q_L and q_U, respectively. Note that since the validity of features can only be defined by comparing their location in different time frames, we do not use the *Add* and *Delete* moves in feature proposals. That is, in order to verify whether a feature is stationary or not, we assume we observe it for at least two adjacent frames. All the newly introduced features are automatically added into the feature set in the time frame, and the validity of features is examined by comparing the observed position and the predicted position using *Stay* and *Leave* moves in the subsequent frames.

Stay: Similarly to the *Stay* move for target proposals, the *Stay* move proposes to keep feature j that was in G_{t-1} but is not in $G_t^{(r)}$. With a slight abuse of notation, let $S_t^{(r)}$ be the set of features which are in G_{t-1} but not in $G_t^{(r)}$ and let one of these be chosen with uniform probability. The location of the feature is drawn from $P_c(G_t^{j(r+1)}|G_{t-1}^j)$, which gives:

$$Q_S(G_t^{(r+1)}; G_t^{(r)}) = \begin{cases} \frac{1}{|S_t^{(r)}|} P_c(G_t^{j(r+1)}|G_{t-1}^j) & \text{if } j \text{ in } S_t^{(r)} \\ 0 & \text{otherwise} \end{cases} \tag{40}$$

Leave: The *Leave* move for geometric features follows the same structure as the *Leave* move for target proposals. Let $L_t^{(r)}$ be the set of features that exist in both $G_t^{(r)}$ and G_{t-1}.

$$Q_L(G_t^{(r+1)}; G_t^{(r)}) = \begin{cases} \frac{1}{|L_t^{(r)}|} & \text{if } i \text{ in } L_t^{(r)} \\ 0 & \text{otherwise} \end{cases} \tag{41}$$

Update: Similarly to the target *Update* proposal, we randomly select a geometric feature and propose a new location for the feature by adding gaussian noise. Since geometric features are defined to be static, it is not necessary to explore different locations for an existing feature. The motion consistency prior for the features is defined to be an indicator function. As a result, any new state $G_t^{j(r+1)}$ that is different from G_{t-1}^j will have 0 probability and, thus, perturbations are only applied to newly added features, $N_t = G_t^{(r)} \setminus G_{t-1}$.

$$Q_G(G_t^{(r+1)}; G_t^{(r)}) = \begin{cases} \frac{1}{|N_t^{(r)}|} Q(G_t^{j(r+1)}; G_t^{j(r)}) & \text{if } j \text{ in } N_t \\ 0 & \text{otherwise} \end{cases} \tag{42}$$

Similarly to the update proposal for targets, $Q(G_t^{j(r+1)}; G_t^{j(r)})$ is modeled by a normal distribution $\mathcal{N}(G_t^{j(r+1)}; G_t^{j(r)}, \Sigma_G)$ parameterized by Σ_G.

Camera Proposal Q_Θ

The final component we need to sample is the camera state. Since there is only one camera, we can model the camera state proposal $Q_\Theta(\Theta_t^{(r+1)}; \Theta_t^{(r)})$ by a simple normal distribution $\mathcal{N}(\Theta_t^{(r+1)}; \Theta_t^{(r)}, \Sigma_\Theta)$.

7.3 Acceptance Ratio

Following the Metropolis Hastings algorithm, we compute the acceptance ratio of the new sample $\Omega_t^{(r+1)}$ by the product of the three ratios:

$$a = \frac{P(I_t|\Omega_t^{(r+1)})}{P(I_t|\Omega_t^{(r)})} \frac{P(\Omega_t^{(r+1)}|I_{1,2,\ldots t-1})}{P(\Omega_t^{(r)}|I_{1,2,\ldots t-1})} \frac{Q(\Omega_t^{r}; \Omega_t^{r+1})}{Q(\Omega_t^{r+1}; \Omega_t^{r})} \tag{43}$$

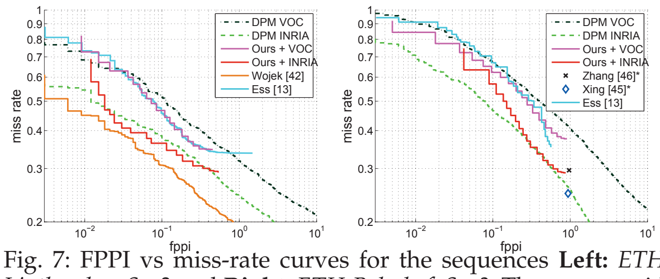

Fig. 7: FPPI vs miss-rate curves for the sequences **Left:** *ETH-Linthescher, Seq2* and **Right:** *ETH-Bahnhof, Seq3*. The papers with * reports only one recall and fppi and require all the images in a batch as an input.

