Report - Current Stage:

The stage 1 was the project we created in the UGP semester.

The problem we faced in the project was our inability to identify all the correct critical points. The critical point detected were incorrect when it did not lie along the x-axis. This crippled the system and we needed to improve it.

The major cause of this was our lack of a thorough search in the critical plane for the critical point. We only checked along the standard axes!

May 2015

Then we moved onto the concept of the intersection of plane and the ellipse. Until this summer I was unable to come to a correct answer to the point of intersection, but here it is:

Let A be a positive-definite $n \times n$ real matrix, $c = (c_1, \dots, c_n)$ a point of $\mathbf{R}^n$, $p = (p_1, \dots, p_n)$ Cartesian coordinates, and $Q : \mathbf{R}^n \to \mathbf{R}$ the (positive-definite) quadratic form

$$Q(p) = (p - c)^t A(p - c).$$

Theorem: The ellipsoid $\{Q(p) = 1\}$ is tangent to the hyperplane $\{p_1 = a\}$ if and only if the upper-left entry of A^{-1} is $(a - c_1)^2$, and in this event the point of tangency is

$$p = c + \frac{1}{a - c_1} A^{-1}\mathbf{e}_1.$$

Proof: The gradient of Q is the vector field

$$\nabla Q(p) = 2A(p - c).$$

Particularly, the gradient is parallel to the first coordinate direction if and only if there is a real number λ such that

$$2A(p - c) = \lambda \mathbf{e}_1.$$

The ellipsoid $\{Q(p) = 1\}$ is tangent to the hyperplane $\{p_1 = a\}$ if and only if there is a p satisfying the system:

$$\begin{aligned}
&(p - c)^t A(p - c) = 1 && p \text{ lies on the ellipsoid,} && (1)\\
&2A(p - c) = \lambda \mathbf{e}_1 && \text{ellipsoid tangent to } \{p_1 = \text{const}\} \text{ at } p, && (2)\\
&(p - c)^t \mathbf{e}_1 = a - c_1 && \text{first coordinate of } p \text{ is } a. && (3)
\end{aligned}$$

(Such a p is necessarily unique since the ellipsoid is strictly convex.)

Substituting (2) into (1) gives

$$\tfrac{1}{2}\lambda(p - c)^t \mathbf{e}_1 = 1.$$

Comparing with (3),

$$\tfrac{1}{2}\lambda(a - c_1) = 1. \tag{4}$$

Solving (2) for $p - c$ and using (4) to eliminate λ,

$$p - c = \tfrac{1}{2}\lambda A^{-1}\mathbf{e}_1 = \frac{1}{a - c_1}A^{-1}\mathbf{e}_1. \tag{5}$$

Transposing and multiplying by $\mathbf{e}_1$ (i.e., dotting both sides with $\mathbf{e}_1$), using (3), and rearranging gives, finally,

$$(a - c_1)^2 = \mathbf{e}_1^t A^{-1}\mathbf{e}_1 = (A^{-1})_{11};$$

that is, the upper-left entry of A^{-1} is $(a - c_1)^2$. The point of tangency is read off (5).

This reduces correctly if $A = \frac{1}{r^2}I$ is the scalar matrix whose "unit ellipsoid" is a sphere of radius r: The hyperplane $\{p_1 = a\}$ is tangent to the sphere of radius r centered at c if and only if $c_1 + r = a$, i.e., $r^2 = (a - c_1)^2$, and this is the upper-left entry of A^{-1}.

Now, using the following equation we have the critical points due to each of the obstacle ellipses.

$$p = c + \frac{1}{a - c_1}A^{-1}\mathbf{e}_1.$$

Code for the calculation of the critical points and the creation of the road-map using this equation was what I worked on in May.
I experienced some bugs in the final road-map created and I am searching for those.

