

# An Advanced Real-Time Multiple Object Tracker in Variant Outdoor Environments

Hamed Moradi Pour\* and Saeid Fazli

Electrical Engineering Department, Zanzan University, Zanzan, Iran

## Abstract

Tracking of humans in dynamic scenes has been an important topic of research. There has been considerable work in tracking humans and other objects in recent years. A real-time method for tracking multiple moving objects based on effective Gaussian Mixture Model (GMM), and identifying the moving objects with Joint Probability Data Association Filter (JPDAF) is proposed in this paper. Most tracking algorithms have better performance under static background but get worse results under background with fake motions. Therefore, most of the tracking algorithms are used in indoor environment. An adaptive Gaussian Mixtures has a nice property in resolving this problem. This paper uses recursive equations to constantly update the parameters of a Gaussian Mixture Model and to simultaneously select the appropriate number of components for each pixel. Therefore, this method is more time and memory efficient than the common GMM with the fixed component number. In tracking multiple moving objects, problems occur when objects pass across each other. The JPDAF method is used in this paper to solve this problem. Moreover, it can effectively deal with the various scenes such as the indoor scene, the outdoor scene, and the cluttered scene. The experimental results on our test sequence demonstrate the high efficiency of the proposed method.

**Keywords:** Adaptive background modeling, Maximum likelihood, Recursive equations, Parameter estimation

## Introduction

Multi target tracking, the association of detected points into sequences over time, is an important NP-hard (non deterministic polynomial-time hard) problem. Considerable efforts have been conducted to design tractable methods by reducing its complexity. Multi target tracking has a broad range of applications including some well-known application such as video surveillance and monitoring [1] sonar based tracking of sea animals or submarines, robot control [2] eye tracking [3] video based identification and tracking of people and car for surveillance or security purposes, and many more [4,5].

## Motion detection

Nearly every visual surveillance system starts with motion detection. Subsequent processes such as tracking and behavior recognition are greatly dependent on it. Accurate and real-time motion segmentation will greatly improve the performance of object tracking. The difficulty in motion segmentation lies in how to reconstruct the background image to adapt to the scene changes, and how to remove the false foreground pixels that may lead to inaccurate segmentation results. Most back grounding methods involve continuously estimating a statistical model of the variation for each pixel. A common method of adaptive back grounding is averaging the images over time, creating a background approximation which is similar to the current static scene except where motion occurs. While this is effective in situations where objects move continuously and the background is visible a significant portion of the time, it is not robust to scenes with many moving objects particularly if they move slowly.

In [6] a nonparametric kernel density estimation method is presented, with higher sensitivity in detecting foreground regions. This method updates the background from the sample data directly by using the Gaussian kernel Parzen estimator. If the sample size is large enough, this method can very accurately describe the background variations. So, the performance of this method is based on the sample size. The more precisely the background is estimated, the more memory and CPU time will be needed.

The authors in [7,8] use mean-shift analysis to track the distribution of a target object. The method shows good performance

in crowded scenes and cluttered backgrounds. However, the object shape is constrained to be an ellipse, which restricts the applicability of the approach when dealing with deformable objects. Also, the performance of the approach drastically decreases under variations in the appearance of the objects.

In [9] proposed a method for detecting and tracking multiple moving objects based on discrete wavelet transform and identifying the moving objects by their color and spatial information. Many tracking algorithms have better performance under static background but get worse results under background with fake motions.

Therefore, most of the tracking algorithms are used indoors instead of outdoor environment. Since discrete wavelet transform has a nice property that it can divide a frame into four different frequency bands without loss of the spatial information, it is adopted to solve this problem due to the fact that most of the fake motions in the background can be decomposed into the high frequency wavelet sub-band. In tracking multiple moving objects, many applications have problems when objects pass across each other. Color and spatial information are used in this paper to solve this problem.

A mixture Gaussian model can represent the complex distribution of each pixel, and it became a popular technique for modeling a dynamic background. Unfortunately, conventional Gaussian mixture model (GMM) suffers from slow convergence at the initial stage [10-13] and the segmentation accuracy largely depends on the threshold parameters.

## Object tracking

After motion detection, surveillance systems generally track

\*Corresponding author: Hamed Moradi Pour, Electrical Eng. Department, Zanzan University, Zanzan, Iran, E-mail: [h.moradi@znu.ac.ir](mailto:h.moradi@znu.ac.ir)

Received April 17, 2012; Accepted July 26, 2012; Published July 31, 2012

Citation: Pour HM, Fazli S (2012) An Advanced Real-Time Multiple Object Tracker in Variant Outdoor Environments. J Appl Computat Math 1:118. doi:10.4172/2168-9679.1000118

Copyright: © 2012 Pour HM, et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

moving objects from one frame to another in an image sequence. Tracking over time typically involves matching objects in consecutive frames using features such as points, lines or blobs. A visual-based multi-target tracking system should be able to track a variable number of objects in a dynamic scene and maintain the correct identities of the targets regardless of occlusions and any other visual perturbations. It is a very complicated and challenging problem; extensive research work has been done.

The most commonly used framework for tracking is that of Bayesian Sequential Estimation. This framework is probabilistic in nature, and thus facilitates the modeling of uncertainties due to inaccurate models, sensor errors, environmental noise, etc. The general recursions update the posterior distribution of the target state, also known as the filtering distribution, through two stages: a prediction step that propagates the posterior distribution at the previous time step through the target dynamics to form the one step ahead prediction distribution, and a filtering step that incorporates the new data through Bayes' rule to form the new filtering distribution. In theory the framework requires only the definition of a model for the target dynamics, a likelihood model for the sensor measurements, and an initial distribution for the target state. In most practical tracking applications the sensor yield unlabelled measurements of the targets. This leads to a combinatorial data association problem that is very challenging when targets have a small separation compared to the measurement errors. Furthermore, clutter measurements may arise due to multi-path effects, sensor errors, etc., further increasing the complexity of the data association problem.

