

# COM-711 Homework 5

October 28, 2011

This assignment is *due December 9<sup>th</sup>, 2011*. The goal of this assignment is to implement a particle filter as discussed in class, and apply it to track objects in three video sequences. Turn in *your code* [7 pts] and your results on each of the three video sequences electronically to Kevin Smith at [kevin.smith@lmc.biol.ethz.ch](mailto:kevin.smith@lmc.biol.ethz.ch).



Sequence 1. Red toy car [1 pt]

Sequence 2. Girl in Pink [1 pt]

Sequence 3. Head Tracking [1 pt]

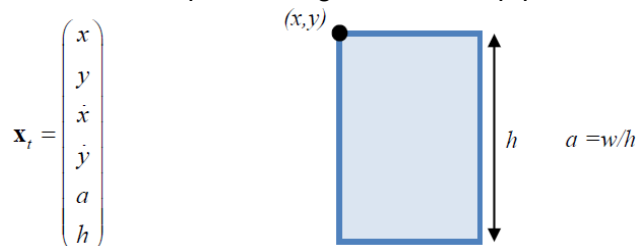
In sequence 1, the goal is to track the small red toy car. In sequence 2, the goal is to track the girl in pink. In sequence 3, the goal is to track the head of the person on the left. You will not lose points for tracking errors in your results sequences, so long as it is evident that you have implemented the particle filter correctly. The raw image sequences are available on the course web site,

<http://cvlab.epfl.ch/teaching/topics/files/sequence1.zip>

<http://cvlab.epfl.ch/teaching/topics/files/sequence2.zip>

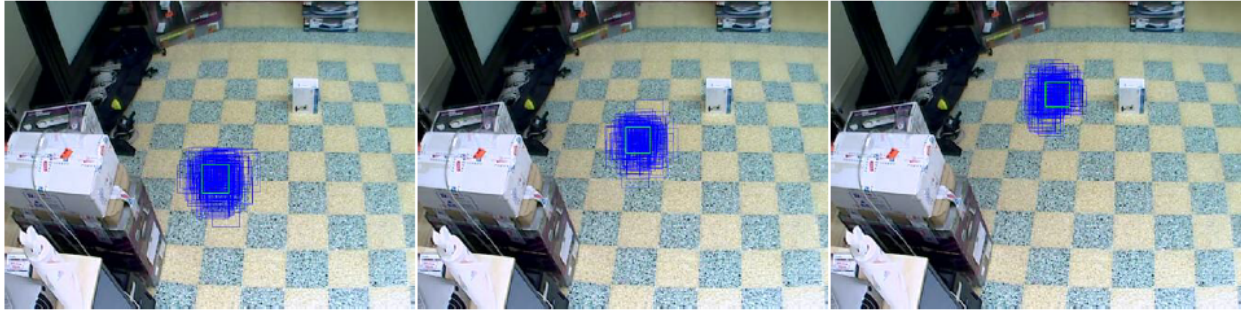
<http://cvlab.epfl.ch/teaching/topics/files/sequence3.zip>.

Each of the sequences contains challenging problems such as occlusion, abrupt motion changes, and distracting backgrounds. Your particle filter should estimate the state of the object to be tracked over the entire sequence,  $X_t = \{x_1, x_2, \dots, x_t, \dots, x_T\}$ , where  $x_t$  is the state of the object at time  $t$ . The state space at a given time step parametrizes a bounding box,



which includes the  $(x, y)$  location of the upper left corner of the bounding box in the image, the velocity terms, the aspect ratio  $a$ , and the height  $h$ .

Your resulting video sequences should display the state of each particle (either as a point or as a bounding box), as well as the final estimated state, as shown below.



Matlab skeleton code and supplementary functions have been provided. Functions are provided to accomplish tasks such as reading files, displaying images, initializing the tracker, and writing results to file. Hints are also provided in the comments of the skeleton code (you may find the hints useful if you implement the particle filter in another language). These materials can be downloaded at <http://cvlab.epfl.ch/teaching/topics/files/materials5.zip>.

To implement a particle filter, you will need to accomplish the following steps:

1. **Represent the particle set.** Each particle should consist of a state estimate and associated weight. The skeleton code represents particles using a Matlab structure.
2. **Perform resampling.** This step is necessary to prevent the degeneracy problem.
3. **Model the dynamics.** Use a linear dynamic model with additive Gaussian noise to predict the new state given the previous state. It will be necessary to tune the covariance matrix  $\mathbf{Q}$  to determine the “spread” of the particles.
4. **Define an observation model.** The observation model computes the likelihood that observed data from the image supports the estimated state hypothesized by a particular particle. In the provided skeleton code, this is done by comparing a color histogram extracted from the particle’s bounding box to a known color model extracted beforehand. The included function LABhistogram.m is used to extract the LAB histogram. The skeleton code models the likelihood as  $p(z_t|x_t) = \exp(-\lambda \text{dist}(h, h^*))$ , where  $\text{dist}$  is the KL divergence of two histograms,  $h$  is a color histogram corresponding to the observation  $z_t$  associated to particle  $x_t$ , and  $h^*$  is a known color model. The parameter  $\lambda$  adjusts the influence of the likelihood function.
5. **Reweight the particle set** using the likelihoods obtained from the observation model.
6. **Infer a solution.** After performing all of the particle filtering step, a final solution must be extracted from the approximate distribution represented by the particle set. I used the mean state vector computed from all particles.

When implementing the particle filter, consider the following questions (for yourself): What happens to performance if the resampling step is removed? How does changing the parameter  $\lambda$  in the observation model change the performance? What happens if the velocity terms are removed from the state vector? How do the values of the covariance matrix  $\mathbf{Q}$  influence the performance of the particle filter? What are some other ways to infer a solution from the particle set?

## Bonus Points

There is a possibility to have bonus points added to the total score of your assignment for accomplishing the following tasks (please notify me in your email if you have attempted either):

1. [+2 pts] Automatically initialize the tracker (eg. using background subtraction, a detector, etc)
2. [+2 pts] Use an alternative observation model (eg. adapt the color histogram model at each time step, model different regions of the object, etc)