2008 IEEE Swarm Intelligence Symposium
St. Louis MO USA, September 21-23, 2008

Image-Based Tracking with Particle Swarms and Probabilistic Data Association

Edward Kao, Peter VanMaasdam, John Sheppard
The Johns Hopkins University
{ekao3, pvanmaa1, jsheppa2}@jhu.edu

Abstract - The process of automatically tracking people within video sequences is currently receiving a great deal of interest within the computer vision research community. In this paper we contrast the performance of the popular Mean-Shift algorithm's gradient descent based search strategy with a more advanced swarm intelligence technique. Towards this end, we propose the use of a Particle Swarm Optimization (PSO) algorithm to replace the gradient descent search, and also combine the swarm based search strategy with a Probabilistic Data Association Filter (PDAF) state estimator to perform the track association and maintenance stages. Performance is shown against a variety of data sets, ranging from easy to complex. The PSO-PDAF approach is seen to outperform both the Mean-Shift + Kalman filter and the single-measurement PSO + Kalman filter approach. However, PSO's robustness to low contrast and occlusion comes at the cost of higher computational requirements.

I. INTRODUCTION

Automated detection and tracking of people within video sequences is a topic that is currently receiving a great deal of interest within the computer vision research community. For many surveillance problems, such as border or other perimeter security applications, the region under video surveillance is simply too large for continuous human observation of the video streams. Thus, some form of automation is required to alert the human observers to the presence of objects (e.g. people or vehicles) within the surveillance region and to maintain track on these objects while the human decides what further action to take. Persistent tracking is an important objective under such an application. It allows for observations on the subject of interest over time that may be used to infer the intent and activities of the subject.

In order to perform persistent tracking in a real world application, the tracking algorithm must be robust to difficult operating conditions such as low contrast (between the objects and competing clutter) and occlusions of the objects being tracked. Another practical constraint is the need for long-range acquisition and tracking, which usually results in a reduced number of pixels on target when compared to most image-based tracking experiments presented by the research community.

At present, the most widely used method for image-based tracking is the Mean-Shift algorithm with a Kalman filter state estimator [4]. This algorithm has been very successful, but does have several limitations. The most prominent limitation is that, as a gradient descent based search strategy, the Mean-Shift algorithm is susceptible to converging to a local optimum instead of the global optimum, a problem that often occurs when objects with similar appearance surround the object being tracked. In addition, Mean-Shift tracking, even with the help of a Kalman filter prediction step, often fails to initialize the gradient descent search at a location close enough to the object in the new frame under challenging conditions such as occlusion and rapidly changing object velocity. This can lead to a loss of track because the gradient descent does not perform an adequate search to re-discover the object's location in the new frame.

In this paper, we present a swarm intelligence based tracking approach whereby Particle Swarm Optimization (PSO) replaces the Mean-Shift's gradient descent search strategy, while maintaining the use of the Mean-Shift feature in calculating the fitness function. In addition, we replace the traditional Kalman filter state estimator with a Probabilistic Data Association Filter (PDAF). We hypothesize that this approach will adequately address the Mean-Shift issues stated above while avoiding an exhaustive search in the region of interest. Comparisons are made using color imagery from a single camera against a variety of scenarios, ranging from simple to complex.

The remainder of the paper is organized as follows. Section II describes the basic formulation of an image-based tracking system. Section III describes the Mean-Shift feature and gradient descent search, as well as the alternative PSO search strategy. In Section IV, we propose a novel PSO-PDAF approach for image-based tracking. Section V consists of a description of the test sequences, with Section VI reporting the results of the experiments. Conclusions and future work are discussed in Section VII.

II. IMAGE-BASED TRACKING

The classical single camera image-based tracking problem can usually be described as accomplishing the following tasks. *Detection* is performed on the current image frame to generate Regions of Interest (ROIs). This detection can take the form of a simple double or triple window Constant False Alarm Rate (CFAR) detector, a matched filter, Moving Target Indication (MTI), etc [10]. For this study, the initial detection consists of a ground truth location at which to initiate tracking, which allows us to contrast algorithm performance free of any initial biases introduced by a detection algorithm. We assume that, in a practical implementation, a more autonomous detection component would replace the ground truth locator.

Detections are then assigned to existing tracks in the *Association* phase. This step, within a correlation-based tracker, is often performed simultaneously with the detection phase, where a small region about each predicted track location in the image frame is searched for a peak *correlation* with the template. Many methods exist to perform this

978-1-4244-2705-5/08/$25.00 ©2008 IEEE

Authorized licensed use limited to: MONTANA STATE UNIV BOZEMAN. Downloaded on January 9, 2009 at 11:30 from IEEE Xplore. Restrictions apply.

correlation, which simply seeks to minimize the cost between the current frame and a prior reference frame in a brute force search about the predicted target location. These methods are typically referred to as *template matchers*.

Unfortunately, template matching is computationally intensive and vulnerable to appearance changes including articulations and intensity variations. As a result, feature-based techniques have been proposed to address these issues. A classic example is the Kanade-Lucas-Tomasi (KLT) tracker [14] that extracts corner features that maximize "trackability" (i.e. saliency, or discriminating information). Optimal correlation of each feature is found using a gradient descent search which is much faster than the exhaustive search in template matching. The KLT tracker is robust to illumination changes but still vulnerable to object articulations such as arm and leg movements. Therefore, KLT tracking is found to be most successful when tracking primarily rigid objects. The Mean-Shift tracker [3] has been successful at tracking articulated objects such as humans, especially in color imagery, and a detailed description is provided in Section III.

