

# Multi-Resolution Tracking in Space and Time

Sumantra Dutta Roy

Department of Electrical Engineering  
Indian Institute of Technology Delhi, INDIA  
sumantra@cse.iitd.ac.in

Larry Steven Davis

Department of Computer Science  
University of Maryland College Park, USA  
lsd@cs.umd.edu

Son Dinh Tran

Department of Computer Science  
University of Maryland College Park, USA  
sontran@cs.umd.edu

B. Sreenivasa Vikram

Bharti School of Telecom Tech & Management  
Indian Institute of Technology Delhi, INDIA  
bsvikky@yahoo.com

## Abstract

*This paper proposes efficient and robust methods for tracking a moving object at multiple spatial and temporal resolution levels. The efficiency comes from optimising the amounts of spatial and temporal data processed. The robustness results from multi-level coarse-to-fine state-space searching. Tracking across resolution levels incurs an accuracy-versus-speed trade-off. For example, tracking at higher resolutions incurs greater processing cost, while maintaining higher accuracy in estimating the position of the moving object. We propose a novel spatial multi-scale tracker that tracks at the optimal accuracy-versus-speed operating point. Next, we relax this requirement to propose a multi-resolution tracker that operates at a minimum acceptable performance level. Finally, we extend these ideas to a multi-resolution spatio-temporal tracker. We show results of extensive experimentation in support of the proposed approaches.*

## 1. Introduction

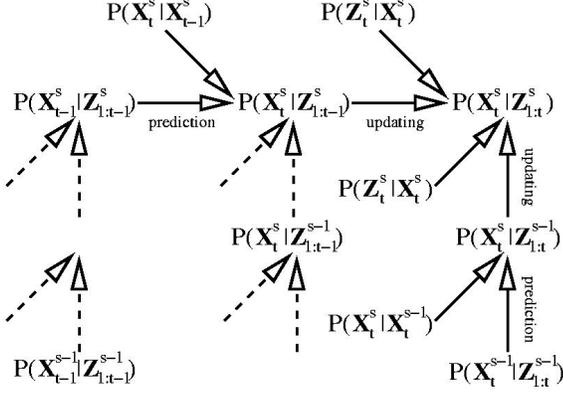
We present novel methods for tracking a moving object at multiple levels of spatial and temporal resolution<sup>1</sup>. The motivation is the accuracy-versus-speed trade-off of tracking at different resolutions. For example, a tracker operating at the highest spatial resolution has better accuracy compared to one that tracks at lower spatial resolutions, at the cost of greater processing time per frame. We propose a function to find an optimal operating point for a tracker in terms of the scale. Further, we propose a faster multi-

---

<sup>1</sup>Here, we use the terms multi-resolution and multi-scale interchangeably

scale (spatial) tracker based on a minimum acceptable performance limit. Finally, the paper also proposes a spatio-temporal multi-scale tracker.

Coarse-to-fine, or hierarchical search is a common Computer Vision technique to deal with large search spaces (e.g., [1]). The state variable decomposition in the layered sampling approach of Sullivan et al. [16] involves an independence assumption on the decomposed variables. This is contrary to the intuitive notion of an explicit relationship existing between representations at different scales. Hua and Wu's multi-scale tracking approach also makes a similar independence assumption [12]. Our approach considers an explicit modelling of the above. Further, our approach does not incur the memory and processing overhead of explicitly constructing and maintaining a Dynamic Markov network, as in [12]. Explicit modelling of hierarchical search space pruning typically leads to a hierarchy of state models. Stenger et al. [15] define a fixed tree-based structure whose levels partition the state space at different levels of granularity. The authors use Bayesian filtering (in time) at each tree level, and prune the state space based on the probability density of the (coarser) parent partitions. In contrast to their approach, we explicitly model the Bayesian filtering across levels, and consider observations at multiple levels of resolution. An earlier work of a subset of the authors in this paper presents some very preliminary results of a minimum performance multi-scale tracker [18]. This paper develops not just this idea much further, it also proposes a novel accuracy-speed optimal tracker, and a new spatio-temporal multi-scale tracker.



