Real time Object Tracking Using Evolutionary Computation Approaches: State Space Exploration using Static & Dynamic Particle Like Sensors

Intekhab Alam
Department of Electronics, Electrical & Computer Engineering
University of Birmingham
Edgbaston, Birmingham, B152TT, U.K
Intekhab.Alam.GB@ieee.org

ABSTRACT: *A Novel methodology has been proposed in this paper that enabled us to track objects of interest in video sequences at a much higher rates of 25-30 frames per second. This is particularly useful in applications where a real time intervention is of paramount importance for e.g. in safety critical applications and in video surveillance environment. This simplistic but very robust tracking algorithm is based on static and dynamic particle like sensors that also march along the object in video frames. The direction of propagation of these sensors is associated with different components including previous motion history and the maneuverability of the object in question and the track characteristics. A group of particles each of a pixel size bunch together to form a super particle and its mean position in the frame is used to direct other subordinate particles to explore the search space further to determine the exact boundary profiles of the object of interest. The super particle position is similar to groups best velocity profile in Particle Swarm Optimization. The main difference in this approach is that the birth of particles takes place in a circular like search space around the groups best position. This evolutionary approach inspired by biological life forms like flock of birds and a bee hive outperforms the standard particle filter based approach where the state dynamics are predetermined and are usually influenced by Newtonian Physics. Our approach dramatically reduces the number of iteration required in each frame and needs even less than 30 particles (300-500 in standard particle filter) to effectively track object of interest in a frame of medium resolution. This approach also recovered the object effectively in real time if it was partially or fully occluded for a significant number of frames and when it underwent large maneuverability which original mean shift algorithm had been unable to cope with.*

Keywords: Particle Filter Tracking, Particle Swarm Optimization, Histogram Back Projection, Kalman Filter, Particle Like Sensors

Received: 2 June 2013, Revised 5 July 2013, Accepted 8 July 2013

1. Introduction

The recent decade has been revolutionizing the domestic consumer market mainly due to availability of economical SoCs (system on chip) and embedded microcontrollers that need minimal components to interface with the analog sensors quantifying the outside world. By the end of this decade we could assume (safely) that vision based algorithms/application would form an integral part of domestic consumer market/products from automatic lawn mowers to security and remote surveillance of the domestic premises [1] [2]. There are some similar products available that provide e.g. driving aid to the road users, smart TV that eliminates the need of using a

remote control device and resource allocation systems based on detecting walking posture and gesture recognizing with unique cornea-eye profiles and eye movements [3] [4]. The real time capabilities of vision algorithms also had been improving over time but still there are some fundamental weaknesses due to which big blue chip companies are avoiding introducing automation based on vision algorithms and somewhat is still in a pre-mature stage of evolution and development.

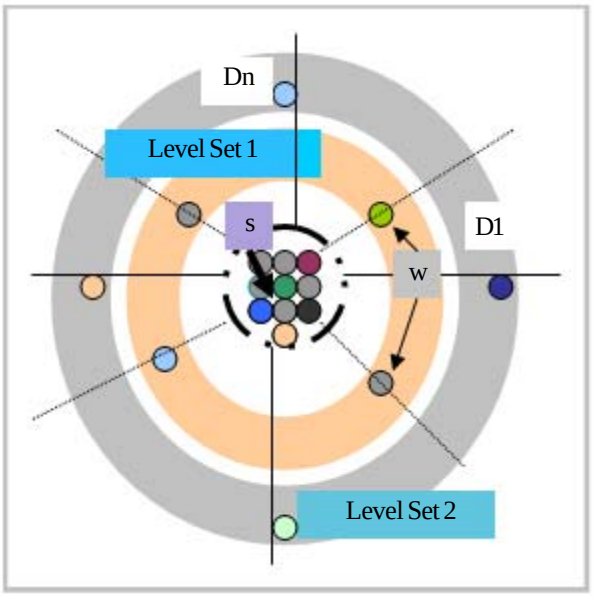

Figure 1. Particle initial positions and corresponding movement vectors are shown in the above diagram. The group of fast moving particles form a bunch in the middle to act as a super particle '*S*'. The worker particles have relatively slower velocity profiles and ideal to explore the directed search space

Traditionally in the past three to four decades computer vision algorithms had been strongly influenced by analytical mathematics and stochastic techniques including Kalman & Particle filter [5] [6] played a vital role in the relative research literature. However, the basic inherent flaw still remains and is related to the inability of Kalman filter (and to some extent standard particle filter as well) to address situations when the object does not follow a pattern of motion expected from the linear Gaussian like distribution of the state space and the state transition models employed in variations of particle filters.