Following the Detection and Association steps, *Track Maintenance* is performed. This step consists, first, of adding any new tracks for unassociated detections. All tracks that have been associated with a measurement are then processed in the *State Estimation* module. This module often consists of a Kalman filter (KF) [5]. The Kalman filter is an iterative algorithm that estimates the state of a process, given measurements, and predicts the state estimates one step forward in time, all in the presence of Gaussian noise. Equation (1)-(5) show the steps in a Kalman filter, where $\mathbf{K}$ is the Kalman Gain, $\hat{\mathbf{x}}$ is the state estimate, $\hat{\mathbf{P}}$ is the state covariance estimate, and $\mathbf{\Phi}$ is the state transition matrix governing propagation of the state forward in time from discrete time k to k+1. $\mathbf{H}$ is the measurement matrix that relates measurements to the state, $\mathbf{R}$ is the *a priori* measurement noise covariance matrix, $\mathbf{Q}$ is the *a priori* process noise covariance matrix, and $\mathbf{z}$ is the measurement vector. The + sign indicates the prediction of state and covariance estimates one step forward in time.

$$\mathbf{K}(k) = \mathbf{P}^{+}(k)\mathbf{H}^{\mathrm{T}}(k)\left[\mathbf{H}(k)\mathbf{P}^{+}(k)\mathbf{H}^{\mathrm{T}}(k) + \mathbf{R}(k)\right]^{-1} \qquad (1)$$

$$\hat{\mathbf{x}}(k) = \mathbf{x}^{+}(k) + \mathbf{K}(k)\left[\mathbf{z}(k) - \mathbf{H}(k)\mathbf{x}^{+}(k)\right] \qquad (2)$$

$$\hat{\mathbf{P}}(k) = \left[\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)\right]\mathbf{P}^{+}(k) \qquad (3)$$

$$\mathbf{x}^{+}(k+1) = \mathbf{\Phi}(k)\hat{\mathbf{x}}(k) \qquad (4)$$

$$\mathbf{P}^{+}(k+1) = \mathbf{\Phi}(k)\hat{\mathbf{P}}(k)\mathbf{\Phi}^{\mathrm{T}}(k) + \mathbf{Q}(k) \qquad (5)$$

Implicit in the derivation of the Kalman filter is that each track is updated with a single measurement. Much of the difficulty in an image-based tracking problem lies not within the state estimation (although that can be complex for targets and sensing platforms with rapidly changing velocities) but with the data association and search stage. If the association incorrectly assigns measurements to tracks, the state estimates will rapidly degrade, potentially causing a complete loss of track. In fact, both template and feature-based methods (including Mean-Shift) are susceptible to track degradation, as they typically provide a fairly noisy measurement due to the matching difficulties caused by target articulation and lighting changes.

One of the first methods proposed for tracking in the presence of noisy measurements is known as the Probabilistic Data Association Filter (PDAF) [1]. This method augments the general Kalman filter to allow multiple measurements to update each track, resulting in a less noisy state estimate. An "average" measurement is computed, where the association probabilities are computed based on the distance, in state space, to the predicted estimate. The authors of [13], while extending the PDAF to create the Joint Likelihood Filter, suggested a novel idea with respect to image-based tracking, which they called *measurement sampling*. Rather than seeking the best (brute-force) correlation peak of the template with respect to the scene, the *n* best correlation peaks are used to update the track state. The correlation score is computed at random locations about the predicted track location according to a normal distribution. This can be seen as a simple form of swarm intelligence where, at a single time step, the *n* best members of a population are taken as representative of the whole population. A more rigorous description of the PDAF and how it is used in our proposed PSO-PDAF tracker will be introduced in section IV.

Once each track's state estimate has been updated, further maintenance is performed, consisting of track deletion, incrementing of track status, and other steps based on the track's likelihood score. We do not consider multi-target tracking within this study, but such methods are required in an end-to-end system. Within such a framework, special logic is often required to determine when multiple tracks begin to converge, and take appropriate action in response. We discuss some of these actions within the future work section.

III SEARCH STRATEGIES FOR IMAGE-BASED TRACKING

As discussed in Section II, a key element of an image-based tracking system is the search strategy, wherein the best match to a reference frame (or feature set) is found in the current frame. This search typically takes the form of a gradient descent strategy, regardless of the nature of the reference frame (or feature set). Within this section, we describe the Mean-Shift algorithm, as well as the canonical PSO search strategy.

A. Mean-Shift Search

The Mean-Shift tracker [3] has been successful at tracking articulated objects, such as humans, especially in color imagery. Like the KLT tracker, Mean-Shift tracking performs a gradient descent search to maximize feature correlation. However, the feature used within Mean-Shift is a density

Authorized licensed use limited to: MONTANA STATE UNIV BOZEMAN. Downloaded on January 9, 2009 at 11:30 from IEEE Xplore. Restrictions apply.

estimation of the pixels within the track region. The Mean-Shift density estimation can be viewed as a weighted histogram where the center of the window has higher weight than the periphery. In the case of color imagery, the density is three-dimensional. The Mean-Shift density estimation at location **y** is given in Equation (6) below. The intensity at each pixel $\mathbf{x} = [row, col]^T$ in a window of size **h** determines the bin u to which it belongs within the (color or grayscale) histogram. This mapping is denoted as $b(\mathbf{x})$ in the equation. Each pixel is weighted by its distance to **y**, with C a normalization constant, $K(\|\mathbf{x}\|^2)$ is the Epanechnikov kernel that decays quadratically around zero, and δ is the Kronecker delta function.

$$\hat{p}_u(\mathbf{y}) = C \sum_{i=1}^{n} K\left(\left\| \frac{\mathbf{y} - \mathbf{x}_i}{\mathbf{h}} \right\|^2 \right) \delta[b(\mathbf{x}_i) - u] \tag{6}$$

$$K(x) = \begin{cases} \frac{2}{\pi^2}(1 - x) & x < 1 \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The Mean-Shift feature for each object is computed when a track is initiated (the learned target Mean-Shift feature) and can be updated periodically to account for appearance changes of the object. Due to the relatively short duration of the image sequences examined in this study, we do not update the feature following track initiation. This also allows us to provide an unbiased comparison between the search strategies.