**Figure 1. Sequential Bayesian filtering across spatial resolution and time**

## 2. Sequential Bayesian Filtering across Spatial Resolutions

Given an observation sequence  $\mathbf{Z}_i$ ,  $i = 1, 2, \dots, t$ , a tracker updates its state  $\mathbf{X}_t$  at time instant  $t$  as follows (e.g., [7]):

$$P(\mathbf{X}_t | \mathbf{Z}_{1:t-1}) = \int P(\mathbf{X}_t | \mathbf{X}_{t-1}) P(\mathbf{X}_{t-1} | \mathbf{Z}_{1:t-1}) d\mathbf{X}_{t-1} \quad (1)$$

$$P(\mathbf{X}_t | \mathbf{Z}_{1:t}) \propto P(\mathbf{Z}_t | \mathbf{X}_t) P(\mathbf{X}_t | \mathbf{Z}_{1:t-1}) \quad (2)$$

This process is recursive, starting from an assumed prior  $P(\mathbf{X}_0)$ , which may be a Gaussian, a mixture of Gaussians [10], or a weighted particle set [13], for example. We use the same methodology to model tracking across multiple spatial scales. If we use superscripts to denote the scale, we can for example, obtain  $P(\mathbf{X}_t^s | \mathbf{Z}_{1:t}^{1:s})$  from  $P(\mathbf{X}_t^{s-1} | \mathbf{Z}_{1:t}^{1:s-1})$  and the observation  $\mathbf{Z}_t^s$  as follows:

$$P(\mathbf{X}_t^s | \mathbf{Z}_{1:t}^{1:s-1}) = \int P(\mathbf{X}_t^s | \mathbf{X}_t^{s-1}) P(\mathbf{X}_t^{s-1} | \mathbf{Z}_{1:t}^{1:s-1}) d\mathbf{X}_t^{s-1} \quad (3)$$

$$P(\mathbf{X}_t^s | \mathbf{Z}_{1:t}^{1:s}) \propto P(\mathbf{Z}_t^s | \mathbf{X}_t^s) P(\mathbf{X}_t^s | \mathbf{Z}_{1:t}^{1:s-1}) \quad (4)$$

Fig. 1 gives a diagrammatic representation of sequential Bayesian filtering across space and time, showing the possible transition paths between spatial resolution and time. For simplicity here, we assume integer values for the scale  $s$ .

We model our multi-scale tracker in a particle filtering (CONDENSATION [13]) framework. Three components of a tracker are specifying the the state vector, the State/Process Dynamics Model, and the Observation/Measurement Model.

**State Vector Specification:** A common motion model is a 2-D affine transformation, since it can model commonly observed object deformation effects such as rotation, translation, scaling and shear. We have also experimented with a 5-element restricted affine state vector as in [14], where we take the elements of the state vector as indicators of an oriented rectangle: the centroid of the colour blob, width,

length, and orientation angle.

**State/Process Dynamics Model** In our experiments, we use a random walk model for the state dynamics. This is not restrictive in any way, since it is easy to frame higher order AR models (e.g., a constant acceleration) in this framework [9]. Most existing works use simple models for their experimentation [14], [19], or learn the parameters of a second order AR process from representative video sequences [13]. As in Fig. 1, we need models for filtering across spatial resolution scales. One may use any 2-D linear transformation for this purpose. We take  $f(s, s-1)$  to be the ratio of the image dimensions corresponding to the two scales  $s$  and  $s-1$ , and  $\mathbf{U}^s$  to be a zero-mean White Gaussian noise vector  $\mathcal{N}(\mathbf{0}, \mathbf{Q}^s)$ . We take  $\mathbf{Q}^s$  to be a constant diagonal matrix, whose parameters are learnt from representative sequences. We use this equation when going up in scale from  $s-1$  to  $s$ . For the 5-parameter restricted affine case, we similarly scale all parameters except the orientation by this factor  $f(s, s-1)$ , which is a constant for a particular pair of scales. The authors in [16] show this to be a good choice. When we go in the opposite direction, we assume a deterministic model for the state update. For the general affine case for example,  $\mathbf{X}_t^s = f(s-1, s) \mathbf{X}_t^{s-1}$ . We explicitly model the dependence between parameters at different scales - which is quite intuitive, unlike the independence assumption of Hua and Wu [12]. Further, our approach does not incur the extra memory and processing requirement of a Dynamic Markov network, which they use in the above work. We have similar models for diagonal movement across scales.  $\tilde{\mathbf{X}}_t^s = f(s, s-1) \tilde{\mathbf{X}}_{t-1}^{s-1} + \mathbf{U}'$  and  $\tilde{\mathbf{X}}_t^{s-1} = f(s-1, s) \tilde{\mathbf{X}}_{t-1}^s + \mathbf{U}^{s-1}$ . Here,  $f(s-1, s)$  is the multiplicative inverse of  $f(s, s-1)$ , and the last term in the above two cases accounts for the noise.