A large number of strategies are available to solve the data association problem. These can be broadly categorized as either single frame assignment methods, or multi-frame assignment methods. The Multiple Hypotheses Tracker (MHT) attempts to keep track of all the possible association hypotheses over time. The idea of MHT [14] is to associate each measurement with one of the existing tracks, or to form a new track from the measurement. The MHT algorithm calculates the likelihoods of the measurements and the posterior probabilities of the hypotheses, storing only the most probable hypotheses. To enhance the computational efficiency, heuristic methods such as gating, hypothesis merging, clustering and several other strategies can be employed.

In this respect the Joint Probabilistic Data Association Filter (JPDAF) [15,16] is more appealing. At each time step infeasible hypotheses are pruned away using a gating procedure. A filtering estimate is then computed for each of the remaining hypotheses, and combined in proportion to the corresponding posterior hypothesis probabilities. JPDA approximates the posterior distributions of the targets as separate Gaussian distributions for each target. If the number of targets is T, then T separate Gaussian distributions are maintained. The number of Gaussian distributions is kept constant by integrating over the distribution of data associations of the previous step. This results in an algorithm where each of the target estimates gets updated by every measurement with weights that depend on the predicted probabilities of the associations. Gating is used for limiting the number of measurements for each track. If the predicted probabilities are too low (i.e., below a predefined threshold) for certain targets, those targets are not updated at all. Clutter measurements can be modeled similarly.

### Gaussian Mixture Model

A GMM is a statistical model that assumes the data originate from a weighted sum of several Gaussian distributions. There are two important problems when GMM is used to model multivariate data: the selection of the number of components and the initialization. In this paper, use a flexible method for foreground segmentation that

proposed in [17,18] and [19-21]. We still use the finite GMM to model the scene background, but a stochastic approximation procedure is used to recursively estimate the parameters of the GMM, and to simultaneously obtain the asymptotically optimal number of the mixture components. Therefore, this method is highly memory and time efficient. Moreover, this method can effectively deal with many scenes, such as the indoor scene, the outdoor scene, and the clutter scene.

The value of a pixel at time t in RGB is denoted by  $\bar{x}^{(t)}$ . Some other color space or some local features could also be used. We choose a reasonable time adaptation period T. At time t we have  $X_T = \{x^{(t)}, \dots, x^{(t-T)}\}$ . For each new sample we update the training data set  $X_T$  and reestimate the density. It is said that  $X_T$  follows a M components finite mixture distribution if its probability density function can be written as:

$$p(\bar{x}; \bar{\theta}) = \sum_{m=1}^M \pi_m p_m(\bar{x}; \theta_m), \quad \text{with} \quad \sum_{m=1}^M \pi_m = 1 \quad (1)$$

Where  $\pi_1, \dots, \pi_m$  are the mixing weights, each  $\bar{\theta}_m$  is the set of parameters defining the component, and  $\bar{\theta} = \{\pi_1, \dots, \pi_m, \bar{\theta}_1, \dots, \bar{\theta}_m\}$  are the parameters. We use a GMM with M components:

$$\hat{P}(\bar{x} / X_T, BG + FG) = \sum_{m=1}^M \hat{\pi}_m N(\bar{x}, \hat{\mu}_m, \hat{\sigma}_m^2 I) \quad (2)$$

Where  $\mu_1, \dots, \mu_M$  are the estimates of the means and  $\sigma_1^2, \dots, \sigma_M^2$  are the estimates of the variances that describe the Gaussian components and  $\bar{\theta}_m = [\mu_m, \Sigma_m]$ .

The estimated mixing weights denoted by  $\hat{\pi}_m$  are non-negative and add up to one. The type of the components may be various, but the d-dimension Gaussian function is chosen here. The d-dimension Gaussian component in the mixture is given by:

$$p(\bar{x} / \theta_m) = p(\bar{x} / \mu_m, \Sigma_m) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_m|^{\frac{1}{2}}} e^{-\frac{1}{2}(\bar{x} - \mu_m)^T \Sigma_m^{-1} (\bar{x} - \mu_m)} \quad (3)$$

### Update equations

Given a new data sample  $\bar{x}^{(t)}$  at time t the recursive update equations are:

$$\hat{\pi}_m \leftarrow \hat{\pi}_m + \alpha (o_m^{(t)} - \hat{\pi}_m) \quad (4)$$

$$\hat{\sigma}_m^2 \leftarrow \hat{\sigma}_m^2 + o_m^{(t)} \left( \frac{\alpha}{\hat{\pi}_m} \right) (\bar{\delta}_m^T \bar{\delta}_m - \hat{\sigma}_m^2) \quad (5)$$

$$\hat{\mu}_m \leftarrow \hat{\mu}_m + o_m^{(t)} \left( \frac{\alpha}{\hat{\pi}_m} \right) \bar{\delta}_m \quad (6)$$

Where,  $\bar{\delta}^{(t)} = \bar{\mu}_m$ . Instead of the time interval T that was mentioned above, here the constant  $\alpha$  defines an exponentially decaying envelope that is used to limit the influence of the old data. We keep the same notation having in mind that effectively  $\alpha = \frac{1}{T}$ . For a new sample the ownership  $o_m^{(t)}$  is set to 1 for the "close" component with largest  $\hat{\pi}_m$  and the others are set to zero. We define that a sample is "close" to a component if the Mahalanobis distance from the component is, for example, less than three. The squared distance from the m<sup>th</sup> component is calculated as:

$$D_m^2(\bar{x}^{(t)}) = \frac{\delta_m^T}{\hat{\sigma}_m^2} \quad (7)$$

If there are no “close” components, a new component is generated with  $\pi_{M+1} = \alpha, \mu_{M+1} = \bar{x}^{(t)}$  and  $\hat{\sigma}_{M+1} = \sigma_0$  where  $\sigma_0$  is some appropriate initial variance.

