A hybrid opto-inertial Tracking System Prototype

Faisal Kalim
Faisal.kalim@tum.de

Supervisor: Benjamin Busam
Partners: IFL & FRAMOS

Final Presentation

Computer Aided Medical Procedures (CAMP) Technische Universität München, Germany July 13, 2016
Project Outline

- Motivation
  - Different Tracking Solutions
  - Each have pros and cons
  - Hybrid solution for overcoming limitations

- Problem Statement
  - To develop a software prototype that fuses the data streams of both an inertial measurement unit (IMU) and an optical tracking system (OTS)

Image Courtesies: B. Busam – Optical Tracking for Medical Applications (July 8, 2016)
Slideshare (July 8, 2016)
Requirements and Specifications

- Tracking
  - Tracking of device based on IMU data
  - Data Fusion

Evaluation

- Compare results of IMU based tracking to OTS

GUI

- Display individual + hybrid pose values
Project Outline

➢ Project Plan

A hybrid opto-inertial Tracking System Prototype - Faisal Kalim

Made with TeamGantt
Main
serverSocket: igtl::ServerSocket::Pointer
port: int
+ main(argc, char* argv[])
+ initialize()
+ ReceiveTransform(igtl::Socket * socket, igtl::MessageHeader * header)
+ ReceivePosition(igtl::Socket * socket, igtl::MessageHeader * header)

IMU
+ position[3]: float
+ velocity[3]: float
+ quaternion[4]: float
+ relativeRotMat[9]: float
+ spatial_simple()

QuatRotation
+ CalculateRelativeRotation(const float q[4], const float p[4], double R[9])
+ void quatern2rotMat(const float q[4], double R[9])

MahonyAHRS
- q0: float
- q1: float
- q2: float
- q3: float
+ MahonyAHRSUpdateIMU(float gx, float gy, float gz, float ax, float ay, float az, float quaternion[4])

KalmanFilter
- p_est[81]: double
- x_est[9]: double
+ kalmanfilterCustom(const double z_data[], double y[3])
+ kalmanfilterCustom_init(void)
Software Design

A hybrid opto-inertial Tracking System Prototype - Faisal Kalim

July 18, 2016  Slide 6

Initialize IMU

Wait for attachment

Set Callbacks

Initialize Servers

Send Hybrid Position

Get Data from OTS

Update Position, Velocity and Transformation

Update position, velocity, and orientation

IMU not attached

IMU attached

Client Connected

Client disconnected

Client disconnected

Client disconnected
Software Design

- Strategies
  - Server-Client: top-down
  - 6 DOF position estimation: bottom-up

- Server-Client Communication
  - OpenIGTLink Protocol, using TransfromMessage

- Separate Threads
  - OpticalServer, GUIServer, HybridServer

- Environment
  - Windows, Visual Studio, C++, phideget, OpenIGTLink, Matlab
Project Results

A hybrid opto-inertial Tracking System Prototype - Faisal Kalim
Project Results

A hybrid opto-inertial Tracking System Prototype - Faisal Kalim
Project Results
Project Outlook

- All tasks completed
  - IMU Integration
  - IMU calibration
  - IMU Data filtering
  - Position and Orientation estimation
  - Server Implementation
  - Co Calibration of IMU and optical tracker
  - Data Fusion
  - Analysis
  - Visualization

- Future Work
  - Calibration of IMU from more readings
  - Accelerometer bias correction from optical data
  - Adaptive filtering of accelerometer data
  - More testing and evaluations
  - Navigation grade IMU
Project Retrospective

➢ Main Challenges
  ➢ Underestimated tasks
  ➢ Issues with third party libraries

➢ Tasks
  ➢ All mandatory tasks completed

➢ 6DOF pose estimation, IMUs, OpenIGTLink, Kalman filtering, Software
  Project Management, Presentation Skills

➢ Lessons Learned
  ➢ Integration of modules in different platforms
- Questions
- Comments
- Ideas