**Observation/Measurement Model** We use a scheme similar to [14] for our observation/measurement model. We need a function that is smooth and a function of the current scale parameter,  $s$ . The motivation behind this is three-fold. First, a smooth likelihood function is useful for guiding the state space search from coarse to fine, as in [6]. Secondly, the reliability of the data is reduced at lower resolutions, hence the function should have a higher degree of tolerance. Third, when there is a loss of track, a smooth likelihood function at a low spatial resolution level is important for regaining tracking. In this paper, we use a scale-specific number of levels,  $K(s)$ , with  $K(s)$  being smaller at lower resolution levels. We compute the likelihood as

$$P(\mathbf{Z}_t^s | \mathbf{X}_t^s) \propto \exp(-\lambda \sum_{k=1}^{K(s)} D^2(\mathbf{Z}_t^s(k), \mathbf{Z}_0^{s_{\max}}(k))) \quad (5)$$

$\lambda$  is a constant, and  $D(\mathbf{Z}, \mathbf{Z}_0)$  is the distance between the observed histogram and that of the target object of interest, based on the Bhattacharyya Coefficient [5]:

$$D^2(\mathbf{Z}, \mathbf{Z}_0) = 1 - \sum_{i=1}^N \sqrt{p_i^{\mathbf{Z}} p_i^{\mathbf{Z}_0}} \quad (6)$$

### 3. Tracking at the Optimal Scale: Accuracy-Speed Considerations

A high spatial resolution enables one to estimate the position of the object of interest the best. This, however, incurs more processing cost due to the large number of pixels to be processed. The opposite happens at low resolutions. *For every frame, we would like to find the optimal scale  $s$ , which balances these two contradictory requirements.* An added problem is the formulation of a criterion which will enable the selection of such an optimal  $s$ . We examine the problem in a constrained optimisation framework. We define a **Demerit Function**  $\mathcal{F}(s, \mu)$  as follows:

$$\mathcal{F}(s, \mu) \triangleq \mu \mathcal{A}(s) + (1 - \mu) \mathcal{S}(s) \quad (7)$$

$\mathcal{A}(s)$  and  $\mathcal{S}(s)$  are the **Accuracy term** and the **Speed term**, respectively, and  $\mu$  is a linear combination coefficient,  $\mu \in [0, 1]$ . We define the **Speed term**  $\mathcal{S}(s)$  as:  $\mathcal{S}(s) \triangleq (s/s_{max})^2$ . We define the **Accuracy term**  $\mathcal{A}(s)$  as follows:

$$\mathcal{A}(s) \triangleq 1 - \exp(-\lambda \sum_{k=1}^{K(s)} D^2(\hat{\mathbf{Z}}_t^s(k), \mathbf{Z}_0^{s_{max}}(k))) \quad (8)$$

Here,  $\hat{\mathbf{Z}}_t^s$  is the observation at the prediction - the estimated value of the object state,  $\hat{\mathbf{X}}_t^s | \mathbf{Z}_{1:t-1}^s$ .

#### 3.1 Computing the Optimal Scale

We use a constrained optimisation framework to compute the optimal scale - we look for local maxima of the Demerit Coefficient  $\mathcal{F}(s, \mu)$ , and values along the boundary conditions as well. One can either treat  $\mu$  as a system-dependent constant, or a variable. Based on system requirements, and empirical observations, a system designer may want to set a specific weighing value for  $\mu$  in Eq. 7. Thus, we need to consider (discrete) partial derivatives of  $\mathcal{F}$  with respect to  $s$ : the points where  $|\frac{\Delta \mathcal{F}}{\Delta s}|$  is a minimum, close to 0. We assume the scale  $s$  to take on integer values between  $s_{min}$  and  $s_{max}$  (as in Sec. 2 and Fig. 1, for instance). (This is not restrictive in any way, since we can switch between this notation, and one for a normalised value of the scale - seamlessly.) For a better approximation of the above discrete derivative to the continuous one, we consider the least possible  $\Delta s$  i.e., 1. For local extrema, we can check for maxima using either the second derivative test, or plugging in values of  $s$  into the function itself (we have to do the latter for the boundary conditions, anyway). For a variable  $\mu$ , we consider derivatives of  $\mathcal{F}(s, \mu)$  with respect to both  $\mu$  and  $s$ . Taking  $\frac{\partial \mathcal{F}(s, \mu)}{\partial \mu} = 0$  gives us an extremum at that  $s$  for which  $|\mathcal{A}(s) - \mathcal{S}(s)|$  is minimum. This gives us  $s_{opt}$ ,

the optimal scale. The other partial derivative is a discrete one:  $|\frac{\Delta \mathcal{F}(s, \mu)}{\Delta s}|$  should be a minimum, at  $s = s_{opt}$ .

