**HW3: Particle Filter Tracking**

**Part A:**

1. The formula 𝑝𝑜𝑠𝑡𝑒𝑟𝑖𝑜𝑟 ∝ 𝑝𝑟𝑖𝑜𝑟 × 𝑙𝑖𝑘𝑒𝑙𝑖h𝑜𝑜𝑑 represents the Bayes' rule.  
   In Bayesian inference, we start with a prior probability distribution that represents our beliefs about a certain event or parameter before observing any data. Then, we observe new evidence, represented by the likelihood function, which tells us how probable is the observed data given a certain parameter or hypothesis. Using Bayes' rule, we can update our prior beliefs based on this new evidence and calculate the posterior probability distribution, which represents our beliefs after observing the data.  
   In the context of the Kalman filter or particle filter, the posterior represents our estimate of the true state of a system given the observations and the model. The prior represents our estimate of the state before observing the new data, based on the previous state and the model dynamics. The likelihood function represents the probability of observing the new data given the current state estimate and the observation model.
2. Per section:
   1. We use the histogram to score the patches because it is a robust way to compare the distribution of pixel values between two patches. The histogram is a summary of the pixel values in a patch, and it is not affected by small changes in the brightness or contrast of the patch (after quantization). This makes it a good way to compare patches that may be slightly different, but that still represent the same object. We assume the background has the same texture and color, while the object we want to track can have different posters, the position of each pixel in the patch doesn’t affect the histogram.

Pros:

* + - It is a robust way to compare patches that may be slightly different.
    - It is computationally efficient.

Cons:

* + - It can be sensitive to changes in the brightness or contrast of the patch.
    - It can be inaccurate when to comparing patches of different sizes.
  1. We did not use the SSD between the patches because it is not a robust way to compare patches that may be slightly different in size or color. The SSD is sensitive to changes in the brightness and contrast of the patch, and it can also be affected by noise. The histogram is a more robust way to compare patches that may be slightly different.
  2. Yes, there are other methods to compare patches. For example, using a pre trained neural network to get a meaningful embedding for the patch, and compare between embedding instead of between pixels. The pro of using a NN is that it won’t be affected by minor brightness or sizes changes of the patch. The con is that NN might require more HW like GPUs.

1. Yes, particle filter can work when the tracked object changes its scale or viewpoint.
   1. Scale change- When an object changes scale, the particle filter can be made more robust by adding the scale parameter to the state vector
   2. Viewpoint change- When comparing histograms of patches different viewpoints won’t be supported, for example if the back of the shirt has a completely different color. Using a different comparing method can solve that issue, for example a NN.
2. A possible way to update the representation of the object to solve changes in brightness or viewpoints is to consistently update the representation on each frame where our confidence of the state is high, from both the model and the patch comparisons. We can define a threshold for high confidence. We think updating the representation often can help handling those changes, while not require more computation because we already calculate the representation of the patch for every patch.