Exercise 4
Math foundation of
computer graphics and vision

Amrollah Seifoddini

Part 1)

Q1: In the dual quaternion formula, real numbers represent rotation and dual numbers (with epsilon) represents translation. It codes the rigid transformation the same way that quaternion represents rotation.

Q2: The advantage is that, dual quaternion generates valid transformation (comparing to averaging) and it has constant speed, shortest path (falls inside manifold) and coordinate invariance features which is desired for blending.

Q3: For 2D rotations, quaternion based shorted path method will result in discontinues mesh at some middle points of rotations compared to linear blend skinning which just average them and guarantees smoothness. This problem is because there is one quaternion needed for every rotation.

Q4:

$$\hat{q}^{t} = e^{t\log(\hat{q})} = e^{\frac{t}{2}\hat{s}\hat{\theta}} \quad \Bigg\} = e^{\frac{t}{2}\hat{D}} = \cos\left(\tfrac{t}{2}\|\hat{D}\|\right) + \frac{\hat{D}}{\|\hat{D}\|}\sin\left(\tfrac{t}{2}\|\hat{D}\|\right) \quad (1)$$

then $$\hat{s}\hat{\theta} = \theta_0 s_0 + \varepsilon(\theta_0 s_\varepsilon + \theta_\varepsilon s_0) \overset{\text{we name}}{=\!=} \hat{D}$$

$$\|\hat{D}\| = \|\theta_0 s_0\| + \varepsilon\frac{\langle \theta_0 s_0, (\theta_0 s_\varepsilon + \theta_\varepsilon s_0)\rangle}{\|\theta_0 s_0\|} = \hat{\theta}$$

by replacing $\|\hat{D}\| \to \hat{\theta}$ in (1) $\Downarrow$

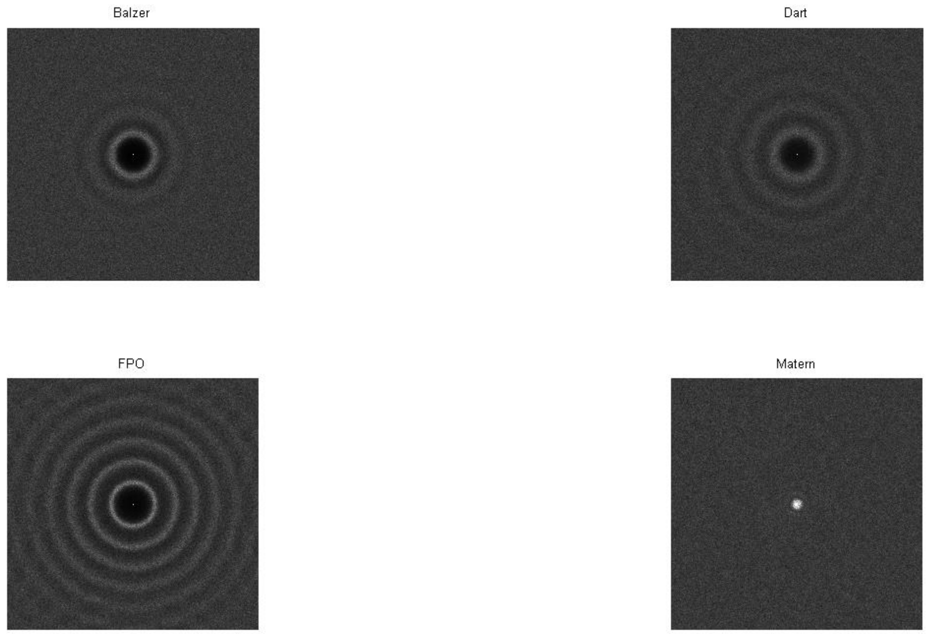
$$\boxed{\cos\left(\tfrac{t}{2}\hat{\theta}\right) + \hat{s}\sin\left(\tfrac{t}{2}\hat{\theta}\right)}$$

Q5: For coding the rotation and translation in x-y plane, we can multiply two dual quaternions. The first one is a pure rotation (with s^=0) around the z-axis, and the second one is a pure displacement (with rotation^=0) along an axis which is toward the destination point.

Part 2)