In this case the operational basin or the search space during the prediction step expands multiple times to process and the tracking window starts drifting away from the true solution and sometimes gets stuck at a position of a local minima. The update step then is left with a gigantic data to process and re-sampling stage takes the bulk of processing resources. This as a consequence looses the track of the object in subsequent frames. Also, as the dynamics is usually based on assumed transition models the effective number of particles n_E is typically very low (0.3-0.7). An example of state transition matrix has been shown in the equation 1 below as adopted from [9].

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ x_t \\ y_t \\ s_{t+1} \end{bmatrix} = \begin{bmatrix} 2 & 0 & -1 & 0 & 0 \\ 0 & 2 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ y_t \\ x_{t-1} \\ y_{t-1} \\ s_t \end{bmatrix} + w_t \qquad (1)$$

During the course of these experiments tracking employing the state transition models as in equation 1 (for a frame of a medium-small size resolution) took an average of between 4.5 and 9 seconds to establish a successful track of the object in question once it has drifted away due to added noise and occlusion. This eventually ended up carrying out a blind search procedure which undermines the automation process in the first place. Alternatively, the same object when using suggested methodology in Figure 1 kept correct track at the frame rate of 15-20 per second with an average discrepancy (RMSE) of 0.48-0.92. The RMSE

(error in these experiments) is an indicator of how much in reality the anticipated cluster centre deviated from its manually calculated ground truth values. The cluster centre is the mean position of the detected features in RGB color space. The primary attribute we selected in our experiments was the simple normalized RGB color distribution of the object and a fast mean shift clustering method based on frame differencing alone and that climbs to the peak of this density in a iterative, robust but quicker fashion based on information integration and summing up evidence and measurements rather than depending on prediction theory as in most stochastic filters. The color distribution modeled by discrete probability density function is invariant to rotation and scale changes and by utilizing simple bin tracking the shift in color distribution due to change in lightening conditions could also be addressed. Normally, there is a strong pattern how the whole color histogram shifts and mostly follows a linear scale.

This work primarily focuses on devising an evolutionary real time tracking and is also a continuation of the original biologically inspired algorithms including [7] [8] that had been gaining popularity in addressing computer vision problems in most recent decade. One of the higher level tasks in vision algorithms whether it is activity recognition or manufacturing process is to track an object of interest in sequence of images either in a live feed or using a pre-recorded format [9]. This therefore does need only minimal intervention of human operator and is a cost effective technique for e.g. all the security cameras could be controlled wirelessly from a single embedded platform when the subject moves out of the scope of any particular camera.

To design an effective tracking algorithm, designers need to be aware about the variations the contextual factor could cause and play towards the success rate and there is a dire need to understand the importance of distinct motion models being employed either through active learning in the scene conditions or a pre assumed model with a safe margin that could address the non linear elements of motion. In simple words an object tracking algorithm that had been successfully implemented in one environment e.g. in automatic premises surveillance, may fail drastically in traffic monitoring applications and therefore in our approach prime focus had been on fast algorithms for evidence gathering before the state is updated.

In real time applications especially in safety critical applications we are encountered with an added responsibility to explore a trade-off between complexity and robustness of the tracking algorithm.

To address these issues our tracking algorithm is mainly inspired by evolutionary and naturally occurring phenomenon including how birds fly across a search space in a random fashion to find food but with effective communication among them to call over their colleagues if they succeed in locating it and also in bee-hives where the job are split among different classes of insects from identification of a potential danger and its defense to birth of newer members of the swarm.

Similar idea has been portrayed in Figure 1, where in the middle there is a swarm of fast moving particles that also formulate a cluster like structure and we called this a super particle. A super particle based swarm like structure could develop (through newer births) from a single super pixel size element that satisfactorily obeyed all the rules described in a predefined bag of descriptors. Although, many authors also use textual format to describe it but we have used simple byte size information of Boolean logic to express the qualities needed to become a super particle. Each super particle has at its disposal a number of subordinate particles as well that could be ordered to fly across in certain random directions to collect information as shown in Figure 1. The uniqueness of our algorithm is that the particle speed and direction (velocity) is dynamically controlled depending on the level set values (LSV). The LSV are similar to the level set values of a function but here the literal meaning is that how far it is away from the true solution. The different color circles also specify the information regarding the scale of the object to be tracked. With this approach we have successfully managed to dynamically adjust the scale of the tracking window and therefore by reducing the number of loops required in terms of programming constructs and real time objective were also achieved. The attributes associated with workers as have been represented by subscript 'w' and population size ranging from d_1 to d_n define how fast the solution could be propagated in time between different frames and when the subject moves out of the scope of one camera. Beside dynamic particles we also seeded the refined search space using static particle sensors that were found particularly useful in locating the object once it undergoes partial to full occlusion or during points of extreme maneuverability.

