

Dynamic Resource Allocation by Ranking SVM for Particle Filter Tracking

Changkyu Song
no5nono@postech.ac.kr
Jeany Son
jeany@postech.ac.kr
Suha Kwak
mercury3@postech.ac.kr
Bohyung Han
bhhan@postech.ac.kr

Computer Vision Lab.
Dept. of Computer Science and
Engineering
POSTECH
Pohang, Republic of Korea

Abstract

We propose a dynamic resource allocation algorithm based on Ranking Support Vector Machine (R-SVM) for particle filter tracking. We adjust the number of observations in each frame adaptively, where tracker performs measurement for a subset of particles to preserve mode locations in the posterior and allocates the rest of particles to maintain the diversity of the posterior without actual measurements. The number of measurements is determined by a ranking classifier, which evaluates the quality of each particle and counts the number of *good* ones. The ranking classifier is trained by R-SVM algorithm and universally applicable to every object and sequence because it is learned based on observation likelihoods, not image features. Our algorithm is useful to reduce observation cost and improve sampling quality in particle filtering. We integrated the proposed technique into l_1 minimization tracking based on sparse representation and validated its effectiveness with several challenging videos.

1 Introduction

Particle filtering is a Sequential Monte Carlo (SMC) technique to estimate the dynamic state given observations in a Bayesian framework. The posterior density of each step is estimated based on the prior and the measurement density functions, and is given by

$$\begin{aligned} p(\mathbf{x}_t | \mathbf{z}_{1:t}) &\propto p(\mathbf{z}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{z}_{1:t-1}) \\ &= p(\mathbf{z}_t | \mathbf{x}_t) \int p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} | \mathbf{z}_{1:t-1}) d\mathbf{x}_{t-1}, \end{aligned} \quad (1)$$

where \mathbf{x}_t ($t = 0, \dots, T$) and \mathbf{z}_t ($t = 1, \dots, T$) are the state and measurement variable, respectively. In practice, the posterior density is represented with a set of particles and their weights, $\{(\mathbf{x}_t^i, \omega_t^i)\}_{i=1:N}$, where N is the number of particles.

Although particle filter is useful to propagate a general posterior density function over time, but has critical limitations. A lot of observations are required to estimate the target state

accurately, especially when the dimensionality of the state space is high. Also, particle filter typically suffers from *loss of diversity* or *degeneracy* [1, 2]; only a small number of particles have non-trivial weights and the weights of other samples are close to zero. Therefore, it is important to maintain the diversity of particles as well as propagate the modes of the posterior for tracking performance improvement. Various algorithms have been proposed to alleviate the limitations such as resampling [1, 7], unscented transformation [14], using continuous density functions [2, 5], and so on. Note that most solutions are focused on particle diversity by increasing randomness or improving proposal distribution.

Considering the balance between accuracy and diversity of the posterior in particle filtering, it is probably a good idea to observe only a minimal number of particles if their qualities are good—in other words, they have high likelihoods, and give the rest of particles uniform weights without observations to improve sample diversity. Note that a subset of well-located particles is often sufficient to preserve the mode locations of the posterior density in visual tracking problems.

The question is how to evaluate the observation quality of each particle. In this paper, we propose an algorithm to measure and rank the quality of particles based on Ranking Support Vector Machine (R-SVM) [8]. We claim that the posterior density function can be approximated effectively with a small number of highly ranked particles, where the rank of each particle is determined by a ranking classifier trained off-line. The number of particles in each frame is typically determined by object dynamics and appearance variations. We allocate resources—samples in particle filtering context—dynamically by R-SVM. The proposed algorithm has the following advantages:

- The number of observations for the posterior estimation is determined dynamically, which may reduce computational cost and/or improve tracking accuracy. For example, the posterior can be approximated efficiently by observing a small subset of particles when the inter-frame motions and appearance variations are small. On the other hand, when the target is involved in significant geometric or photometric transformations, more resources are to be allocated for sophisticated handling of the frame.
- Our algorithm learns the quality of observations based on the likelihood vectors, whose elements correspond to patch likelihoods within target. Since the features for classification are not visual features directly available from image but likelihoods, the trained classifier can be applied to any objects and sequences without re-training.
- The diversity of the particles are improved by minimizing redundant observations and giving uniform weights to unobserved particles, which reduces degeneracy and potentially improves tracking accuracy.

