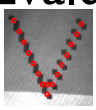
Evaluation and Optimization of  irtual Rigid Body

Project 14

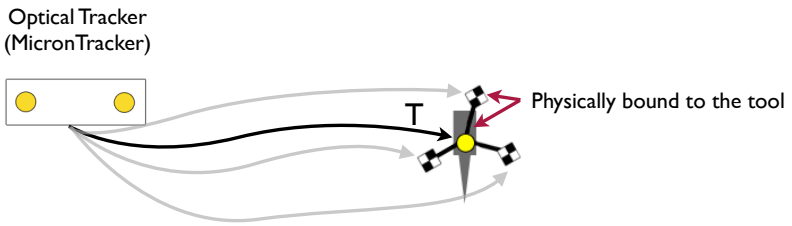
David Lee (dslee@cis.jhu.edu)
Mentors: Alexis Cheng, Dr. Emad M. Boctor

Checkpoint Presentation
April 3rd, 2014

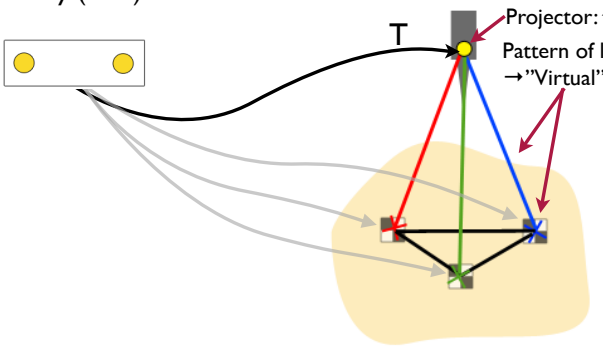
Recapitulation : Concept

•Pose (T = [R, t]) of the surgical tool in optical tracker coordinates?

- Conventional physical rigid body (PRB)

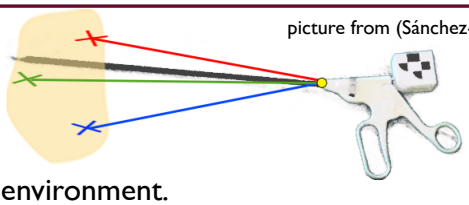


- Virtual rigid body (VRB)

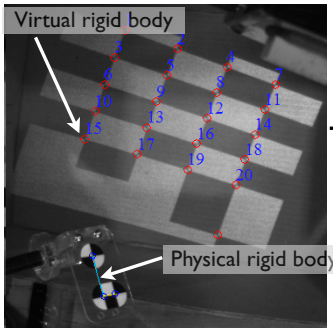


Recapitulation : Merits

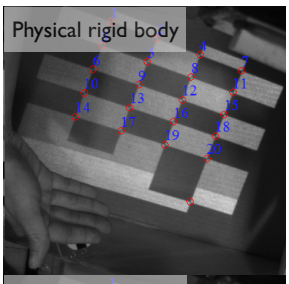
picture from (Sánchez-Margallo et. al, 2013)

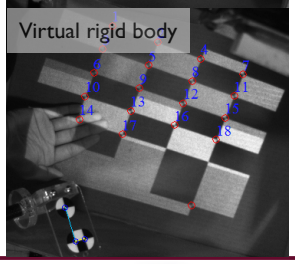


- Freedom of size and shape
  - VRB can be projected in any form.
  - Prospect for higher accuracy
- Freedom of space
  - "Virtual" : does not interfere with environment.
  - Applicable to dense surgical situation such as laparoscopy.
- Robustness to occlusion
  - Less likely to be occluded
  - Easy to implement redundancy



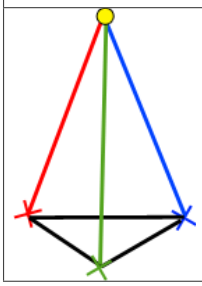
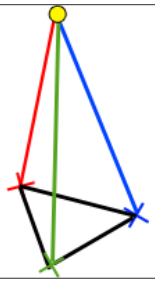
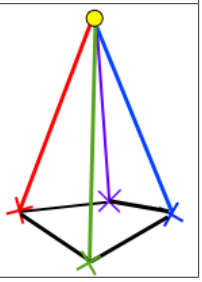
Occlusion





