DSS Studio

A Novel Mixed Reality Navigation System for Precise Tunnel Localization in ACL Joint Bone Puncture Surgery: A Navigation Plugin

Diyu Zhou (dz372@cornell.edu)
Shuhan Liang (sl3537@cornell.edu)
Sissi Zheng (wz472@cornell.edu)

Plugin Overview

This plugin, based on the Fologram framework, is designed as an AR visualization tool for bone marrow puncture guidance using inputs from different skeletal models. By optimizing the model and converting point clouds, it generates features such as bone marrow pathways, path guidance, distance and angle indicators, and dangerous area warnings. The goal is to enhance surgical precision and streamline the surgical process for orthopedic surgeons.

Supported Features (Version 1.0)

- Precise multi-target bone tunnel puncture point surgical navigation
- Real-time multi-target precise distance measurement
- Optimal path navigation
- Puncture angle calculation
- Implant distance and depth calculation
- Danger zone calculation

Installation

bone_tunnel

Software required

- Rhino 8.0
- Grasshopper
- Fologram

Hardware required

- PC
- HoloLens

Installation Step-by-step

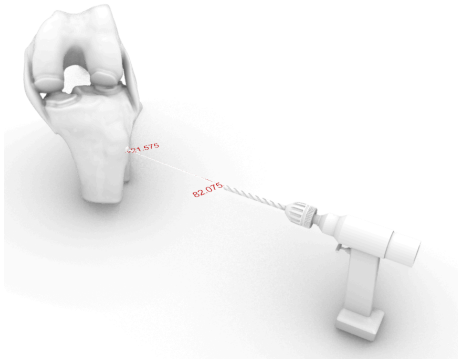
- Download and open rhino8
- Download and open grasshopper
- Set up”bone_tunnel” plug
- Set up“Fologram”plug