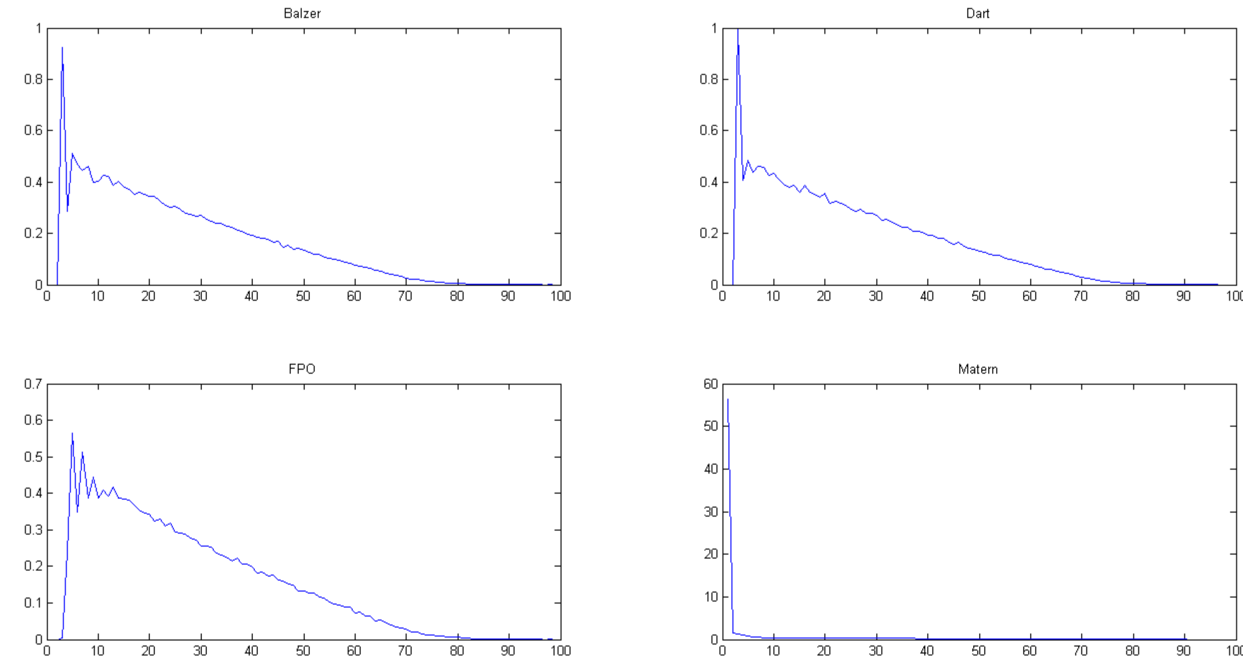
Task 1:

For this task, I followed the guidelines of exercise sheet. Here are the result for four sample patterns:

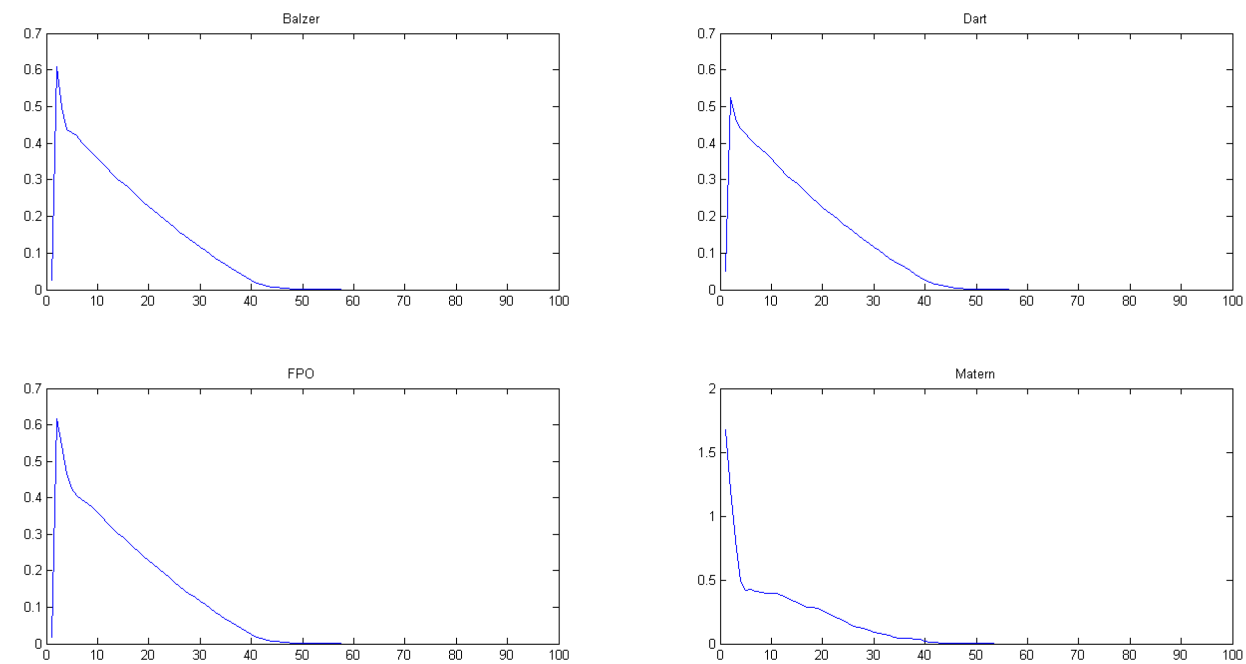


Task 2)

For this task, I again followed the guidelines of exercise sheet. Here are the result for four sample patterns: (I had to change parameters of model to get following images. Parameters are in code)



Or yet another result with slightly different parameters (last parameters are in the code):



Task 3)

Q2: For PCF we do not need to average the result among different point pattern instances, because we are already using a smoothing function (Gaussian) while computing the PCF values. So, the result graphs are indistinguishable for different instances of the same point pattern. But for periodograms we have to average over several instances because they all produce slightly different periodograms. This is not the case for PCF.

Q3: PCF's are sufficient to describe the point pattern, even though they are one-dimensional. That's because they are taking into account distances between all pairs of points in the pattern. Then, generate a density distribution over all distances and normalize the result. So, we have the same of information that we could get by periodograms for a point pattern. In peridogram it's like we are moving a Driac function with our point pattern over the image, and in PCF we are using different windows (distances) to count the sample points which fall into that. Then we generate the distribution of those distances. We can see that some distances are more frequent than others, and those are the wave bands in periodogram.