Recapitulation : Purpose

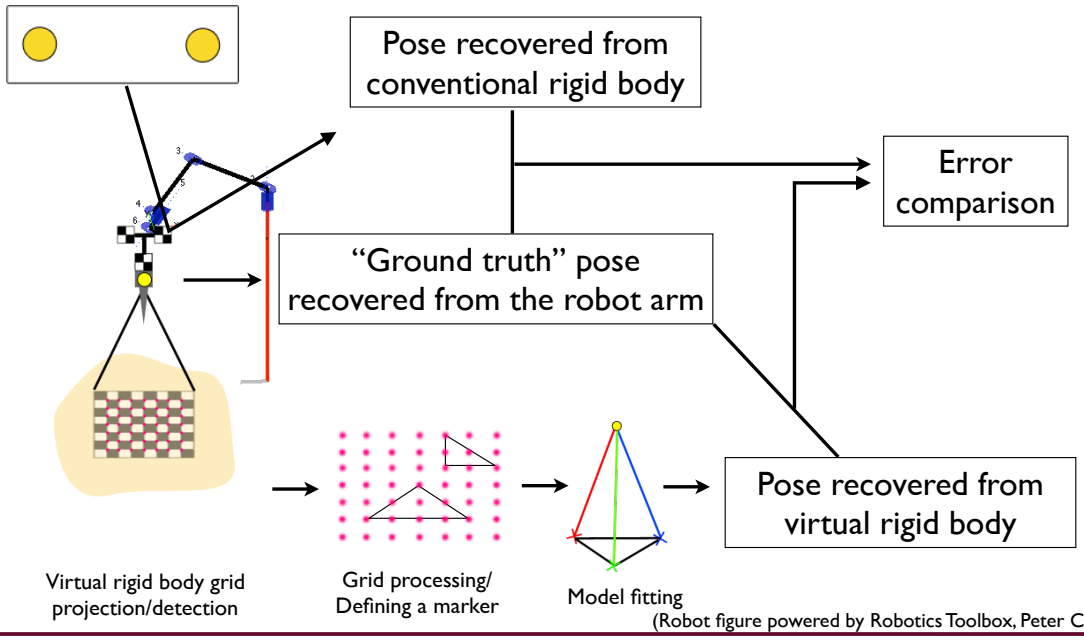
- Investigate the operating condition of virtual body including,
  - -Range of motion
    - -Translational, rotational, composite
  - -Virtual rigid body characteristics

Typical model	Size	Shape	Number of Projections
			

- Compare the accuracy between virtual body and physical rigid body

Overall study design

For a given pose of the projector,



(Robot figure powered by Robotics Toolbox, Peter Corke)

"Pyramid Model": Pose estimation

- Input :
  - $\vec{p}_i$:Virtual rigid body position in MT coordinate
- Pyramid model:
  - $T_a = [\boldsymbol{R}, \vec{t}\,]$: Unknown pose of apex in tracker coordinate
  - $\vec{u}_i$: Known unit projection vectors from apex
  - Distance between projected bodies
    - $d_{ij} = ||\vec{p}_i - \vec{p}_j||$: from tracker coordinate
    - $||s_i\vec{u}_i - s_j\vec{u}_j||$: from apex coordinate
  - Find sets of s_i such that optimization problem $d_{ij}^2 = ||s_i\vec{u}_i - s_j\vec{u}_j||^2$ is satisfied.
  - T_a is recovered by registration between $\vec{p}_i$ and $s_i\vec{u}_i$
- Output :
  - Estimated pose of the projector

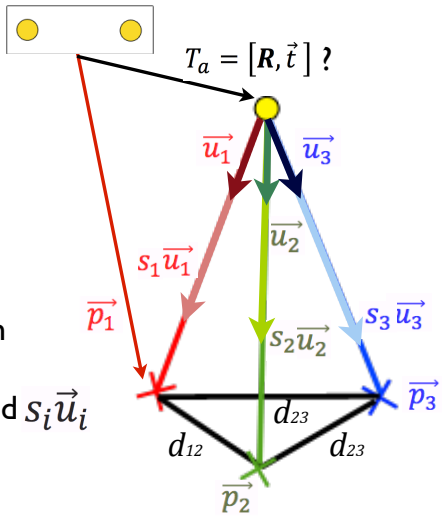


Figure redrawn from (Cheng et. al., 2014)

Deliverables - modification

JOHNS HOPKINS UNIVERSITY

Original

- Minimum (Mar 31st)
  - Marker grid
  - Experimental routines in python or C++ codes
  - Experimental data
- Expected (Apr 15th)
  - Analysis and evaluation of different sets of virtual markers
  - Optimal design of virtual rigid body
- Maximum (Apr 30th)
  - Publication
  - Experimental data on non-level surfaces.
  - Introductory ideas on projector design.