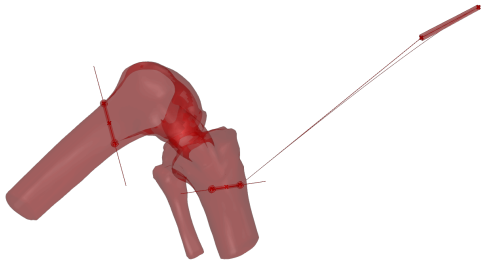
Guide of Functionality

Input

Spatial points
3D skeletal model
Bone spindle

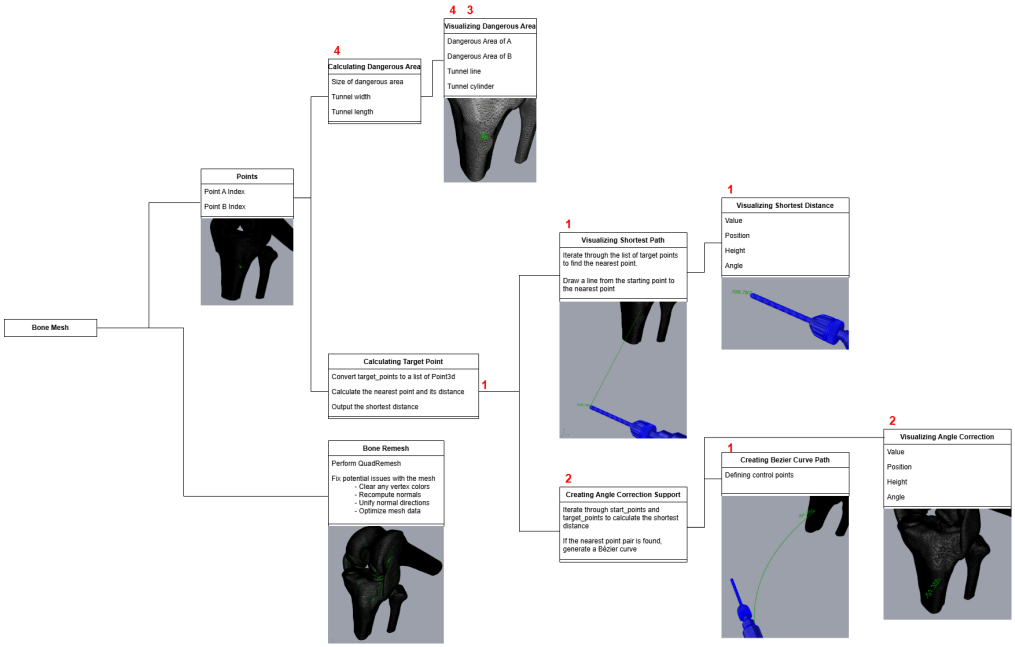
Output





Surgical path
Distance between bone tunnel points
Bone tunnel depth
Bone marrow canal opening angle

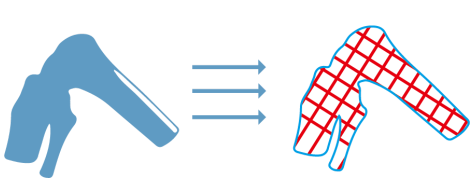
Feature Pseudo Code

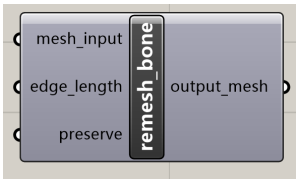


[1] Real-time multi-target optimal distance measurement & navigation
[2] Puncture angle calculation
[3] Implant depth calculation
[4] Danger zone calculation

—

Detail of Functionality





Remesh_Bone

This component is used to optimize the input skeletal mesh to facilitate subsequent calculations and optimizations.

Input- edge_length
Type: float

Description
The `edge_length` parameter determines the length of the edges of the target mesh (measured in the current model units of Rhino).
This parameter controls the resolution of the mesh, affecting its density and level of detail.

Functionality**:**
A smaller `edge_length` generates a higher-resolution mesh (with more faces).
A larger `edge_length` generates a lower-resolution mesh (with fewer faces).

Input-Preserve
Type: bool (Boolean)

Description:
The `preserve` parameter determines whether to maintain the boundary shape of the input mesh (usually referring to sharp edges or boundary lines of the mesh).

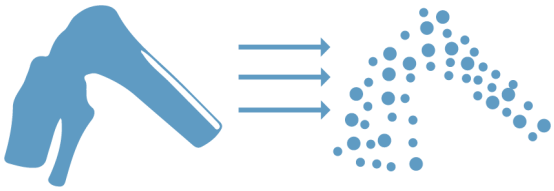
Functionality**:**
When set to `True`, QuadRemesh will attempt to preserve the original mesh's boundary shape as much as possible.
When set to `False`, QuadRemesh will not enforce boundary preservation, allowing the mesh to be freely adjusted based on the target edge length.

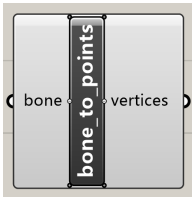
Input-mesh_input
Type: Rhino.Geometry.Mesh

Description:
The mesh_input parameter represents the original mesh to be remeshed, defining the geometry to be reconstructed.

Output-mesh_output
Type: Rhino.Geometry.Mesh

Description:
The output is an optimized mesh, making it convenient for subsequent point set conversion.





Bone_to_Point

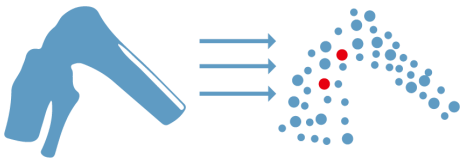
This component is used to convert the input skeletal mesh into a point cloud to facilitate subsequent calculations and optimizations.

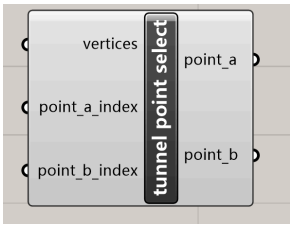
input-Bone
Type: Rhino.Geometry.Mesh

Description:
The input value can be context-dependent, either the original mesh bone or the optimized skeleton mesh.