In [3], tracking is performed by maximizing the similarity between the learned target Mean-Shift feature and the candidate Mean-Shift feature shifted through different locations in the current frame using a gradient descent search. Similarity between candidate and target Mean-Shift features is computed using the Bhattacharyya coefficient shown in Equation (8):

$$B(\mathbf{y}) = \sum_{u=1}^{m} \sqrt{\hat{p}_u(\mathbf{y}) \hat{q}_u} \tag{8}$$

where **y** is the current candidate location, $\hat{p}_u(\mathbf{y})$ the candidate Mean-Shift feature at location **y**, $\hat{q}_u$ the target Mean-Shift feature, and m is the total number of bins in the feature density estimate. Gradient descent is performed using the following formula:

$$\mathbf{y}_{\mathbf{t+1}} = \frac{\sum_{i=1}^{n} \mathbf{x_i} w_i g\left(\left\| \frac{\mathbf{y_t} - \mathbf{x_i}}{\mathbf{h}} \right\|^2 \right)}{\sum_{i=1}^{n} w_i g\left(\left\| \frac{\mathbf{y_t} - \mathbf{x_i}}{\mathbf{h}} \right\|^2 \right)} \tag{9}$$

where $g(x) = K'(x)$, the derivative of $K(x)$ with respect to x, and:

$$w_i = \sum_{u=1}^{m} \delta[b(\mathbf{x_i}) - u] \sqrt{\frac{\hat{q}_u}{\hat{q}_u(\mathbf{y_t})}} \tag{10}$$

The Mean-Shift feature, while powerful, has several limitations. First, and most importantly, the gradient descent search may be insufficient at locating the global optimum when the fitness landscape has multiple local optima. This will be demonstrated in our sample runs for tracking people in such environments. Second, in the attempt to abstract away from edge and template based techniques, all spatial information regarding the object has been lost. While this is desirable when the object exhibits articulations, it is not desirable when the object retains the same shape and simply changes intensity distributions (i.e. from shadows and other intensity changes within the scene). Finally, Mean-Shift lacks a "segmentation" step to separate object pixels from background pixels. In general if the initial track box is not tight around the object, track can be lost when the object moves from a dark to a light background. We do not address the final two limitations within this paper as they are inherent in the feature computation itself, not the search. Our results demonstrate that merely improving the search strategy is sufficient to maintain track even in difficult conditions. However, it is worth noting that the Mean-Shift feature and similarity metric was proposed so that a closed-form solution can be derived for a gradient descent search strategy (i.e. equation (9)). A stochastic search strategy like the PSO relaxes such constraint and allows for more flexibility in the feature representation.

Following development of the standard Mean-Shift techniques described above, Comaniciu extended the concept to incorporate a Kalman filter to predict the location of the track box on subsequent frames [4]. He also further extended the approach to include a de-weighting of background pixels as well as some additional methods to address the drawbacks discussed above. However, no statistical analysis has been shown on the performance, i.e. only four sequences demonstrating successful track are shown.

B. Particle Swarm Optimization (PSO) Search

PSO is a swarm intelligence algorithm for solving optimization problems, first developed by Kennedy and Eberhart [8]. The method consists of randomly initializing a set of particles within the search space, with each particle representing a (local) candidate solution to the optimization problem. The particles then "fly" through the search space, evaluating themselves and the population as a whole with respect to some objective function, while searching for the function's optimum.

For the general PSO formulation, each particle within the swarm carries an estimate of its location within the search space, as well as a velocity estimate within that space. These terms are defined, for time t and particle i, as the vectors $\mathbf{x}^i(t)$ and $\mathbf{v}^i(t)$ respectively. For image-based tracking, we use the particles to find the best measurements for a state estimator in the image space; therefore, the particle state is simply the row

Authorized licensed use limited to: MONTANA STATE UNIV BOZEMAN. Downloaded on January 9, 2009 at 11:30 from IEEE Xplore. Restrictions apply.

and column position of the object. The velocity vector and new position (of the particles, not of the state estimator) are adjusted towards the best solutions encountered so far, in a probabilistic fashion, by:

$$\mathbf{v}^{\mathbf{i}}(t) = \mathbf{v}^{\mathbf{i}}_{\mathbf{old}} + \mathbf{r_1} + \mathbf{r_2} \tag{11}$$

$$\mathbf{x}^{\mathbf{i}}(t) = \mathbf{x}^{\mathbf{i}}(t - 1) + \mathbf{v}^{\mathbf{i}}(t) \tag{12}$$

$$\mathbf{v}^{\mathbf{i}}_{\mathbf{old}} = w\mathbf{v}^{\mathbf{i}}(t - 1) \tag{13}$$

$$\mathbf{r}_1 = c_1 \times rand() \times [\,\mathbf{p}^{\mathbf{i}}_{\mathbf{best}} - \mathbf{x}^{\mathbf{i}}(t-1)\,] \tag{14}$$

$$\mathbf{r_2} = c_2 \times rand() \times [\,\mathbf{g}_{\mathbf{best}} - \mathbf{x}^{\mathbf{i}}(t-1)\,] \tag{15}$$

where c_1 and c_2 are parameters that weight velocity vector terms, w is a convergence decay constant, *rand*() is a uniform distribution random number generator in [0,1), $\mathbf{p}^{i}_{best}$ is the best (highest fitness) solution found so far for particle *i*, $\mathbf{g}_{best}$ is the global best solution found among all particles.

