

Tracking Multiple Humans in Crowded Environment

Tao Zhao
Sarnoff Corporation
201 Washington Road
Princeton, NJ 08543
tzhao@sarnoff.com

Ram Nevatia
IRIS
University of Southern California
Los Angeles, CA 90089
nevatia@usc.edu

Abstract

Tracking of humans in dynamic scenes has been an important topic of research. Most techniques, however, are limited to situations where humans appear isolated and occlusion is small. Typical methods rely on appearance models that must be acquired when the humans enter the scene and are not occluded. We present a method that can track humans in crowded environments, with significant and persistent occlusion by making use of human shape models in addition to camera models, the assumption that humans walk on a plane and acquired appearance models. Experimental results and a quantitative evaluation are included.

1 Introduction

Tracking of humans in video sequences is important in dynamic scene analysis as they are the principal actors in daily activities of interest. There has been considerable work in tracking humans and other objects in recent years. Isolated objects or small number of objects having transient occlusion can be tracked fairly reliably in some systems. However, tracking in a more crowded situation where large number of people are present and exhibit persistent occlusion, remains challenging.

The goal of our work is to develop a general framework to detect and track humans in conditions with persistent, and temporarily heavy, occlusion. We assume a stationary camera (or moving camera after stabilization) so that motion can be detected by comparison with a background model. We do not require that humans be isolated when they first enter the scene. A snapshot of our results is shown in Fig.1.

Tracking blobs, detected as connected components in the foreground mask obtained by change detection, is a common way to track objects (e.g., [10]). However, such blobs do not always correspond to objects; single objects may split into multiple blobs and multiple objects may merge into a single blob (Fig.1). Tracking multiple objects with frequent occlusions becomes difficult with such approaches. Some approaches (e.g. [3]) require the objects to be initialized before occlusion happens, usually from blobs which may be erroneous. Some methods perform initialization based on segmentation



Figure 1: Left: a snap shot of our result overlaid on the input frame. Right: output of standard change detection shows the challenges to blob tracking and blob-based initialization.

by some heuristics (e.g., vertical projection of the foreground [4], head candidates by boundary analysis [11]). Their utility in crowded environments is likely to be limited. Particle filter based tracking ([7, 5]) has been popular recently. It keeps a non-parametric distribution of joint state probability and thus scales poorly as the dimensionality increases due to large number of objects.

Our approach to detection and tracking of multiple humans emphasizes on the use of models. Most important is the use of shape models for objects being tracked. In typical surveillance video, shape is relatively invariant across humans and is characterized by a small number of parameters. In addition, we use knowledge of camera models and the assumption that motion is on a known plane; this allows us to make inferences in 3D and account for changes in image due to perspective effects. We also use the appearance models acquired from the images but do not require that the object appear un-occluded when it first enters the scene; we do, however, require knowledge of the entrances and exits (typically just the boundary of the image).

We formulate the problem of detection and tracking as one of Bayesian inference to find the best interpretation given the image observations, the prior models and the estimates from previous frame analysis (i.e., the maximum *a posteriori*, MAP, estimation). The state to be estimated at each frame includes the number of objects, their correspondences to the objects in the previous frame (if any), their parameters (e.g., positions) and the uncertainty of the parameters. The color-based joint likelihood model considers all the objects and the background together and encodes both the constraint that the object should

¹This work was done while the first author was a PhD student in USC. This research was supported, in part, by the Advanced Research and Development Activity of the U.S. Government under contract No. MDA-908-00-C-0036.

be different from the background and that the object should be similar to its correspondence. Using this likelihood model gracefully integrates detection and tracking, and avoids a separate, sometimes ad hoc, initialization step.

The image is modeled as a composition of an unknown number of possibly overlapping objects and a background model. The solution space contains subspaces of varying dimensions, corresponding to different object number; the solution also contains both discrete and continuous variables. We use a Markov chain Monte Carlo (MCMC)-based method to compute the MAP estimate. MCMC-based methods have been recently used for many computer vision problems such as image parsing [9] and articulated body tracking [6]. We design reversible dynamics for multi-object tracking problem. We also use various direct image features to make the Markov chain efficient. Direct image features alone do not guarantee optimality because they are usually computed locally or using partial cues. Using them as proposal probabilities of the Markov chain has both the computational efficiency of image features and the optimality of a Bayesian formulation. The sequential nature of MCMC can make more in-depth analysis of the solution distribution. The explicit optimization makes it less sensitive to dimensionality compared to particle filters. Our experiments show that the described approach works robustly in very challenging situation with affordable computation. We have used similar concepts in an earlier paper [12] applied to a single frame, the new approach extends the method to video sequences. Even though we present results for human tracking only, the method easily generalizes to other objects.

2 A Bayesian Problem Formulation

Tracking is to estimate the state of the system at time t ($\theta^{(t)}$) given the observations up to time t ($I^{(1)}, \dots, I^{(t)}$) and all the previous estimates ($\theta^{(1)}, \dots, \theta^{(t-1)}$). It is commonly simplified as

$$P(\theta^{(t)} | I^{(1)}, \dots, I^{(t)}, \theta^{(1)}, \dots, \theta^{(t-1)}) = P(\theta^{(t)} | I^{(t)}, \theta^{(t-1)}, Bg^{(t-1)})$$

where $Bg^{(t-1)}$ is a background model estimated from $I^{(1)}, \dots, I^{(t-1)}$ and $\theta^{(1)}, \dots, \theta^{(t-1)}$.