We integrate the adaptive resource allocation strategy by ranking algorithm to maintain the posterior density in the particle filtering framework. Note that the advantages induced by our algorithm is more significant when the observation cost per each particle is high as in l_1 minimization tracking based on sparse representation [10].

There are several studies about adaptive resource allocation in visual tracking scenarios. The interaction groups of multiple targets are defined by the Similarity Graph (SG) and the adaptive amount of resources is allocated to each group based on the status of the groups and the associated targets [9]. Tran and Davis [16] introduce a probabilistic framework for multiple resolution tracker in both spatial and temporal dimension, and attempt to achieve the robustness and the efficiency of a tracker. An adaptive particle allocation technique is

proposed for multi-camera tracking [6], where the number of particles for each camera is determined dynamically based on the visibility of the target. Pan and Schonfeld [12] have proposed an articulated object tracking technique, which adjusts the number of particles for individual body parts and image frames within a particle filter framework.

The rest of the paper is organized as follows. Section 2 presents our tracking algorithm with dynamic resource allocation and section 3 describes the method to train the ranking classifier. The experimental results will be demonstrated in section 4.

2 Adaptive resource allocation in particle filtering

Our dynamic resource allocation strategy based on R-SVM can be applied to many general tracking algorithms in particle filtering framework. We describe the representation of the posterior density and the observation method based on the sparse representation.

2.1 Posterior density representation

The main difference in our tracking algorithm with adaptive resource allocation is that we may perform the observations only for a small subset of particles. Note that the observation cost can be reduced significantly by the partial observation strategy but the state estimation error is minimal as long as a sufficient number of *good* particles are maintained and observed. In our algorithm, the posterior density function is represented by a mixture model based on observed particles and unobserved uniformly weighted particles, which is given by

$$\begin{aligned} p(\mathbf{x}_t | \mathbf{z}_{1:t}) &\equiv \sum_{k \in \{o, u\}} \pi_t^k p_k(\mathbf{x}_t | \mathbf{z}_{1:t}) \\ &= \pi_t^o \frac{p_o(\mathbf{z}_t | \mathbf{x}_t) p_o(\mathbf{x}_t | \mathbf{z}_{1:t-1})}{\int p_o(\mathbf{z}_t | \mathbf{x}_t) p_o(\mathbf{x}_t | \mathbf{z}_{1:t-1}) d\mathbf{x}_t} + \pi_t^u u(\mathbf{x}_t) p_u(\mathbf{x}_t | \mathbf{z}_{1:t-1}) \quad (2) \\ &\approx \pi_t^o \sum_{i=1}^{N_t^o} \omega_i^o \delta(\mathbf{x}_t - \mathbf{x}_t^i) + \pi_t^u \sum_{i=1}^{N_t^u} \frac{1}{N_t^u} \delta(\mathbf{x}_t - \mathbf{x}_t^i) \quad (3) \end{aligned}$$

where $u(\mathbf{x}_t)$ is the uniform distribution over \mathbf{x}_t , and $\sum_i^{N_t^o} \omega_i^o = 1$. Note that π_t^o and π_t^u are the normalized mixture weights for observed and unobserved density, which are given by

$$\pi_t^o = \max\{\alpha, N_t^o / N\} \text{ and } \pi_t^u = 1 - \pi_t^o, \quad (4)$$

where $N = N_t^o + N_t^u$ and α is a constant. The operation of the particle filter in this paper is almost identical to Condensation algorithm [7] except the posterior representation.

The mixture model in particle filtering has been employed for visual tracking in [11, 17], where the advantage of the explicit mixture model is the effectiveness in the representation of the posterior density function; it is useful to maintain the multi-modality of the density function. In our framework, the mixture representation is even more useful since we adaptively maintain two components in the posterior density function—one is the density based on the particles with real observations and the other is the one with the particles of uniform weights. Another advantage is the potential improvement of the sample quality, especially in challenging situations; it is because the diversity of particles is increased by the uniform term in the posterior, and our algorithm may suffer less from degeneracy problem or can recover from failures.