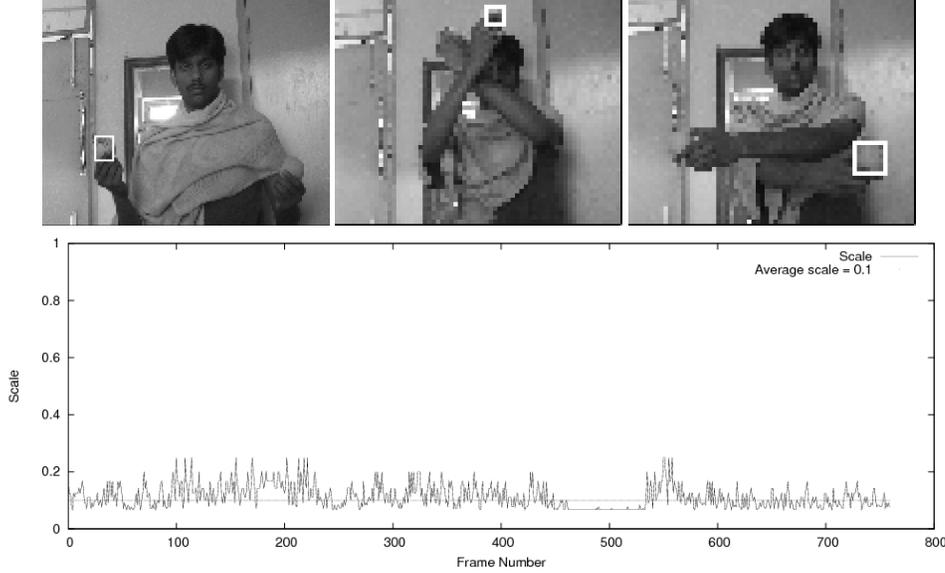
Fig. 2 shows the main steps in the algorithm. We start

ALGORITHM OPT_ACC-SPEED-MULTI-SCALE
A. INITIALISE scale $s := s_{min}$
B. Start with sample set corr to assumed prior $P(\mathbf{X}_0^s)$
C. Delineate moving object
REPEAT FOR ALL remaining frames:
1. PREDICT new sample set: State Dynamics Eqns. (*Sec. 2*)
2. obtain MEASUREMENT (*Sec. 2*)
3. UPDATE particle weights acc to $P(\mathbf{Z}_t^s   \mathbf{X}_t^s)$ ; renormalise
4. SCALE CHANGE CHECK: (*Sec. 3*) IF loss-of-track THEN $s := s_{min}$ ; ELSE $s_{opt} := \text{compute-optimal-scale}$ ;

**Figure 2. Optimal Scale tracker based on Accuracy-Speed considerations.**

at  $s_{min}$ , with an assumed prior density (Steps A and B in Fig. 2). If there is a loss of track (Step 4 in Fig. 2), the system reverts back to the minimum scale. This big jump across resolution levels can occur in an occlusion example (the tracker waits for the object to reappear), or to broaden the sampling range (to cause the state to quickly leave the current sampling volume: Details in Section 2). *It is important to note that this rule allows a scale-space tracker to be better than one operating at the maximum resolution.* The same idea applies to the other trackers proposed in this paper (Sections 4 and 5).

The top row in Fig. 3 shows excerpts from the successful use of the optimal scale tracker on a long sequence with occlusions - the two identical tennis balls cross each other in front of a person wearing a similarly coloured dress. As can be seen from these experiments, the average scale is 0.1, a small fraction. These experiments consider  $\mu = 0.5$ . While the formulation calculates the optimal scale, the optimisation itself takes some time - slightly less than a frame a second, on a 2.8GHz P-IV machine without any hardware acceleration, 1GB of memory, and running MATLAB 7.0. In the next section, we relax the stringent requirement of finding the optimal scale - to one of tracking at a minimum acceptable performance. This gives a a very fast spatial multi-resolution tracker.



**Figure 3. The optimal scale tracker based on accuracy-speed considerations (Sec. 3): sample excerpt from a long sequence with occlusions, and variation in the normalised scale  $\langle s \rangle$  (the range  $0 < s_{min} < s_{max}$  normalised to  $0 < \langle s_{min} \rangle < 1$ ), with frame number. Average  $\langle s \rangle = 0.1$ .**