### Background subtraction for foreground

The GMM with  $M$  components as (3) models both the foreground object and the scene background without distinction, that is, some of the mixture components model the foreground objects, and others model the scene background. If one mixture component occurs frequently (with high  $\pi_m$ ) and does not very much (with low  $\sigma_m$ ), it could be deemed to be background [22]. Therefore, the  $K$  mixture components are ordered based  $\bar{\theta}_{ML} = \arg \max_{\theta} (\log p(X|\bar{\theta}))$  on moreover, the first  $B$  components are chosen as a model of the scene background where  $B$  is estimated as:

$$B = \arg \min_b \left( \frac{\sum_{t=1}^b \pi_t}{\sum_{m=1}^m \pi_m} > T \right) \quad (8)$$

$T$  is the threshold, the fraction of the total weight given to the background model. Background subtraction is performed, by marking any pixel of the input frame that is more than 2.0 standard deviations away from any of the  $B$  components as a foreground.

### Parameters estimation

Given a set of  $t$  independent and identically distributed samples  $X = \{\bar{x}^{(1)}, \dots, \bar{x}^{(t)}\}$  the Maximum Likelihood

(ML) estimation of the parameter values is:

$$\bar{\theta}_{ML} = \arg \max_{\theta} \{\log p(X|\bar{\theta})\} \quad (9)$$

Unfortunately, it is difficult to analytically find the estimation from (9) [18]. But the Bayesian maximum a posteriori (MAP) criterion:

$$\hat{\theta}_{MAP} = \arg \max_{\theta} \{\log p(X|\bar{\theta}) + \log p(\bar{\theta})\} \quad (10)$$

Is flexible for the parameters estimation [12]. Where  $p(\bar{\theta})$  is the prior for the mixture parameters. The usual choice for obtaining ML or MAP estimation of mixture models parameters is the EM algorithm, which is an iterative procedure for searching local maxima. The EM algorithm is based on the interpretation of  $X$  as incomplete data. For finite mixture, the missing part is a set of  $t$  label  $Z = \{Z^{(1)}, \dots, Z^{(t)}\}$  associated with the  $t$  samples, indicating which component produced each sample. The log-likelihood of the complete data is:

$$\log p(X, Z|\theta) = \sum_{t=1}^t \sum_{m=1}^M \bar{z}_m^{(t)} \log[\pi_m p(\bar{x}|\bar{\theta}_m)] \quad (11)$$

The EM algorithm starts with some initial parameters  $\theta_{(0)}$ . If to denote the set of the missing part  $Z$ , the estimation  $\hat{\theta}_m$  from the  $k^{\text{th}}$  iteration of the EM algorithm is obtained using the previous estimation  $\hat{\theta}_{m-1}$  by alternately applying the E-step and the M-step.

• **E-step.** Computes the conditional expectation of the complete log-likelihood, given  $X$  and the previous estimation  $\hat{\theta}_{m-1}$ .

$$\begin{aligned} Q(\bar{\theta}, \bar{\theta}_{(m-1)}) &= E_z [\log p(X, Z|\bar{\theta}, \bar{\theta}_{(m-1)})] \\ &= \sum_{\text{all possible } z} p(Z|X, \bar{\theta}_{(m-1)}) \log p(x, z|\bar{\theta}) \end{aligned} \quad (12)$$

• **M-step.** Updates the parameters estimation in the case of MAP estimation, according to (10).

$$\theta_m = \arg \max_{\theta} \{Q(\bar{\theta}, \hat{\theta}_{m-1}) + \log p(\bar{\theta})\} \quad (13)$$

### Estimating the number of mixture components

Obviously, in order to use the EM algorithm, the appropriate number of mixture components  $M$  must be known. With too many components, the mixture may over-fit the data, while a mixture with too few components may under-fit. Besides, the appropriate number of components can reveal some important existing underlying structure that characterizes the data. However, introducing more mixture components always increases the log-likelihood. The balance is achieved by introducing  $p(\bar{\theta}(M))$  that penalizes complex solutions.

$$\log p(X|\bar{\theta}(M)) + \log p(\bar{\theta}(M)) \quad (14)$$

Where  $p(\bar{\theta}(M))$  is the previously defined Dirichlet prior as:

$$p(\bar{\theta}(M)) \exp \sum_{m=1}^M c_m \log \pi_m = \prod_{m=1}^M \pi_m^{c_m} \quad (15)$$

Where coefficients  $C_m$  equal to  $-\frac{N}{2}$ , where  $N$  presents the number of parameters per component of the mixture.

### Online parameters estimation

For the ML estimate, the following holds:

$$\frac{\partial}{\partial \bar{\theta}} \log p(X|\bar{\theta}) = 0 \quad (16)$$

The mixing weights are constrained to sum up to 1. Take this into account by introducing the Lagrange multiplier  $\lambda$  and get:

$$\frac{\partial}{\partial \pi_m} (\log p(X|\bar{\theta}) + \lambda (\sum_{m=1}^M \pi_m - 1)) = 0 \quad (17)$$

After getting rid of  $\lambda$ , for  $t$  data samples will get:

$$\hat{\pi}_m^{(t)} = \frac{1}{t} \sum_{i=1}^t o_m^{(i)}(\bar{x}^{(i)}) \quad (18)$$

With the “ownerships” defined as:

$$o_m^t(\bar{x}) = \pi_m^{(t)} p_m(\bar{x}; \theta_m^{(t)}) / p(\bar{x}; \hat{\theta}^{(t)}) \quad (19)$$

Similarly, for the MAP solution, have:

$$\frac{\partial}{\partial \pi_m} (\log p(x|\bar{\theta}) + \log p(\bar{\theta}(M)) + \lambda (\sum_{m=1}^M \pi_m - 1)) = 0 \quad (20)$$

Where  $p(\bar{\theta}(M))$  is the mentioned Dirichlet prior (15). For  $t$  data samples, get:

$$\pi_m^{(t)} = \frac{1}{K} \left( \sum_{t=1}^t o_m^{(i)}(x^{(i)}) - c \right) \quad (21)$$

where

$$K = \sum_{m=1}^M (\sum_{t=1}^t o_m^{(i)}(\bar{x}^{(i)}) - c) = t - Mc \quad (22)$$

The parameters of the prior are  $c_m = -c$  and  $c = N/2$  rewrite (21) as:

$$\pi_m^{(t)} = \frac{\hat{\Pi}_m - c/t}{1 - Mc/t} \quad (23)$$

Where

$$\hat{\Pi}_m = \frac{1}{t} \sum_{i=1}^t 0_m^{(i)}(\bar{x}^{(i)}) \quad (24)$$

the mentioned ML estimate and the bias from the prior are introduced through  $c/t$ .