Modified

- Minimum (Mar 31st) - Pipeline Setup
  1. Virtual rigid body (VRB) grid
  2. Detection component
  3. Processing component
  4. Robot component
- Expected (Apr 23nd) - Experiment/Analysis
  - Run pipeline for data collection
  - Comparison between virtual and physical rigid body
  - Optimal design of virtual rigid body
- Maximum (Apr 30th) - Application
  - Demonstration of virtual rigid body in laparoscopy setting
  - Documentation

Deliverables

JOHNS HOPKINS UNIVERSITY

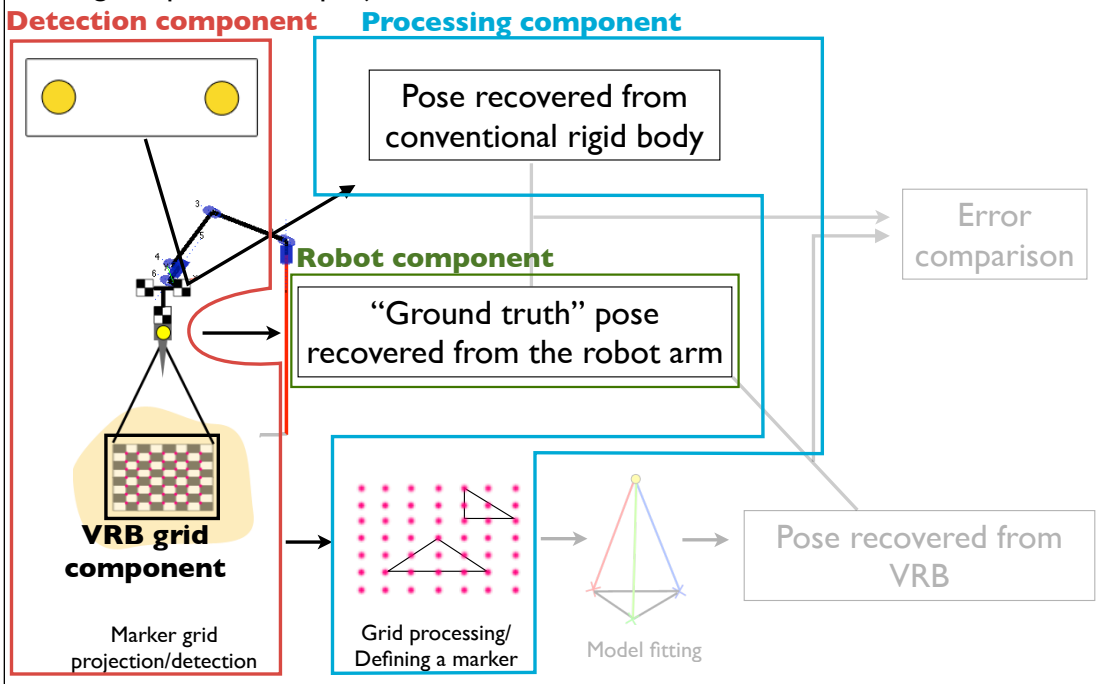
- Minimum (Mar 31st) - Experimental Pipeline Setup ✔
  1. Virtual rigid body (VRB) grid ✔
  2. Detection component ✔
  3. Processing component ✔
  4. Robot component ✔
- Expected (Apr 15th) - Experiment/Analysis ⚠ (by 04/23)
  - Run pipeline for data collection ⚠ (by 04/07)
  - Comparison between virtual and physical rigid body
  - Optimal design of virtual rigid body
- Maximum (Apr 30th) - Application
  - Demonstration of virtual rigid body in laparoscopy setting
  - Documentation