2. Technical Details

The search space after projection of 3d world onto the camera film is essentially a 2d plane represented by $\Omega_{(y,\,x)}$ where y, x coordinates specify the resolution of the image. In tracking applications we usually do not have the luxury to blind search the whole space due to real time constraints and limited computing resources. However, if such a facility is somehow been made

available then this could undermine the utilization of tracking algorithms in the first place. In our view tracking is regarded as a resource efficient method that engineers the most efficient methodology to locate a particular object in subsequent frames incorporating any changes in the profile of the object that is being tracked.

The prime motivation for this task is to devise efficient real time tracking algorithms aswell that are implementable on embedded platforms including ARM9 and PIC range of budget microcontrollers and are also portable with respect to the contextual utilization and also with minimum amount of changes for a variety of scene conditions. By distributed sensors we mean to spread the relevant search space with particles of distinct objective functions. The basic idea is to provide a good level of explorability in a specific sub space once we have some prediction about the localization of the object of interest. In terms of computer time the positional shift produced by a sub pixel level velocity vector is exactly the same as the maximum permissable velocity component that allows a particle to travel from origin to the extremium of any captured image $\Omega_{(y_{max}, x_{max})}$.

This is significantly different from the real physical world where the amount of work done in shifting a sensor from location a to *b* is distance dependent and cost is directly proportional to the travelled path. Therefore, this means that we have the options of selecting a higher velocity component when there is little chance that it will cross the anticipated boundaries or produce the shift in smaller sub steps. In the later case it will consume more computing resources but could be more precise compared to the first approach with a chance of rapid oscillations around the region before settling down to the true solution. In particle swarm optimization such balance is provided by the tuning variables as shown in the equation 2.

$$v_i = v_i + \Phi_1 (P_{bi} - s_i) + \Phi_2 (gB - s_i) \tag{2}$$

The velocity vector of any swarm particle is updated by a factor consisting of linear combination of personal best and global best velocities. If on the other hand the components of motions are not selected to be in the best interest of a real time constraint it could become more resource hungry without providing the desirable tracking results. In our experiments the values of tuning variable could range anything from 0 to 1 and hence with switching characteristics that turns a swarm particle into an ordinary particle filter when it is safe to do so. Therefore any keen reader would observe that the best components of motion satisfying real time constraints are not actually the swarm like characteristics but rather an ability to dynamically alter the fundamental nature of motion associated with a specific particle in motion. The swarm like feature controls the exploration of search space as in Figure 2 only when the super particles fail to predict the solution. In all other cases the best solution is to follow the global best position around specific level sets as in figure 1.

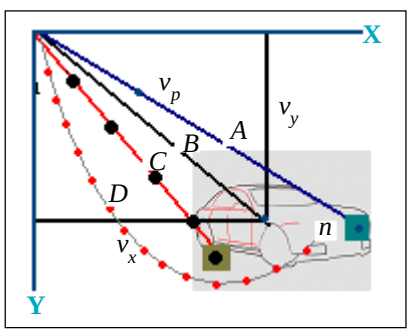

Figure 2. Four distinctive situations have been shown in the diagram where a dynamic particle sensor follows the path of a moving vehicle in a *xy* plane

In the above figure 2 a tracking example for a vehicle is shown that is moving away from the origin and portrays a situation when it has suddenly accelerated and the shift has occurred in between two successive frames. The origin could be the previously successful tracked position. We also assume that the tracking window lagged behind and no significant portion of the vehicle is currently falling inside it. Standard mean shift algorithm does not have the capacity to address this situation as had been identified [11]. The red dots exemplifies a scenario when particle swarm optimization technique would try to find a potential solution compared to when particles are only influenced by the linear combination of *x* and *y* components of a call to a prospected solution either by propagation of a global best using a selective histogram back projection technique or through a systematic information integration as had been explained in [18]. The interesting fact worth mentioning is that it takes same processing power if particle falls inside the region of interest in a single step (B), or consumes 15 times this cost when it iteratively traveled to the region that is being tracked. The linear combination of these velocity components identified in

equation 2 as Φ_1 and Φ_2 (and by subscripts v_y and v_x in the above figure 2) and are also termed as tuning parameters determine a unique path to the object of interest as shown by vectors *A* to *D*. Here in terms of processing cost solution *B* is the most effective one as it has traveled to the region consuming only 1 computer step compared to *A* and *C* where the costs are 2*x* and 4*x* respectively. The path propagated in situation *A* also shows when the dynamic state transition models are used and therefore traveled to the destination saving ½ of the computer time as needed by solution *C*. Each of the smaller squared regions inside the object of interest shows a region where a successful similarity score was achieved indicating the presence of an object like feature. The object like features are the unique attributes associated with particular objects which we would use later on to build a similarity score. In biological and human vision these steps are much trivial and almost happen immediately for e.g. we can pick up a familiar face in a crowd in a fraction of time. In the case of color, scoring could be based on the unique bin numbers associated with the complete discrete probability distribution or the Bhattacharyya measurement. Another profile building measure could be the total variation flow which is unique for a specific region within its boundaries. The most important concept in this regard however and the focal point of the concept the author trying to propose is that adapting a search technique using reciprocating kind of particle movements with unique velocity components (dynamic in nature) is far superior in terms of processing needs compared to the assumed dynamics used in standard stochastic techniques like standard particle and Kalman filtering.

