CS 6491 Computer Graphics

Project 2 Report

Lipeng Wan, Wenjie Yao, 10/24/2018

1. Abstract

In this project, we applied the concepts of constructing curves, elbows and control polygons we learnt in class to create an interesting craft which mimics a band of multiple braids, that could be twisted. We build the final product with several layers of building components, including centerline, control polygon and control vectors, bi-arcs and PCC, etc. User could control the band by moving one of the vertices, choosing the number of vertices of the main control polygon and the number of braids on the main strand, and by dragging to adjust the twist extent.

2. Project Description and Motivation

The primary goal of this project is to develop an interactive application for designing complex braids in 3D that follow a closed-loop curve that can be edited by the user.
The project will help us understand and practice geometric constructions in 3D that use points and vectors to define curves and to move shapes along curves in an optimal way. It can also teach us about the fundamental of minimal-twist parallel transport and about its practical and possible implementations.

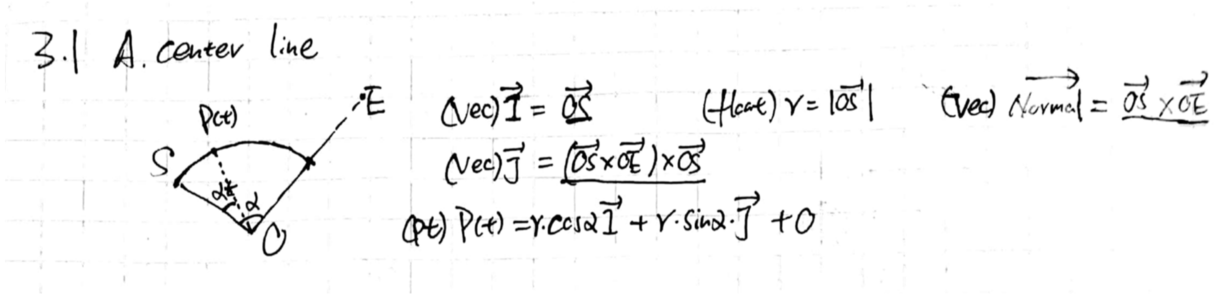
3. Technical Details

We will discuss the detailed methods we used to implement the final product, in the order of the instructions on the project description given by Prof. Jarek. We will highlight the key part in each step.

3.1 Torus Elbow and Natural/Twisted Path on It

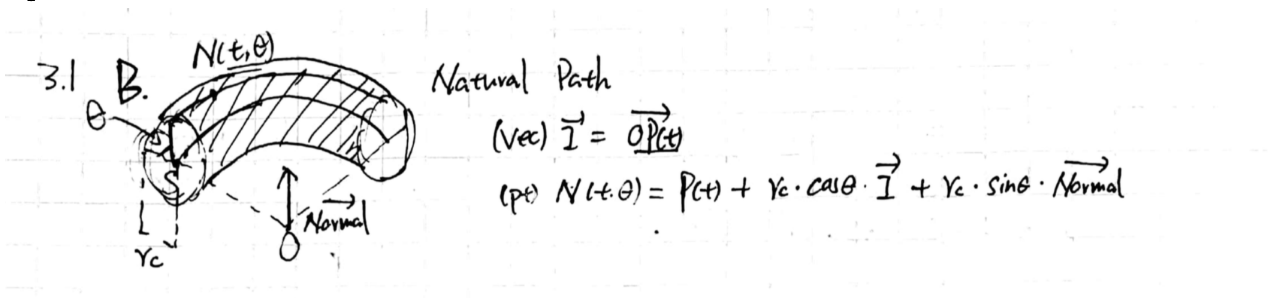
Our goal is to concatenate elbows of the same cross-radius r so that their centerlines (as shown in Figure 3.1 A) form a smooth (tangent continuous) PCC loop in 3D. We will start with a point G1 on the first elbow, trace its natural path to the end of elbow 1 to reach point H1.