We formulate the tracking problem as computing the maximum a posteriori (MAP) estimation $\theta^{(t)*}$ such that

$$\begin{aligned} \theta^{(t)*} &= \operatorname{argmax}_{\theta^{(t)} \in \Theta} P(\theta^{(t)} | I^{(t)}, \theta^{(t-1)}, Bg^{(t-1)}) \\ &= \operatorname{argmax}_{\theta^{(t)} \in \Theta} P(I^{(t)} | \theta^{(t)}, Bg^{(t-1)}) P(\theta^{(t)} | \theta^{(t-1)}) \end{aligned}$$

where $P(I^{(t)} | \theta^{(t)}, Bg^{(t-1)})$ is the likelihood and $P(\theta^{(t)} | \theta^{(t-1)})$ is the prior probability. This enables the prior knowledge on θ and the image observations to be integrated to form an optimal estimate.

In the multi-object tracking problem, we will make the best interpretation of an image frame with a background model, an unknown number of 3D objects with known (*i.e.* being tracked) or unknown (*i.e.* new object) appearances.

The state is parameterized as $\theta^{(t)} = \{\tilde{\mathbf{m}}_1^{(t)}, \dots, \tilde{\mathbf{m}}_n^{(t)}\} = \{(k_1^{(t)}, \mathbf{m}_1^{(t)}), \dots, (k_n^{(t)}, \mathbf{m}_n^{(t)})\}$, where $k_i^{(t)}$ is the ID of object i and $\mathbf{m}_i^{(t)}$ contains its parameters.

2.1 3D human shape model

The knowledge of valid human shape is very important in initializing objects and providing constraints during tracking. We use 3D model (in conjunction of a camera model and the assumption that the objects move on a known ground plane) to make the system applicable for a wide range of view angles. The 3D shape of a human object is approximated with a composition of a number of ellipsoids. Human body is highly articulated; therefore a number of such multi-ellipsoid models can be used to represent a few representative postures given an application domain. The same models have been used successfully in [12] for human segmentation. In this work, we found that using only one model (3-ellipsoid, one for head, one for torso and one for the legs) is sufficient for walking and standing humans. However, the system can readily handle multiple models in a more general setting. The parameters of each human object i are $\mathbf{m}_i = \{\mathbf{u}_i, h_i, f_i, i_i\}$ which are head *position*¹, *height*, *thickness* and 2D *inclination* respectively. The 3D shape model with the parameters fixed, after camera projection, results in a 2D shape model (*i.e.*, a mask).

2.2 Object appearance model

Besides the shape model, we also maintain a color histogram ($\tilde{\mathbf{p}}_i = \{\tilde{p}_1, \dots, \tilde{p}_m\}$, $\sum_{j=1}^m \tilde{p}_j = 1$) of the object as a representation of its appearance which helps establish correspondence in tracking. We use color histogram because humans may undergo non-rigid motion. Furthermore, there exists efficient algorithm (*i.e.*, the mean-shift technique) to optimize histogram-based object function. When gathering the color histogram, a kernel function $k(\cdot)$ is applied to weight pixel locations so that the center has a higher weight than the boundary considering the boundary may be more noisy. Such a representation has been used in [2].

2.3 Background appearance model

The background appearance model is a modified version of a Gaussian distribution. Denote the $(\bar{r}_j, \bar{g}_j, \bar{b}_j)$ and $\sum_j = \operatorname{diag}\{\sigma_{r_j}^2, \sigma_{g_j}^2, \sigma_{b_j}^2\}$ as the mean and the covariance of the color at pixel j . The probability of pixel j being from the background is

$$P_b(I_j) = P_b(r_j, g_j, b_j) \propto \max \left\{ \exp \left\{ - \left(\frac{r_j - \bar{r}_j}{\sigma_{r_j}} \right)^2 - \left(\frac{g_j - \bar{g}_j}{\sigma_{g_j}} \right)^2 - \left(\frac{b_j - \bar{b}_j}{\sigma_{b_j}} \right)^2 \right\}, o \right\}. \quad (1)$$

where o is a small constant. It is a composition of a Gaussian distribution and a uniform distribution. The uniform distribution captures the outliers which are not modeled by the Gaussian distribution to be more robust. The Gaussian parameters are updated continuously by the video stream.

¹The 3D position of the feet can be inferred from 2D position of the height and 3D height, along with the camera model and the ground plane. The feet position can determine the depth order of multiple objects.

2.4 The prior distributions

The prior distribution $P(\theta^{(t)}|\theta^{(t-1)})$ is composed of two parts

$$P(\theta^{(t)}|\theta^{(t-1)}) = P_1(\theta^{(t)})P_2(\theta^{(t)}|\theta^{(t-1)}) \quad (2)$$

where P_1 is the prior independent of time (the previous frame) and P_2 is the prior dependent to the previous frame (i.e., a temporal prior).

$$P_1(\theta) = \prod_{i=1}^n \underbrace{e^{-\lambda_1 A_i} (1 - e^{-\lambda_2 A_i})}_{(a)} \underbrace{P_{\mathbf{u}}(\mathbf{u}_i) P_h(h_i) P_f(f_i) P_i(i_i)}_{(b)}$$

where (a) are the prior probabilities on the image size A_i of an object. The first term penalizes unnecessary overlapping and the second term penalizes very small object size since it is more likely to be noise. (b) are the prior probabilities of the parameters of the objects. $P_{\mathbf{u}}(\mathbf{u}_i)$ is a uniform distribution in the image. $P_h(h_i) \sim \mathcal{N}(\mu_h, \sigma_h^2)$ truncated in the range of $[h_{min}, h_{max}]$, $P_f(f_i) \sim \mathcal{N}(\mu_f, \sigma_f^2)$ truncated in the range of $[f_{min}, f_{max}]$ and $P_i(i_i) \sim \mathcal{N}(0, \sigma_i^2)$. We use a rough adult body size for these parameters.