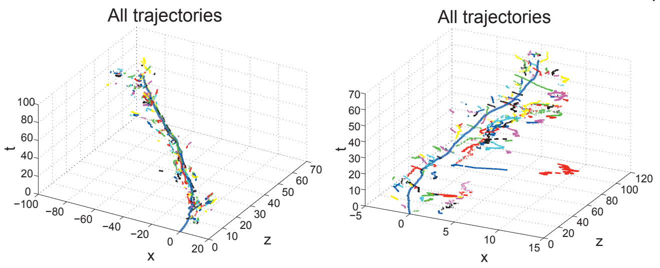

Fig. 8: The camera trajectories (long dark-blue lines) estimated from the sequences *ETH-Linthescher* (left) and *ETH-Bahnhof* (right). Targets' trajectories (short multi-color lines) are also shown for illustration purposes.

The first term expresses the ratio between the image likelihoods; the second term is the ratio between approximated predictions; the last term encodes the ratio between proposal distributions. Since we change the state of only one target's presence or location at a time, most of the factors can be cancelled out in the above computation. This characteristic makes the algorithm efficient and capable of processing videos in real-time.

8 Experimental Evaluation

We demonstrate our proposed algorithm using two different types of data inputs and three datasets. The first dataset is a part of the ETH dataset [12] that includes the sequences *ETH-Linthescher* and *ETH-Bahnhof* (seq02 and seq03 in [12]). This data consists of video sequences recorded with a moving camera in densely populated urban streets with pedestrians. The videos have a frame rate of $\sim$14Hz and a resolution of 640$\times$480 pixels.

The second and third datasets are collected using a Kinect RGB-D camera [29], [33] and consist of video sequences associated with depth maps (RGB-D). Both of the datasets contain longer video segments and tracks than previous datasets, making data association and camera motion estimation more difficult. The first RGB-D dataset (we call it the *Kinect office dataset*) is acquired using a static Kinect mounted approximately 2 meters high (and tilted down) in an office. This set contains 17 videos, typically 2 to 3 minutes long. People in these scenes take on different poses (e.g. sitting on a chair, standing up), are observed from different view points (front, side, 3/4 rotation) and are subject to various degrees of occlusions, inter-occlusions and self-occlusions. The second RGB-D dataset (we call it the *Kinect mobile dataset*) is collected from a Kinect mounted on a mobile platform (a PR2 robot). The robot was driven (teleoperated) around an office building, while sequences of people performing daily activities in offices, corridors, hallways and cafeteria were acquired. The sequences include various configurations where the camera and targets are moving at the same time, the targets are located at different distances from the camera, the number of targets in the scene are changing over time, and targets are subject to occlusions, illumination condition varies in time, etc. This dataset includes 18 video sequences.

In both Kinect datasets, humans are hand-annotated with bounding boxes around upper bodies in each image. Targets' 3D locations are inferred from the bounding boxes and depth images (where available). The annotation is provided on four images every second. In addition, ground truth odometry information of the camera's location in 3D space is also provided for evaluation purposes. In the *Kinect mobile dataset*, the odometry is obtained via the robot localization using the ROS system [2], which utilizes multiple sensor inputs as well as a known building map.

8.1 Implementation Details

The overall system flow is as follows. Given a sensor input at each time frame, a set of weak detection hypotheses X_t of human targets in the scene are generated using the observation cues. The correspondences between the predicted locations of targets, $\hat{Z}_t$, and the weak hypotheses are identified using the Hungarian algorithm [25]. Geometric features (a maximum of 40) are detected using the SURF detector [6] and tracked using the KLT tracker [38]. For the RGB-D data, we include depth information.

The feature trajectories, sensor inputs, detection hypotheses and previous time posterior distribution are passed to the RJ-MCMC algorithm to estimate the posterior at time t. We draw 5000 samples from which the initial 1000 samples are discarded (*burn in*) and only every 100th sample is kept for the posterior distribution to avoid high correlation among samples (*thinning*). The MAP approximation of the new camera and targets' states is given by the mean of the posterior samples, $\{\Omega_t^{(r)}\}_{r=1}^N$. In order to improve the computational efficiency, we remove the trajectories that generate fewer than 10% of the samples. [1]

To account for the different sensor modalities in each of the test sets, the experiments below were run with two different system setups.