The core concept is to improve effective number of particles in standard particle filters based approach and to address the degeneracy issue that could help in achieving our real time objective. Contrary to the arbitrary state transition model (based on Newton laws of motion) in standard particle filters, this works by explicitly definition of the associated velocities. By using a dynamic population level of particles we can also control the processing complexity. If a contour shape is required at a specific frame then simply by the addition/birth of more particles we can even find a denser contour line [12].

The reciprocating process is controlled by a super particle and therefore when a group of such particles is formulated then it is more conceivable to determine the most desirable direction to propagate and to collect the required information from and to adjust the scale of the tracking window thus eliminating any undesirable calculations in the search space and also helps avoiding the solution being trapped into a local minima. The reciprocation process and automatic scale adjustment is shown in a simple diagram below. Such scaling method also outperforms the Camshift tracking algorithm.

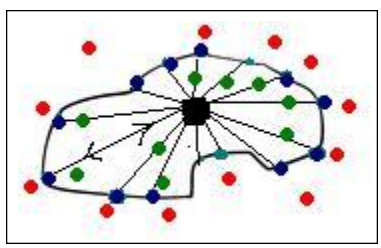

Figure 3. Scale adjustment by reciprocating measurements taken from the super cluster mean position using subordinate worker particles. Green colored dots lie within the object region and red outside. Boundaries are represented by blue dots

In the previous figure 3 the super particles and their mean position is indicated by a dark black region. The subordinate particles reciprocate from their position in the directions shown by the lines. The movement could also be understood in the context of a spring like motion. The red dots indicate the stretched positions where the measurements do not qualify to be grouped as belonging to the object region whereas the green dots are compressed measurements. This kind of motion of a particle would eventually settle down to the correct position when the calculated error falls below a specified threshold. When a higher level of accuracy is desired a linear regression method similar to [19] was used to determine the precise boundary locations. The complexity is directly controlled by using lesser rays and by dynamic velocity vectors. It becomes most helpful if two neighboring rays use $1.5x - 3x$ multiples of unique velocity vector to save computation time as also been highlighted in figure 2 using scenarios *A* to *C*.

In the above PETS sequence tracking at a frame rate of 25*fps* was achieved with automatic scale adjustment and by successive decreasing the number of particles from 30 to only 5 the object (pedestrian in this case) was successfully tracked with an accuracy rate of 0.95-0.80. The super particles oscillate from a core position as projected by selective histogram back projection

algorithm until a final mean position is decided. The subordinate particles were then released to collect the information [13] about the boundary locations to adjust the tracking window size $\begin{bmatrix} w_y \\ w_x \end{bmatrix}$.

Figure 4. The tracking window scale is automatically adjusted by calculating the steady state positions of the particles. To accelerate the process the particle velocities are dynamically varied around the mean calculated by the intersection of histogram back projection and motion estimates

2.1 Kernel weighted selective histogram back projection

We have constructed discrete probability distribution using m-bin histogram method to represent the object model. The fundamental difference in standard mean shift and histogram back projection methodologies is the way these bins are weighted in order to give more preferences to the region lying at the center of the object compared to the ones around the boundaries. This is shown in the expression below. In Figure 3 we get *C* after performing summation in 6, *B* is the associated bin number.

$$\hat{q_u} = C \sum_{i=1}^{n} k(\| x_i \|^2)\, \delta[b(x_i^*) - u] \tag{3}$$

$$\hat{p_u} = \sum_{i=1}^{n} \delta[b(x_i^*) - u] \tag{4}$$

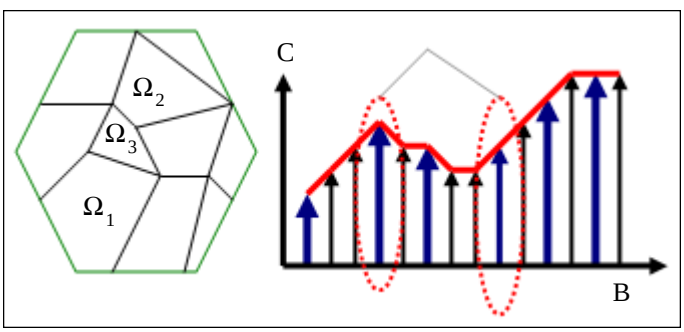

Figure 5. The concept of kernel weighting has been shown in the above figure. The part on the left shows a possible portioning of the object region (green) into cells based on the group of pixels belonging to some specific bin. Only the dark blue colored bins (just two selected by an oval structure around them) are used in selective HBP algorithm to reduce the processing costs