The bias decreases for large data sets. If a small bias is acceptable, we can keep it constant by fixing  $c/t$  to  $c_T = c/T$  with some large  $T$ .

Assume that the parameters estimation is almost invariable when new sample  $x^{(t+1)}$  is added, and  $0_m^{(t+1)} \bar{x}^{(t)}$  can be approximated by  $0_m^{(t)} \bar{x}^{(t)}$ , that is the previous parameters estimation. Then, we can get the recursive update equation:

$$\hat{\pi}_m^{(t+1)} = \hat{\pi}_m^{(t)} + (1+t)^{-1} \left( \frac{0_m^{(t)}(\bar{x}^{(t+1)})}{1 - Mc_T} - \hat{\pi}_m^{(t)} \right) - (1+t)^{-1} \frac{c_T}{1 - Mc_T} \quad (25)$$

Here,  $T$  should be sufficiently large to make sure that  $Mc_T < 1$ . We start with initial  $\pi_m^{(0)} = 1/M$  and discard the  $m^{\text{th}}$  component when  $\pi_m^{t+1} < 0$ .

$$\hat{\mu}_m^{(t+1)} = \hat{\mu}_m^{(t)} + (t+1)^{-1} \left( \frac{0_m^{(t)}(\bar{x}^{(t+1)})}{\hat{\pi}_m^{(t)}} \right) (x^{(t+1)} - \hat{\mu}_m^{(t)}) \quad (26)$$

$$\hat{\sigma}_m^{(t+1)} = \hat{\sigma}_m^{(t)} + (t+1)^{-1} \frac{0_m^{(t)}(\bar{x}^{(t+1)})}{\hat{\pi}_m^{(t)}} ((\bar{x}^{(t+1)} - \hat{\mu}_m^{(t)}) (\bar{x}^{(t+1)} - \hat{\mu}_m^{(t)})^T - \hat{\sigma}_m^{(t)}) \quad (27)$$

## Tracking Multiple Objects

Once segmented, the objects are tracked in the subsequent frames. Tracking is a loop consisting of prediction of the positions from the previous frames, search for the best match, and update of the object representation. Traditionally, the tracking problem is formulated as sequential recursive estimation [23] having an estimate of the probability distribution of the target in the previous frame, the problem is to estimate the target distribution in the new frame using all available prior knowledge and the new information brought by the new frame.

### Bayesian sequential estimation

In the state space formalism, the state of the target is described by a state vector containing parameters such as target position, velocity, angle, or size. The target evolves according to the following discrete-time stochastic model called the dynamic model:

$$X_k = f_{t-1}(X_{t-1}, V_{t-1}) \quad (28)$$

Where  $f_{t-1}$  is a known function and  $(V_{t-1})$  is the process noise. Equivalently, target evolution can be characterized by the transition density  $p(X_t / X_{t-1})$ . The target state is related to the measurements via the observation model:

$$Z_t = h_{t-1}(X_t, W_t) \quad (29)$$

where  $h_{t-1}$  is a known function and  $W_{t-1}$  is the measurement noise. Again, the observation density  $p(Z_t / X_t)$  characterizes equivalently the relationship between the state vector and the measurement. Given the sequence of all available measurements  $Z_t = \{z_i, i = 1, \dots, t\}$ . We seek  $p(X_t / Z_t)$ . The Bayesian estimation allows to  $p(X_t / Z_t)$  find in a recursive manner. In terms of the posterior density at previous time

step  $p(X_{t-1} / Z_{t-1})$  the conceptual solution is found in two steps. Using the transition density one can perform the prediction step:

#### Prediction step:

$$p(X_t / Z_{t-1}) = \int p(X_t / X_{t-1}) dX_{t-1} \quad (30)$$

The prediction step makes use of the available knowledge of target evolution encoded in the dynamic model.

The update step uses the measurement  $Z_t$  available at time  $t$ , to update the predicted density:

#### Update step:

$$p(X_t | Z_t) \propto p(Z_t | X_t) p(X_t | Z_{t-1}) \quad (31)$$

These two steps, repeated for each frame, allow computing recursively the state density for each frame. In the case of multiple targets, each target is parameterized by a state  $X_{k,t}$ ,  $k=1, \dots, K$  which may differ in interpretation over the individual targets. The combined state is constructed as the concatenation of the individual target states,  $X_t = (X_{1,t}, \dots, X_{K,t})$  the individual targets are assumed to evolve independently according to Markovian dynamic models  $p_k(X_{k,t} | X_{k,t-1}), k=1, \dots, K$ . This implies that the dynamics for the combined state factorises over the individual targets as:

$$p(X_t | X_{t-1}) = \prod_{k=1}^K p_k(X_{k,t} | X_{k,t-1}) \quad (32)$$

### Problem statement

The humans which are of interest here, all have the same visual appearance and cannot be discriminated visually. Therefore, to track a particular human over time, the association task becomes crucial. We denote by a  $F_{1:t}$  possible hypothesis of the  $t$ -frame association problem  $F_{1:t} = \{f_1, f_2, \dots, f_t\}$  Where  $f_t$  is a parametric representation of frame  $t$

The optimum answer to the association problem is the maximum a posteriori estimation of  $F_{1:t}$

$$\hat{F}_{1:t} = \arg\{\max_{F_{1:t}} p(F_{1:t} | Z_{1:t})\} \quad (33)$$

Which is NP-hard, with a complexity exponential in  $t$ , so to find the optimal solution is essentially impossible.