ETH datasets: In each ETH sequence, only 2D information is used, therefore we employ the simplified version of the camera projection function (see Section5.1). In addition, the people are often quite small, so we cannot use depth, faces, skin detection or 3D-based motion for this data. Instead, we use the Deformable Parts Model (DPM) detector [15] and color-based meanshift tracker [10] as the observation cues. The detection cues correspond to

1. In practice, a post-processing smoothing step might be added to avoid having "jittery" samples in the posterior distribution. Including this step in the evaluation, however, can hide the performance issues of the underlying tracker, so for transparency we have presented results without smoothing.

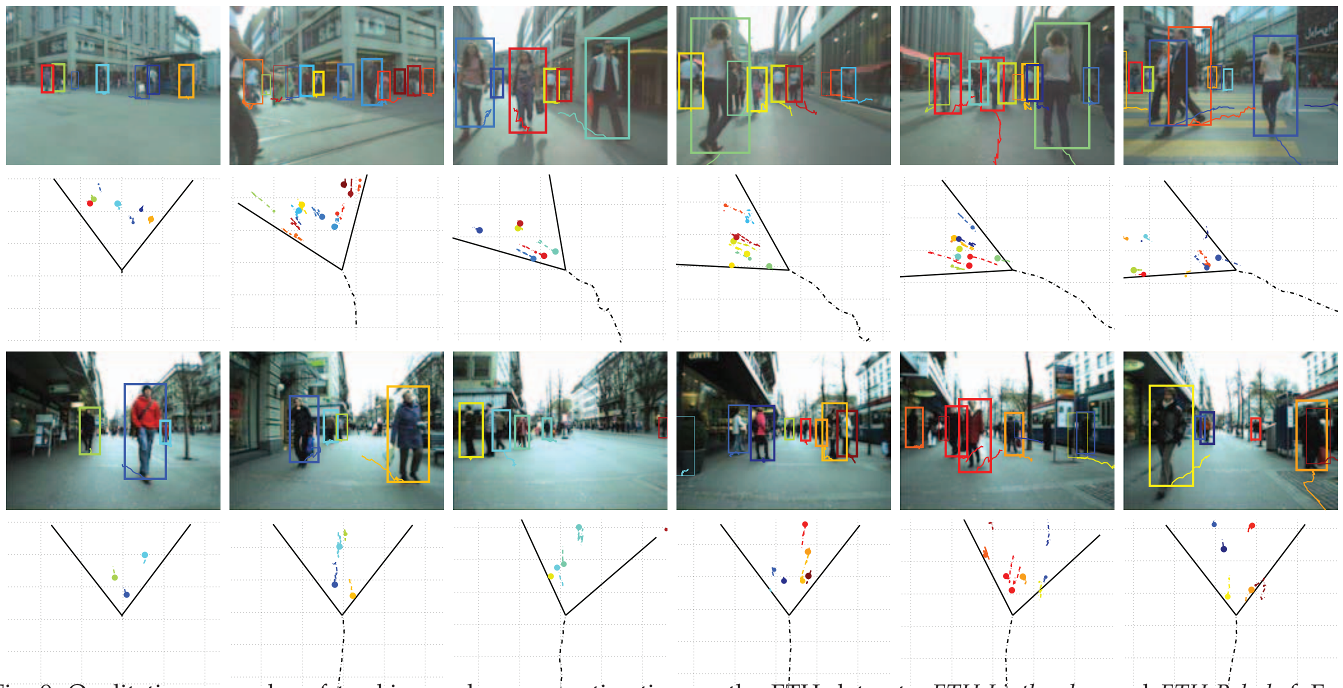
Fig. 9: Qualitative examples of tracking and camera estimation on the ETH datasets, *ETH-Linthescher* and *ETH-Bahnhof*. Each set of tracking examples is shown in two rows: the target trajectory and detection overlaid on the image (**top**) and a top-down projection onto the ground plane (**bottom**). Each target's trajectory is shown in a distinct color. In the top-down view, the V-shaped line indicates the camera's field of view, and the tail behind the V is the camera center's location over time. Notice the long target paths which indicate stable tracking.

DPM detections with confidence greater than 0.5. Note that as shown in [13], depth may help to further improve the detection rate. All of the system parameters are held constant for all of the sequences.