The constant w serves as an inertia factor, with large values favoring exploration of the space, and small values favoring exploitation. Equations (14) and (15) reflect an individual particle's residuals between its current state and the prior best states, locally and globally. The residuals are used to update the particle's velocity as shown in equation (11), driving its states towards the optima found so far. To help avoid premature convergence, the velocity estimates are often clamped at a user-defined maximum velocity $\mathbf{v}_{max}$.

This method of optimization has several properties that distinguish it from traditional optimization techniques. First, unlike traditional cognition models, the particles' beliefs are changed not only with respect to their own experiences but also with respect to their neighbors' experiences. The random initialization combined with the random walk within the update equations helps to search the optimization space rapidly, with the search becoming more focused after several iterations of the swarm.

IV PSO-PDAF IMAGE-BASED TRACKING

Our PSO algorithm for image-based tracking uses the same objective function as in Mean-Shift, merely adopting a different search strategy to seek the location in the new image that best matches the reference. At the beginning of an image sequence, the target Mean-Shift feature is extracted on a person of interest. In the subsequent frame, a swarm of particles is distributed within a search region around the predicted track box location, eventually converging on the location that maximizes the Bhattacharyya coefficient (equation (8)) between the target and candidate Mean-Shift features. The top n particles based on their $\mathbf{p}^{i}_{best}$ values are taken as measurements representing the swarm and used as the input to a PDAF filter that estimates the track's position and velocity. The filter then predicts the track's location in the next frame, with the predicted location being used to center the initial track box. The swarm is then re-distributed about this point, and the process of measurement, estimation, prediction, and swarm generation repeats until the end of the image sequence. This approach is summarized in Figure 1.

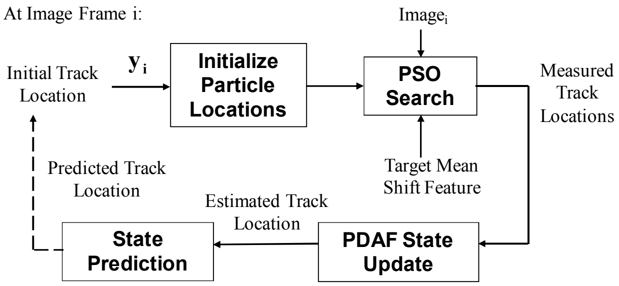

Figure 1: Proposed PSO-PDAF Tracking Paradigm.

For each image, the particle locations are initialized with a uniformly distributed spread around the initial track location. The extent of the spread determines the size of the search region. A larger extent should help recover a track after loss from occlusion and low contrast but will also take more iterations to converge. The particle memory is also reset since the previous $\mathbf{p}^{i}_{best}$ may be invalid due to object motion. The initialization step is very similar to the re-randomization method proposed by Hu and Eberhart in [6] for non-stationary optimization problems.

As discussed previously, the PDAF is an extension to the Kalman filter created to improve tracking performance in noisy conditions. It modifies the filter to account for an update by multiple measurements, weighting each by the distance to the predicted track location. An important concept in tracking is the concept of a 'track gate'. The gate is essentially an elliptical region around the predicted track location, and typically only measurements that fall within this association gate are considered for updating the track filter. Within our system, we simply choose the top n particles as the measurements within the gate. The equations governing the non-parametric PDAF can be seen below:

$$\hat{\mathbf{x}}(k) = \mathbf{x}^{+}(k) + \mathbf{K}(k)\tilde{\mathbf{z}}(k) \tag{16}$$

$$\tilde{\mathbf{z}}(k) = \sum_{\mathrm{i=1}}^{\mathrm{m(k)}} \boldsymbol{\beta}_{\mathbf{i}}(k)\tilde{\mathbf{z}}_{\mathbf{i}}(k) \tag{17}$$

$$\mathbf{S}(k) = \mathbf{H}(k)\mathbf{P}^{+}(k)\mathbf{H}^{\mathbf{T}}(k) + \mathbf{R}(k) \tag{18}$$

$$e_i = P_D \exp\left(-\frac{1}{2}\tilde{\mathbf{z}}_{\mathbf{i}}^{\mathbf{T}}(k)\mathbf{S}^{-1}(k)\tilde{\mathbf{z}}_{\mathbf{i}}(k)\right) \tag{19}$$

$$\boldsymbol{\beta}_{\mathbf{i}}(k) = \begin{cases} \dfrac{e_i}{b + \sum_{\mathrm{j=1}}^{m(k)} e_j} & i = 1,\ldots,m(k) \\[2ex] \dfrac{\mathrm{b}}{b + \sum_{\mathrm{j=1}}^{m(k)} e_j} & i = 0 \end{cases} \tag{20}$$

$$\tilde{\mathbf{z}}_{\mathbf{i}}(k) = \mathbf{z}_{\mathbf{i}}(k) - \mathbf{H}(k)\mathbf{x}^{+}(k) \tag{21}$$

Authorized licensed use limited to: MONTANA STATE UNIV BOZEMAN. Downloaded on January 9, 2009 at 11:30 from IEEE Xplore. Restrictions apply.