The temporal prior reflects the smoothness and the connectivity of the trajectories, and the consistency of other parameters. For the convenience of expression of the temporal prior, we re-arrange $\theta^{(t)}$ and $\theta^{(t-1)}$ as $\tilde{\theta}^{(t)} = \{\tilde{\mathbf{m}}_1^{(t)}, \dots, \tilde{\mathbf{m}}_N^{(t)}\}$ and $\tilde{\theta}^{(t-1)} = \{\tilde{\mathbf{m}}_1^{(t-1)}, \dots, \tilde{\mathbf{m}}_N^{(t-1)}\}$ so that one of $\{k_i^{(t)} = k_i^{(t-1)}, \tilde{\mathbf{m}}_i^{(t)} = \phi, \tilde{\mathbf{m}}_i^{(t-1)} = \phi\}$ is true, where $N = |\theta^{(t)} \cup \theta^{(t-1)}|$. $k_i^{(t)} = k_i^{(t-1)}$ means object $k_i^{(t)}$ is a *tracked* object, $\tilde{\mathbf{m}}_i^{(t)} = \phi$ means object $k_i^{(t-1)}$ is a *dead* object, and $\tilde{\mathbf{m}}_i^{(t-1)} = \phi$ means object $k_i^{(t)}$ is a *new* object.

$$P(\theta^{(t)}|\theta^{(t-1)}) = \prod_{i=1}^N P(\tilde{\mathbf{m}}_i^{(t)}|\tilde{\mathbf{m}}_i^{(t-1)}). \quad (3)$$

The temporal prior of each object follows the definition

$$P(\tilde{\mathbf{m}}_i^{(t)}|\tilde{\mathbf{m}}_i^{(t-1)}) = \begin{cases} P((k_i^{(t)}, \mathbf{m}_i^{(t)})|(k_i^{(t-1)}, \mathbf{m}_i^{(t-1)})) & = P(\mathbf{m}_i^{(t)}|\mathbf{m}_i^{(t-1)}); \\ P((k_i^{(t)}, \mathbf{m}_i^{(t)})|\phi) & = P_{new}(\mathbf{m}_i^{(t)}); \\ P(\phi|(k_i^{(t-1)}, \mathbf{m}_i^{(t-1)})) & = P_{dead}(\mathbf{m}_i^{(t-1)}). \end{cases}$$

We assume that the position and the inclination of an object follow constant velocity models with Gaussian noise and the height and thickness follow Gaussian distributions. Therefore we use Kalman filters for temporal estimation.

$$P(\mathbf{m}_i^{(t)}|\mathbf{m}_i^{(t-1)}) = P(\mathbf{u}_i^{(t)}|\mathbf{u}_i^{(t-1)})P(h_i^{(t)}|h_i^{(t-1)})P(w_i^{(t)}|w_i^{(t-1)})P(i_i^{(t)}|i_i^{(t-1)}) \\ \times \exp\{-(\mathbf{u}_i^{(t)} - \bar{\mathbf{u}})^T \bar{\Sigma}^{-1} (\mathbf{u}_i^{(t)} - \bar{\mathbf{u}})\} \\ \exp\{-(\frac{h_i^{(t)} - \bar{h}}{\bar{\sigma}_h})^2\} \exp\{-(\frac{w_i^{(t)} - \bar{w}}{\bar{\sigma}_w})^2\} \exp\{-(\frac{i_i^{(t)} - \bar{i}}{\bar{\sigma}_i})^2\} \quad (4)$$

where $(\bar{\mathbf{u}}, \bar{\Sigma})$, $(\bar{h}, \bar{\sigma}_h^2)$, $(\bar{w}, \bar{\sigma}_w^2)$, $(\bar{i}, \bar{\sigma}_i^2)$ are the predicted mean and variance (covariance matrix) of the position, height, thickness and inclination of the object i from their respective Kalman filters.

$P_{new}(\mathbf{m}_i^{(t)}) = P_{new}(\mathbf{u}_i^{(t)})$ and $P_{dead}(\mathbf{m}_i^{(t-1)}) = P_{dead}(\mathbf{u}_i^{(t-1)})$ are the penalties of the initialization of a new track and the termination of an existing track respectively. They are set empirically according to the distance of object to the entrances/exits (the boundaries of the image and other areas that people move in/out of the view). The probabilities are high when the object is close to the entrances/exits and vice versa.

2.5 Multi-object joint likelihood

Given a state θ , we partition the image into different regions corresponding to different objects and the background. We denote S_i as the region (a mask) of an object i defined by \mathbf{m}_i , and \tilde{S}_i as the visible part of S_i . The visible part of an object is determined by the depth order of all the objects, which is available given their 3D positions and the camera model. The entire object region $S = \cup_{i=1}^n S_i = \sum_{i=1}^n \tilde{S}_i$, since \tilde{S}_i are disjoint regions. We use \bar{S} to denote the supplementary region of S , or the non-object region. The relationship of the regions is shown graphically using an elliptic object model in Fig.2.