#### 4. A Spatial Multi-Scale Tracker with a Minimum Acceptable Performance Level

A standard technique to analyse the accuracy of a tracker for non-linear dynamical systems is to consider the posterior Cramer-Rao Lower Bound (CRLB, hereafter) [17], [8], [11]. A signal at low resolution  $s$  often comes from linear 2-D aggregation of multiple signals at higher resolution  $s'$  [3], [4]. Hence, it is reasonable to assume that the variations can be modelled as

$$\mathbf{Q}_t^{s'} = f^2(s, s') \mathbf{Q}_t^s, \quad \mathbf{R}_t^{s'} = f^2(s, s') \mathbf{R}_t^s \quad (9)$$

where  $\mathbf{Q}$  and  $\mathbf{R}$  indicate the state and measurement noise, respectively. Let  $\hat{\mathbf{X}}_t^s$  be an unbiased estimator of  $\mathbf{X}_t^s$ . The lower bound on the error covariance is given by the following bound on its expectation:

$$\mathcal{E}\{(\hat{\mathbf{X}}_t^s - \mathbf{X}_t^s)(\hat{\mathbf{X}}_t^s - \mathbf{X}_t^s)^T\} \geq (\mathbf{J}_t^s)^{-1} \quad (10)$$

Here,  $\mathbf{J}_t^s$  is the Fisher Information Matrix, and its inverse  $\mathbf{P}_t^s = (\mathbf{J}_t^s)^{-1}$  is the Cramer-Rao Lower Bound (CRLB). In [17], the authors give a recursive computational mechanism for the non-linear filtering problem in the form of a Riccati equation. This indicates that as scale  $s$  increases, so does  $\mathbf{J}_t^s$ , and thus  $\mathbf{P}_t^s$  decreases. In practice, the signal-to-noise ratio may be small (the effectiveness of a tracker is judged in cluttered backgrounds, and the presence of occlusions). To calculate the CRLB accurately, one needs a

good false alarm model [8], [11]. Clutter and occlusions are not easy to model into false alarm models - in visual tracking applications, however. A simple approximation is to put bounds on the sample covariance matrix  $\Sigma_t^s$ , instead. This speeds up the tracking (as compared to the optimal accuracy-speed tracker in Sec. 3) in that we do not have to evaluate a function across every scale for each frame, if we are satisfied with a norm on the sample covariance matrix e.g., the Frobenius norm. Fig. 4 gives an overview of the alternative tracking strategy. The sample covariance matrix  $\Sigma_t^s$  is an indication of the amount of clutter. We use the following stable continuous adjustment rule [2]: While  $\|\Sigma_t^s - \mathbf{Q}_0^{s_{max}}\| > \|\Sigma_t^{s_{min}} - \mathbf{Q}_0^{s_{max}}\| + \epsilon$ , enforce  $s := s + 1$  (Step 4 in Fig. 4).  $\mathbf{Q}_0^{s_{max}}$  denotes the state noise at the highest level of resolution  $s_{max}$  in the first frame.  $\epsilon$  is a small constant to account for the numerical variability when we compute the statistics of the particle set.

##### 4.1. Performance Analysis of the Minimum Performance Level Multi-Scale Tracker

The loss-of-track rule (Section 3.1) effectively keeps the object within the sampling scope, and the accuracy bounds rule makes this sampling efficient.

ALGORITHM MIN_PERF_MULTI-SCALE
A. INITIALISE scale $s := s_{\min}$ B. Start with sample set corr to assumed prior $P(\mathbf{X}_0^s)$ C. Delineate moving object REPEAT FOR ALL remaining frames: 1. PREDICT new sample set: State Dynamics Eqns. (*Sec. 2 *) 2. obtain MEASUREMENT (*Sec. 2 *) 3. UPDATE particle weights acc to $P(\mathbf{Z}_t^s   \mathbf{X}_t^s)$ ; renormalise 4. CHECK bounds on $\Sigma_t^s$ (*Sec. 4 *) IF loss-of-track THEN $s := s_{\min}$ ; ELSE IF more-accuracy-reqd THEN $s := \min(s + 1, s_{\max})$ ; GOTO 4 ELSE $s := \max(s - 1, s_{\min})$ ;