$$b = \frac{2(1 - P_D P_G)N}{\sqrt{\gamma}} \qquad (22)$$

where $\tilde{\mathbf{z}}_{\mathbf{i}}(k)$ is the difference between the predicted track location $\mathbf{x}^{+}(k)$ and candidate detection i at time step k, $\boldsymbol{\beta}_{\mathbf{i}}$ is the association probability for detection i, computed from the normalized distance e_i, (with $\boldsymbol{\beta}_{\mathbf{0}}$ representing the probability that none of the measurements are correct) and *m(k)* is the number of detections within the track gate. P_D and P_G are *a priori* probabilities of detection and track gate satisfaction (set to 0.9 and 0.99 respectively), N is the number of detections within the track's gate, where the gate is assumed to be spherical with a standard deviation of γ (set to 5 in our experiments), and $\mathbf{S}(k)$ is the track's estimated covariance. $\mathbf{H}(k)$, $\hat{\mathbf{P}}(k)$, $\mathbf{P}^{+}(k)$, $\mathbf{K}(k)$, and $\mathbf{R}(k)$ all take the same meaning as in the standard Kalman filter (see section II).

In addition to the modification of the measurement equation, the covariance update of the filter must be changed to account for each measurement's noise characteristics according to the equations:

$$\hat{\mathbf{P}}(k) = \boldsymbol{\beta}_{\mathbf{0}}(k)\mathbf{P}^{+}(k) + \left[1 - \boldsymbol{\beta}_{\mathbf{0}}(k)\mathbf{P}^{\mathbf{C}}(k) + \mathbf{P}^{\mathbf{I}}(k)\right] \qquad (23)$$

$$\mathbf{P}^{\mathbf{C}}(k) = \mathbf{P}^{+}(k) - \mathbf{K}(k)\mathbf{S}(k)\mathbf{K}^{\mathbf{T}}(k) \qquad (24)$$

$$\mathbf{P}^{\mathbf{I}}(k) = \mathbf{K}(k)\left[\sum_{\mathrm{i=1}}^{\mathrm{m(k)}} \boldsymbol{\beta}_{\mathbf{i}}(k)\tilde{\mathbf{z}}_{\mathbf{i}}(k)\tilde{\mathbf{z}}_{\mathbf{i}}^{\mathbf{T}}(k) - \tilde{\mathbf{z}}(k)\tilde{\mathbf{z}}^{\mathbf{T}}(k)\right]\mathbf{K}^{\mathbf{T}}(k) \qquad (25)$$

The PSO parameters, c_1, c_2, and w, take on the following values for this study, as recommended by the authors of [15]: $c_1 = c_2 = 1.49$ and $w = 0.72$. The maximum particle velocity, $\mathbf{v}_{max}$, is set at 50 pixels per iteration which we believe is adequate for the resolution of our image sequences. The region around the initial search location, $\mathbf{x}_{max}$, is set to one times the track box height and ten times the track box width, as the people being tracked move mostly horizontally within our imagery (note that for airborne downward looking imagery, the constraints should be equal in both dimensions). This search region should be large enough to recover lost tracks while preventing the particles from wondering too far afield. Our baseline PSO implementation uses ten particles, with a global neighborhood (i.e. no speciation). The search convergence criterion is velocity based, i.e. convergence is declared when the average velocity of the particles in the swarm drops below a threshold. We observe that when the average velocity of the particles drops below 0.5**track_box_width* per iteration, the $\mathbf{g}_{best}$ location has essentially converged; therefore, the velocity convergence threshold is set at this value. Figure 2 shows an example of the initial swarm population (with ten particles) and the population at convergence (the particles represented by dark Xs are the top three $\mathbf{p}^{i}_{best}$ selected as measurements for the PDAF). This is also an example of the PSO re-discovering the person coming out of an occlusion.

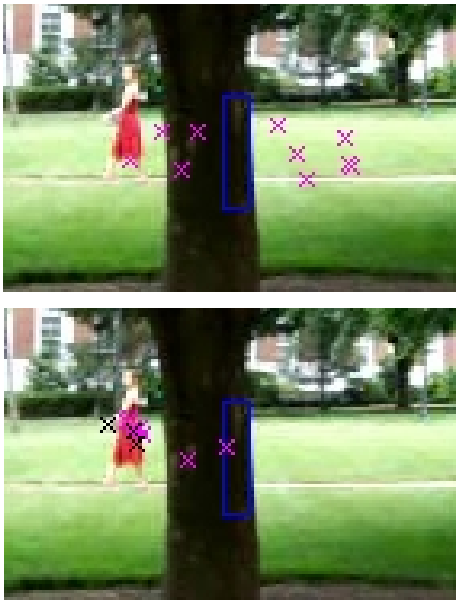

Figure 2: Initial (top) and converged (bottom) PSO populations.

PSO techniques have been examined recently within the context of the video surveillance paradigm; however, most of the current research has tended to focus solely on object detection and not on integrating the swarm results into a state estimation framework. For example, in [12], as the PSO fitness function is a classifier, the authors trained on Harr Wavelet and fuzzy edge-symmetry features to recognize the presence of humans within an image frame. The PSO is used solely to detect humans, and is not integrated into a unified tracking and state estimation paradigm. The authors extend this basic framework into a multi-view system in [11], however their solution does not perform state estimation—the PSO is still used solely as a detector within each view, with the tracks from each view being fused via a point correspondence method. In this case, the 'point' is simply the global best point estimate obtained by the swarm, and not the points from the individual particles.