2.5.1 Single object likelihood

For an isolated object whose parameter is \mathbf{m}_i with a correspondence in the previous frame, we evaluate the likelihood of the image within S_i

$$P(I^{S_i}|\mathbf{m}_i) \propto \exp\{-\underbrace{\lambda_b |S_i| B(\mathbf{p}_i, \mathbf{d}_i)}_{(1)} + \underbrace{\lambda_f |S_i| B(\mathbf{p}_i, \bar{\mathbf{p}}_i)}_{(2)}\}$$

where \mathbf{d}_i is the color histogram of the background image within the object mask, $\bar{\mathbf{p}}_i$ is the color histogram estimated during the object is tracked, both weighted by a kernel function $k(\cdot)$, $|S_i|$ is the area of the object. $B(\mathbf{p}, \mathbf{d})$ is the Bhattacharyya coefficient of two histograms. $B(\mathbf{p}, \mathbf{d}) = \sum_{j=1}^m \sqrt{p_j d_j}$ reflects the similarity of two histograms. Such a metric has been used for color-based tracking in [2].

This likelihood favors both the difference to the background and the similarity with its correspondence in the previous frame, which enables simultaneously detection and tracking in the same object function. We call the two terms *background exclusion* and *object attraction* respectively. λ_b and λ_f weight the two terms and we use $\lambda_b = \lambda_f = 0.5$. The object attraction term is the same as the likelihood function used in [2]. For an object without a correspondence (i.e. a new object), only the background exclusion part is used.

The single object likelihood can be optimized efficiently w.r.t. the position (assuming the object size is a constant in one iteration) using the mean shift technique similar to [2]. The derivation of the position update rule is given in the Appendix. Compared to the original color-based mean shift tracking, the background exclusion term can utilize a known background

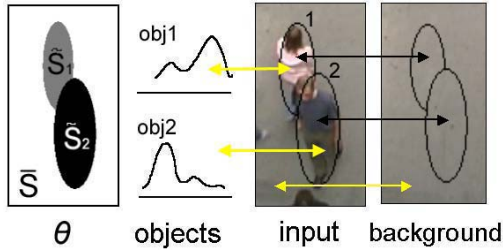


Figure 2: First pane: the relationship of visible object regions and the non-object region. Rest panes: the color likelihood model. In \tilde{S}_i , the likelihood model penalizes the similarity of the input color histograms and the corresponding background color histogram and favors the similarity with its correspondence. In \bar{S} , the likelihood penalizes the difference with the background model. Note that the elliptic models are used for illustration.

model, which is available for a stationary camera. As we observe in our experiments, tracking using the above likelihood is more robust to the change of appearance of the object (*e.g.*, when going into the shadow) compared to using the object attraction term alone.

2.5.2 Multi-object joint likelihood

In case of multiple objects which can possibly overlap in the image, the likelihood of the image given the state cannot be simply decomposed into the likelihood of each individual objects due to the visibility. Instead, a joint likelihood of the whole image given all objects and the background model needs to be considered.

The likelihood of the object region

$$P(I^S|\theta) = \prod_{i=1}^n P(I^{\tilde{S}_i}|\mathbf{m}_i, \theta) \propto \exp\{\lambda_S \sum_i |\tilde{S}_i| B(\mathbf{p}_i, \mathbf{d}_i)\}$$

The likelihood of the non-object region

$$P(I^{\bar{S}}|\theta) = \prod_{j \in \bar{S}} (P_b(I_j))^{\lambda_{\bar{S}}} \propto \exp\{-\lambda_{\bar{S}} \sum_{j \in \bar{S}} e_j\}$$

where $e_j = \log(P_b(I_j))$, as defined in Equ.1.

The likelihood of the entire image is $P(I|\theta) = P(I^S|\theta)P(I^{\bar{S}}|\theta)$. The color likelihood is illustrated in Fig.2. The posterior probability is obtained by combining the prior (Equ.2) and the likelihood.

3 Computing MAP by Efficient MCMC

Computing the MAP is an optimization problem. Due to the joint likelihood, the dimensionality of the state space is proportional to the number of objects in the scene. Since the object number is unknown, the solution space contains subspace of varying dimensions. It also involves both discrete variable (*i.e.* the correspondence) and continuous variables. This has made the optimization challenging.

We use Markov chain Monte Carlo with jump/diffusion dynamics to sample posterior probability. Jumps make the Markov chain to move between different subspaces and traverse the discrete variables; diffusions make the Markov chain to sample continuous variables. In the process of sampling, the optimal solution is found and the uncertainty associated with the solution is also obtained.

The Metropolis-Hasting algorithm can be used to design a Markov chain with stationary distribution $\mathcal{P}(\theta) = P(\theta^{(t)}|I^{(t)}, \theta^{(t-1)}, Bg^{(t-1)})$. At each iteration g , we sample a candidate state θ' according to θ_{g-1} from a *proposal distribution* $q(\theta_g|\theta_{g-1})$. The candidate state θ' is accepted with the following probability:

$$p = \min \left\{ 1, \frac{\mathcal{P}(\theta')q(\theta_{g-1}|\theta')}{\mathcal{P}(\theta_{g-1})q(\theta'|\theta_{g-1})} \right\} \quad (5)$$

If the candidate state θ' is accepted, $\theta_g = \theta'$, otherwise, $\theta_g = \theta_{g-1}$. It can be proven that the Markov chain constructed this way has its stationary distribution equal to $\mathcal{P}(\cdot)$, independent of the choice of the proposal probability $q(\cdot)$ and the initial state θ_0 [8]. However, the choice of the proposal probability $q(\cdot)$ can affect the efficiency of the MCMC significantly. Random proposal probabilities will lead to very slow convergence rate while more informed proposal probabilities ([9]) will make the Markov chain traverse the solution space more efficiently. If the proposal probability is informative enough so that each sample can be thought of as a *hypothesis*, then the MCMC approach can be thought of as a stochastic version of the *hypothesize and test* approach.