Figure 3.1 A


3.1 A. center line
P(t)
S
E
O
dt
α
(vec) $\vec{I} = \vec{OS}$
(float) $r = |\vec{OS}|$
(vec) $\overrightarrow{Normal} = \vec{OS} \times \vec{OE}$
(vec) $\vec{J} = (\vec{OS} \times \vec{OE}) \times \vec{OS}$
(pt) $P(t) = r \cdot \cos\alpha \vec{I} + r \cdot \sin\alpha \cdot \vec{J} + O$

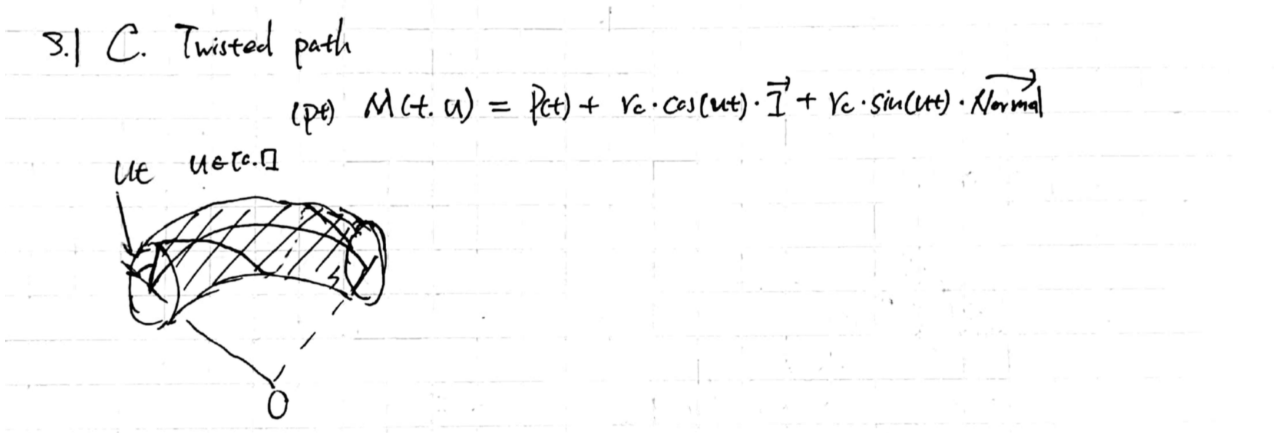
Let the user move a point P along the starting cross-circle. Display a thin, red torus (with cross-radius of r/10 or so) whose centerline is on T. We will use it to visualize the natural, twist-free path followed by P as you increase u from 0 to e, as shown in Figure 3.1 B. The natural path is a circular arc in a plane normal to axis X.

Figure 3.1 B


3.1 B.
$N(t,\theta)$
θ
r_c
O
$\overrightarrow{Normal}$
Natural Path
(vec) $\vec{I} = \overrightarrow{OP(t)}$
(pt) $N(t,\theta) = P(t) + r_c \cdot \cos\theta \cdot \vec{I} + r_c \cdot \sin\theta \cdot \overrightarrow{Normal}$

Allow the user to control a linearly varying twist of the v-parameterization along Cu that is linearly proportional to u. The user controls the total twist at u==e along the elbow, as shown in 3.1 C. This will be used for the total mismatch compensation.

Figure 3.1 C


3.1 C. Twisted path
(pt) $M(t, u) = P(t) + r_c \cdot \cos(ut) \cdot \vec{I} + r_c \cdot \sin(ut) \cdot \overrightarrow{Normal}$
ut
u∈[0.□]
O

3.2 PCC

We will discuss briefly several possible approaches for estimating tangent TC for a vertex C in the sequence of consecutive vertices {A, B, C, D, E}.
Besides the 3-circle approach shown in Figure 3.2 B, we have 3 variants, including using the unit vector of V (A, C), shown in Figure 3.2 A, and using the unit vector of V(B, C), and using the unit vector of V(B, C) + V(C, D).