2.2 Observation with sparse representation

The sparse representation [19] has been widely used in computer vision community recently and the l_1 minimization tracking algorithm [10] is its successful integration for visual tracking. In the l_1 minimization tracking, the appearance reconstructed by the sparse representation with the non-trivial templates is used to compute the likelihood for each particle. Suppose that the target and the trivial templates are given by $\mathbf{T} = [\mathbf{t}_1 \ \mathbf{t}_2 \ \dots \ \mathbf{t}_n] \in \mathbb{R}^{d \times n}$ and $\mathbf{I} = [\mathbf{i}_1 \ \mathbf{i}_2 \ \dots \ \mathbf{i}_d] \in \mathbb{R}^{d \times d}$, respectively. There are additional negative trivial templates to ensure the non-negativity constraint in the coefficient vector for the trivial template. Formally, an observation denoted by $\mathbf{y} \in \mathbb{R}^d$ is decomposed as

$$\mathbf{y} = [\mathbf{T} \ \mathbf{I} \ -\mathbf{I}] \begin{bmatrix} \mathbf{a} \\ \mathbf{e}^+ \\ \mathbf{e}^- \end{bmatrix} \triangleq \mathbf{B}\mathbf{c}, \quad \text{subject to } \mathbf{c} \geq 0, \quad (5)$$

where \mathbf{c} is a non-negative coefficient vector; $\mathbf{a} \in \mathbb{R}^n$ is for \mathbf{T} , and $\mathbf{e}^+, \mathbf{e}^- \in \mathbb{R}^d$ are for \mathbf{I} and $-\mathbf{I}$, respectively. The sparse representation of this problem is obtained by solving a l_1 -regularized least squares problem with regularization parameter λ , which is given by

$$\mathbf{c}^* = \arg \min_{\mathbf{c}} \|\mathbf{B}\mathbf{c} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{c}\|_1, \quad (6)$$

where $\|\cdot\|_1$ and $\|\cdot\|_2$ denote the l_1 and l_2 norm, respectively.

A good target candidate should have a small number of non-zero coefficients for trivial templates and make an accurate reconstruction with non-trivial templates only. The likelihood of a particle is computed by the distance between the observation \mathbf{y}^i for the i -th particle and the reconstruction of \mathbf{y}^i using non-trivial templates, which is given by

$$p(\mathbf{z}_t | \mathbf{x}_t^i) = \exp\left(-\frac{\|\mathbf{T}\mathbf{a}^* - \mathbf{y}^i\|^2}{\sigma^2}\right), \quad (7)$$

where \mathbf{a}^* is obtained by Eq. (6), and σ is a constant. We perform the observation based on partially overlapped 3×3 sub-blocks to generate feature vectors for R-SVM; the use of sub-blocks also helps reduce computational cost, which is discussed in Section 4.1.

2.3 Particle filtering with dynamic resource allocation

We draw N particles in each time step, and divide them into s subsets as

$$S_j = \{(\mathbf{x}_t^i, \omega_t^i)\}_{i=1:\frac{N}{s}}, \quad \text{and } j = 1, \dots, s. \quad (8)$$

The observation is performed in multiple stages—one subset in each stage, where we continue the observations until q highly ranked (top $p\%$) particles are obtained to estimate the posterior. The parameter p corresponds to 90% bounding box overlap in our training dataset, and q is determined empirically.

The likelihood is a weak feature to determine the confidence of each particle, but some weak features such as Haar-like features are employed to face detection [18] successfully by the integration in an appropriate training algorithm. We propose a learning technique to evaluate the quality of a particle based on the observation likelihoods, which is discussed in the next section.

3 Learning a ranking classifier

We predict the quality of each particle by R-SVM based on the likelihoods obtained from Eq. (7), and control the number of observations in each frame. The R-SVM trained off-line can be used to evaluate the particles for any targets and sequences universally since training is performed with the particle likelihoods, not with appearances such as image features. In this section, we review the R-SVM algorithm, and describe how training data are collected and the ranking classifier is learned for visual tracking.