**Figure 4. A spatial multi-scale tracker with a minimum performance level.**

#### 4.1.1 Accuracy Test

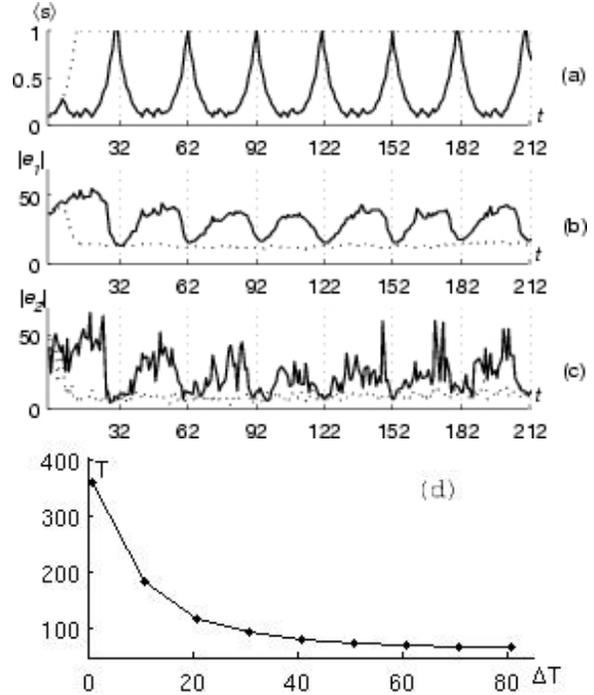
We test the accuracy of our multi-scale tracker using two trackers A and B. A always tracks at the maximum scale  $s_{max}$  (except for a short initialisation period), whereas B follows the minimum acceptable performance criterion (Sec. 4) - except being forced to move up in scale gradually to  $s_{max}$  every  $\Delta T = 30$  frames (as in the expression for the smooth upward diagonal movement across scales in Sec. 2). Fig. 5(a) shows a graph of the resolution varying with time for Tracker A (dotted) and Tracker B (solid curve). Fig. 5(b) shows the weighted mean error of all particles (using the particle weights). Fig. 5(c) is a plot of the error associated with the particle with the maximum likelihood (which often, but not necessarily always, corresponds to the minimum error). The ‘error’ is the norm of the state vector difference. In effect, one can move up in scale at any point in time to increase the accuracy of the tracker.

#### 4.1.2 Efficiency Test

We run Tracker B over the same sequence with  $\Delta T$  taking on values 1, 10, . . . 80, and measure the total running time over the sequence. (When  $\Delta T = 1$ , the two trackers become identical.) Fig. 5(d) shows this variation: we save more time with a large  $\Delta T$ . In general, this time saving is dependent on the smallest resolution permissible, and the complexity of the sequence to be tracked.

#### 4.1.3 Robustness Test

Fig. 6 shows the results of successful tracking over the same long sequence (over 700 frames) as with the optimal scale

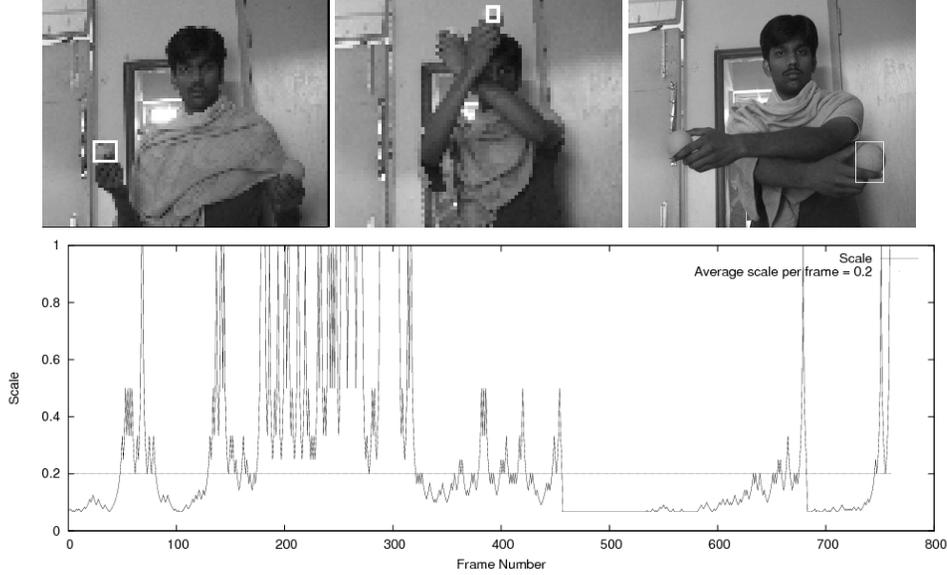


**Figure 5. Accuracy tests with the spatial scale-space tracker with a minimum performance level (Section 4): Plots (versus the frame number) of the following parameters: (a) the normalised scale  $\langle s \rangle$ ; (b)  $|e_1|$ : the weighted mean error of all particles; and (c)  $|e_2|$ : the error associated with the particle with the maximum likelihood. (d) Efficiency test: the total running time  $T$  versus the time step  $\Delta T$ .**

tracker (Section 3).