Similarly, in [9], the authors implement a PSO to detect facial features and track them from frame to frame within the image stream. Tracking is performed solely by detecting the facial features on subsequent frames in a region close to where they were detected on previous frames. Again, no state estimation is performed.

In [16], the authors propose tracking objects in video sequences using PSO to perform image-based search and use a Bayes filter as an estimator. They employ a four-dimensional search space to account for changing track box size and use a histogram-based objective function. The histogram is essentially a gray-scale version of the color Mean-Shift feature described here. While their work shares some similarities with ours in the setup of the PSO search, significant differences exist in our methods to perform estimation and track maintenance. We utilize a standard Kalman filter, and directly contrast the performance with a more advanced PDAF state estimator as well as the Mean-

Authorized licensed use limited to: MONTANA STATE UNIV BOZEMAN. Downloaded on January 9, 2009 at 11:30 from IEEE Xplore. Restrictions apply.

Shift approach. They also did not provide any statistical performance evaluation, showing track success on only three image sequences.

V TEST DESCRIPTIONS

The test cases used for performance evaluation consist of three sets of image sequences, ranging in duration from 20 to 75 frames, at a frame rate of approximately 7.5 Hz. The input images are of 640 pixels by 480 pixels in dimension and the people being tracked vary in area size from 200 to 2000 pixels. The first set consists of eight sequences of unobstructed good contrast imagery used to establish the baseline "best-case" performance. The second is another set of eight sequences, with the people being tracked having low contrast (i.e. similar color distribution) either with the background or other close by people. The final set also contains eight sequences and represents challenging obscuration conditions, either from other people or objects in the scene. The people within the video sequences have had their centroids hand labeled for each frame, as ground truth is required to compute track performance. Figure 3 shows sample image frames from one sequence out of each category, with a zoomed in view of the person being tracked.

We first compare the PSO-based search method to Mean-Shift's gradient descent search, with both approaches using the same state estimation process (a Kalman filter with a single update). Our proposed hybrid PSO-PDAF approach is then contrasted to the single measurement PSO-KF approach, with the number of measurements used by the PDAF (three) being determined empirically to achieve good average track quality on the test sequences. Filter tunings (initial covariances, measurement noise, process noise) are the same for all experiments.

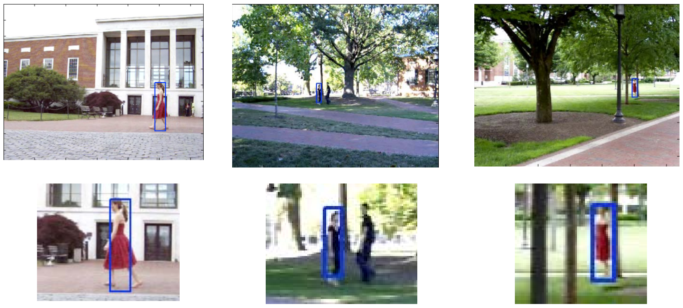

Figure 3: Example image frames from (left to right) Easy, Low Contrast, and Occlusion sets.

We use two metrics to determine the relative performance of the algorithms; the first is track quality. This metric represents the number of frames that the algorithm was determined to have a valid track divided by the total number of frames within the sequence. A valid track is defined as having the Root Mean Square (RMS) error between the filter's estimated target location and the ground truth location, normalized in both row and column by the window size so as to represent a unit track box size, below a threshold of $\sqrt{2}$ which corresponds to the distance from the centroid of the box to one of its corners. For the low contrast and occlusion tracks, evaluation begins after the low contrast or occlusion event has taken place. As the PSO is a stochastic algorithm, a total of 100 runs against each image sequence are used to establish statistics on performance. The second metric is the average number of Mean-Shift feature evaluations required per image frame. The Mean-Shift feature evaluation not only comprises the majority of the computation but also represents a single fitness evaluation in the search space, and is therefore representative of the algorithm's computational requirements.

VI RESULTS

Figure 4 illustrates the fitness landscape in a region about the true person of interest for two of the frames. The first frame comes from a sequence labeled "Easy 1," and the second comes from a sequence labeled "Low Contrast 1."

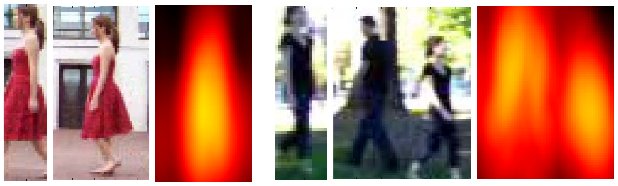

Figure 4: Example Fitness Landscapes.

The figure shows the target region from the initial frame in the sequence, a region about the person being tracked in a later frame, and the brute-force fitness (i.e., Bhattacharyya coefficient) landscape. The frames from the "Easy" sequence show a well-defined high-value peak at the expected location. For the frames from the "Low Contrast" sequence, there are two peaks in the region, one on the person to be tracked (which is the global peak), and another on the person nearby with a very similar Mean-Shift feature. This shows a case where a gradient descent approach could climb the wrong hill and easily be fooled into tracking the second person, demonstrating Mean-Shift's potential to find a local optimum instead of the global optimum in more complex scenarios. Similarly, Mean-Shift even with a Kalman filter almost always fails to recover from track occlusion after the peak has been lost for a few frames.

The number of particles to be used with the PSO-based approaches was determined empirically. Figure 5 shows the results of these experiments. For the PSO-PDAF approach, the performance begins to level off with 10 particles in the swarm. For the PSO-KF approach, performance levels off with 15 particles in the swarm. The figure also shows that the PSO-PDAF approach is superior to the PSO-KF approach, at least with respect to the average performance across all image sequences.