Here the function $b(x_i)$ plays a pivotal role. It assigns the data point under observation to one specific bin and hence formulates a discrete representation of the object probability density function. It is evident after comparison of the equation 5 and 6 that bins are weighted according to some associated weights in kernel weighted histograms as in standard mean shift. This is the most processing expensive operation in standard mean shift. By carefully choosing only some selective bins we can reduce the processing cost by a significant margin without compromising the end result. In this approach we have used a rule based approach to incorporate the kernel weighting process depending on the bin numbers. A two dimensional lookup table is constructed that allocates weights depending on the bin number b. In figure 3 the region marked as Ω_3 lies in the center of the kernel and therefore relative bin had been allocated highest weight of 0.6. The sum of all kernel weights is normalized to 1.

However in selective HBP tracking care must be taken to include bins that are evenly selected by the inclusion of border regions as well [12].

It is also worth mentioning that real time tracking of object has been made possible by adapting a look-up table kind of approach and by utilizing alone as an objective function in some selective sequences. In the tracking scenario shown in the figure below the man wearing blue shirt is tracked successfully by using 12-15 particles only with an objective to find the correct values from a look up table. However at points of greater maneuverability tracking using complete object model becomes mandatory. In this approach the mean position of the particles is used as the centroid of the tracking window '*w*'. The predicted position of this window was initially calculated from its associated velocity profile observed during the earlier frames. If 10 random measurements associated with each particle fail to find relevant bins then the objective function was altered to find the positions where significant motions were present using frame differencing. Thus the window is shifted to the new position and update process starts once again. The tracking window is only superimposed on the frame after the update process completes.

Figure 6. Four frames have been shown above from the sequence taken from PETS dataset. Here only binary look-up table is used to track the man wearing sky blue shirt. Random kernel search is successfully utilized to track at 15-25 fps

In selective bin HBP tracking Bhattacharyya similarity measure [15] was used to compare target model and target candidate region which then iteratively climbs to the peak of the density. This is done by translating the tracking window using the newly calculated velocity vector.

2.2 State transition with distributed measurements

In our experiments we have selected the following format of state vector.

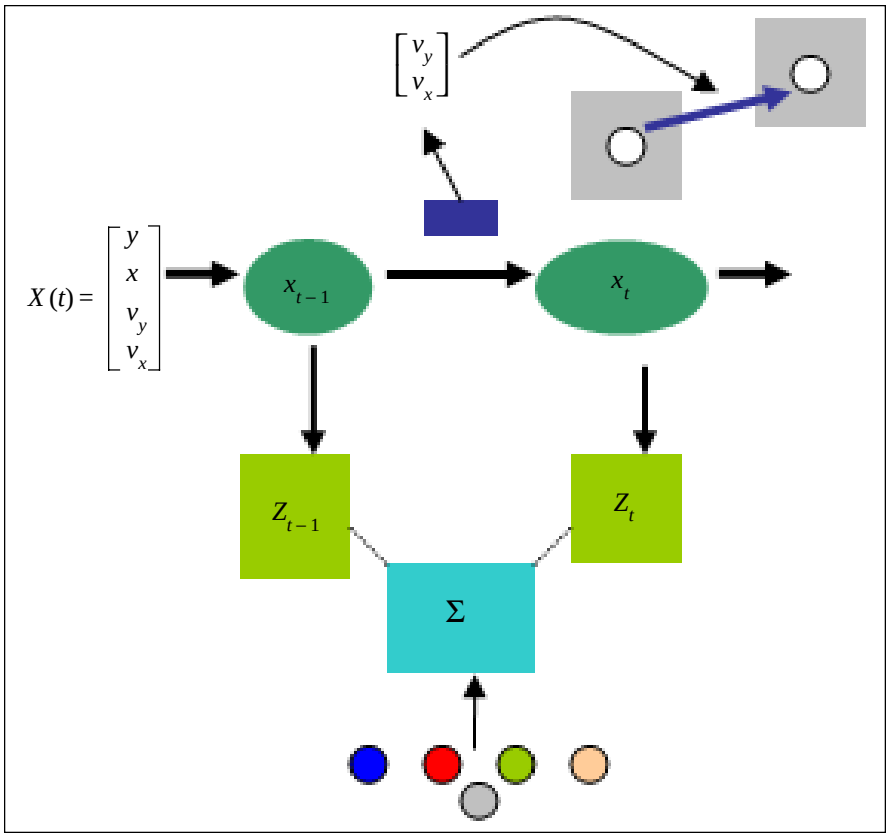

Figure 7. State transition model is calculated by a selective HBP as shown by a small blue rectangle between two states. The state measurements are based on a technique of combining evidence through particles with different objective functions spread in the relevant search space around the centroid