3.1 Ranking Support Vector Machine

The R-SVM [8] learns a ranking function; the goal of the R-SVM is to construct a ranking model based on the partial order of training data. The model is employed to report the ranks—instead of the class labels—of individual testing data.

Denote by \mathbf{d}_k ($k = 1, \dots, M$) a vector, and by r_k a rank of \mathbf{d}_k . Consider a ranking function f satisfying the following condition:

$$\mathbf{d}_{k_1} \prec \mathbf{d}_{k_2} \iff f(\mathbf{d}_{k_1}; \mathbf{w}) < f(\mathbf{d}_{k_2}; \mathbf{w}), \quad (9)$$

where \prec denotes partial order between two vectors. We order the input vectors based on the projection of kernelized data using the weight vector \mathbf{w} , and the ranking function is actually the dot product of $\Phi(\mathbf{d}_k)$ and \mathbf{w} , where $\Phi(\cdot)$ is a kernel function. In case of the linear R-SVM, the ranking relation is represented simply by

$$\mathbf{d}_{k_1} \prec \mathbf{d}_{k_2} \iff \mathbf{d}_{k_1}^\top \mathbf{w} < \mathbf{d}_{k_2}^\top \mathbf{w}. \quad (10)$$

Training R-SVM is the optimization procedure to maximize the margins of the support vectors and minimize the error penalty. The property in Eq. (9) or (10) can be realized by introducing a pairwise SVM regularization term. The objective function is given by

$$\min_{\mathbf{w}, \xi} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{k_1, k_2} \xi_{k_1, k_2} \right\}, \quad (11)$$

subject to

$$f(\mathbf{d}_{k_1}; \mathbf{w}) - f(\mathbf{d}_{k_2}; \mathbf{w}) \geq 1 - \xi_{k_1, k_2}, \quad \forall (\mathbf{d}_{k_1}, \mathbf{d}_{k_2}) \in \mathbf{r} \quad (12)$$

$$\xi_{k_1, k_2} \geq 0, \quad \forall k_1, k_2, \quad (13)$$

where ξ_{k_1, k_2} is a non-negative slack variable, \mathbf{r} is a set of ranks of given data and C is a trade-off parameter between margin size and training error. In testing phase, the rank of a new data is obtained by the ranking function with trained projection direction.

3.2 Learning particle quality for visual tracking

We employ R-SVM to evaluate the quality of each sample in particle filter tracking, so the training data need to be constructed based on realistic tracking results. The feature vectors for our R-SVM are constructed based on the likelihoods obtained from the regions associated with particles. Instead of observing the entire region defined by a particle, we divide the region into partially overlapping small blocks and perform the measurement for each block. There are a few crucial reasons to divide the regions into small pieces for observations. First, multi-block representation is useful to handle the multi-modality of the target

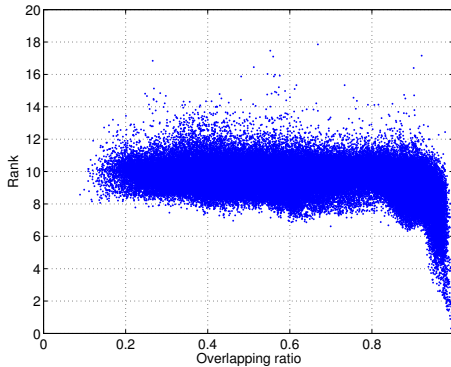


Figure 1: Performance evaluation of R-SVM

appearance, which should be considered to learn the R-SVM. Second, the collection of the partial observations are less sensitive to partial occlusions and noises than the total observations. Third, the observation cost may be reduced significantly depending on the choice of the measurement function, *e.g.*, l_1 minimization via sparse representation.¹