3.1 Markov chain dynamics

In order for the Markov chain to traverse the solution space, we design the following reversible dynamics. We assume that we have the sample in $(g-1)$ -th iteration $\theta_{g-1}^{(t)} = \{(k_1, \mathbf{m}_1), \dots, (k_n, \mathbf{m}_n)\}$ and now propose a candidate θ' for the g -th iteration (t is omitted where there is no ambiguity).

Object addition We sample the parameters of a new human hypothesis $(k_{n+1}, \mathbf{m}_{n+1})$ from $q_a(k, \mathbf{m})$ and add it to θ_{g-1} .

$$q(\theta'|\theta_{g-1}) = p_{add}q_a(k_{n+1}, \mathbf{m}_{n+1})$$

Object removal Randomly select an existing human hypothesis $r \in [1, n]$ with a uniform distribution and remove it. If k_r has a correspondence in $\theta^{(t-1)}$, then that object becomes *dead*.

$$q(\theta'|\theta_{g-1}) = p_{remove}q_r(r) = p_{remove}(1/n)$$

Establish correspondence Randomly select a *new* object r in $\theta_{g-1}^{(t)}$ and a *dead* object in $\theta^{(t-1)}$, and establish their temporal correspondence.

$$q(\theta'|\theta_{g-1}) = p_{establish}q_e(r, r')$$

We choose $q_{x2}(r, r') \propto \frac{1}{\|\mathbf{u}_r - \mathbf{u}_{r'}\|^2}$ for all the qualifying pairs.

Break correspondence Randomly select an object r where k_r is in $\theta^{(t-1)}$ with a uniform distribution and change k_r to a *new* object (and same object in $\theta^{(t-1)}$ becomes *dead*).

$$q(\theta' | \theta_{g-1}) = p_{break}(1/n')$$

where n' is the number of objects in $\theta_{g-1}^{(t)}$ that have correspondences in the previous frame.

Exchange identity Exchange the IDs of two close-by objects. Randomly select two objects $r_1, r_2 \in [1, n]$ and exchange their IDs.

$$q(\theta' | \theta_{g-1}) = p_{exchange} q_x(r_1, r_2)$$

We choose $q_x(r_1, r_2) \propto \frac{1}{\|\mathbf{u}_{r_1} - \mathbf{u}_{r_2}\|^2}$. One of them can be a new object. Identities exchange can also be realized by the compositions of establishing/breaking correspondence. It is used to ease the traversal since establishing and breaking correspondences may lead to a big decrease in the probability and are less likely to be accepted.

Parameter update Update the continuous parameters of an object. Randomly select an existing human hypothesis $r \in [1, n]$ with a uniform distribution, and update its continuous parameters

$$q(\theta' | \theta_{g-1}) = p_{diff}(1/n) q_d(\mathbf{m}'_r | \mathbf{m}_r)$$

The first 5 are *jump* dynamics and the last one is *diffusion*. In each iteration, one of the above dynamics is chosen randomly. We use $p_{add} = 0.1$, $p_{remove} = 0.1$, $p_{establish} = 0.1$, $p_{break} = 0.1$, $p_{exchange} = 0.1$ and $p_{diff} = 0.5$.

3.2 Informed proposal probabilities

In theory, the proposal probability $q()$ does not affect convergence. However, different $q()$ lead to different performances. The speed of the Markov chain strongly depends on the proposal probabilities. In this application, the proposal probability of adding a new object ($q_a(k, \mathbf{m}) = q_a(k, \mathbf{m} | I^{(t)}, Bg^{(t-1)}, \theta^{(t-1)})$) and the update of the object parameters ($q_d(\mathbf{m}'_i | \mathbf{m}_i) = q_d(\mathbf{m}'_i | \mathbf{m}_i, I^{(t)}, Bg^{(t-1)}, \theta^{(t-1)})$) are the two most important ones. We use the following informed proposal probabilities to make the Markov chain more intelligent and thus have a higher acceptance rate.

3.2.1 Object addition

We use two ways to add new objects.

$$q_a(k, \mathbf{m}) = \lambda_{a1} q_{a1}(k, \mathbf{m}) + \lambda_{a2} q_{a2}(k, \mathbf{m}).$$

$q_{a1}()$ samples \mathbf{m} first and then k , $q_{a1}(k, \mathbf{m}) = q(\mathbf{m})q(k | \mathbf{m})$. $q(\mathbf{m}) = q(\mathbf{u})q_h(h)q_f(f)q_i(i)$. $q(\mathbf{u})$ answers the question "where to add a new human hypothesis". We have shown in [12] that human hypothesis can be generated efficiently by various image features: 1) head candidates from boundary analysis, 2) head candidates from edge analysis, and 3) projection

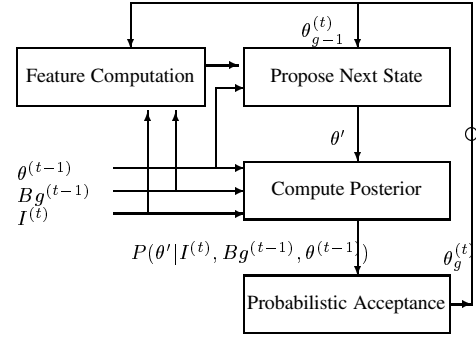


Figure 3: The block diagram of the MCMC tracking algorithm.

analysis of the foreground residue (*i.e.*, foreground with the existing objects carved out). The details can be found in [12]. The rest of the parameters (h, f, i) are sampled from their respective prior distributions.