Kinect datasets: All the observation cues described in Section 5 are used in these experiments. We incorporate the upper and full body HOG detectors as trained by [16] and [11], and as implemented in OpenCV [1] to run on the GPU. Although the DPM detector [15] is more accurate, the speed of the GPU-based HOG detector is required. A 640x480 pixel image can be processed in 100∼200 milliseconds. We also use the face detector implemented in OpenCV. Skin pixels are identified by thresholding HSV values between (2, 60, 40) and (15, 200, 200). Finally, the octree-based motion detector is discretized to 3cm. The weak detection hypotheses X_t consist of the HOG detections (upper and full-body), face detections, as well as 3D point clusters [35].

8.2 Evaluation on the ETH dataset

We first study the single-frame detection accuracy on the video sequences *ETH-Linthescher* and *ETH-Bahnhof* (Fig.7) and compare it to the baseline methods: the DPM detector [15], the method by Wojek et al. [42] and the system by Ess et al. [13]. As a metric, we compute single-frame detection accuracy via the overlap ratios between the ground truth bounding boxes and the tracked bounding boxes. True and false positives are identified among such detections following the PASCAL challenge protocol [14]. The confidence of each target is measured as the number of valid samples. Note that we use the extended annotation from [42] for the evaluation of *ETH-Linthescher* but we use the annotation in [13] for the evaluation of *ETH-Bahnhof*. The extended annotation decreases the minimum person-size from 60 pixels to 48 pixels. As in [42], [13], we discard detections and annotations that are smaller than 60 pixels in the evaluation.

To show the adaptability of our system, we show experiments using two different DPM models as learned from the the INRIA [11] and the VOC09 [14] datasets. The results in Figure 7(left) show that our method (Ours+INRIA and Ours+VOC curves) improves detection accuracy over the two DPM baselines (DPM INRIA and DPM VOC), and obtains better or comparable results than the system in [13]. Note that, as observed by Ess [13], tracking algorithms often produce inferior detection results to their baseline detector since the tracker requires multiple frames to initiate tracking and also holds on to targets a few frames after they disappear (shown as thin bounding boxes in Figure 9). Nevertheless, our system produces better detections than the baseline detector. Wojek et al. [42] produce better detections than our system, however, they do not perform tracking, nor do they estimate the camera's trajectory. Also, Xing et al. [45] reported slightly better accuracy than ours, but the algorithm require all the images are given in a batch. We observe that our method (OURS+INRIA) performs slightly worse than the DPM baseline (DPM INRIA) as shown in Figure 7(right). We believe that this is due to the projection error that is introduced by the lens (barrel) distortion in the *ETH-Bahnhof* which contains un-rectified images as stated by the authors [2]. Such projection error prevents the algorithm from estimating targets 3D states accurately, thus producing an overall inferior detection accuracy.

2. Please see http://www.vision.ee.ethz.ch/~aess/dataset/ for details. Notice that the *ETH-Linthescher* appears to be rectified.

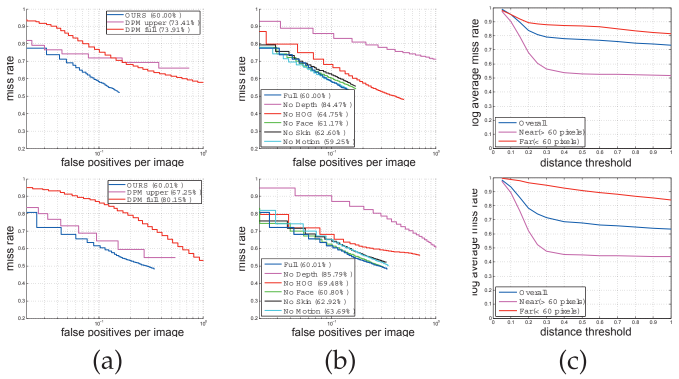

Fig. 10: Top row: results on the *Kinect office dataset*. Bottom row: results on the *Kinect mobile dataset*. (a) Baseline comparison versus the Deformable Parts Model (DPM) [15], [16]. Our system outperforms both the full- and the upper-body DPMs. (b) System analysis where contributions of each observation cue are visualized with different plots. The 'Full' observation includes all components. The other curves show the results obtained by removing specific components (such as the face detector). Notice that the depth mask is the most important cue, followed by the HOG detectors. The other components' contributions are situation-dependent and on average they appear less important. (c) Log Average Miss Rate (LAMR) over different distance thresholds. Detections were considered true positives if they were within 30cm in height and the distance threshold in depth from the ground truth. Results are shown for all the data, and also broken down for two distance ranges: near (detections larger than 60 pixels in height) and far (smaller than 60 pixels).