For each of the three image categories, Figure 6 shows a comparison of the PSO approaches (KF and PDAF) using 10 particles in the swarm with the Mean-Shift+KF approach in terms of average track quality and number of evaluations required.

Authorized licensed use limited to: MONTANA STATE UNIV BOZEMAN. Downloaded on January 9, 2009 at 11:30 from IEEE Xplore. Restrictions apply.

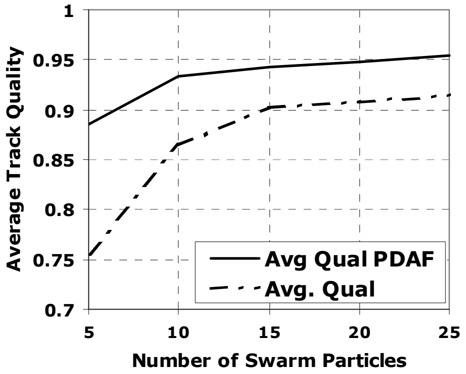

Figure 5: Track Quality Metric vs. Population Size.

Not shown for space reasons are the detailed results of a 2-sided difference of means *t*-test performed between each tracking approach. In summary, while there was no statistical difference between the algorithms for the Easy cases, both the PSO-KF and the PSO-PDAF were statistically better than the MS-KF using a 95% confidence threshold for all of the Low-Contrast and Occlusion sequences. In addition, the PSO-PDAF was seen to be better than the PSO-KF using the same confidence interval for almost 2/3 of the Low-Contrast and Occlusion sequences.

As expected, the performance of each algorithm is similar for the Easy scenarios. In these cases, Mean-Shift is seen to maintain an effective track while requiring very few fitness evaluations. Therefore, for scenarios consisting of high contrast non-occluded and widely spaced objects, the Mean-Shift remains the preferred tracking method.

However, the PSO approach (either KF or PDAF) clearly achieves superior performance against the low contrast and occlusion sequences. The PSO approach searches a much larger area than the Mean-Shift's gradient descent approach and thereby outperforms Mean-Shift in the runs where the local fitness landscape has multiple peaks or the optimal peak is far away from the initial track position. These situations usually arise when the background changes drastically, when there are confusers (people or background) with similar appearance (i.e. low contrast), or when there has been occlusion for many frames. This gain in the track quality comes at a significant cost in computation when using the PSO. A more feasible system can be constructed using a hybrid approach, where the PSO search strategy is only employed when the Mean-Shift fails to converge on a location with high enough Bhattacharyya coefficient. We have created an initial implementation of such a hybrid algorithm, and have seen performance similar to the PSO approach while requiring much fewer evaluations on average (~65 evaluations per frame instead of 200+). Even so, the PSO search alone at present is still near 'real time'. A single Mean-Shift fitness evaluation on a 2000 pixel window size is seen to require <1ms on a single 3GHz Intel Xeon processor using an un-optimized implementation. Runtime could be significantly reduced by using an optimized standard library such as the Intel Integrated Performance Primitives (IPP) [7].

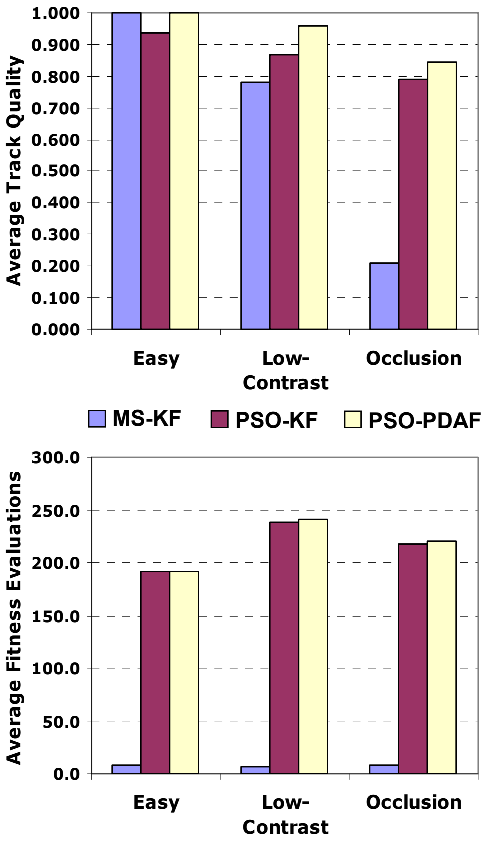

Figure 6: A comparison between the Mean-Shift and PSO search strategies with respect to track quality and computational requirements.

The PSO's ability to perform a wider search can also lead to stolen tracks by nearby confusers, especially when the person of interest is occluded. Often, a stolen track does not have the opportunity to recover after the occlusion since the track has strayed too far away. The occurrence of these "stolen" tracks is the reason why Mean-Shift outperforms PSO-KF in certain runs.

This "stolen track" issue is greatly mitigated by the use of the PDAF. The PDAF takes *n* measurements from the PSO population. This provides a greater diversity of measurements, each weighted by the probability that the given measurement applies to the current track. Therefore, measurements that are likely from confusers often receive much less weight in the evaluation. As a result, the track does not converge to the location of a confuser as rapidly and provides a greater chance to recover the intended person after emerging from the occlusion. Results clearly show that with respect to average track quality, the PDAF estimator outperforms the KF estimator, at essentially the same computational load.