The Markov assumption is also in place regarding state transition and observation model. States form a hidden Markov model [16] and are only partially visible. The prediction and update steps therefore could be written as

$$P(x_t \mid z_{t-1}) = \int P(x_t \mid x_{t-1}) \,.\, P(x_{t-1} \mid z_{t-1})\, dx_{t-1} \tag{5}$$

$$P(x_t \mid z_t) = \alpha P(z_t \mid x_t) \,.\, P(x_t \mid z_{t-1}) \tag{6}$$

The first term on the right hand side of the equation and inside the integral is the state transition by HBP algorithm and as in above figure 5. In equation 6, $P(x_t \mid z_t)$ is formulated by combining the evidence systematically using an independent voting system. As in standard conflict resolution strategies all particles cast votes once any particle raises a binary flag $f = 1$ using its assigned objective function indicating the presence of the object region. By majority voting the conflict becomes resolved. Also with the effective inclusion of right proportions of particles types we can provide better opportunity to explore the relevant search space.

2.3 Particle like sensors with multiple objective functions

In our approach we have utilized five different objective functions and spread the task randomly among the whole population of particles in the search space formulated in the prediction step as pointed out in earlier section. One of the most important one that provides a expanding or a shrinking motion is the log of the probabilities of object and back ground regions log ($P1/P2$). Here $P1/P2$ are the multivariate colour pdf values of the object and background regions.

$$P(x \mid \Omega) = \frac{1}{(2\pi)^{d/2} |\Sigma_i|} e^{-\frac{1}{2}(x-\mu_i)^t \Sigma_i^{-1} (x-\mu_i)} \tag{7}$$

We also have particles with a job to explore search space with an objective to find significant movement coming from frame difference, constant brightness assumption, $b(x_i)$ and relevant probability values including the log (x) as explained above. Conclusively, there are almost endless possibilities and strategies for setting up the initial position of these sensors and is a crucial step in reducing the processing requirements for real time tracking. Two possible scenarios are shown in the figure below.

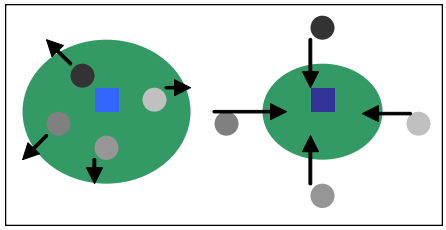

Figure 8. Expanding/Shrinking motion of particles produced by log ($p1/p2$) in context to the square rectangle (centroid calculated by HBP)

2.4 Conflict Resolution

In figure 6 four particle like sensors were shown (infactuated by the flock of bird like behaviour) that either move towards the food (blue region) or fly away from it. A counter is used to stop oscillations when they their enter their target region. This then could be used to give further births to particles in the close vicinity to explore this sub-space futher. This also gives clear indication about the scale of the region (green) especially when the object is moving either towards or away from the camera. At very close proximity we may have a conflict like situation which could be resolved using weighted averaging technique as shown in expression 8. In another tracking application we successfully used majority voting that also becomes madatory when any particle like sensor raises a binary flag to 1 indicating compliance with its primary objective.

$$B(x_1, x_2, \ldots, x_n, O) = \sum_{i=1}^{n} \omega_i \rho(O \mid x_i) \qquad (8)$$

Here $\rho(O \mid x_i)$ is probability that observation came from x.

3. Tracking results

In this section we have presented some further tracking results by adapting specific dynamics for each and every particle based sensor as influenced by the Selective Histogram Back Projection Algorithm. In majority cases the effective particle rate η_E has = been recorded to be above 0.9 at mean error rates of between 0.05 and 0.15 which is comparatively better than Hybrid Particle filter [15]. The PETS sequence utilizes mean seperation distance between final particle positions to dynamically adjust the scale of the HBP window. A robust technique based on static and reciprocating particle sensors is also shown in figure [10]. When the walking man was partially occluded by the white board static particles form a defensive line by the most probablistic positions the object is likely to emerge after occlusion based on its velocity profile and other unique characteristics. The red lines indicate there might be a change in elevation or if the subject reverses its direction. The static particles have a limited life-time from 0.05 to 0.1 seconds depending on the frame update rate and the velocity characteristics of the object in motion. By limiting the life time and thus by generating similar particles reduces the need to spend computing resources on the associated dynamics. However, the complete discussion would go beyond the scope of this paper and would be addressed in a forth coming publication in a relevant conference.