$q(k | \mathbf{m}) = q(k | \mathbf{u})$ is to sample k from $\{k_{d_1}^{(t-1)}, \dots, k_{d_{n_d}}^{(t-1)}, new\}$ according to $P(\mathbf{u} | \mathbf{u}_{d_i}^{(t-1)})$ (see Equ.4), $i = 1, \dots, n_d$ and $P_{new}(\mathbf{u})$, where n_d is the number of dead objects.

$q_{a2}()$ samples k first and then \mathbf{m} , $q_{a2}(k, \mathbf{m}) = q(k)q(\mathbf{m} | k)$. $q(k)$ randomly samples a dead object $k_{d_r}^{(t-1)}$ and sample \mathbf{m} from $P(\mathbf{m} | \mathbf{m}_{d_r}^{(t-1)})$ (Equ.4).

3.2.2 Parameter update

We use two ways to update the model parameters.

$$q_d(\mathbf{m}'_r | \mathbf{m}_r) = \lambda_{d1} q_{d1}(\mathbf{m}'_r | \mathbf{m}_r) + \lambda_{d2} q_{d2}(\mathbf{m}'_r | \mathbf{m}_r)$$

$q_{d1}()$ uses stochastic gradient decent to update the object parameters.

$$q_{d1}(\mathbf{m}'_r | \mathbf{m}_r) \sim \mathcal{N}(\mathbf{m}_r - k \frac{dE}{d\mathbf{m}}, \mathbf{w})$$

where $E = -\log P(\theta^{(t)} | I^{(t)}, \theta^{(t-1)}, Bg^{(t-1)})$ is the energy function, k is a scalar to control the step size and \mathbf{w} is random noise to avoid local maximum.

A mean shift vector computed in the visible region provides an approximation of the gradient of the object likelihood w.r.t. the position. Since other components of the posterior probability changes relatively slowly, they can be absorbed in the noise term. The mean shift has an adaptive step size and has a better convergence behavior than numerically computed gradients. The rest of the parameters follow their numerically computed gradients.

$q_{d2}()$ moves the position of the current object to a computed head candidates close-by while keeping the rest of the parameter unchanged.

3.3 Summary of the algorithm and filtering with adaptive measurement noise

The diagram of the the algorithm is shown in Fig.3. This iterative process starts from an initial state. In each iteration,

a candidate is proposed from the state in the previous iteration assisted by image features. The candidate is accepted probabilistically according to Equ.5. The computation of the joint likelihood at each iteration can be done incrementally in the neighborhood of the object which is being changed, which is in contrast to the full computation of particle filters. After a number (M) of iterations called *burn-in* period, the samples $\theta_g^{(t)}$ become independent of the initial state and can be regarded as unbiased samples from the posterior probability. The state corresponding to the maximum posterior value up to the current iteration is recorded and it becomes the solution when the given number (N) of iterations is reached. The number of iterations needed to obtain satisfactory results depends on the complexity of the scene. More iterations are needed for a scene containing more humans and more occlusion. The appearance models of the tracked objects are updated after each iteration using an IIR filter.

Since the Markov chain generates samples from the posterior distribution of $\theta^{(t)}$, besides obtaining the MAP estimation, we can also compute other statistics from the samples. Assume we have $N - M$ samples of $\{\theta_{M+1}^{(t)}, \dots, \theta_N^{(t)}\}$ from the posterior probability of $\theta^{(t)}$, with the first M samples discarded. Object k only appeared in N_k samples $\{\theta_{k_1}^{(t)}, \dots, \theta_{k_{N_k}}^{(t)}\}$. The expectation related to object k can be computed as

$$E[f(\mathbf{m})] = \int_{\mathbf{m}} f(\mathbf{m})P(\mathbf{m})d\mathbf{m} \sim \frac{1}{N_k} \sum_{i=1}^{N_k} f(\mathbf{m}_{k_i}) \quad (6)$$

We compute the mean ($f(\mathbf{m}) = \mathbf{m}$) and the covariance (or variance, $f(\mathbf{m}) = (\mathbf{m} - \bar{\mathbf{m}})(\mathbf{m} - \bar{\mathbf{m}})^T$) of the position, height, thickness and inclination.

As mentioned earlier, we use Kalman filters to filter and predict the states of each object. The previous uses of Kalman filters usually assume that the measurement noise is fixed (estimated or empirically given). Here, the covariance of the real measurement noise is estimated by the samples from the posterior probability distribution which results in optimal filtering.