✔ complete
⚠ delayed

Experimental Pipeline

JOHNS HOPKINS UNIVERSITY

For a given pose of the projector,

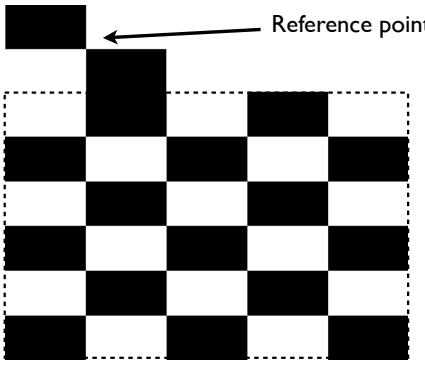


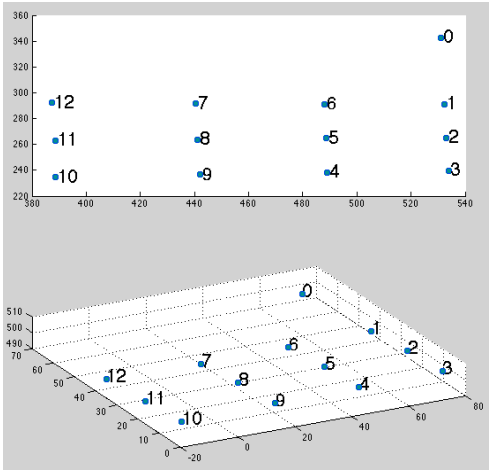
Virtual Rigid Body Grid

JOHNS HOPKINS UNIVERSITY

- Components
1. VRB grid
2. Detection
3. Processing
4. Robot

- Considerations
  - has to be detected well
    - ➡ Classic black and white for maximal contrast
  - correspondence has to be assigned
    - ➡ Rectangle : width/height ratio = 2
    - ➡ Reference point attached to a rectangular grid





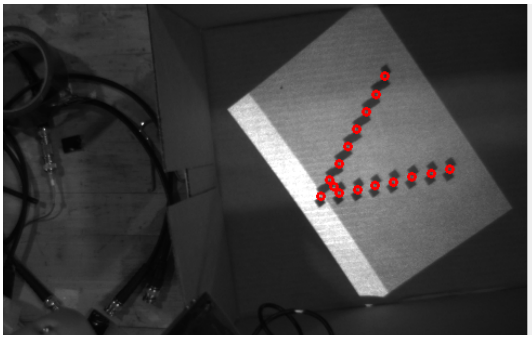
Correspondence assignment

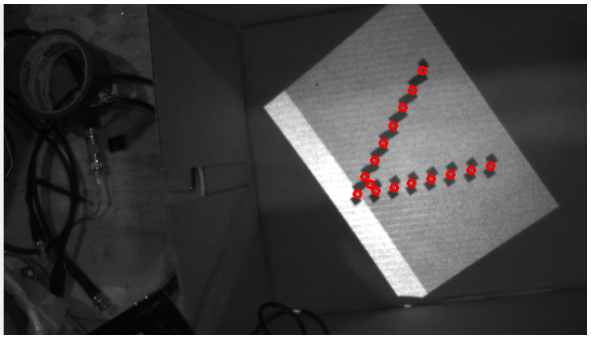
Detection Component MicronTracker Interface

JOHNS HOPKINS UNIVERSITY

- Components
  1. Virtual body grid
  2. Detection
  3. Processing
  4. Robot

- Input
  - Stereo images from MicronTracker (MT) hardware





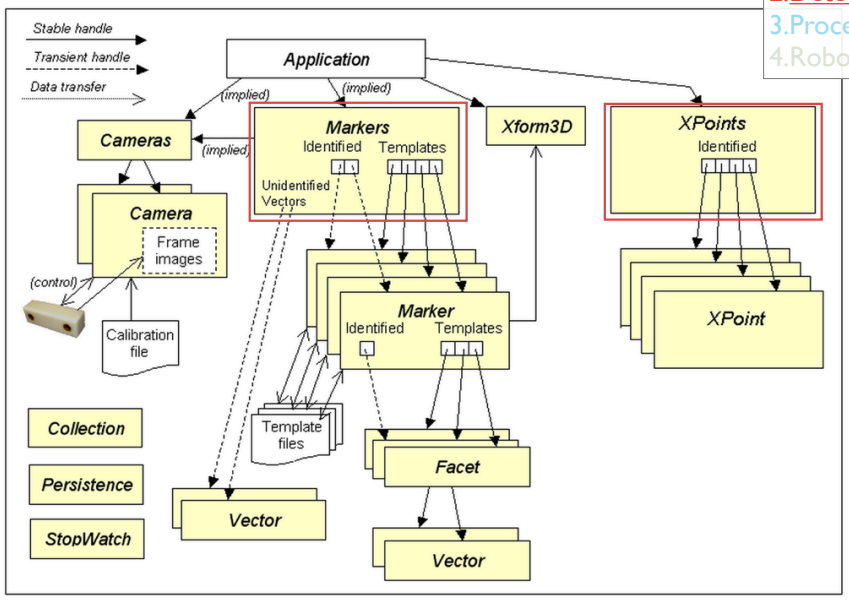
MUSiiC Research Laboratory · Center for IMAGING SCIENCE