However, since the 3rd week of May, I have been stuck on the following issue:

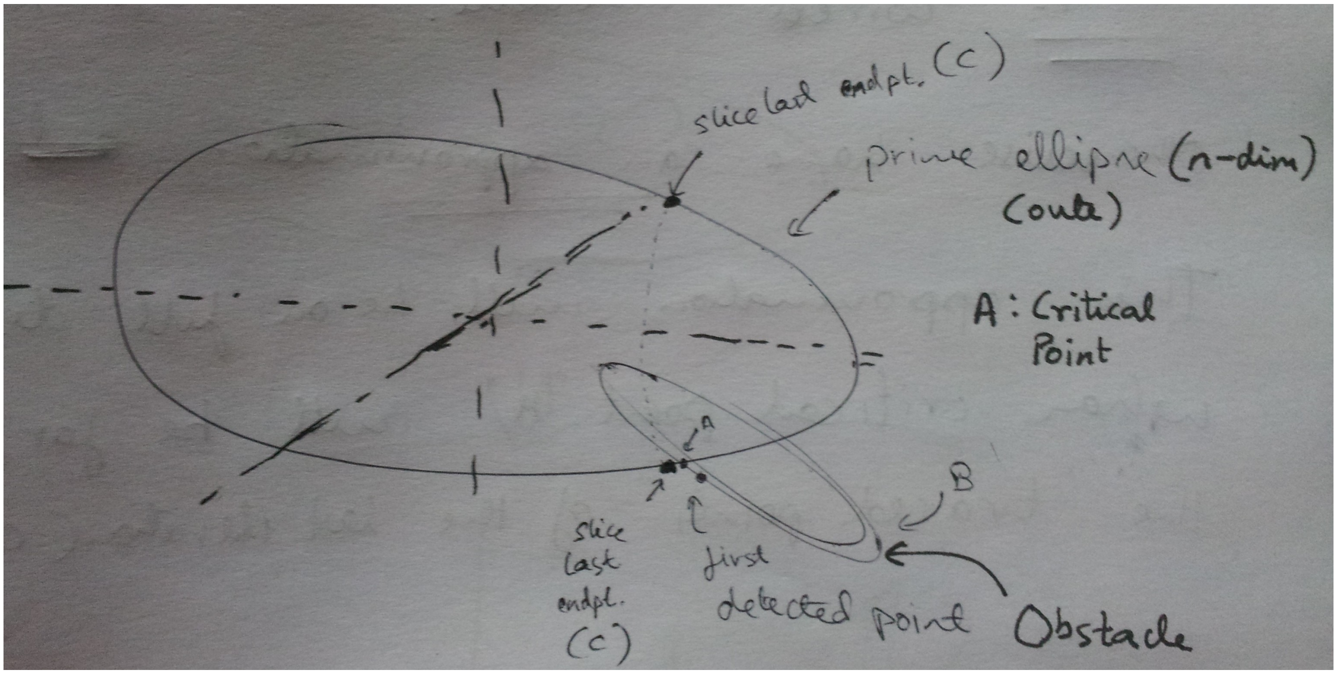
The above equation works when we have the equation of the ellipsoid and the plane in n-dimensions.
When we run recursion, say for dimension *n*, we need the ellipsoid in the *n-1* dimensions that is created by the created by the intersection of the ellipsoid in n dimensions and the plane.
This is not trivially found and will be needed to run the recursion from 4D to 3D and from 3D to 2D and so on.

This is the first major issue. Completing this part of the code will firmly complete our code and project.

Future Work:

Another issue is the calculation of exact critical points that are created when two ellipses intersect each other.
This can be done by finding the *n-1* dimensional ellipsoid equation that describes the intersection of the two *n* dimensional ellipsoids. But, there are many complications, what if there are three intersecting?

The present system is a close approximation and will work well with high accuracy. But the calculation of the critical point, which is shown below (A), will guarantee an absolutely correct implementation.



Proposal

I have started working on finding the bug and want to have a phone meeting as soon as possible to address the first issue in this report. This is would at least bring our project to some sort of conclusion.