## 5. Multi-Scale Spatio-Temporal Tracker

Surveillance applications typically involve processing of long videos. One needs to pay more attention to the few possible cases of unexpected motions and actions, rather than expend time on the predictable and normal parts, which would usually occupy a significant length of the video. This is applicable for post-facto off-line processing, or on-line processing with a buffer. Thus, one needs to sample the video in an efficient manner, so as to skip predictable portions, and yet not miss out on the portions of interest: loss of track owing to unexpected activity. Fig. 7 gives an overview of the proposed spatio-temporal multi-scale tracker for surveillance applications.  $\Delta t$  refers to the number of frames to skip (‘the skip factor’). We start from a



**Figure 6. The spatial multi-scale tracker with a minimum performance level (Sec. 4): sample output excerpt from the same long sequence with occlusions, and variation in the normalised scale  $\langle s \rangle$  with the frame number. Average  $\langle s \rangle = 0.2$ .**

fixed value  $\alpha$ . (For our experiments, we fix  $\alpha = 10$  - as empirically observed over a large number of representative surveillance sequences. We fix this based on the variation of the total number of tracking errors for a long sequence, with  $\alpha$  - see Fig. 7.) The tracker has a constant forward skip until loss of track ('a forward error'). The roll-back is then exponential (Step 5 in Fig. 7). *Thus in the worst case, the tracker actually examines all frames* - the same as a tracker which does not have this temporal skip methodology. *Further, the tracker operates at an efficient spatial scale* - according to the methodology of the optimal accuracy-speed tracker (Sec. 3, Fig. 2); or the minimum acceptable performance tracker (Sec. 4, Fig. 4).

### 5.1. Effectiveness of the Proposed Temporal Multi-Scale Tracker

From observations over a large number of representative surveillance sequences, we can model the probability of error in going in the forward direction as a small constant  $p$ . Another important empirical observation for large surveillance videos is that large rollbacks are rare. Here, we observe that large rollbacks are very few. Thus, it is reasonable to model the probability of a particular total roll-back length  $\tau(i)$  given a forward error, as a discretised exponential distribution (for ease in mathematical representation, one may

consider  $\alpha$  to be a power of 2):

$$P(\tau(i)|\text{fwd-err}) = \frac{1}{\nu} \exp(-\nu\tau(i)), \text{ where}$$

$$\tau(i) = \sum_{j=\log \alpha - i}^{\log \alpha - 1} 2^j; 1 \leq i \leq \log \alpha \quad (11)$$

The maximum number of times a particular frame can be seen is  $\log \alpha$ . *It is not easy to compute the average number of times a particular frame is examined - this is an important performance parameter. However, one can easily compute the probability of the worst-case scenario - when each frame encounters a full rollback.* For a video with  $N$  frames, this works out to be  $[p P(\tau(\log \alpha)|\text{fwd-err})]^{[N/\alpha]\alpha}$ . For surveillance videos with suitably chosen motion models,  $p$  is typically a very small fraction, so is  $P(\tau(\log \alpha))$ . For large  $N$ , the resultant value is negligibly small.

This methodology reduces processing costs by a huge extent. Some sample statistics are - for 750 frames ( $720 \times 576$ ) tracked at an average spatial scale of 0.1 processes only 281 frames out of 750, resulting in 1153224 pixels processed. This is 0.0037 of the total number of pixels at the full resolution for the complete video, 311040000.

## 6. Conclusions

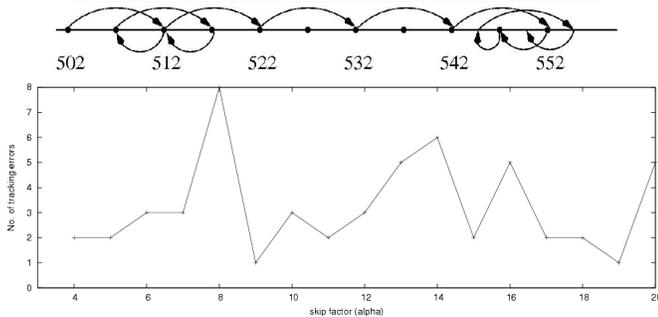
This paper presents an integrated approach toward scalable tracking in space and time, with a quantitative evalu-

### ALGORITHM SPATIO-TMPRL\_MULTI-SCALE

```

A. INITIALISE scale  $s := s_{\min}$ 
B. Use either tracker (Fig. 2, 4)
   for the first  $\alpha$  frames
C.  $\Delta t := \alpha$ ;  $t := t + \Delta t$ ; (* init skip *)
REPEAT UNTIL video over:
0.  $t_{last} := t$  (* mark frame for rollback limit *)
1. PREDICT new sample set:time  $t$ 
   State Dynamics Eqns. (*Sec. 2 *)
2. obtain MEASUREMENT (*Sec. 2 *)
3. UPDATE particle weights acc to
 $P(\mathbf{Z}_t^s | \mathbf{X}_t^s)$ ; renormalise
4. CHECK: IF loss-of-track THEN
   handle as in Tracker 1 or 2;
5. IF still loss-of-track THEN
 $\Delta t := \Delta t / 2$ ;  $t := t + \Delta t$ ; REPEAT 1-4
   UNTIL NO loss-of-track OR
 $t == t_{last} + 1$ ;
   ELSE  $\Delta t := \alpha$ ;  $t := t + \Delta t$ ;

