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.
Department of Computer Science and Engineering,
POSTECH
Pohang, Republic of Korea

Particle filtering is a Sequential Monte Carlo (SMC) technique to estimate the dynamic state given observations in a Bayesian framework. 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 a high dimensional state space is involved. Also, particle filter typically suffers from *loss of diversity* or *degeneracy*; 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 perfomance improvement.

In this context, we propose a dynamic resource allocation algorithm based on Ranking Support Vector Machine (R-SVM) [1] for particle filter tracking. We adjust the number of observations in each frame adaptively and automatically, where tracker performs measurement for a subset of highly ranked particles in likelihood to preserve mode locations in the posterior and allocates the rest of particles to maintain the diversity of the posterior without actual measurements.

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. We represent the posterior density function with a mixture model based on observed particles and unobserved uniformly weighted ones, which is given by

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

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

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

where $N = N_t^o + N_t^u$ and α is a constant. The mixture representation of the posterior in Eq. (1) enhances the diversity of particles by the uniform term, so our algorithm may suffer less from degeneracy problem or can recover from failures. Also, our framework speeds up particle filtering by observing only particles with high observation likelihoods; this advantage is more significant when the measurement cost per each particle is high as in l_1 minimization tracking based on sparse representation [2].

The posterior density function of Eq. (1) is estimated by the following procedure. We draw N particles in each time step, and divide them into s subsets as

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

The observation is performed in multiple stages—one subset in each stage, where we continue the observations to estimate the posterior until q highly ranked (top p%) particles are obtained. We employ l_1 minimization technique to obtain the observations via sparse representation, and the quality of each particle is predicted by R-SVM based on likelihoods.

The R-SVM trained off-line can be used to evaluate the particles from any targets and sequences *universally* since training is performed with the particle likelihoods, not with appearances such as image features. The

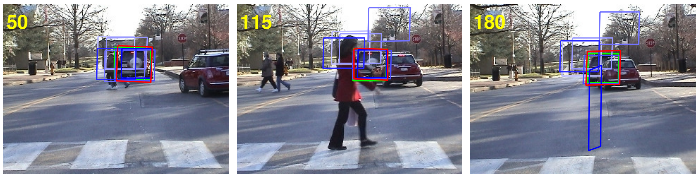

Figure 1: Tracking results of the *campus* sequence. The groundtruth is given by green rectangles. Ours are with red rectangles, and l_1 minimization tracking with different number of particles—30, 60, 100, 200, 400, and 600 are represented by blue rectangles with different intensities.

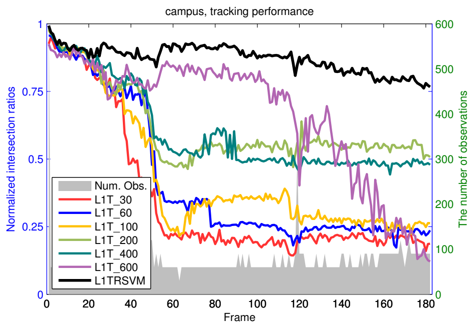

Figure 2: Normalized intersection ratios over time for ours (L1TRSVM) and l_1 minimization tracking with different numbers of particles in the *campus* sequence.

feature vectors for the classification 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 3×3 small blocks and perform the measurement for each block. 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

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

where d_b is the number of pixels in a block. The quality (rank) of a particle is measured by the bounding box overlapping ratio between groundtruth and candidate regions; the rank of each particle is defined by

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

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 R-SVM.

We validated our particle ranking technique, which is integrated into l_1 minimization tracking, with several challenging videos, e.g., Figure 1. More experiments and comparisons with other tracking algorithms in different sequences are illustrated in our paper. To summary, our algorithm showed superior or comparable results with only a small subset of observations as presented in Figure 2; it is useful to reduce observation cost and improve sample quality in particle filtering.

[1] 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.

[2] X. Mei and H. Ling. Robust visual tracking using l_1 minimization. In *Proc. 12th Intl. Conf. on Computer Vision,* Kyoto, Japan, 2009.