Figure 3.2 A

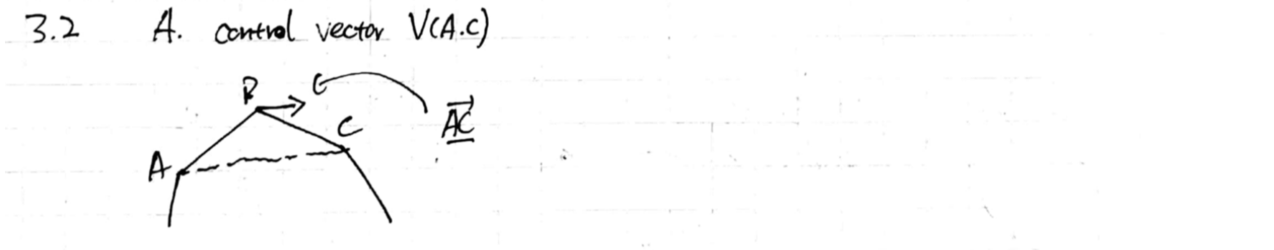
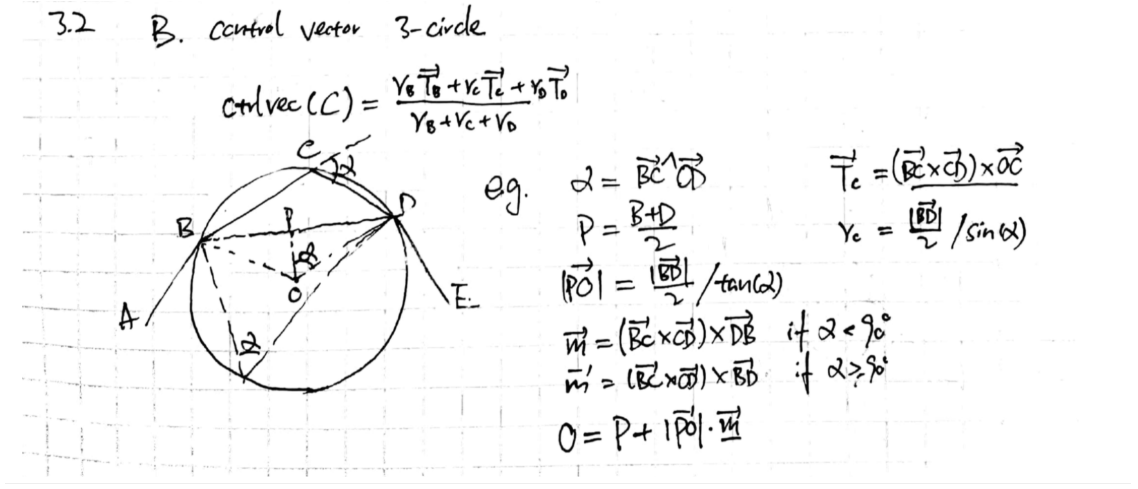


Figure 3.2 B



3.2 B. control vector 3-circle

$$\text{ctrlvec}(C) = \frac{r_B\vec{T_B} + r_C\vec{T_C} + r_D\vec{T_D}}{r_B + r_C + r_D}$$

e.g. $\alpha = \vec{BC} \wedge \vec{CD}$ $\quad \vec{T_c} = \underline{(\vec{BC}\times\vec{CD})\times\vec{OC}}$

$P = \frac{B+D}{2}$ $\quad r_c = \frac{|\vec{BD}|}{2} / \sin(\alpha)$

$|\vec{PO}| = \frac{|\vec{BD}|}{2} / \tan(\alpha)$

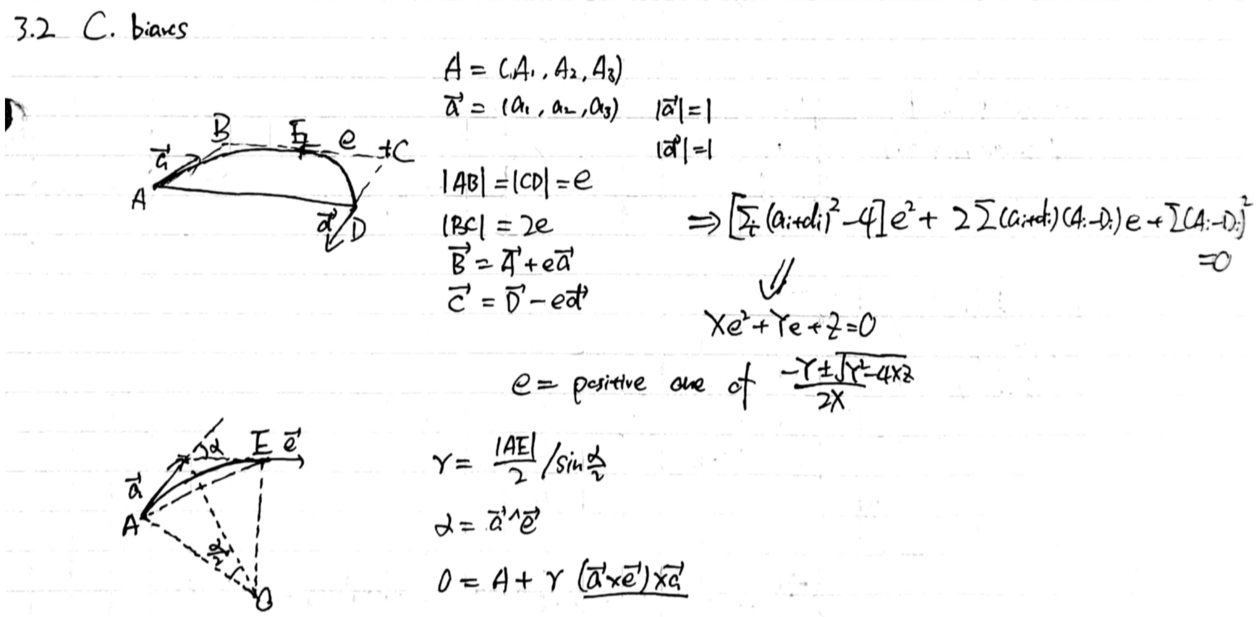
$\vec{m} = (\vec{BC}\times\vec{CD})\times\vec{DB}$ if $\alpha < 90°$