We also show the results of camera estimation in Figure 8 (long dark-blue lines). The (x, z) plane is defined along the camera coordinate system in the first frame of each video sequence, and the third dimension is time. Although no ground truth is available for the camera, we can qualitatively see that in the *ETH-Linthescher* sequence, the camera makes a left turn around the 150^{th} frame, which matches what we observe in the video sequence. Afterwards the camera moves approximately straight ahead until the end of the video. The camera motion in the *ETH-Bahnhof* sequence is correctly estimated as going roughly straight through the crowd.

8.3 Evaluation on the Kinect datasets

Next we demonstrate our method using the two Kinect datasets. As before, we begin by examining the detection accuracy. We compare our system against the DPM full body and upper body detectors as trained in [16]. Figure 10(a) shows the FPPI vs miss-rate curves for the three approaches. In the legend, we also provide the *log-average miss rate* (LAMR) proposed by [42]. As in [42], the LAMR is computed by drawing equally spaced samples in log space of the FPPI. We incorporate two evaluation protocols to determine a true positive. The first is based on the bounding box overlap protocol from PASCAL [14]. The second is based on a 3D distance threshold for localization.

Our algorithm outperforms both baseline methods significantly; there is 13% improvement in LAMR over both baselines on the *Kinect office dataset*, and 7% over the upper body DPM detector and 20% over the full body detector on the *Kinect mobile dataset*. Notice that we achieve such improvement even though we employ the weak HOG detector for detecting targets. As expected, the full body detector does not work well in the indoor scenario due to frequent occlusions, tight field of view, and unconventional poses such as sitting.

Next, we compare the contribution of each observation cue to our system. In this experiment, we turn off one detection cue at a time and compare the resulting detection accuracies in Figure 10(b). Turning off the depth shape detector (the No Depth curve) is the most detrimental to the system. Turning off both of the HOG detectors also results in a clear decrease in performance. Turning off the other observation cues has less obvious impact. This can be explained by the fact that the other cues are situation-dependent, and so their contribution is not evident when averaging over the dataset. For example, the face detector is a very strong and reliable cue when there is a large frontal face in the scene. However, often the person is far from or turned away from the camera, making the face detector useless, or worse, creating noise in the system. A similar argument can be made about motion detection. The fact that our system is able to perform well despite the variability of its individual components is a testament to its robustness. As future work, we would like to i) incorporate other types of detectors, such as the DPM [15] or a Hough Voting-based detector [27], and ii) study a principled way to learn the model weight w_j associated with each detector (Eq.5) in order to further improve robustness of the system.

Finally, we evaluate our algorithm's localization accuracy. In Figure 10(c), we show the LAMR measure over different 3D distance thresholds. Our method is more accurate in detecting people less than approximately 5 meters from the camera than those past 5 meters. This is an expected effect since the Kinect provides virtually no depth information past 5 meters, and in fact the depth information past 3 meters is very noisy.

Overall, experiments show that our algorithm outperforms state-of-the-art detectors. In addition, the fusion of multiple detection cues provides a more reliable final result and is capable of handling the variable performance of each individual detector. Selected tracking examples are shown in Figure 13. As shown in these results, the proposed method can reliably detect and track people in challenging indoor scenarios including occlusion between people, people in various poses, truncated body parts, and clutter.

Camera Estimation: Finally, we evaluate our system's ability to estimate the camera parameters. We compare our results against a baseline method which is constructed as follows. Given the feature trajectories, we compute the rotation matrix R_t^{t-1} and translation vector T_t^{t-1} of the camera between consecutive frames. Using the depth provided by the RGB-D sensor, we can compute R_t^{t-1} and T_t^{t-1} using the orthogonal Procrustes problem [18]. To cope with dynamic elements in the

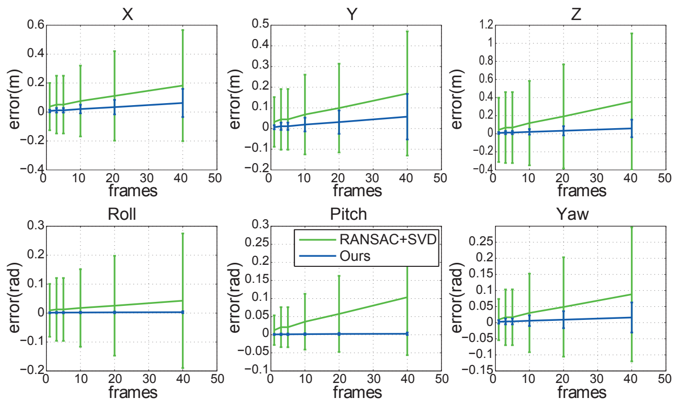

Fig. 11: Quantitative evaluation of camera localization. The mean and standard deviation of the error in each camera parameter estimation over different time spans.