Instead, to solve the association problem, different methods have been introduced to solve the problem sub optimally either by finding the most likely hypothesis from a limited hypothesis set over multiple frames:

$$\{F_{1:t}^h | h = 1, 2, \dots, \} \quad (34)$$

Thus finding the best member

$$\hat{F}_{1:t} = F_{1:t}^{\hat{h}} \text{ where } \hat{h} = \arg\{\max_h p(F_{1:t}^h | Z_{1:t})\} \quad (35)$$

as the solution such as the multi hypothesis tracking (MHT) algorithm, or a frame by frame solution based on single-frame associations over time  $[t-1, t]$  such as JPDA [14].

The single-frame association method is a feasible approach which has been widely used. However even in the single-frame case, virtually all approaches propose to find solutions over a reduced hypothesis space for frame  $t$ .

$$\{f_t^h | h = 1, 2, \dots, \} \quad (36)$$

And searching for the best member:

$$\hat{f} = f_t^{\hat{h}} \text{ where } \hat{h} = \arg\{\max_h p(f_t^h | Z_{1:t})\} \quad (37)$$

As the solution, the optimal single-frame solution is found if it is included among the hypotheses of that frame.

$$\arg\{\max_h p(f_t^h | Z_t)\} \in \{f_t^h\} \quad (38)$$

One of the widely used and well known single-frame algorithms to solve the multi target tracking problem in a reduced hypothesis space is JPDA.

### Joint probabilistic data association

To solve (33), JPDA has been widely applied to multi target tracking. There have been considerable efforts to generalize JPDA and overcome its shortcomings by using the extended Kalman filter (EKF) to linearize modest nonlinear systems [24]. JPDA assumes the number of targets to be known with the following constraints:

1. Each measurement originates from only target or clutter.
2. A measurement can be associated at most to one target.
3. At most one measurement can be associated to a target.

To make the association problem tractable, JPDA reduces the number of possible association hypotheses and keeps a reasonable subset of them as valid association hypotheses using a gating strategy which JPDA employs to validate the measurements and to generate a subset of association hypotheses, valid associations, based on validated measurements. Gating keeps the measurements which fall inside the validation gate of each target as valid measurements; hence the measurements that fall outside of the target's validation gate are not considered as association candidates and are thrown away. Moreover to manage the dimensionality and so the complexity of the problem, JPDA employs a recursive strategy and updates the filter distribution for each target.

Instead of maintaining the filtering distribution for the joint state  $p(X_t | Z_{1:t})$ , the JPDAF effectively combats the curse of dimensionality by recursively updating the marginal filtering distributions for each of the targets  $p(X_{k,t} | Z_{1:t}), k=1, \dots, K$  through the Bayesian sequential estimation recursions in (30) and (31). The prediction step proceeds independently for each target as

$$p_k(X_{k,t} | Z_{1:t-1}) = \int p_k(X_{k,t} | X_{k,t-1}) p_k(X_{k,t-1} | Z_{1:t-1}) dX_{k,t-1} \quad (39)$$

Due to the data association uncertainty the updating step cannot be performed independently for the individual targets. The JPDAF gets around this difficulty by performing a soft assignment of targets to measurements according to the corresponding posterior probabilities of these marginal associations. Moreover to manage the dimensionality and so the complexity of the problem, JPDA employs a recursive strategy and updates the filter distribution for each target. JPDA sequentially estimates the marginal distribution  $p_k(X_{k,t} | Z_{1:t})$ , given the measurements  $Z_{1:t}$ . From Bayes' rule

$$p_k(X_{k,t} | Z_{1:t}) = \frac{p_k(X_{k,t} | Z_{1:t-1}) \cdot p_k(Z_t | X_{k,t}, Z_{1:t-1})}{p(Z_t | Z_{1:t-1})} \quad (40)$$

The information of the measurements  $Z_{1:t}$  Has already been used to generate hypotheses  $X_{1:t-1}$  is implicitly embedded in  $X_t$  so that  $p_k(X_{k,t} | Z_{1:t})$ , thus all of the information of  $Z_{1:t}$  is implicitly embedded in  $X_t$  so that  $p_k(Z_t | X_{k,t}, Z_{1:t-1})$  can be simplified to  $p_k(Z_t | X_{k,t})$ . Moreover,  $p_k(Z_t | Z_{1:t-1})$  is a fixed constant so we can write:

$$p_k(X_{k,t} | Z_{1:t}) = c_k \cdot p_k(X_{k,t} | Z_{1:t-1}) p_k(Z_t | X_{k,t}) \quad (41)$$

Where  $c_k$  is normalization constant. The first term of (41),  $p_k(X_{k,t} | Z_{1:t-1})$ , is a prediction step. The second term of (41),  $p_k(Z_t | X_{k,t})$  is the likelihood of measurement  $Z_t$ , given hypothesis  $X_{k,t}$  and is computed in standard JPDA as:

$$p_k(Z_t | X_{k,t}) = \left[ \beta_k^0 + \sum_{m \in [1, M_k]} \beta_k^m \cdot p_k(Z_t^m | X_{k,t}) \right] \quad (42)$$

Where  $\beta_k^m$  is the marginal posterior probability of associating measurement  $m$  to target  $k$ . So that  $\beta_k^0$  is the probability of no measurement to be associated to target  $k$ .

### The Proposed Method

Tracking and surveillance applications require the segmentation of objects (regions of interest separate to the background) from the scene. Typically, systems divide the scene into two regions, foreground and background, where only the foreground contains events of interest, and the background is relatively unchanging over time. To achieve this separation of foreground and background, techniques such as Adaptive Gaussian Mixture Model are used.