Output-Vertices
Type: Point

Description:
The output consists of vertices converted from the mesh skeleton.





Tunnel Point Select

The selection of specific bone tunnel points on the skeletal mesh is done through the

point set on the skeletal mesh.

input-vertices
Type: Point 3D

Description:
This input value is the point set generated from the skeletal mesh.

input-point_a_index
Type: Int

Description:
Points are selected using the index, with `point_a` being the starting point of the bone tunnel..

input-point_b_index
Type: Int

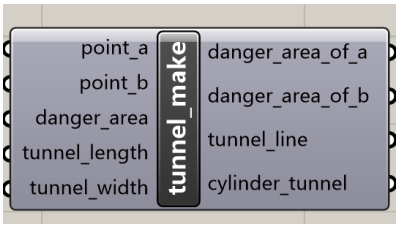
Description:
Points are selected using the index, with `point_b` being the endpoint of the bone tunnel.

Output-point_a
Type: Point

Description:
`point_a` is the starting point of the bone tunnel.

Output-point_b
Type: Point

Description:
`point_b` is the endpoint of the bone tunnel.



Tunnel_make

The input points and set values are used to generate the danger zones and bone tunnels at each point.

Input-point_a
Type: Point

Description:
`point_a` is the starting point of the bone tunnel.

Input-point_b
Type: Point

Description:
`point_b` is the endpoint of the bone tunnel.

Input-danger_area
Type: Float

Description:
The input value is the radius length of the danger zone.

Input-tunnel_length
Type: Float

Description:
The input is the length of the bone marrow canal.

Input-tunnel_width
Type: Float

Description:
The input is the width of the bone marrow canal.

Output-danger_area_of_a
Type: Geometry

Description:
The danger zone generated based on a as the center and the area radius.

Output-danger_area_of_b
Type: Geometry

Description:

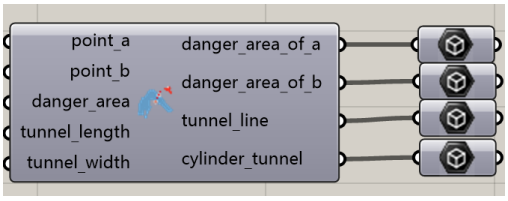
The danger zone generated based on b as the center and the area radius.

Output-tunnel_line
Type: Line

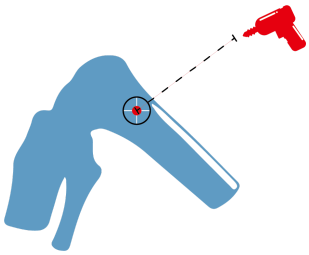
Description:
The bone tunnel guide line generated based on the point locations inside the bone.

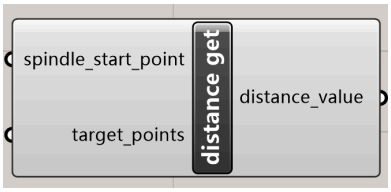
Output-cylinder_tunnel
Type: Geometry

Description:
The bone tunnel guide line generated based on the point locations and extension length outside the bone.



***All of these outputs should be connected with synchronized geometry in fologram, in an attempt to facilitate AR display and synchronization with HoloLens.**





distance get

This is used to calculate the distance between the bone drill bit point and the nearest target point among multiple bone tunnel target points.

Input-spindle_start_point
Type: Point

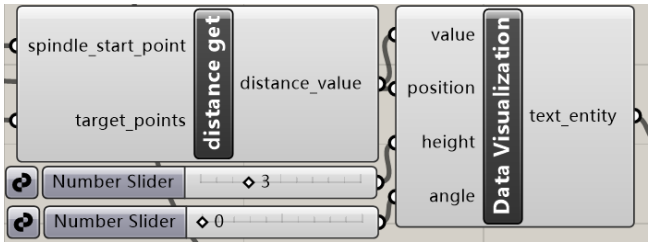
Description:
This point is the drill bit point of the bone drill.