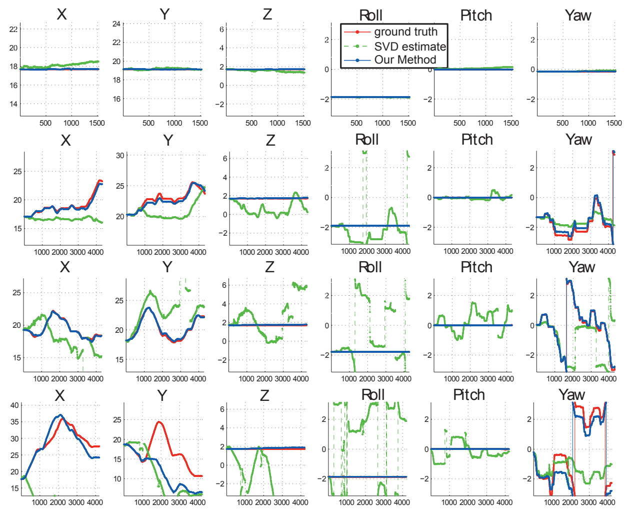

Fig. 12: Each row represents the 6 estimated camera parameters over time for selected sequences. Our method reliably estimates the camera in most cases, but can fail if there are no features (e.g. camera faces a featureless wall for time frames $1000 \sim 1500$ in the last sequence.)

scene and add robustness, we add a RANSAC [17] step on top of the estimator.

The comparison is presented in Figure 11. Since our method localizes the camera online, we measure the difference between parameters in consecutive time stamps. For all pairs of time stamps t_i and t_j with temporal gap t_g ($t_j = t_i + t_g$), we compute the transformation that maps the camera coordinate system of t_i to t_j. Such transformations are obtained for both the ground truth and the two estimations. The error between the transformations of ground truth and each estimation is reported for different time intervals (t_g). Figure 11 shows the mean and standard deviation of the error. The amount of error tends to increase with the time span due to error accumulation in the estimation. We report the estimation accuracy for each variable: translation (x, y, z) and rotation (roll, pitch, yaw). Each row in Figure 12 shows the estimated parameters over time for four different sequences. The ground truth is in red, the baseline in green, and our system is in blue.

As demonstrated in these results, our method is capable of robustly estimating camera motion under difficult conditions in which the baseline method fails to localize the camera. These scenes are challenging due to 1) lack of dominant stationary scene elements, 2) lack of a motion model for the camera or targets. Our method is able to cope with such challenges by 1) jointly identifying moving targets and static features in the estimation process, 2) using high level semantics (targets) as well as local features, and 3) incorporating the camera's motion prior. We observe that our method can localize the camera very accurately except for few very hard cases; e.g. the camera was facing a featureless wall around the 1000th frame of the 4th example in Figure 12.

9 Conclusion

Tracking multiple people, in different environments, performing different tasks and with different relationships will always be a challenging problem. Even humans have a great deal of trouble performing this task; only the best of athletes can predict how their team will move on the field, and the ability requires years of training, a deep knowledge of the team, and the constrained rules of a specific sport. In this paper, we have laid the groundwork for a general person tracking system and applied it to two specific environments - tracking people from a moving, ground-level camera, and tracking people indoors from a robot platform. We argue that the system is adaptable enough to be applied to other scenarios due to the following characteristics.

The joint formulation of all variables: The relationship between the camera, targets' and geometric features' states is combined into a novel probability model, allowing them to influence and improve each other's estimate during inference.

The combination of multiple observation cues: By combining multiple detection cues from different sensor modalities in a principled fashion, our system is adaptable to different sensor configurations and different environments.

Allowing people to interact: We do not assume that people move independently, instead we model interaction with two modes: repulsion and group movement. By automatically selecting between interaction modes, the system adapts to different scenarios.

Automatically detecting people: Our system automatically detects people, removing the need for manual initialization. Since the detection is probabilistic, the tracker can also recover from missed detections or false positives via motion model and sampling.

Automatic detection of static features for camera estimation: Since it estimates the camera's motion, our system can be applied on sequences acquired from a moving camera, even under the assumption that the odometry of the camera is unknown or poorly specified. The camera estimation is performed automatically using stationary features from the environment.