Adaptive Gaussian mixtures are commonly chosen for their analytical representation and theoretical foundations. For these reasons, they have been employed in real-time surveillance systems for background subtraction and object tracking. But there are two open problems when Gaussian Mixture model are used for model multivariate data: the selection of the number of components and the initialization. The proposed method uses recursive equations [17] to constantly update the parameters of a Gaussian Mixture Model and to simultaneously select the appropriate number of components for each pixel. This GMM algorithm can automatically select the needed number of components per pixel that leads to the improvement in the segmentation results. Therefore, this method is highly memory and time efficient. Moreover, our method can effectively deal with various scenes such as the indoor scene, the outdoor scene, and the cluttered scene.

Motion correspondence is a fundamental problem in computer vision where it is referred to as the data association problem. The target tracking and surveillance community has extensively studied the motion correspondence problem and a number of statistical data association techniques have been developed. In this paper, JPDAF algorithm is used to solve the data association problem [24].

Most of the single-frame tracking methods, including JPDAF, reduce the number of possible association hypotheses to make the association problem tractable. Results show that our algorithm performs well for a surveillance application where we are attempting to locate a moving person within the flow images. The algorithm is capable of running in near real-time speed (15 fps) on a 3 GHz workstation. Experimental results are presented to show the effectiveness of the proposed method. Person tracking results show that our system is able to track people accurately.

The proposed multiple object tracking algorithm is as follows:

**(1) Algorithm initializing**

- a) In our experiments, the initial component value is  $M = 1$ .
- b) The first frame can be immediately used to initialize the mean  $\hat{\mu}_m^{(0)}$ .
- c) The initial covariance is  $\hat{\sigma}_m^{(0)} = 0.2$
- d) the initial mixing weights are set  $\hat{z}_m^{(0)} = 1/M$

**(2) Background model learning.** the mixture parameters of the background model can be learned by the recursive procedures with (25), (26) and (27).

- (a) The ownership can be simplified as  $o_m^{(i)}(\bar{x}^{i+1}) = \begin{cases} 1 & \text{match} \\ 0 & \text{otherwise} \end{cases}$
- (b) Define two rules for adaptively estimate the number of the mixture components

(1) Generation rule: if not 'match' any mixture components, generate a new component, with  $M = M + 1$ .

(2) Extinction rule: if  $\hat{z}_m^{(i+1)} < 0$ , discard this component  $m$  with  $M = M - 1$

**(3) Background subtraction.** We can extract the foreground object by background subtraction. Discussed in earlier sections.

**(4) Object tracking.** For tracking we use Bayesian estimation.

- (a) Compute the prediction step for each target (39).
- (b) Compute the likelihood of measurement  $p_k(Z_i | X_{k,t})$ .
- (c) Compute the update step for each target (41).

**Algorithm 1: Multiple Object Tracking Algorithms**

**Experimental Results**

In this section, we performed several experiments to prove the feasibility of the proposed tracking method.

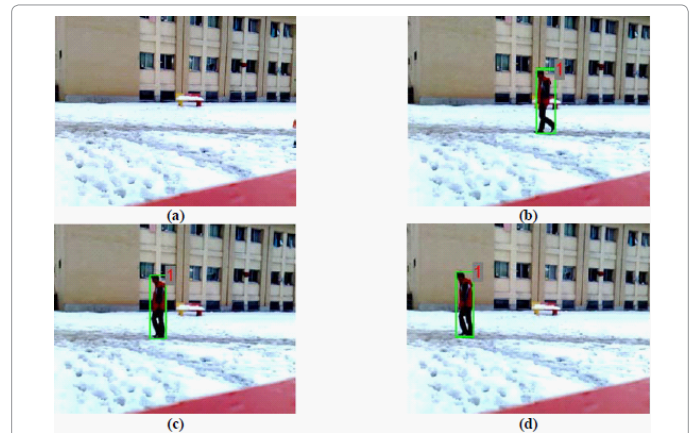
We used an entry-level video camera to capture test sequences. The experimental results are shown for various conditions such as a single moving object in outdoor environment with static background, a single moving object in outdoor environment with varying background and multiple outdoor objects. The results demonstrate a high performance of the proposed algorithm in the cluttered outdoor scene. The image size of all test sequences is 320\*240 pixels. These experiments are implemented on the MATLAB 7 platform with the Pentium4, 3 GHz and 1.0 GB memory PC. The experimental results show that the proposed method is efficient, effective and near real time.

**Tracking single object**

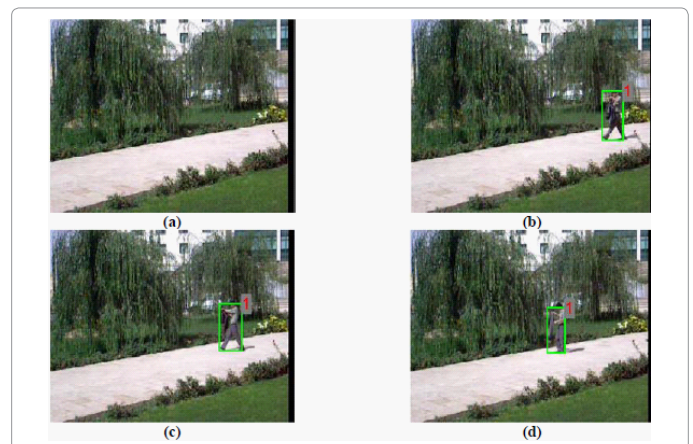
We did several experiments to simulate several conditions for moving objects such as single object in outdoor environment with static background and single object in outdoor environment with varying background. Figure 1 shows an example of the tracking for a single object in outdoor environment with static background. Figure 2 shows another example for a single object in outdoor environment with varying background.

**Tracking multiple objects**

In order to test the proposed tracking and identification method for multiple objects, we captured five video sequences in outdoor scenes with multiple moving objects crossing each other. In sequence



**Figure 1:** Tracking a single object in outdoor environment with static background: (a) the scene with no moving object; and (b)–(d) an object enters the surveillance system and is tracked by the system.