Input-target_points
Type: Point

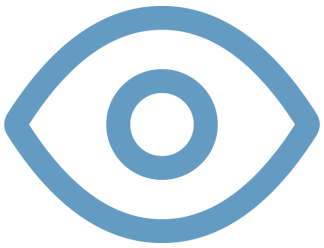
Description:
This point is the target point, which can be a single point or a collection of multiple points.

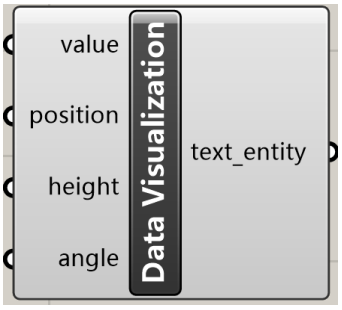
Output-distance_value
Type: Value

Description:
The output value is the distance between the nearest target point and the bone drill bit point.



*__This output should be connected with data visualization.__





Data Visualization

Convert the calculated value into a visualizable text_entity.

Input-value
Type: value

Description:
The input value is the calculated distance or angle.

Input-position
Type: Point

Description:
The location where the text_entity needs to be generated.

Input-height
Type: Value

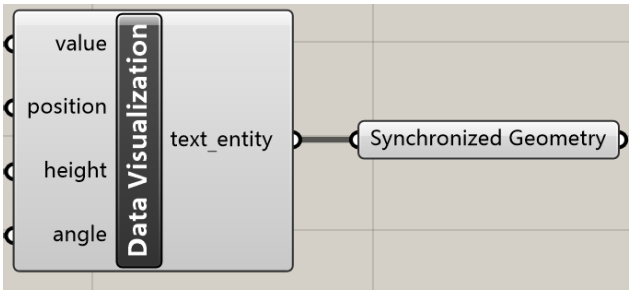
Description:
The size of the text_entity to be generated.

Input-angle
Type: Value(0-360)

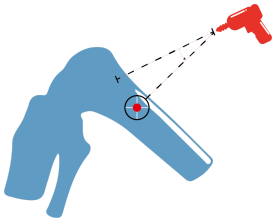
Description:
Rotate the text_entity around the generated point location as the center.

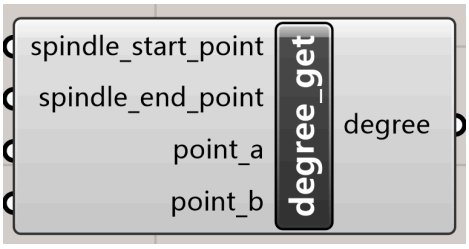
Output-text_entity
Type: Entity

Description:
Visualize the value as an entity.



***All of these outputs should be connected with synchronized geometry in fologram, in an attempt to facilitate AR display and synchronization with HoloLens.**





degree_get

By picking the start and end points of the drill bit and the start and end points of the bone tunnel, calculate the angle between the two straight lines of the bone drill and the bone tunnel to facilitate calibration.

Input-spindle_start_point
Type: Point

Description:
This point is the drill bit point of the bone drill.

Input-spindle_end_point
Type: Point

Description:
This point is the drill end point of the bone drill.

Input-point_a
Type: Point

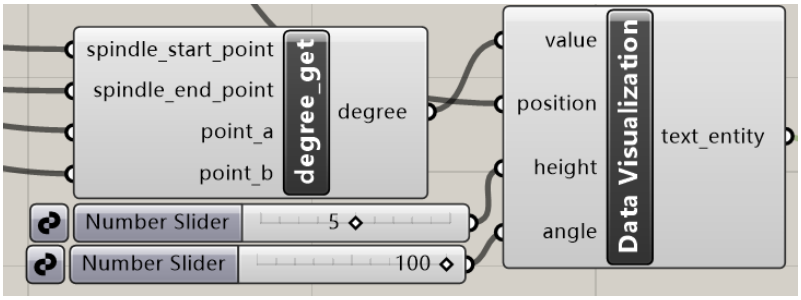
Description:
`point_a` is the starting point of the bone tunnel.