In our algorithm, the target and the candidate region are divided into partially overlapped 3×3 blocks. Let $\{\mathbf{b}_m\}_{m=1:9}$ be the blocks and $l(\mathbf{b}_m)$ be the likelihood of the m -th block. The 9-dimensional feature vector \mathbf{d}_k is constructed for training the R-SVM classifier as

$$\mathbf{d}_k = [l(\mathbf{b}_1), l(\mathbf{b}_2), \dots, l(\mathbf{b}_9)]^\top, \quad \mathbf{b}_m \in \mathbb{R}^\ell, \quad (14)$$

where ℓ is the number of pixels in a block. The quality of a particle is measured by the bounding box overlapping ratio between groundtruth and candidate regions. For training purpose, the rank of each particle is given by

$$r_k = \tau(\beta o(\mathbf{x}^k)), \quad (15)$$

where $\tau(\cdot)$ is a monotonic step function, $o(\mathbf{x}^k)$ is the overlapping ratio between the groundtruth and the k -th particle, and β is a constant. We collect a number of feature vectors \mathbf{d}_k and the corresponding ranks r_k from real tracking results, and train the R-SVM based on the objective function in Eq. (11).

The absolute values of particle likelihoods are typically not so meaningful to identify the quality of particles because the likelihoods typically decrease over time but the posterior can still be represented effectively with relative weights of particles. However, the use of tracking algorithms with adaptive appearance modeling can ameliorate such problem. In practice, the trained R-SVM for our tracking algorithm can classify good and bad particles with the likelihood-based feature vectors, which is illustrated in Figure 1.

4 Experiment

In this section, we first present the simulation results for particle filter tracking with dynamic resource allocation. After that, comparative tracking results in real videos are demonstrated for the performance evaluation of the proposed algorithm.

¹The details will be discussed in Section 4.1.

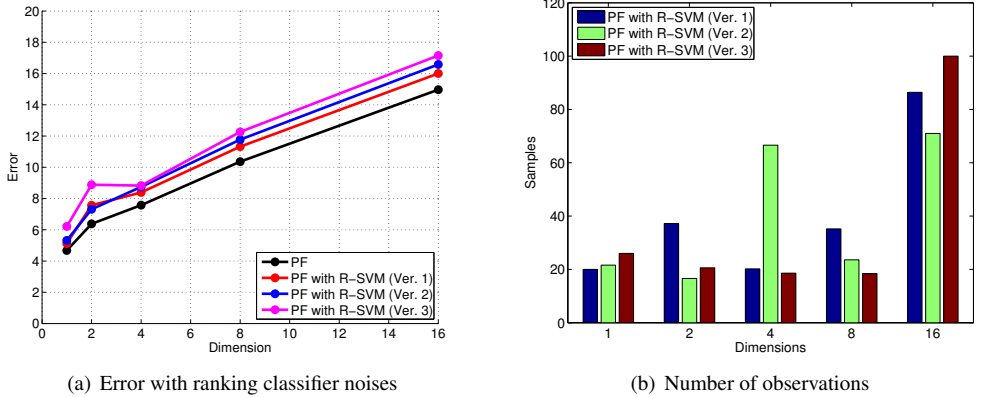


Figure 2: Comparisons between Particle Filter (PF) with 1000 particles and our method. Among our method, the ranking prediction error is smallest in Ver. 1 and largest in Ver. 3. Note that the average number of observations in our algorithm is much less than PF.

4.1 Simulations

Performance of adaptive resource allocation We compared the accuracies of condensation algorithm [7] and our method by simulation. The process and measurement functions for the simulation are from Section VI in [1], and the evaluation was conducted for multiple dimensions. We assumed that the prediction error in the R-SVM is normally distributed and tested the performance of our tracking algorithm with three different levels of standard deviations—0, 3, and 6—in the R-SVM predictions. The simulation results are summarized in Figure 2, which suggests that our tracking algorithm with adaptive resource allocation is comparable to particle filtering with much less observations and little sensitive to the noises of the learned ranking classifiers.