$\vec{m}' = (\vec{BC}\times\vec{CD})\times\vec{BD}$ if $\alpha \geq 90°$

$O = P + |\vec{PO}| \cdot \underline{\vec{m}}$

Given point A and unit tangent vector U and point D with unit tangent vector V, compute an oriented, tangent-continuous bi-arc (a curve made of two smoothly connected circular arcs) that passes through A with tangent U and that passes through D with tangent V. Note that vectors U, V, and AB are not coplanar. Hence, the bi-arc will not lie in a plane.

Figure 3.2 C



3.2 C. biarcs

$A = (A_1, A_2, A_3)$

$\vec{a} = (a_1, a_2, a_3)$ $\quad |\vec{a}| = 1$

$|\vec{d}| = 1$

$|AB| = |CD| = e$

$|BC| = 2e$

$\vec{B} = \vec{A} + e\vec{a}$

$\vec{C} = \vec{D} - e\vec{d}$

$$\Rightarrow \left[\sum_i (a_i + d_i)^2 - 4\right]e^2 + 2\sum(a_i+d_i)(A_i - D_i)e + \sum(A_i - D_i)^2 = 0$$

$$\Downarrow$$

$$Xe^2 + Ye + Z = 0$$

$e =$ positive one of $\frac{-Y \pm \sqrt{Y^2 - 4XZ}}{2X}$

$r = \frac{|AE|}{2} / \sin\frac{\alpha}{2}$

$\alpha = \vec{a} \wedge \vec{e}$

$O = A + r\,\underline{(\vec{a}\times\vec{e})\times\vec{a}}$

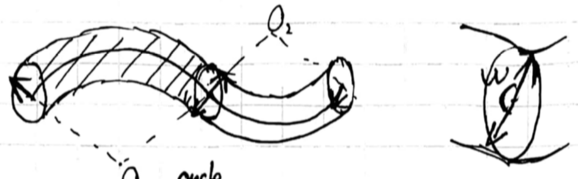
3.3 Twist-free Alignment and Control of Elbows Along PCC

Let the user pick the location of point O on that circle. Use the current position of O to align the stripes of the first elbow and to draw the natural path of O along it as a thin red tube.

Implement the propagation of the path through the elbows and use it to compute a twist-free alignment of the colored stripes of the elbows and to show the thin red tube of the parallel transport of O, as shown in

Figure 3.3 A

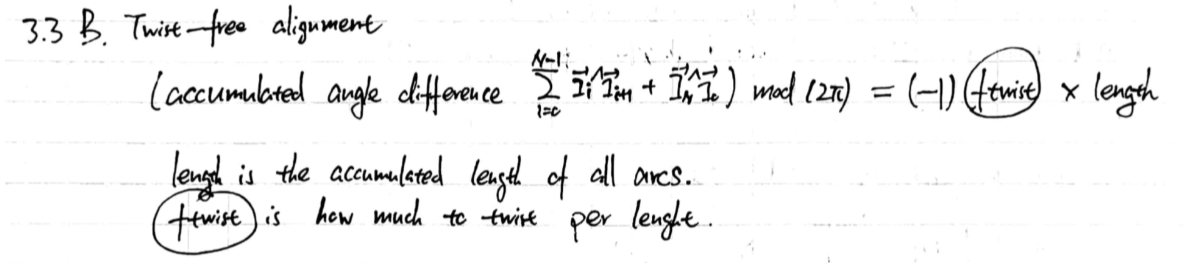
3.3 A. Parellel transport



calculate the $\underset{\text{angle}}{\text{[}}$difference between former and later elbow's $\vec{OS}$ to set the initial phase position for pipe rendering for the later one

We write and a function that change the twist that is also useful for braids, as shown in Figure 3.3 B.