As we apply this system to additional scenarios in the future, we would like to learn what is the best combination of observation cues for a given sensor suite and environment from a training data. This system can

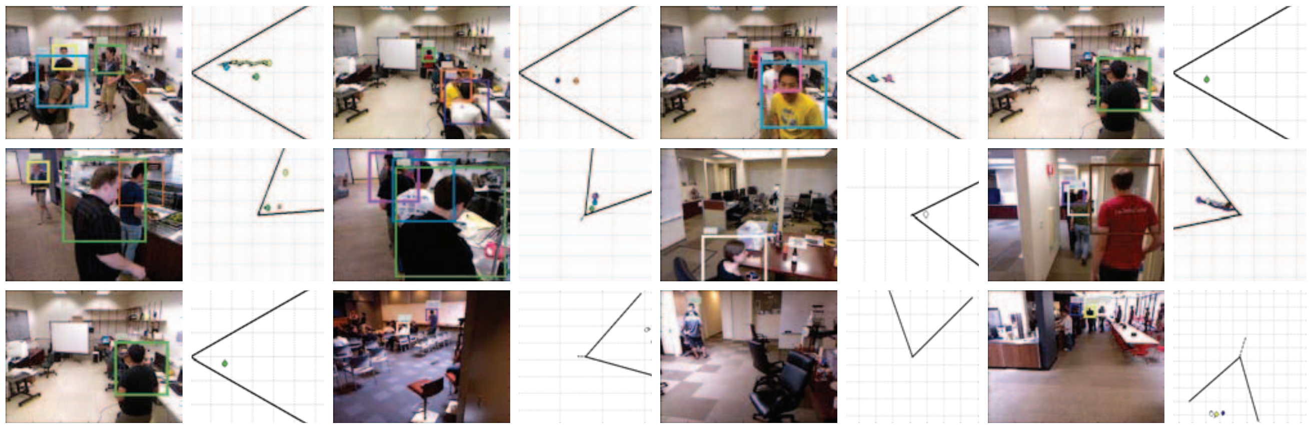

Fig. 13: Examples of tracking results. First row: results on the *Kinect office dataset*. Second row: results on the *Kinect mobile dataset*. Detections are shown as boxes in images, and dots projected onto the ground plane in the top-down view. Each color is a person. Note that our system detects people in various poses, truncated by the image, and despite the severe occlusions between people that are common in indoor environments. The last row shows examples of challenging scenes where the people appear beyond the Kinect's range or under extreme lighting conditions.

be used as a building block to learn and recognize high level semantics about human activities by providing trajectories of people.

