EX3\_203200480\_320521461

Part A

1. First we will explain what is posterior, prior and likelihood:
2. **Posterior**: a posterior refers to the chances that event B will occur given that event A happened.

For example :

Let A be the amount of time the student spent on studying **measured \ known variable**

Let B be the grades of a student **hidden variable that needs to be estimated**

Obviously there is a relation between the two but A is the variable that one can measure and B is the variable that we want to estimate.

So if we assume that we want to estimate the grades of a student given amount of time spent on studying then the posterior will be P(B|A)

Or in simple words what is the probability that a person got grade A given that he spent B hours studying.

1. **Prior**: is the probability distribution that we can make about our believes about a certain event before the event has occurred and some evidence were taken to account, the prior is defined by P(B).

For example it resembles the task of estimating the time a person grade without knowing how much time he spent studying. And therefore we can estimate a persons grade based on “belief” that relies on previous data distribution (i.e the grades students got last semsters)

1. Likelihood – this is the evaluation of a statistical model to a sample of the data for the given of the unknown parameters the model tried to estimate.

In our example it resembles the task that we already know a person grade and we need to determine how likely is that the same person spent B hours studying.

The likelihood is defined as :P(A|B)

Now we can use the Bayes therom and get the following eq :

2.

1. We use the histogram to score the patches because histograms are easy to implement and fast to compute ,it’s a basic concept in computer vision and is known method in images similarity.

The pro of this method is that it is fast to compute (good for real time systems), and if the object we are tracking is very different than its surrounding (a white ship on the blue ocean) we can really that the histogram patch with ship will be very different than all other patches in the image, and we can detect the ship easy.

The cons is that if the score is not affected by visual features on the patch, patches with very different visual features but same colors (intensity values) will get similar scores (although the chances of that happing are quit small).

1. It is possible to SSD to compare frames in the video. But SSD is proven to be less effective than compering histogram when there is a task that calculates area loss (as in compering two bounded objects), meaning two images can have a big difference when it comes to SSD but still be visually very similar.

For example if we wish to track a person going through different kind of scenery like going from the street to the park the SSD in this case will be large and could make our tracking algo fail. We can use other comparison tools such as histogram of the gradient (SIFT) to compare 2 images meaning histogram is not the best comparison tool but Histogram is better than SSD.

1. Another method to compare and encode visual features is SIFT which is to compare the histogram of the gradients of the divided interest areas in the two images.

The pros of this method is that it is insensitive to illumination changes and scaling difference

The cons of this method is that it is more complex to calculate rather than just compering the interest areas histogram.

1. Using particle filter can work even if the scale of object changes from frame to frame.

But for that to work some changes need to be applied in order for this to work, we will need to change the size of the rectangle bounding box that we use in order to compare frames according to scale change we face in the video, which is quit trivial if we know that the object size is going to change then so does the compared area.

As for the viewpoint change, since our method of matching the two images is to compare their histograms than we are more likely to fail

1. We suggest to create an average template when after every time step we would like to update the average template values, since we use the smoothness assumption than the relevant patch wont change to much from frame to frame so in every frame we update the average template based on the current frame prediction that way our compared histogram will update itself with the changes in the light, viewpoint, …

We could update the patch after each frame, and we will determine the new patch based on all previous data we can devis an equation were the update rule is:

And preform various tries to find the best