# Video Processing – HW3

* 1. Posterior – the probability of the event given the observed data
  2. Prior – the probability of the event before observing the data
  3. Likelihood – the probability of the observed data given the event

This formula represents Bayes’ theorem. We want to get an estimation based on our initial belief (prior probability) and on the observed data likelihood.

Eventually we get the posterior probability which improves our initial belief using the observed data.

1. Measurement steps explanations:
   1. We’re using the histogram to score the patches so we could determine the similarity between the patches and the target object. We have the original object (target object) histogram and we’re comparing it to the histogram of the current patch to measure similarity between them.

\*The histogram of a patch represents the color distribution of it.

Pros:

* Computing histogram is easy to implement and efficient computationally.
* Can be easily applied on different image sizes.
* Distribution of color values (rather than absolute values) allow using histograms be more robust to illumination changes.

Cons:

* It will be harder to distinguish between objects with similar color distributions. Different object can have similar color distribution with different spatial arrangement, so histogram won’t be able to distinguish between those objects.
* Noise can influence the histogram results significantly and give wrong results.
  1. Using SSD to calculate distance between patches is not a good idea because it is computationally costly, and it can give us wrong results. When calculating SSD can include more than one object. For example, some object and the background. In this case, if between two patches the background is different, the SSD results may confuse us to this there is no match between those patches.
  2. One way that could help us to compare between patches, is using neural networks dedicated for this task.
     1. Pros: The results would probably be very accurate.
     2. Cons: Computationally it would probably be too expensive.

1. Particle filter could work when the tracked object changes its scale or its viewpoint, but there some modifications required for it.

Regarding scale changing, we’ve seen on the internet that there is an approach called scale-space particle filtering which by applying multiple particle filters at different scales and adjusting the weights, it can overcome the changed scale challenge.

Changing viewpoint can be treated with an approach called view-based particle filtering. Similarly, to the scale change, here it is required to use multiple particle filters at different viewpoints.

1. According to the internet, a possible way to determine when to update the patch of the object is use a tracking error metric which measures difference between predict and actual location of the object. If the error is too big, then one can know the patch should be updated. It is hard to know what the best way is to determine the new patch and it could vary according to the task itself (the content in the frames themselves). One way we can think of is taking the mean of the last “x” patches and use this information to get a more stable patch.

And regarding template update, a common approach is to use an adaptive update mechanism. The idea behind the adaptive update mechanism is to update the template only when the tracking performance drops below a certain threshold.