**Figure 2:** Tracking single object in outdoor environment with varied background: (a) the scene with no moving object; and (b)–(d) an object enters the surveillance system and is tracked by the system.

1, there are two persons in the test video stream, and they enter into the surveillance system. Figure 3 shows how these two persons move in the video sequence. The results of objects identification are shown in table 1. As can be seen in the table 1, the error in the tracking is less than 5% for all sequences. We compare our proposed method with the algorithm described in [25]. In this paper proposed a method for detecting and tracking multiple moving objects based on discrete wavelet transform and identifying the moving objects by their color and spatial information. From table 1, we notice that some objects are not correctly identified in some frames of the sequences. The wrong identification occurs in two types. First, it occurs when the moving object just enters or leaves the scene. When the moving object is just entering the surveillance system, the system will detect the moving object. However, the moving object is detected and tracked at the border of scene; the extracted features of the moving object in this case cannot represent the moving object very well. The main reason is that only partial of the features of the moving objects are extracted. The second kind of erroneous identification is called type 2. This kind of error occurs when the moving object is slowing down. In this situation, the inter-frame difference image of the object will become smaller and we will get smaller bounding box and less moving pixels. Therefore, the extracted features will lose its representative. This kind of erroneous identification arises from the detection error. In our proposed

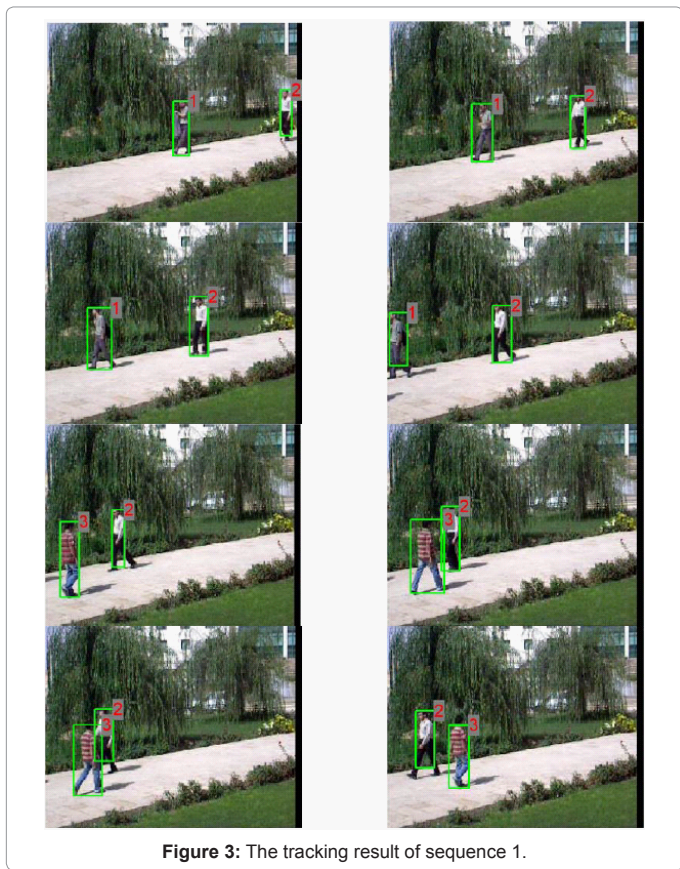


Figure 3: The tracking result of sequence 1.

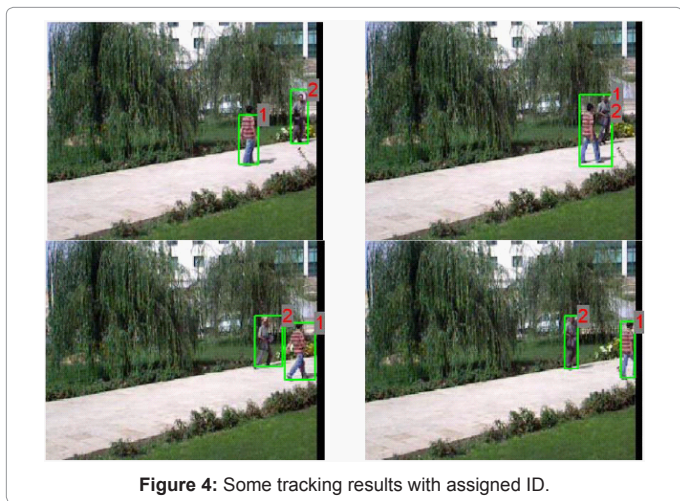


Figure 4: Some tracking results with assigned ID.

method wrong identification occurs just in type 2 therefore tracking performance are better. More over our algorithm is capable of running in near real-time speed (15 fps) on a 3 GHz workstation.

### Tracking performance

Some selected frames from the results are shown in figure 3. The algorithm is running at 15 frames per second which is near real time. The reported processing time is for 320\*240 pixel images and measured on a 3GHz PC. The ID of each human is shown on his beside in the picture. The proposed method works well with no identification error. The system is able to handle occlusion correctly and the object

Sequence Number	Method	Frames	Moving Objects	Objects Detected	Tracking Error (object)	Correct Tracking Rate (%)
1	Our Method	540	774	774	30	96.12
	Compared Method	540	774	774	49	93.66
2	Our Method	420	620	620	29	95.32
	Compared Method	420	620	620	40	93.54
3	Our Method	360	511	511	18	96.47
	Compared Method	360	511	511	35	93.15
4	Our Method	407	564	564	23	95.92
	Compared Method	407	564	564	32	94.32
5	Our Method	427	560	560	18	96.78
	Compared Method	427	560	560	36	93.57

Table 1: The result of multiple object tracking.



Figure 5: Selected frames of an object, entering the surveillance system.