```



**Figure 7. The proposed spatio-temporal multi-scale tracker: An overview; and a sample traversal of frames: this shows a portion with many roll-backs; and at the bottom: selecting the skip factor (Details in Sec. 5)**

ation of the tracker performance. We propose three types of trackers in this paper. The first formulation considers tracking at a accuracy-speed optimal spatial scale, the second works with a minimum performance guarantee, while the last formulation is for spatio-temporal multi-resolution tracking.

## References

- [1] E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid Methods in Image Processing. *RCA Engineer*, 29:33 – 41, 1984.
- [2] Y. Bar-Shalom, X. Rongli, and T. Kirubarajan. *Estimation with Application to Tracking and Navigation*. Wiley-Interscience, 2001.
- [3] P. J. Burt and G. S. van der Wal. An Architecture for Multi-Resolution, Focal, Image Analysis. In *Proc. ICPR*, pages 305 – 311, 1990.
- [4] K. C. Chou, A. S. Willsky, and A. Benveniste. Multi-scale Recursive Estimation, Data Fusion, and Regularization. *IEEE Transactions on Automatic Control*, 39(3):464 – 478, 1994.
- [5] D. Comaniciu and V. Ramesh. Mean Shift and Optimal Prediction for Efficient Object Tracking. In *Proc. IEEE ICIP*, pages III:70 – 73, 2000.
- [6] J. Deutscher, A. Blake, and I. Reid. Articulated Body Motion Capture by Annealed Particle Filtering. In *Proc. IEEE CVPR*, pages 2126 – 2133, 2000.
- [7] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice*, volume 10. Kluwer Academic Publishers, 2000.
- [8] A. Farina, B. Ristic, and L. Timmoneri. Cramer-Rao Bound for Nonlinear Filtering with  $p_d < 1$  and Its Application to Target Tracking. *IEEE Transactions on Signal Processing*, 50(8):1916 – 1924, 2002.
- [9] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2002.
- [10] B. Han, D. Comaniciu, Y. Zhu, and L. S. Davis. Incremental Density Approximation and Kernel-Based Bayesian Filtering for Object Tracking. In *Proc. IEEE CVPR*, pages I:638 – 644, 2004.
- [11] M. Hernandez, T. Kirubarajan, and Y. Bar-Shalom. Multi-sensor Resource Deployment Using Posterior Cramer-Rao Bounds. *IEEE Transactions on Aerospace and Electronic Systems*, 40(2):399 – 416, 2004.
- [12] G. Hua and Y. Wu. Multi-scale Visual Tracking by Sequential Belief Propagation. In *Proc. IEEE CVPR*, pages I:826 – 833, 2004.
- [13] M. Isard and A. Blake. CONDENSATION - Conditional Density Propagation For Visual Tracking. *International Journal of Computer Vision*, 28(1):5 – 28, 1998.
- [14] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-Based Probabilistic Tracking. In *Proc. ECCV*, pages 661 – 675, 2002.
- [15] B. Stenger, A. Thayananthan, P. H. S. Torr, and R. Cipolla. Filtering Using a Tree-Based Estimator. In *Proc. IEEE ICCV*, pages II:1063 – 1070, 2003.
- [16] J. Sullivan, A. Blake, M. Isard, and J. MacCormick. Bayesian Object Localization in Images. *International Journal of Computer Vision*, 44(2):111 – 135, 2001.
- [17] P. Tichavsky, C. Muravchick, and A. Nehorai. Posterior Cramer-Rao Bounds for Discrete-Time Nonlinear Filtering. *IEEE Transactions on Signal Processing*, 46(5):1386 – 1396, 1998.
- [18] S. D. Tran and L. S. Davis. Object Tracking at Multiple Levels of Spatial Resolutions. In *Proc. ICIAP*, pages II:1063 – 1070, 2007.
- [19] S. K. Zhou, R. Chellappa, and B. Moghaddam. Visual Tracking and Recognition Using Appearance-Adaptive Models in Particle Filters. *IEEE Transactions on Image Processing*, 13(11):1491 – 1505, November 2004.