Figure 9. Tracking result from the blue ball sequence through the Initialization of zero level set curve at various locations on blue ball. It only took 4, 5 iteration to recover the ball shape in the first and last part above

3.1 Framework to represent Level Set/Shape Curves

In figure 10, a shape curve has been implemented along with the square window representing the kernel window size. The level set or the shape curve could be represented in linked list format as in [10]. The shape curve is translated onto the current frame

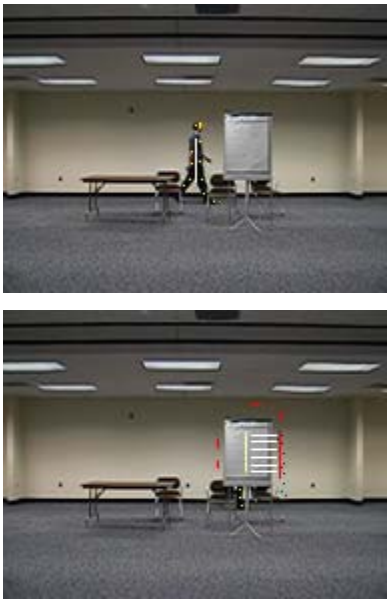

Figure 10. Two frames have been shown from the lecture theatre sequence. The position of static and dynamic sensors had been shown using red and yellow color lines. The white lines show the propagation of particle (shown using blue dots) that successfully recovered the object after occlusion

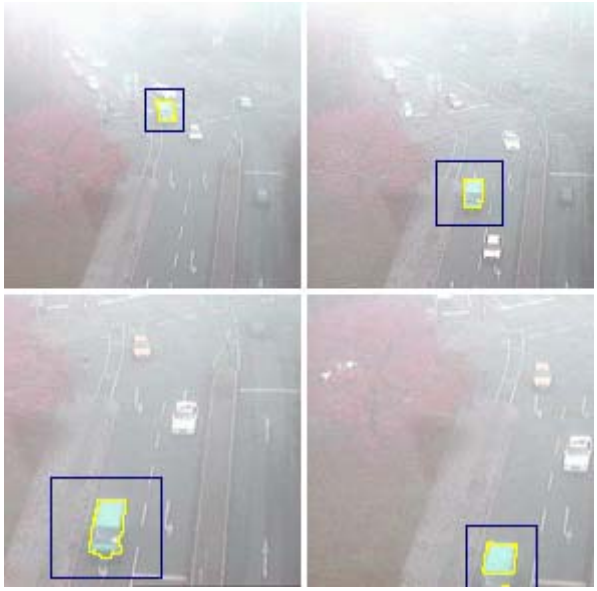

Figure 11. Some frames from the traffic sequence. The shape curve is implemented using level set format used in [10]

using the parameters of the motion estimated by the methods used in this paper. The idea has been portrayed in the following ball sequence tracking below where it shows that by aligning the shape with the motion estimation module a considerable amount of computation time could be saved.

4. Conclusions

A selective Histogram back projection algorithm was utilized as an indicator towards a globel best solution and an initial position

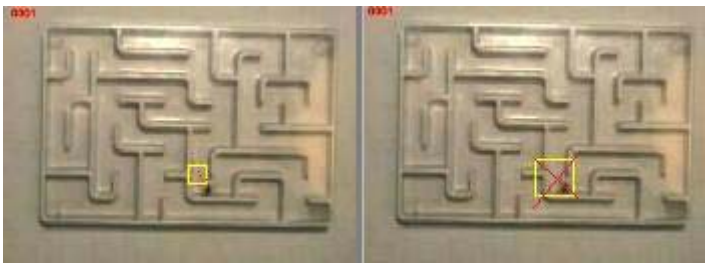

Figure 12. An exemplification of dynamic windows and the search strategy used in our algorithms to reduce processing overheads

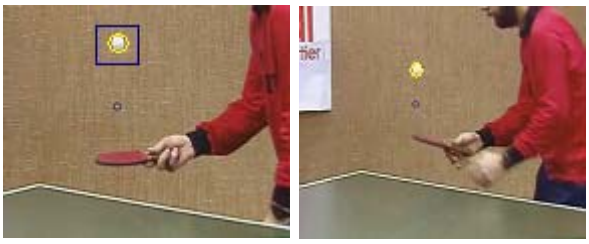

Figure 13. Two frames from table tennis sequence. The proposed algorithm successful tracks as compared to the standard mean shift presented in [15]

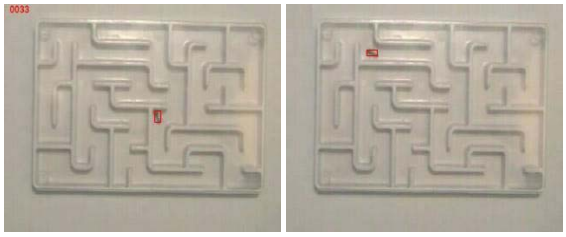

Figure 14. The successful tracking of an Ant in a maze at 20fps is shown above

Figure 15. Multi objective tracking has been applied in this PETS sequence. Multiple pedestrians are being tracked at 13fps with dynamic window size