Detection Component MicronTracker Interface

JOHNS HOPKINS UNIVERSITY

- Components
  1. Virtual body grid
  2. Detection
  3. Processing
  4. Robot

- Language/Library
  - C++/"MTC", accompanied with MT hardware



(Courtesy of Claron Technology Inc.)

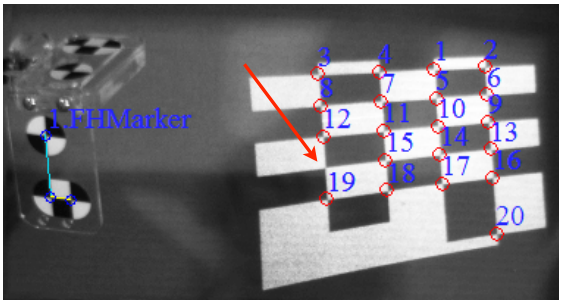
MUSiiC Research Laboratory · Center for IMAGING SCIENCE

Detection Component MicronTracker Interface

JOHNS HOPKINS UNIVERSITY

- Components
  1. Virtual body grid
  2. Detection
  3. Processing
  4. Robot

- Functionality :
  1. Detects projected virtual rigid body
     - "XPoints", intersection of two lines are detected.
  2. Detects physical rigid body
     - "Template marker" defined specific configuration of physical virtual body.
     - Template marker used for clear distinction from the detected virtual bodies.
- Multiple (30) frames are captured for a given pose since in a single frame,
  1. not all points are detected
  2. susceptible to noise/jitter



MUSiiC Research Laboratory

Center for IMAGING SCIENCE

Detection Component MicronTracker Interface

JOHNS HOPKINS UNIVERSITY

- Components
  1. Virtual body grid
  2. Detection
  3. Processing
  4. Robot

- Output: .txt of virtual and physical rigid bodies'
  - 3D positions in MT coordinate
  - 2D position in the image coordinate

marker_pose1.txt
marker_pose2.txt
marker_pose3.txt
marker_pose4.txt
marker_pose5.txt
marker_pose6.txt
marker_pose8.txt
marker_pose9.txt
marker_pose11.txt
marker_pose12.txt
xpoints_pose1.txt
xpoints_pose2.txt
xpoints_pose3.txt
xpoints_pose4.txt
xpoints_pose5.txt
xpoints_pose6.txt
xpoints_pose7.txt
xpoints_pose8.txt
xpoints_pose9.txt
xpoints_pose10.txt
xpoints_pose11.txt
xpoints_pose12.txt