Computational cost reduction by target partitioning We divide the target area into 3×3 partially overlapping sub-blocks, and use the likelihood of each block to construct a 9D feature vector for the input to the R-SVM. Also, note that the likelihood of each particle is computed by the aggregation of sub-block likelihoods. When the likelihood of each sample is computed by sparse representation as in [10], we can achieve additional speed-up by the observation based on the partitioned target because the observation cost of the l_1 minimization tracking is affected by the number of dimensions significantly. Figure 3 illustrates the benefit of several versions of target partitioning; our tracking algorithm with dynamic resource allocation would be much faster than the original l_1 minimization tracking. Note that the observation in the l_1 minimization tracking is computationally very expensive.

4.2 Tracking results and performance evaluation

We evaluated our l_1 minimization tracking algorithm with dynamic resource allocation by R-SVM (LITRSVM) using real video sequences and compared it with the original l_1 tracking algorithm (LIT) [10], incremental subspace learning (IVT) [13], and three boosting-based trackers—on-line boosting (Boost) [4], semi-supervised on-line boosting (SBoost) [3] and beyond semi-supervised boosting (BSBoost) [15]. We also tested LIT with several different numbers of particles.

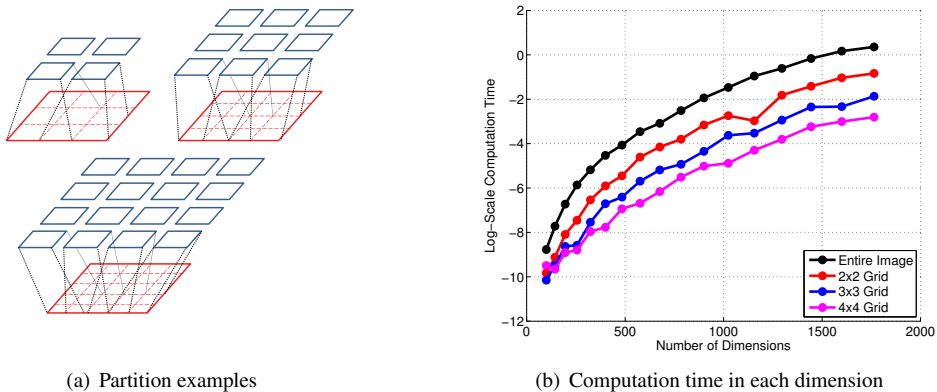


Figure 3: Effect of target partitioning in computational cost

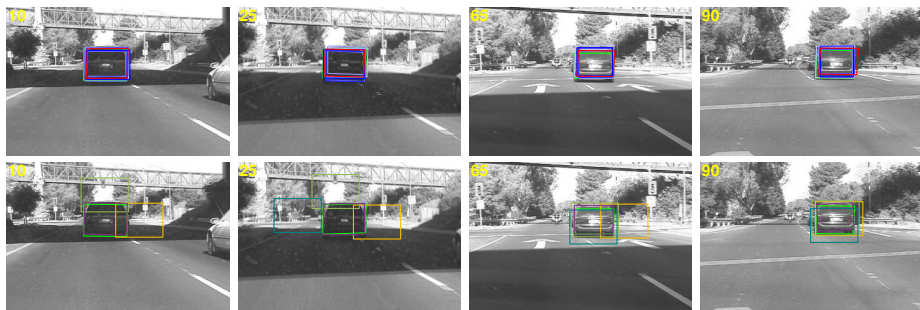


Figure 4: Tracking results of the *car* sequence. The groundtruth is given by green rectangles. **(Top)** L1TRSVM (red) and L1T with different number of particles—30, 60, 100, 200, 400, and 600 (blue with different intensities; 600 is darkest). **(Bottom)** IVT (purple), Boost (orange), SBoost (light green), and BSBoost (dark green).

The first sequence—*car* sequence—is for tracking a moving vehicle in the presence of significant illumination changes. Both our method and L1T successfully track the target while boosting-based trackers lose the target as illustrated in Figure 4 and 6(a). Interestingly, the performance of L1T is hardly affected by the number of particles—tracking results with only 30 particles are almost identical to the results with 600 particles. The performance L1TRSVM is comparable to L1T regardless of the number of particles. However, our method consistently utilizes a small number of particles and the number is determined by R-SVM adaptively and automatically; this fact is presented in Figure 6(a).