Figure 3.3 B



3.3 B. Twist-free alignment

(accumulated angle difference $\sum_{i=0}^{N-1} \widehat{\vec{I_i}\vec{I}_{i+1}} + \widehat{\vec{I_N}\vec{I_0}}$) mod $(2\pi) = (-1)\,\boxed{\text{ftwist}} \times \text{length}$

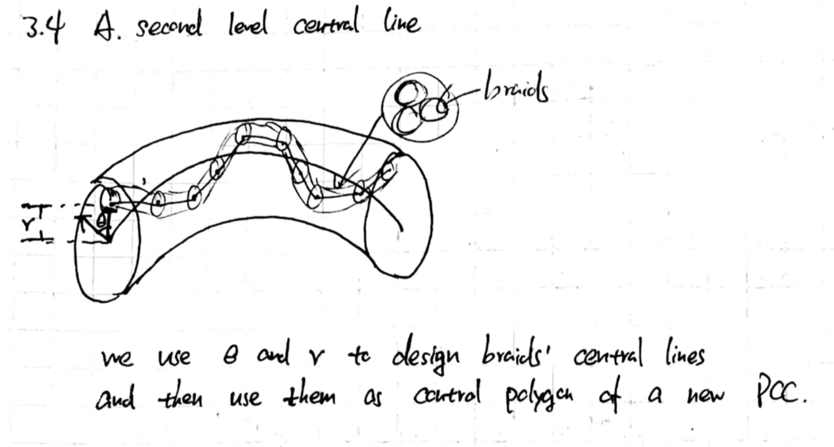
length is the accumulated length of all arcs.

$\boxed{\text{ftwist}}$ is how much to twist per lenght.

3.4 Braids

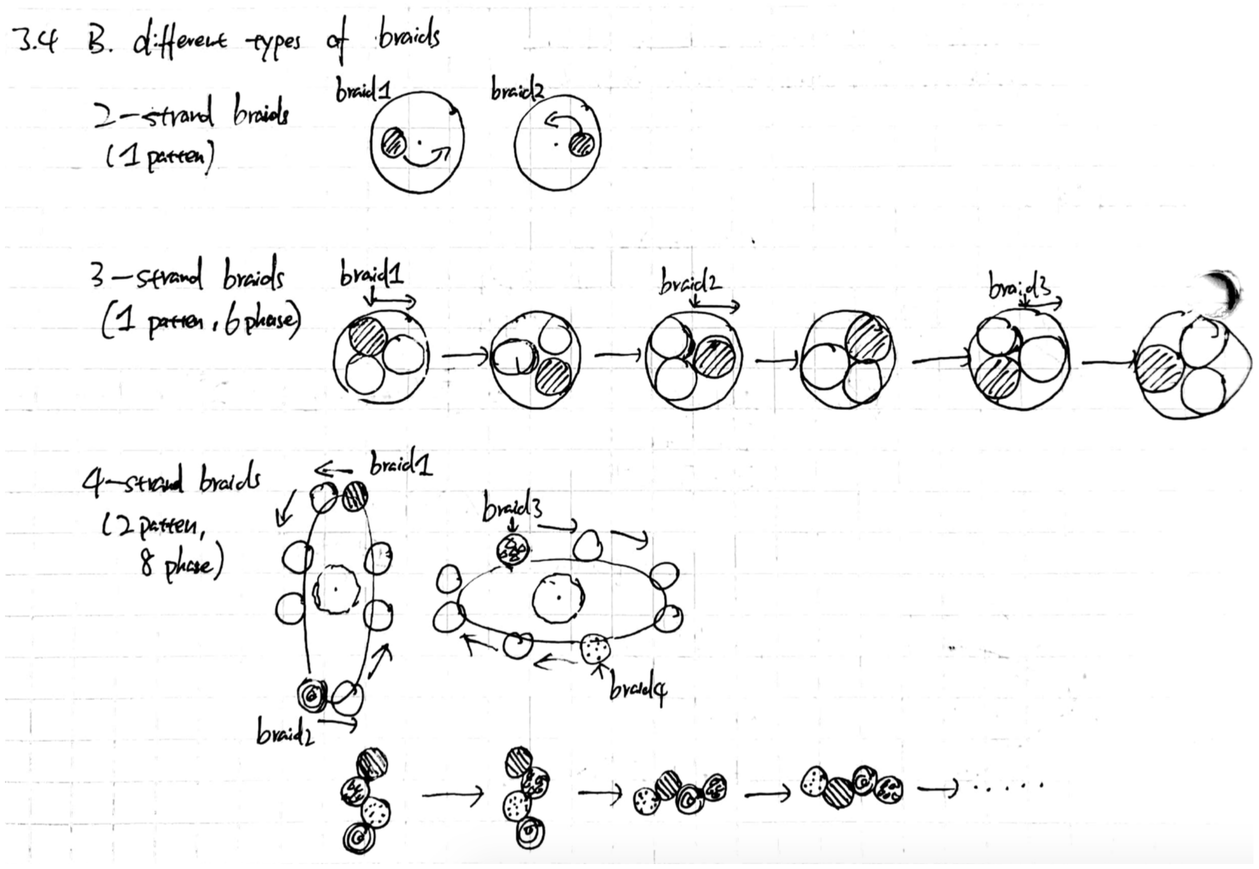
Implement path for 1, 2, 3, and possibly more strand braids.
First, we need a second level centerline (as shown in Figure 3.4 A) on the main centerline that would be needed for the braids on the main PCC.

