Project Management and Software Development

FMtrack - Optical Tracking System for Medical Procedural Simulator

Final Presentation – 09.02.2017

Project Supervisors: Benjamin Busam, Patrick Wucherer, Felix Achilles

Presented by: Ramona Schneider
Problem Statement

- Automatic identification and tracking of non-rigid medical instruments
- Simulation of medical images
- Goal: Creation of a training/assessment tool for minimally-invasive procedures

Simulation setup [1]

Non-rigid instrument [2]

## Requirements

<table>
<thead>
<tr>
<th>requirement</th>
<th>achieved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real-time processing of data</td>
<td>yes</td>
</tr>
<tr>
<td>Tracking and identification of medical instruments (multiple, rigid, non-rigid)</td>
<td>Rigid: works</td>
</tr>
<tr>
<td></td>
<td>Multiple: not tested yet</td>
</tr>
<tr>
<td></td>
<td>Non-rigid: not working yet</td>
</tr>
<tr>
<td>Simulated images (current instruments, correct position/shape)</td>
<td>Instruments, position: works</td>
</tr>
<tr>
<td></td>
<td>Shape: not working yet</td>
</tr>
</tbody>
</table>
Components

- Medability
  - Application
    - Core
      - SFML
      - MetalIO
      - Chai3d
      - OpenIGTLink
      - OpenGL
Class Design

```
FramosTracker
-instrument_listener: Thread
-simulator_listener: Thread
-m_calibrated: boolean
+FramosTracker(instrument_port: int, simulator_port: int)
+update(): void
+calibrate(): void
+isCalibrated(): boolean
+listen(): boolean
+getTrackerToWorldPosition(): chai3d::cVector3d
+getLocalPosition(): chai3d::cVector3d
+getWorldPosition(): chai3d::cVector3d
-computeTipPosition()

Model
-m_ots: FramosTracker
```

```
QThread

Thread
-m_port: int
-m_calibrate: boolean
-matrices: std::vector<std::pair<chai3d::cMatrix3d, chai3d::cVector3d>>
-rot: chai3d::cMatrix3d
-pos: chai3d::cVector3d

+Thread(port: int)
+getMatrices(): std::vector<std::pair<chai3d::cMatrix3d, chai3d::cVector3d>>
+getRotation(): chai3d::cMatrix3d
+getPosition(): chai3d::cVector3d
+setCalibration(calibrate: boolean): void
-run()
-ReceiveTransform(socket: igtl::Socket, header: igtl::MessageHeader): int
```
Setup

Application

Instrument with markers

Tracking Cameras
Future Work

• More testing

• Tracking of multiple instruments

• Non-rigid tracking
Summary

• **Achieved:**
  – OpenIGTLink Interface / optical tracking integrated in Medability software
  – Rigid instruments can be tracked
  – Position can be shown

• **Problems:**
  – Time
  – Linking problems

• **Still to do:**
  – Tracking of multiple instruments
  – Non-rigid tracking