The next video, *campus* sequence, involves two occlusions—one is mild and the other is very severe, and the same set of tracking algorithms are applied to this sequence. L1TRSVM is successful but all other tracking algorithms including all versions of L1T fail to track the target after occlusions as in Figure 5 and 6(b). It is probably because our algorithm divides the target into 3×3 sub-blocks and computes the likelihood for each particle based on the distance between the block-wise observations and reconstructions while L1T utilizes the reconstructed image of an entire observation. Therefore, our algorithm is more robust to partial occlusions than the original L1T. The diversity of particles also helps escape from local optima and find the target when it reappears after total occlusion. The benefit of our

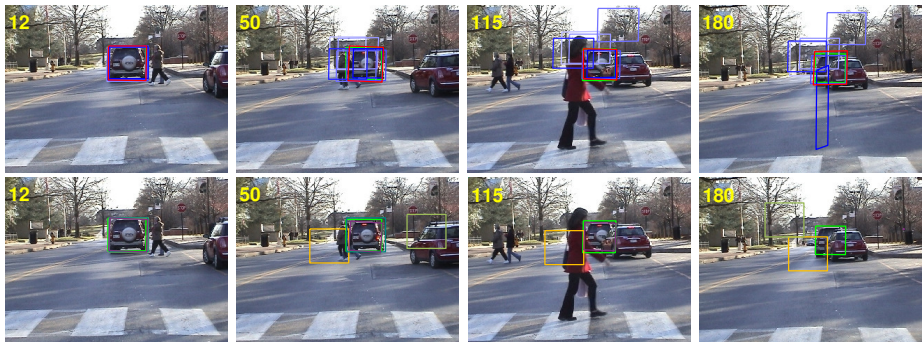


Figure 5: Tracking results of the *campus* sequence. **(Top)** LITRSVM and LIT with different number of particles. **(Bottom)** IVT, Boost, SBoost, and BSBoost.

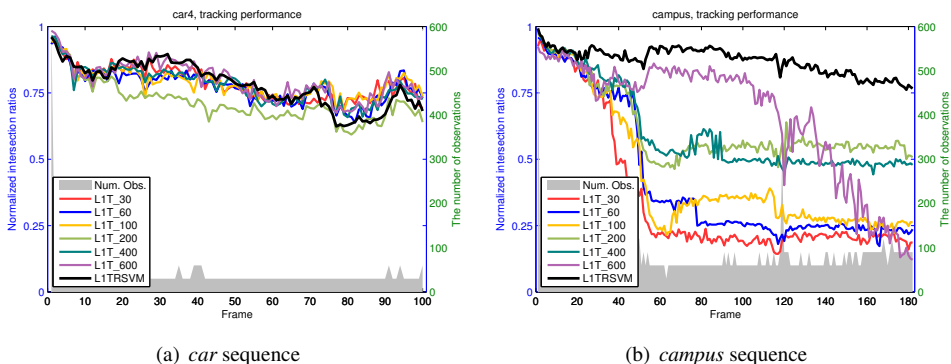


Figure 6: Normalized intersection ratios over time of LITRSVM and LIT with different numbers of particles. **(a)** Accuracies of all algorithms are comparable. However, LITRSVM adjusts the number of observations and only 30 observations are made in most frames. **(b)** Only LITRSVM is successful after the severe occlusion around frame 120. The number of observations in LITRSVM is small but temporarily increased during the severe occlusion.

dynamic allocation technique is also significant in this sequence; in most of the frames, our algorithm makes a small number of observations as illustrated in Figure 6(b).

The last sequence (*CAVIAR—OneShopOneWait1cor*) is for tracking a non-vehicle object, which is a walking human with partial occlusions. LITRSVM shows decent performance with a small number of observations.

The summary of quantitative results are presented in Table 1, and it suggests that our algorithm has better or comparable performance with only a small number of observations (given inside parentheses) by dynamic resource allocation based on R-SVM, whereas the number of particles in LIT and IVT is 600. Note that the accuracies of our algorithm are most consistent.