Figure 16. An example of boundary construction with only limited number of particles using simple linear interpolation

of a super particles with reciprocating subordinate paticle like sensors with multiple objective functions .The effective positioning

of paticles with their dynamics controlled by this vector and by using reciprocating movements of the subordinate particles around the calculated mean postion leads saving considerable processing costs compared to traditional particle filters and hence to meet our prime objective of real time tracking. This technique also requires limited particles (10-25) at much higher accuracy rates then standard meanshift based approaches. The scale adjustment using oscillatory subordinate particles perform at much faster rate then the standard camshift based technique. Evidence or measurement based technique based on static sensors has been found particularly helpful during phases of partial to complete occlusion. In fact it had been found through experiments that the overwhelming component of prescribed motion could be discovered using only successive static sensor like particles with limited life spans. The shape propagation is a very comprehensive and elaborate procedure and would be addressed in detail in future publications. However, any keen reader is advised to consult paper [10] for an introduction to the subject. Our future research would also address how particle based sensors and shape alignment could be brought together utilizing a common frame work.

References

[1] http://www.mobileye.com/.

[2] http://www.amazon.com/NightWatcher-Robotic-Security-Light-Camera-LED/dp/B0040720F2.

[3] http://www.samsung.com/us/2013-smart-tv/.

[4] Bull, S., Cooke, N. Mabbott, A. (2007). Visual Attention in Open Learner Model Presentations An Eye-Tracking Investigation, in C. Conati, K. McCoy & G. Paliouras (eds), User Modeling 2007:11th International Conference, Springer-Verlag, Berlin Heidelberg, p. 187-196.

[5] Xin Li, Kejun Wang, Wei Wang, Yang Li. (2010). A multiple object tracking method using Kalman filter, *Information and Automation* (*ICIA*), 2010 *IEEE International Conference on* , p. 1862,1866, 20-23 June.

[6] HwangRyol Ryu, Huber, M. (2007). A particle filter approach for multi-target tracking, *Intelligent Robots and Systems. IROS* 2007. *IEEE/RSJ International Conference on* , 2753, 2760, Oct. 29 -Nov. 2.

[7] Kennedy, J., Eberhart, R. (1978). Particle swarm optimization, *Neural Networks, In:* Proceedings., *IEEE International Conference on*, 4, p. 1942, Nov/Dec.

[8] Gupta, D. K., Arora, Y., Singh, U. K., Gupta, J. P. (2012). Recursive Ant Colony Optimization for estimation of parameters of a function, *Recent Advances in Information Technology* (*RAIT*), 2012 1st *International Conference on*, p. 448, 454, 15-17 March.

[9] http://en.wikipedia.org/wiki/Recursive_Bayesianestimation,Vankatesh Babu, Perez, R., Bouthemv, P. (2005). Robust tracking with motion estimation and kernel based color modelling, Image Processing, ICIP 2005, *IEEE international Conferen*ce, p. I, 717-20, 11-14 Sep.

[10] Shi, Y., Kerl, W. (2005). A Fast Level Set Method without solving PDEs, *In*: Proceedings of the ICASSP2005, Philadelphia, PA, USA, March.

[11] Comaniciu, D., Ramesh, V., Meer, P. (2003). Kernel-based object tracking, Pattern Analysis and Machine Intelligence, *IEEE Transactions on*, 25 (5) 564, 577, May.

[12] Mingfei Wang, Linlin Ci, Ping Zhan, Yongjun Xu. (2007). Acoustic Source Localization in Wireless Sensor Networks, *Intelligent Information Technology Application, Workshop on*, p. 196,199, 2-3 Dec.

[13] Bruce D. Lucas. (1984). Generalized Image Matching by the Method of Differences, doctoral dissertation, tech. report , Robotics Institute, Carnegie Mellon University, July.

[14] Jung-ho Lee, Woong-hee Lee, Dong-Seok Jeong. (2003). Object tracking method using back-projection of multiple color histogram models, *Circuits and Systems, ISCAS '03. In:* Proceedings of the 2003 International Symposium on, 2, p. II-668, II-671, 2, p. 25-28 May.

[15] Rahman, M. M., Bhattacharya, P., Desai, B. C. (2005). Statistical similarity measures in image retrieval systems with categorization & block based partition, Imaging Systems and Techniques. *IEEE International Workshop*, p. 92,97, 13 May.

[16] http://en.wikipedia.org/wiki/Hidden_Markov_model

[17] Maggio, E., Cavallaro, A. (2005). Hybrid Particle Filter and Mean Shift tracker with adaptive transition model, Acoustics, Speech, and Signal Processing, *In:* Proceedings. (ICASSP '05). *IEEE International Conference*, 2, p. 221,224, March 18-23.

[18] Alam, I. (2013). Tracking using Information Fusion Principle' CEEC 2013, University of Colchester, Essex, September.
[19] http://en.wikipedia.org/wiki/Linear_regression.