Figure 3.4 A

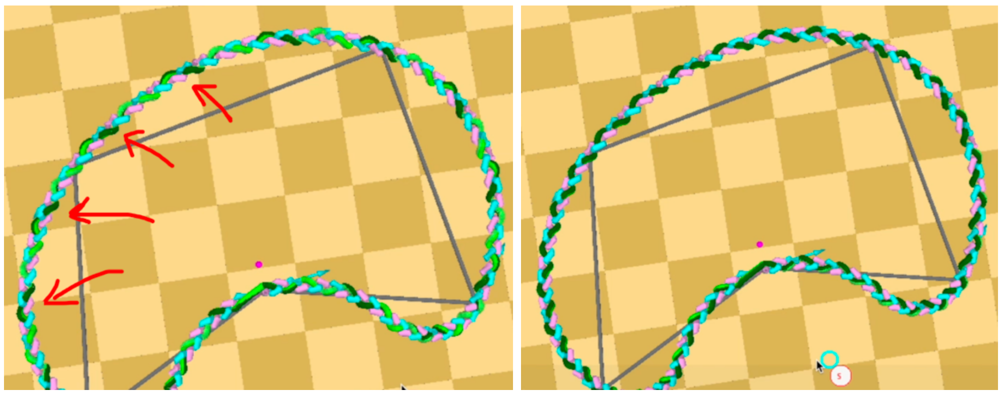

3.4 A. second level central line
braids
r
θ
we use θ and r to design braids' central lines
and then use them as control polygon of a new PCC.

After implementing the 2, 3 and 4 braids, we found different patterns of them. We demonstrate the patterns in Figure 3.4 B.

Figure 3.4 B



5. Discussion



The figures above show 3 braids case, where the green braid is striped equally by dark green, green and lime. In the left figure, we can see that the braids show a periodical pattern in its natural twist, as marked out by red arrows. While the right figure shows what happen after pre-twist for the green braid.

We think, according to physics and the way we computer arcs, the left pattern is more natural and stable, because the strip follows the natural path. However, the pre-twisted one may be better-looking, when users need their single braid striped with multiple colors.

6. Further Research

We could use the twist-compensated solution to define a series of cross-sectional frames {point C, vector I, vector J} where C moves along the centerline. Use it to instantiate a SQUINT mesh and connect the corresponding nodes (spheres) of consecutive SQUINTs with longitudinal beams to create a closed-loop SQUINT lattice. This is interesting because we can better understand SQUINT and the connection of two projects.

7. References

Rossignac, J. R., & Requicha, A. A. (1987). Piecewise-circular curves for geometric modeling. *IBM Journal of Research and Development, 31*(3), 296-313. doi:10.1147/rd.313.0296

8. Appendix

Revised Plan

Wenjie Yao: 3.1, 3,2, 3.3, 3.4, Report (discussion and figures)

Lipeng Wan: 3.2, 3.3 Report, Video