4 Experiments and Evaluations

The above approaches are implemented and experiments are performed on real-life data. In processing each frame, we choose the initial state to be a predicted state (the parameters of each object predicted by their Kalman filters). We show here the result on a sequence which we also used to evaluate the segmentation algorithm in [12]. It provides a direct comparison of the gain of the tracking algorithm. This 900-frame sequence is captured from a camera above a building gate with the camera tilt angle = 40° . A large tilt angle results in significant perspective effect on human shape in images, which can show the strength of the use of 3D shape model. The sequence contains dense traffic with 23 people going out of and 10 people going into the building. It contains multiple people walking closely resulting in persistent overlapping. Due to the spatial proximity, the blobs of the foreground contain

multiple people in most cases. In many case, multiple people enter the scene together, this will fail the tracking algorithms which rely on initializing objects when they are isolated. The maximum number of people in the scene is 13; such a dimensionality may require an infeasibly large number of samples in a particle filter-based approach.

We show in Fig.4 the selected frames from the result of the sequence. The ID of each human is shown on his head. The readers are advised to view the result in video format. Most of the tracks are correct and the initializations of the objects are prompt. The following evaluation will show the performance quantitatively.

Firstly we evaluate the results by the trajectory-based errors. Trajectories whose lengths are less than 10 frames are discarded and not counted in the evaluation. Among the 33 human objects, trajectories of 3 objects are broken once (ID 28→ID 35, ID 31→ID 32, ID 30→ID 41, all between frame 387 and frame 447, as marked with colored arrows in the images), and the rest of the trajectories are correct. The trajectory-based error rate is 9.1% (please note the upper limit of this error rate is not 100%). Usually the trajectories are initialized once the humans are fully in the scene, some even start when the objects are partially in. Only the initializations of three objects (objects 31, 50, 52) are noticeably delayed (by 50, 55, 60 frames respectively after they are fully in the scene). Partial occlusion (objects 31, 50, 52) or/and lack of contrast with the background (object 31, 52) are the causes of the delays. Secondly we perform the frame-based evaluation and compare the results with the segmentation results in [12]. The detection rate and the false alarm rate is 98.13% and 0.27% respectively. The detection rate and the false alarm rate of the same sequence by using segmentation alone are 92.82% and 0.18%. With tracking, not only the temporal correspondences are obtained, but also the detection rate is increased by a large margin while the false alarm rate is kept low.

The computation needed for the tracking is also reduced compared to segmentation since the correlation of the adjacent frames is considered. For each frame of the above sequence, we use 300 iterations in contrast to 1000 iterations as in the segmentation. Now the system runs 3 fps on a P4 2.7GHz PC, with un-optimized C++ code.

5 Conclusion and Future Work

We have presented a principled approach to simultaneously detect and track humans in a crowded scene acquired from a stationary camera. Our contribution in this work is: 1) a Bayesian framework of the multi-object tracking problem, including a color-based joint likelihood which enables simultaneously detection and tracking; 2) an efficient MCMC-based approach to compute the optimal solution: the design of reversible dynamics to explore the solution space and the use of informed proposal probabilities from image features for faster convergence; 3) the extension of the mean-shift tracking to incorporate background information in the context of a stationary camera. Experiments and evaluations on challenging real-



Figure 4: Selected frames of the tracking results on a sequence. The numbers on the heads show identities. (Please note that the two people who are sitting on two sides are in the background model, therefore not detected.) The colored arrows points to the three objects whose trajectories are broken.

life data show promising results.

This work could be improved/extended as the following. 1) We are interested in extend the system to track multiple class of objects (*e.g.*, humans and cars). It can be enabled by adding model switching in the dynamics. 2) Tracking, operating in a 2-frame interval, has a very local view therefore ambiguities inevitably exist, especially in the case of tracking multiple close-by or overlapping objects. The analysis in the level of trajectories may resolve the local ambiguities. The analysis may take into account the prior knowledge on the valid object trajectories including their starting and ending points.

References

- [1] R.T. Collins, Mean-shift Blob Tracking through Scale Space, *Proc. CVPR*, 2003.
- [2] D. Comaniciu, V. Ramesh and P. Meer, Kernel-Based Object Tracking, *IEEE Trans. PAMI*, vol.25, no.5, 2003.
- [3] A. M. Elgammal and L. S. Davis, Probabilistic Framework for Segmenting People under Occlusion, *Proc. ICCV*, 2001.
- [4] S. Haritaoglu, D. Harwood and L. S. Davis, W4: Real-Time Surveillance of People and Their Activities, *IEEE Trans. PAMI*, vol.22, no.8, 2000.
- [5] M. Isard and J. MacCormick, BraMBLe: A Bayesian Multiple-Blob Tracker, *Proc. ICCV*, 2001.
- [6] C. Sminchisescu, and B. Triggs, Kinematic Jump Processes For Monocular 3D Human Tracking, *Proc. CVPR*, 2003.
- [7] H. Tao, H. S. Sawhney and R. Kumar, A sampling algorithm for tracking multiple objects, *Proc. Workshop Vision Algorithms, with ICCV 99*.
- [8] L. Tierney, Markov chain concepts related to sampling algorithms, *Markov Chain Monte Carlo in Practice*, Chapman and Hall, pp. 59-74, 1996.
- [9] Z.W. Tu, X. Chen, A. Yuille and S.C. Zhu, Image Parsing: Segmentation, Detection and Recognition, *Proc. ICCV*, 2003.
- [10] C.R. Wren, A. Azarbayejani, T. Darrell and A.P. Pentland, Pfunder: Real-time Tracking of the Human Body, *IEEE Trans. PAMI*, vol.19, no.7, 1997.
- [11] T. Zhao, R. Nevatia and F. Lv, Segmentation and Tracking of Multiple Humans in Complex Situations, *Proc. CVPR*, 2001.
- [12] T. Zhao and R. Nevatia, Bayesian Multiple Human Segmentation in Crowded Situations, *Proc. CVPR*, 2003.