Table 1: The summary of average normalized intersection ratios in each algorithm

	Ours	LIT_600	IVT	Boost	SBoost	BSBoost
<i>car</i>	0.78 (38.10)	0.80	0.85	0.25	0.36	0.21
<i>campus</i>	0.87 (74.51)	0.66	0.54	0.18	0.37	0.37
<i>CAVIAR</i>	0.72 (33.43)	0.79	0.35	0.61	0.50	0.12

5 Conclusion

We present a dynamic resource allocation algorithm based on R-SVM for particle filter tracking, where the number of observations is adaptive and the unobserved particles are used to increase the diversity of particles. The R-SVM is learned with observation likelihoods so it is applicable to any objects and sequences once it is trained. Our algorithm is useful to reduce observation cost and improve sample quality in particle filtering. The proposed algorithm is integrated into an l_1 minimization tracker, and shows superior or comparable results with only a small subset of observations.

Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (2010-0003496).

References

- [1] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking. *IEEE Trans. Signal Process.*, 50(2):174–189, 2002.
- [2] A. Doucet, N. de Freitas, and Neil Gordon. *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.
- [3] C. Leistner H. Grabner and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *In Proc. European Conf. on Computer Vision, Marseille, France, 2008*.
- [4] M. Grabner H. Grabner and H. Bischof. Real-time tracking via on-line boosting. In *Proc. British Machine Vision Conference, 2006*.
- [5] B. Han, Y. Zhu, D. Comaniciu, and L.S. Davis. Visual tracking by continuous density propagation in sequential bayesian filtering framework. *IEEE Trans. Pattern Anal. Machine Intell.*, 31(5):919–930, 2009.
- [6] B. Han, S.-W. Joo, and L.S. Davis. Multi-camera tracking with adaptive resource allocation. *Intl. J. of Computer Vision*, 91(1):45–58, 2011.
- [7] M. Isard and A. Blake. Condensation - Conditional density propagation for visual tracking. *Intl. J. of Computer Vision*, 29(1), 1998.
- [8] T. Joachims. Optimizing search engines using clickthrough data. In *Proc. of the 8th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pages 133–142, 2002.
- [9] A. Kembhavi, W.R. Schwartz, and L.S. Davis. Resource allocation for tracking multiple targets using particle filters. In *Workshop on Visual Surveillance, in conjunction with ECCV 2008, 2008*.

- [10] X. Mei and H. Ling. Robust visual tracking using l_1 minimization. In *Proc. 12th Intl. Conf. on Computer Vision*, Kyoto, Japan, 2009.
- [11] K. Okuma, A. Taleghani, N. de Freitas, J.J. Little, and D.G. Lowe. A boosted particle filter: Multitarget detection and tracking. In *Proc. European Conf. on Computer Vision*, Prague, Czech Republic, May 2004.
- [12] P. Pan and D. Schonfeld. Adaptive resource allocation in particle filtering for articulated object tracking. In *Proc. IEEE Conf. on Acoustics, Speech and Signal Processing*, 2008.
- [13] D. Ross, J. Lim, and M. Yang. Adaptive probabilistic visual tracking with incremental subspace update. In *Proc. European Conf. on Computer Vision*, Prague, Czech, volume II, pages 470–482, May 2004.
- [14] Y. Rui and Y. Chen. Better proposal distributions: Object tracking using unscented particle filter. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Kauai, HI, volume II, pages 786–793, 2001.
- [15] H. Grabner S. Stalder and L. van Gool. Beyond semi-supervised tracking: Tracking should be as simple as detection, but not simpler than recognition. In *OLCV 09: 3rd On-line learning for Computer Vision Workshop*, Kyoto, Japan, 2009.
- [16] S.D. Tran and L.S. Davis. Object tracking at multiple levels of spatial resolutions. In *Proc. of Intl. Conf. on Image Analysis and Processing*, pages 149–154, 2007.
- [17] J. Vermaak, A. Doucet, and P. Perez. Maintaining multi-modality through mixture tracking. In *Proc. 9th Intl. Conf. on Computer Vision*, Nice, France, volume II, 2003.
- [18] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Kauai, HI, pages 511–518, 2001.
- [19] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, and Yi Ma. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Machine Intell.*, 31(2):210–227, 2009.