Input-point_b
Type: Point

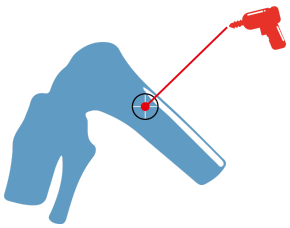
Description:
`point_b` is the endpoint of the bone tunnel.

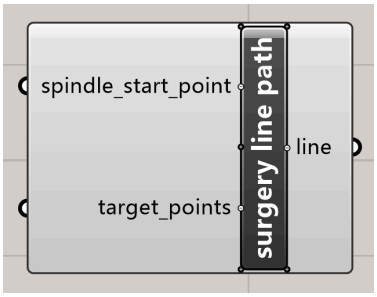
Output-degree
Type: value

Description:
Calculate the angle between the two straight lines of the bone drill and the bone tunnel.



*__This output should be connected with data visualization.__





surgery line path

This component is used to generate the shortest path for the bone drill to reach the nearest target point.

Input-spindle_start_point
Type: Point

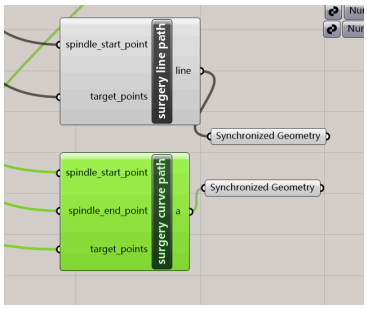
Description:
This point is the drill bit point of the bone drill.

Input-target_points
Type: Point

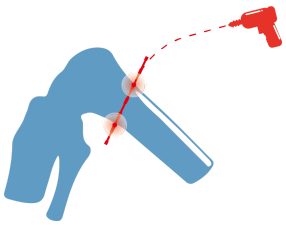
Description:
This point is the target point, which can be a single point or a collection of multiple points.

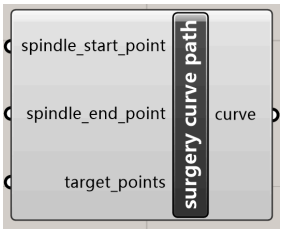
Output-line
Type: line

Description:
Generate the shortest path to the nearest target point.



***This output should be connected with synchronized geometry**





surgery curve path

This component is used to generate a curved path for the bone drill to reach the nearest target point, which helps the user calibrate the angle and path.

Input-spindle_start_point
Type: Point

Description:
This point is the drill bit point of the bone drill.

Input-spindle_end_point
Type: Point

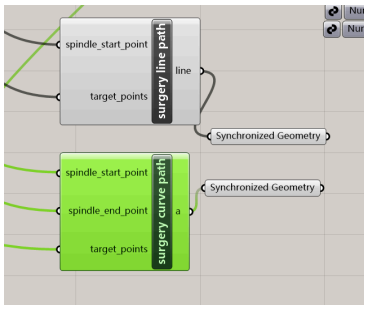
Description:
This point is the drill end point of the bone drill.

Input-target_points
Type: Point

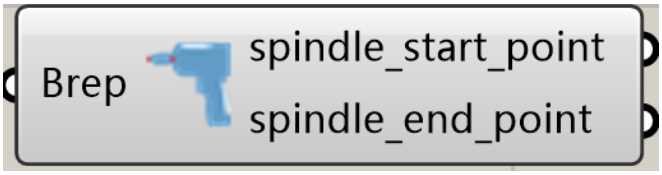
Description:
This point is the target point, which can be a single point or a collection of multiple points.

Output-curve
Type: curve

Description:
Generate a Bézier curve to the nearest target point.



***This output should be connected with synchronized geometry**



Extract the bone drill point location

This is a cluster file used to extract the preset start point and end point of the bone drill. We will include it along with the bone drill brep file in the appendix to ensure proper usage.

Input-Brep Type:
Type: Brep

Description:
Preset bone drill.

Output-spindle_ start_point
Type: Point

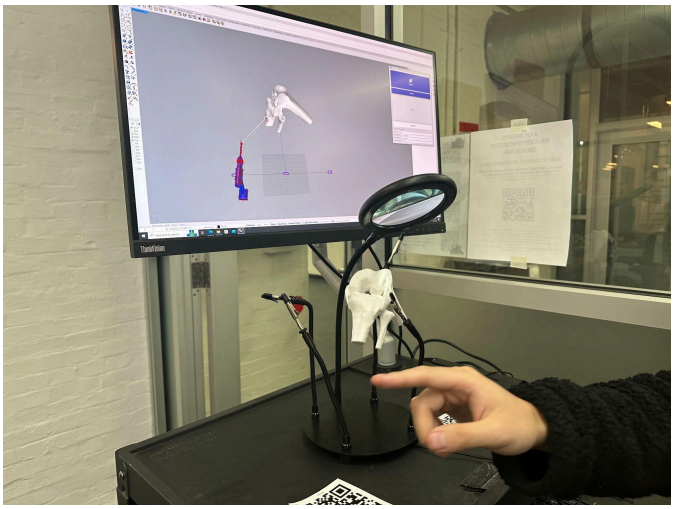
Description:
Output the bone drill bit point location.

Output-spindle_ end_point
Type: Point

Description:
Output the bone drill end point location.

Usage ExamplesDetailed guide of functionality.

https://youtu.be/z2X9pDw00ao



Results From Using Our Tool

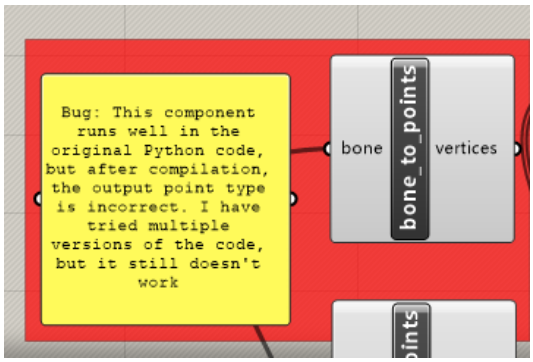
We have achieved the precise localization of bone tunnels, which significantly enhances recovery from joint diseases and improves the success rate of surgeries. In this context, MR-based precise bone puncture surgery navigation plays a crucial role in assisting surgical procedures. Unlike traditional methods, we have leveraged the capabilities of MR-based surgical navigation, enabling surgeons, with the aid of MR-SNS, to accurately locate the ideal tunnel positions planned preoperatively and precisely drill the bone tunnels.

Additionally, we have explored the potential of generating an optimal path in static bone puncture surgery, which can notably reduce the cognitive load on surgeons during prolonged operations. This provides a key auxiliary tool within AR's precise navigation system, supporting orthopedic surgeons in performing accurate surgeries.

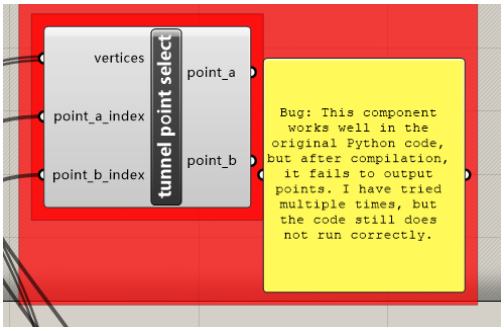
limitation

Additionally, we have explored the potential of generating an optimal path in static bone puncture surgery, which can notably reduce the cognitive load on surgeons during prolonged operations. This provides a key auxiliary tool within AR's precise navigation system, supporting orthopedic surgeons in performing accurate surgeries.

Bugs



(1) **bone_to_points:** This component runs well in the original Python code, but after compilation, the output point type is incorrect. We have tried multiple versions of the code, but it still does not work.



(2) **tunnel point select:** This component runs well in the original Python code, but after compilation, it falls to output points. We have tried multiple times, but the code still does not run correctly.