REFERENCES

[1] OpenCV, The Open Source Computer Vision library. http://opencv.willowgarage.com/wiki/.
[2] ROS, The Robot Operating System. http://www.ros.org/.
[3] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, 2008.
[4] M. S. Arulampalam, S. Maskell, and N. Gordon. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2002.
[5] S. Avidan. Ensemble tracking. In *PAMI*, 2007.
[6] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. *CVIU*, 2008.
[7] C. Bibby and I. Reid. Robust real-time visual tracking using pixel-wise posteriors. In *ECCV*, 2008.
[8] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L. V. Gool. Robust tracking-by-detection using a detector confidence particle filter. In *ICCV*, 2009.
[9] W. Choi and S. Savarese. Multiple target tracking in world coordinate with single, minimally calibrated camera. In *ECCV*, 2010.
[10] D. Comaniciu and P. Meer. Mean shift : A robust approach toward feature space analysis. In *PAMI*, 2002.
[11] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.
[12] A. Ess, B. Leibe, K. Schindler, and L. van Gool. A mobile vision system for robust multi-person tracking. In *CVPR*, 2008.
[13] A. Ess, B. Leibe, K. Schindler, and L. van Gool. Robust multi-person tracking from a mobile platform. *PAMI*, 2009.
[14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, June 2010.
[15] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 32(9), Sept. 2010.
[16] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, 2008.
[17] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6), 1981.
[18] J. Gower and G. Dijksterhuis. *Procrustes Problems*. Oxford University Press, 2004.
[19] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
[20] D. Hoiem, A. A. Efros, and M. Herbert. Putting objects in perspective. *IJCV*, 2008.
[21] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In *UIST*, 2011.
[22] M. Jones and J. Rehg. Statistical color models with application to skin detection. In *CVPR*, 1999.
[23] J. Kammerl. Octree point cloud compression in PCL. http://pointclouds.org/news/compressing-point-clouds.html, 2011.
[24] Z. Khan, T. Balch, and F. Dellaert. MCMC-based particle filtering for tracking a variable number of interacting targets. *PAMI*, 2005.
[25] H. W. Kuhn. The hungarian method for the assignment problem. In *Naval Research Logistics Quarterly*, 1955.
[26] S. Kwak, W. Nam, B. Han, and J. Han. Learning occlusion with likelihoods for visual tracking. In *ICCV*, 2011.
[27] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Statistical Learning in Computer Vision, ECCV*, 2004.
[28] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. In *ICIP*, 2002.
[29] Microsoft Corp. Kinect for XBOX. http://www.xbox.com.
[30] P. Minvielle, A. Marrs, S. Maskell, and A. Doucet. Joint target tracking and identification-part i: sequential monte carlo model-based approaches. In *International Conference on Information Fusion*, 2005.
[31] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *ICCV*, 2009.
[32] H. Pirsiavash, D. Ramanan, and C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011.
[33] PrimeSense. NITE natural interaction middleware. http://www.primesense.com/?p=515.
[34] D. Ramanan, D. Forsyth, and A. Zisserman. Tracking people by learning their appearance. *PAMI*, Jan. 2007.
[35] R. B. Rusu and S. Cousins. 3d is here: Point cloud library (pcl). In *ICRA*, Shanghai, China, May 9-13 2011.
[36] P. Scovanner and M. Tappen. Learning pedestrian dynamics from the real world. In *ICCV*, 2009.
[37] H. B. Shitrit, J. Berclaz, F. Fleuret, and P. Fua. Tracking multiple people under global appearance constraints. In *ICCV*, 2011.
[38] C. Tomasi and T. Kanade. Detection and tracking of point features. In *Carnegie Mellon University Technical Report*, 1991.
[39] O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on riemannian manifolds. In *CVPR*, 2007.
[40] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2003.
[41] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *ICCV*, 2003.
[42] C. Wojek, S. Walk, S. Roth, and B. Schiele. Monocular 3d scene understanding with explicit occlusion reasoning. In *CVPR*, 2011.
[43] C. Wojek, S. Walk, and B. Schiele. Multi-cue onboard pedestrian detection. In *CVPR*, 2009.
[44] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. In *IJCV*, 2007.
[45] J. Xing, H. Ai, and S. Lao. Multi-object tracking through occlusions by local tracklets filtering and global tracklets association with detection responses. In *CVPR*, 2009.
[46] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008.

Wongun Choi received his MS degree in Electrical and Computer Engineering in the University of Michigan, Ann Arbor in 2011. Currently, he is working toward his Ph.D degree at the Computer Vision Lab in the University of Michigan, Ann Arbor. His research interests include object tracking, object detection, scene understanding and activity recognition.

Caroline Pantofaru received the BSc degree in Mathematics and Computer Science from the University of Toronto, and the PhD degree in Robotics from Carnegie Mellon University in 2008, advised by Martial Hebert. She is currently a Research Scientist at Willow Garage, Inc. Her research interests include object recognition, people detection and tracking, and scene understanding. She is a member of the IEEE.

Silvio Savarese is an Assistant Professor of Electrical Engineering at the University of Michigan, Ann Arbor. After earning his Ph.D. in Electrical Engineering from the California Institute of Technology in 2005, he joined the University of Illinois at Urbana-Champaign from 20052008 as a Beckman Institute Fellow. He is recipient of a TWR Automotive Endowed Research Award in 2012, an NSF Career Award in 2011 and Google Research Award in 2010. In 2002 he was awarded the Walker von Brimer Award for outstanding research initiative. He served as workshops chair and area chair in CVPR 2010, and as area chair in ICCV 2011 and CVPR 2013. Silvio Savarese has been active in promoting research in the field of object recognition, scene representation and activity understanding. He co-chaired and co-organized the 1st, 2nd and 3rd edition of the IEEE workshop on 3D Representation for Recognition (3dRR-07, 3dRR-09, 3dRR-11) in conjunction with the ICCV. He was editor of the Elsevier Journal in Computer Vision and Image Understanding, special issue on “3D Representation for Recognition” in 2009. He co-authored a book on 3D object and scene representation published by Morgan and Claypool in 2011. His work with his students has received several best paper awards including a best student paper award in the IEEE CORP workshop in conjunction with ICCV 2011, a Best Paper Award from the Journal of Construction Engineering and Management in 2011 and the CETI Award at the 2010 FIATECH’s Technology Conference.