ICBV – HW2

Brahan Wassan, Dan Zlotnikov

**Question 1 – Edge Detection**

**Section A**

Given

And the following Edge Detectors:

= edge dector based on the local maximum of

= Edge detector based on finding the ZC of the Laplacian

Compute analytically the edge points detected by each of the edge detectors:

For E1:

Eval :

Now we want to find the extremum points, we got an expression that depends only on y, so we can ignore x, in further computations:

So our maximum values are for

Now lets evaluate the second derivative, at the point we found:

**Question 2 – Parametric Curves**

**Section A**

The parametric equation of a straight line in 3D is

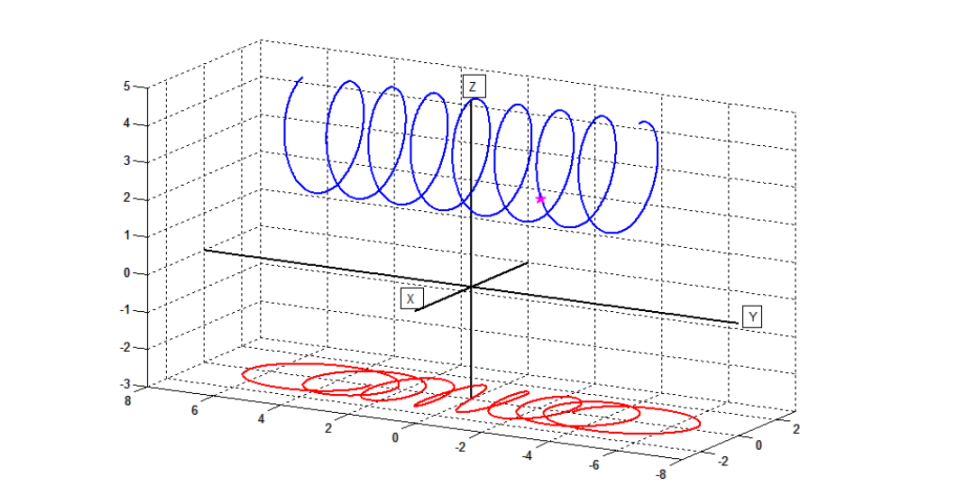
The parametric equation of a straight line in 2D is

Without loss of generality, we assume that the center of projection is at the origin. Upon projection onto the plane z = f, the image of the point is:

Dividing we get:

And we know that is a straight line in 3D, so for every t, for some constant m which is the slope of **. So, the projection also has a constant slope, hence it is a straight line.**

**Section B**

****

1. The projection of the star is marked on the image plane as a pink star.
2. First, we have that:

We can see that this trace is created by a circle which is continuously moved along the y axis, so the x and z coordinates will be some variation of cos and sin of some expression.

We have

And so we know that when s =, then x = 0.

We also know that when s = 0, then, x = 0. In order to get 8 full circles in the span of 16 *,* we have to have the expression cos(s and in order to adjust for the starting and 0 points, we add to the angle, leaving us with cos(s+. The final adjustment is for the radius which is 1.5 (we have we get:  
**x = 1.5cos(s+**

Y changes constantly and ranges from so we get:

**y =**

The definition of z comes similarly to x; the only difference is that the circle radius is at a distance of 3.5 from the XY plane, because and so we get:

**z = 3.5 +1.5sin(s+**

1. ϐ(s) is the perspective projection of α(s), so for every point p we will define its projected point so we substitute the terms and get: