Dyna-Eye: A dynamic 2D-3D Stereo Viewer

Final Presentation – 08.02.2018

Project Supervisor: Benjamin Busam

Presented by: Ruiqi Gong
Recap

- Stereo viewer with crossed eyes

[YouTube](https://www.youtube.com/watch?v=3D_without_glasses_Cross-Eye_HD) “3D without glasses, Cross-Eye HD” (07.11.2017)

[Donald Simanek](http://www.donaldsimanek.com) “How to view 3D without glasses”. (05.11.2017)
Method: local model

- Notation:
  - $e$: distance of two eyes
  - $f$: distance to focus point

- Preprocess:
  - Calculate centroid of the 3D model
  - Translate the object onto the origin
  - Normalize volume of the object
Method: local model

- **Notation:**
  - $e$: distance of two eyes
  - $f$: distance to focus point

- **Preprocess:**
  - Calculate centroid of the 3D model
  - Translate the object onto the origin
  - Normalize volume of the object

- **Click! Click!**
  - Take two pictures
Method: global model

- Notation:
  - \( l \): distance to screen
  - \( W_s \): width of screen (in centimeters)
  - \( x \): 3D model’s position in viewport

- Split the window into two viewports
- Display “two pictures” on correct positions
Method: global model

- Notation:
  - \( l \): distance to screen
  - \( W_s \): width of screen (in centimeters)
  - \( x \): 3D model’s position in viewport

- Split the window into two viewports
- Display “two pictures” on correct positions
- Shift “two pictures” according to the distance to focus point
Method: global model

- Notation:
  - $l$: distance to screen
  - $W_s$: width of screen (in centimeters)
  - $x$: 3D model’s position in viewport

- Split the window into two viewports
- Display “two pictures” on correct positions
- Shift “two pictures” according to the distance to focus point
  - Problematic: no “picture-shifting” function
  - Solution: achieved by shifting the cameras
Method: rotation compensation

- Notation:
  - $\Delta x$: horizontal shift
  - $\alpha$: angle of rotation

- Model is in 3D, and its image changes when the camera moves horizontally.
- Rotate the model and let it face to the camera as before.
UML Diagram

User:
Ask_3D_Model_Path
Input_Path: String

Stereo Viewer:
Ask_Parameters
Input_Parameters: Float [ ]
Show_Object_3D: Images

Description
Starting his/her own work;
Look at the screen once in a while;
Adjust focus point by scrolling.

Operating System:
Request_Load_File
Return_File: PointCloud<PointType>

Description
Start rendering the 3D object;
Keep responding to user's behavior;
Keep updating the display.

Mouse_Scroll_Down
Scene_Update: Focus Point closer
Mouse_Scroll_Up
Scene_Update: Focus Point further
Achievement

✓ Gain knowledge of PCL, VTK and Cmake
✓ Build up the math model
✓ Implement the rendering pipeline from 3D model to binocular images
✓ Choose and create appropriate 3D models
✓ Implement the focus helper
✓ Correlate the focus helper with the 3D model
✓ ✓ Runtime focus point movement (preliminarily completed)
Future work

• Faster and smoother runtime focus point movement
• Better calibration between PCL coordinates system and real-world metrics
• Further testing on screens with different sizes
• Complete documentation
Gantt Chart

Start:
- Phone Meeting
- Read code of previous project
- First Meeting

Mid:
- Preparation for the first Presentation
  - Make Slides & Review
  - Rehearsal
  - Requirements Presentation

End:
- Project Development
  - Dev Environment
  - Trivial Experiments
  - Implementation
  - Testing and Documentation

Key:
- Green: Project Development
- Dark Blue: Implementation
- Orange: Preparation for the first Presentation

Timeline:
- November: Week 42, Week 43, Week 44
- December: Week 45, Week 46, Week 47, Week 48
- January: Week 49, Week 50, Week 51, Week 52
- February: Week 01, Week 02, Week 03, Week 04, Week 05, Week 06, Week 07, Week 08, Week 09, Week 10

- Dyna-Eye
- Preperation for the final presentation
  - Final Presentations Vol. 2
Thank you for your attention.

Question?