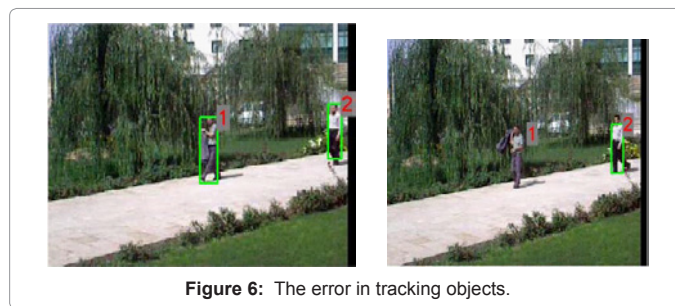


Figure 6: The error in tracking objects.

ID's remaining correct as shown in figure 4. The ID of each object is assigned automatically as it enters the surveillance system and then remains fixed. No error is observed in identifying multiple objects in the test sequences (Figure 5).

### Conclusion

A real-time method for detecting and tracking multiple moving objects based on adaptive Gaussian Mixture Model and corresponding the moving objects by JPDAF algorithm is proposed in this paper. A flexible GMM is used for foreground segmentation. In the proposed algorithm, the mixture component number is not a preset and fixed value and is optimally online estimated. Therefore, this method is

more time and memory efficient than the traditional GMM with the fixed component number. In tracking multiple moving objects, many applications have some problems when objects pass across each other. The JPDAF algorithm is used in this paper to solve this issue. No identification error has been observed on the test sequences. The algorithm is capable of running in near real-time speed (15 fps) on a 3 GHz workstation. Experimental results are presented to show the effectiveness of the proposed method.

## References

1. Tao D, Li X, Wu X., Maybank SJ (2006) Elapsed time in human gait recognition: a new approach. IEEE International Conference on Acoustics, Speech, and Signal Processing 2.
2. Aghili F, Namvar M (2006) Adaptive control of manipulators using uncalibrated joint-torque sensing. IEEE Trans Robotics 22: 854–860.
3. Huff M, Papenmeier F, Jahn G, Hesse FW (2010) Eye movements across viewpoint changes in multiple object tracking. Vis Cogn 18: 1368-1391.
4. Rife JH, Rock SM (2006) Design and validation of a robotic control law for observation of Deep Ocean jelly fish. IEEE Trans Robotics 22: 282–291.
5. Allili MS, Ziou D (2008) Object tracking in videos using adaptive mixture models and active contours. Neuro computing 71: 2001-2011.
6. Elgammal A, Harwood D, Davis L (2000) Non-Parametric model for background subtraction. Computer Science 1843: 751-767.
7. Shan C, Tan T, Wei Y (2007) Real-time hand tracking using a mean shift embedded particle filter. Pattern Recogn 40: 1958-1970.
8. Yang C, Duraiswami R, Davis L (2005) Efficient mean-shift tracking via a new similarity measure. Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition 1: 176-183.
9. Cheng FH, Chen YL (2006) Real time multiple objects tracking and identification based on discrete wavelet transform. Pattern Recogn 39: 1126 – 1139.
10. Song X, Chen J, Cheng H, Zhou X (2010) A Robust Moving Objects Detection Based on Improved Gaussian Mixture Mode. International Conference on Artificial Intelligence and Computational Intelligence 2: 54-58.
11. Lee D (2005) Effective Gaussian Mixture Learning for Video Background Subtraction. IEEE Trans. Pattern Anal. Machine Intell 27: 827-832.
12. Juan HX, Min ZJ, Bo-yang L (2010) Moving objects detection approach based on adaptive mixture Gaussian background model. Computer Applications.
13. Allili MS, Bouguila N, D. Ziou (2007) Finite General Gaussian Mixture Modelling and Application to Image and Video Foreground Segmentation. Electronic Imaging 183-190.
14. Stone LD, Barlow CA, Corwin TL (1999) Bayesian Multiple Target Tracking.
15. Rasmussen C, Hager GD (2001) Probabilistic data association methods for tracking complex visual objects. IEEE Transactions on Pattern Analysis and Machine Intelligence 23:560-576.
16. Hui M, Ng, BW-H (2006) Distributive JPDAF for Multi-Target Tracking in Wireless Sensor Networks. IEEE Region 10 Conference 1-4.
17. Zivkovic Z, van der Heijden F (2004) Recursive unsupervised learning of finite mixture models. IEEE Trans Pattern Anal Mach Intell 26: 651-656.
18. Figueiredo MAT, Jain AK (2002) Unsupervised learning of finite mixture models. IEEE Transactions on Pattern Analysis and Machine Intelligence 24: 381–396.
19. Cheng J, Yang J, Zhou Y, Cui Y (2006) Flexible background mixture models for foreground segmentation. Image Vision Comput 24: 473-482.
20. Lee EB, Nguyen TB, chung ST (2010) Improved Real-time Implementation of Adaptive Gaussian Mixture Model-based Object Detection Algorithm for Fixed-point DSP Processors. Measurement Science Instrumentation 1.
21. KadewTraKuPong P, Bowden R (2001) An Improved Adaptive Background Mixture Model for Real-time Tracking with Shadow Detection. 2nd European Workshop on Advanced Video-Based Surveillance Systems.
22. Allili MS, Ziou D, Bouguila N, Boutemedjet S (2010) Image and Video Segmentation by Combining Unsupervised Generalized Gaussian Mixture Modeling and Feature Selection. IEEE Trans on Circuits and Systems for Video Technology 20: 1373-1377.
23. Daronkolaie AG, Shiry S, Menhaj MB (2007) Multiple Target Tracking for Mobile Robots Using the JPDAF Algorithm. 19th IEEE International Conference on Tools with Artificial Intelligence 1: 137-145.
24. Kachouie NN, Fieguth PW (2007) Extended-Hungarian-JPDA: Exact Single-Frame Stem Cell Tracking. IEEE Transaction on Biomedical Engineering 54: 2011-2019.
25. Chen F, Chen YL (2006) Real time multiple objects tracking and identification based on discrete wavelet transform. Pattern Recogn 39: 1126-1139.