VII CONCLUSION

In this paper, we have described and evaluated the use of a PSO approach to improve tracking performance over the Mean-Shift algorithm. We have shown that the PSO approach

Authorized licensed use limited to: MONTANA STATE UNIV BOZEMAN. Downloaded on January 9, 2009 at 11:30 from IEEE Xplore. Restrictions apply.

generally results in better performance with respect to the track quality, especially when combined with a PDAF state estimator instead of a Kalman filter. However, the PSO approach (with either estimator) requires an order of magnitude more fitness function evaluations than the standard Mean-Shift. Thus, the performance increase of the PSO methods, although still being near 'real-time', is somewhat offset by the computational resources required.

The reasons for the improvement in track quality, as discussed, stem mainly from two attributes of the PSO. The first is the ability to reach a better optimum within the fitness landscape, rather than the local optimum the gradient descent Mean-Shift finds. The second is the ability of the PSO to perform a rapid stochastic search to re-acquire the object after a loss of track for many frames often caused by occlusion. In this case, the Mean-Shift method has no hope of re-acquiring the track. Also, due to the weighted nature of the PDAF's measurement equation, the use of the top n $\mathbf{p}^i_{best}$ locations as measurements tends to mitigate the problem of tracks being stolen by other people and similar background clutter within the initial PSO search region.

Future work includes integrating the PSO-PDAF approach into a more complete multi-object tracking system. The swarm's search strategy will need to be modified to take into account the "joint" information when multiple objects are within the same search region. A natural extension would be to replace the PDAF with the Joint-PDAF [1] estimator. Also, we plan to further examine a hybrid system using the Bhattacharyya coefficient to detect loss of track (e.g. when the coefficient falls below a certain threshold) when using standard Mean-Shift, and correspondingly switch to the PSO method in order to attempt to re-establish track. Re-initialization of the target histogram periodically to adapt to target appearance changes (only if the Bhattacharyya coefficient is sufficiently large) will be investigated, as will methods of using the PSO to evolve on some of the kernel parameters (histogram bin size, target size, etc) rather than setting them at fixed values. Going away from the gradient descent search strategy allows for more flexible feature representation and similarity metrics which should be investigated. The impact of different methods of convergence schemes should also be examined. Possible schemes include spread of the particles or fitness change below a threshold. Finally, we would like to compare the performance between the Particle Filter as a search strategy in place of the PSO.

The initialization of the swarm population at each new image frame also deserves further investigation. Setting the spatial extent of the initial swarm population based on the error covariance of the filter may provide an adaptive search region depending on the confidence on the current estimated track location. Moreover, it may be beneficial to retain some of the $\mathbf{p}^i_{best}$ in the population since information from the search in the previous frame may not be entirely obsolete, as demonstrated by work on non-stationary optimization problems such as in [2]. Also, local neighborhoods in the PSO increase diversity in the measurements. Such diversity may be beneficial for the PDAF by providing multiple peaks in the fitness landscape.

VIII REFERENCES

[1] Y. Bar-Shalom and T. Fortman, *Tracking and Data Association*, Academic Press, 1988

[2] J. Carlisle, *Applying the Particle Swarm Optimizer to Non-stationary Environments*, PHD Thesis, Auburn University, Alabama, 2002.

[3] D. Comaniciu and V. Ramesh, "Mean-Shift and Optimal Prediction for Efficient Object Tracking", *Proceedings of the IEEE Conference on Image Processing*, vol. 3, pp. 70-73, Vancouver, Canada, 2000.

[4] D. Comaniciu, V. Ramesh, and P. Meer, "Kernel-Based Object Tracking", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 564-577, May 25th 2003.

[5] A. Gelb (ed.), *Applied Optimal Estimation*, Cambridge, MA: MIT Press, 1974.

[6] X. Hu, and R. Eberhart, "Adaptive Particle Swarm Optimization: Detection and Response to Dynamic Systems", *Proceedings of IEEE Congress on Evolutionary Computation*, vol. 2, pp. 1666-1670, 2002.

[7] Intel IPP: http://intel.com/software/products/ipp

[8] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence*, San Francisco, CA: Morgan Kaufman Publishers, 2001.

[9] C. Middendorf and M. Scheutz, "Real-Time Evolving Swarms for Rapid Pattern Detection and Tracking", *Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems (Artificial Life X)*, 2006.

[10] A. Mohan, C. Papageprgiou, and T. Poggio, "Example-Based Object Detection in Images by Components", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 349-361, April 2001.

[11] Y. Owechko, S. Medasani, P. Saisan, "Multi-View Classifier Swarms for Pedestrian Detection and Tracking", *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol.3, pp.18, 2005.

[12] Y. Owechko, S. Medasani, N. Srinivasa, "Classifier Swarms for Human Detection in Infrared Imagery", *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp.121, 2004.

[13] C. Rasmussen and G. Hager, "Probabilistic Data Association Methods for Tracking Complex Visual Objects", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no.6, pp. 560-576, June 2001.

[14] C. Tomasi and T. Kanade, "Detection and Tracking of Point Features", Technical Report CMU-CS-91-132, April 1991.

[15] F. van den Bergh and A. Engelbrecht, "A New Locally Convergent Particle Swarm Optimizer", *Proceedings of IEEE International Conference on Systems, Man, and Cybernetics*, pp. 96-101, 2002.

[16] Y. Zheng and Y Meng, "Adaptive Object Tracking using Particle Swarm Optimization", *Proceedings of the IEEE Conference On Computational Intelligence in Robotics and Automation*, pp. 43-48, Jacksonville, FL, 2007.

Authorized licensed use limited to: MONTANA STATE UNIV BOZEMAN. Downloaded on January 9, 2009 at 11:30 from IEEE Xplore. Restrictions apply.