Appendix: Single object tracking with background knowledge using mean shift

The object is represented by an elliptical region (we use the minimum ellipse which contains the object). The object is normalized into a unit circle for the convenience of derivation. Denote $\tilde{\mathbf{p}}$ as the learnt color histogram of the object, $\mathbf{p}(\mathbf{u})$ as the object color histogram with the object center at \mathbf{u} and $\mathbf{b}(\mathbf{u})$ as the color histogram of the background at the corresponding region. Let $\{\mathbf{x}_i\}_{i=1,\dots,n}$ be the pixel locations in the region with the object center at \mathbf{u} . A kernel with profile $k(\cdot)$ is used to assign smaller weights to the pixels farther away from the center, considering those closer to the boundary may contain more noise. An m -bin color histogram $\mathbf{p}(\mathbf{u}) = \{p_j(\mathbf{u})\}_{j=1\dots m}$, $\sum_{j=1}^m p_j(\mathbf{u}) = 1$, is constructed as $p_j(\mathbf{u}) = \sum_{i=1}^n k(\|\mathbf{x}_i\|^2) \delta[b_f(\mathbf{x}_i) - j]$, where function $b_f(\cdot)$

maps a normalized pixel location to the histogram bin of the color of that pixel location, and δ is the delta function. Similar for $\tilde{\mathbf{p}}$ and \mathbf{d} .

We would like to optimize

$$L(\mathbf{u}) = -\lambda_b \underbrace{B(\mathbf{p}(\mathbf{u}), \mathbf{b}(\mathbf{u}))}_{L_1(\mathbf{u})} + \lambda_f \underbrace{B(\mathbf{p}(\mathbf{u}), \tilde{\mathbf{p}})}_{L_2(\mathbf{u})}$$

where $B(\cdot)$ is the Bhattachayya coefficient.

By applying Taylor expansion at $\mathbf{p}(\mathbf{u}_0)$ and $\mathbf{d}(\mathbf{u}_0)$ (\mathbf{u}_0 is a predicted position of the object), we have

$$\begin{aligned} L_1(\mathbf{u}) &= B(\mathbf{p}(\mathbf{u}), \mathbf{d}(\mathbf{u})) = B(\mathbf{u}) \\ &\approx B(\mathbf{u}_0) + B'_p(\mathbf{u}_0)(\mathbf{p}(\mathbf{u}) - \mathbf{p}(\mathbf{u}_0)) \\ &\quad + B'_d(\mathbf{u}_0)(\mathbf{d}(\mathbf{u}) - \mathbf{d}(\mathbf{u}_0)) \\ &= c_1 + \sum_{u=1}^m \sqrt{\frac{d_u(\mathbf{u}_0)}{p_u(\mathbf{u}_0)}} p_u(\mathbf{u}) + \sum_{u=1}^m \sqrt{\frac{p_u(\mathbf{u}_0)}{d_u(\mathbf{u}_0)}} d_u(\mathbf{u}) \\ &= c_1 + \sum_{i=1}^n k(\|\frac{\mathbf{u} - \mathbf{x}_i}{h}\|^2) w_i^b \end{aligned}$$

where $w_i^b =$

$$\sum_{u=1}^m \left(\sqrt{\frac{d_u(\mathbf{u}_0)}{p_u(\mathbf{u}_0)}} \delta[b_f(\mathbf{x}_i) - u] + \sqrt{\frac{p_u(\mathbf{u}_0)}{d_u(\mathbf{u}_0)}} \delta[b_b(\mathbf{x}_i) - u] \right)$$

Similarly, also in [2],

$$\begin{aligned} L_2(\mathbf{u}) &= B(\mathbf{p}(\mathbf{u}), \tilde{\mathbf{p}}) \\ &\approx \frac{1}{2} \sum_{u=1}^m \sqrt{p_u(\mathbf{u}_0) \tilde{p}_u} + \frac{1}{2} p_u(\mathbf{u}) \sqrt{\frac{\tilde{p}_u}{p_u(\mathbf{u}_0)}} \\ &= c_2 + \sum_{i=1}^{n_h} w_i^f k(\|\frac{\mathbf{u} - \mathbf{x}_i}{h}\|^2), \end{aligned}$$

where $w_i^f = \sum_{u=1}^m \sqrt{\frac{\tilde{p}_u}{p_u(\mathbf{u}_0)}} \delta[b_f(\mathbf{x}_i) - u]$. Therefore,

$$L(\mathbf{u}) = c_1 + c_2 + \sum_{i=1}^n \underbrace{(\lambda_f w_i^f - \lambda_b w_i^b)}_{w_i} k(\|\frac{\mathbf{u} - \mathbf{x}_i}{h}\|^2)$$

The last term of $L(\mathbf{u})$ is the density estimate computed with kernel profile k at \mathbf{u} , with weights that can be computed. The mean-shift algorithm with negative weight [1] applies. By using the Epanechikov profile ([2], $L(\mathbf{u})$ will be increased with the new location moved to

$$\mathbf{u}' \leftarrow \frac{\sum_{i=1}^n \mathbf{x}_i w_i}{\sum_{i=1}^n |w_i|}. \quad (7)$$