Physical rigid body

Virtual rigid body

Number of detected points

(x3, y3, z3, x2, y2) - *3 : 3D position in tracker, *2 : 2D position in image

```
19
-60.880044, -168.596140, 827.952640, 323.996335, 67.972801
-138.034985, -162.876964, 838.508828, 249.081785, 74.732995
-1.985415, -145.721557, 825.847761, 381.799470, 90.586825
-72.818764, -138.290419, 833.994193, 311.779505, 97.460532
53.253668, -124.538215, 824.300700, 435.963554, 112.511571
-13.439937, -116.364054, 830.036436, 370.041858, 119.243188
-84.492008, -107.766944, 837.676231, 299.752916, 127.387193
-162.585143, -99.049044, 847.653320, 224.528801, 136.390879
41.811326, -95.501654, 823.156340, 424.865796, 140.382242
-24.916747, -87.120700, 834.935636, 358.131433, 147.892714
-96.401731, -77.584112, 843.926689, 287.715465, 157.082927
-175.314717, -66.063781, 850.996221, 211.952673, 168.097254
30.802009, -67.583347, 828.184496, 413.301029, 167.797696
-36.048707, -58.034232, 837.160441, 346.642159, 176.350194
-108.114547, -46.550988, 845.605271, 275.727382, 187.206551
20.122966, -39.436150, 832.046599, 402.079617, 195.722951
-47.462460, -28.088445, 841.837715, 334.783350, 206.015182
9.163013, -10.527956, 833.586507, 390.727150, 224.045411
-12.347978, 47.076978, 844.617281, 367.767791, 280.294380
16
-125.922399, -194.932536, 833.531178, 261.368049, 43.931647
-60.829637, -168.515079, 827.604713, 324.045435, 67.986596
-1.926094, -145.780408, 825.883580, 381.855949, 90.458395
-72.817363, -138.386176, 834.137299, 311.784247, 97.418636
-150.218032, -130.393416, 841.232252, 236.613746, 105.664084
53.135217, -124.257499, 822.738923, 436.056172, 112.637278
-13.527076, -116.222024, 828.892325, 370.019489, 119.180813
-84.223642, -107.484461, 835.936905, 299.966497, 127.368692
-162.823639, -99.087978, 847.813596, 224.321489, 136.257002
41.960399, -95.649538, 823.582693, 424.962493, 140.137724
```

frame 1

frame 2

MUSiiC Research Laboratory

Center for IMAGING SCIENCE

Detection Component MicronTracker Interface

- *"Multiple (30) frames are captured for a given pose since in a single frame,*
  1. *not all points are detected*
  2. *points are susceptible to noise/jitter"*

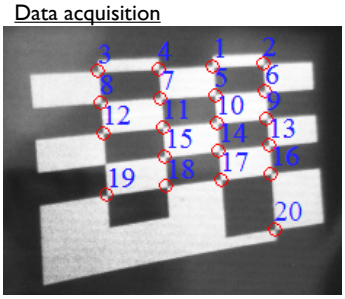
- Implication :
  - Dynamic accuracy evaluation is difficult with current setup.
  - This project will be confined on *static* poses.

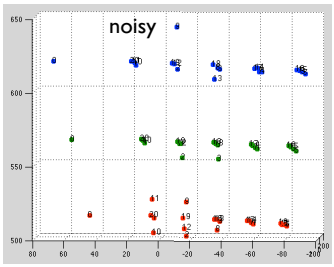
- Why bother to use MicronTracker?
  - Provides a unified platform to analyze the performance of two rigid bodies.

•Components
1.Virtual body grid
2.Detection
3.Processing
4.Robot

Processing component Tracker Data Processing

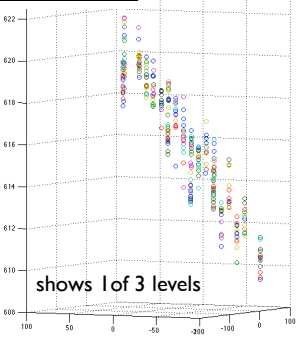
Data acquisition





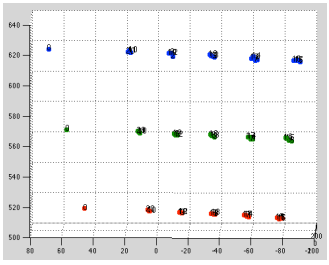
collate
30 frames

Frame collation

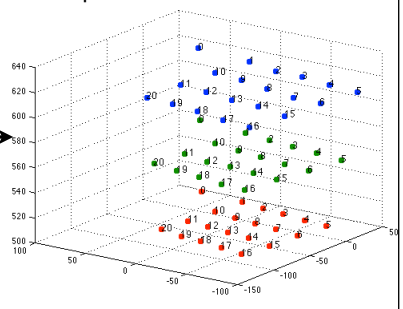


truncate
z > 2,
average

Outlier rejection



Correspondence



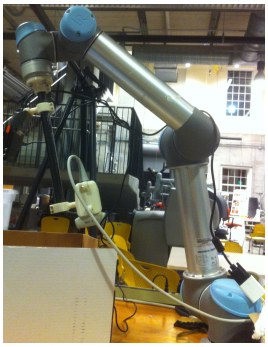
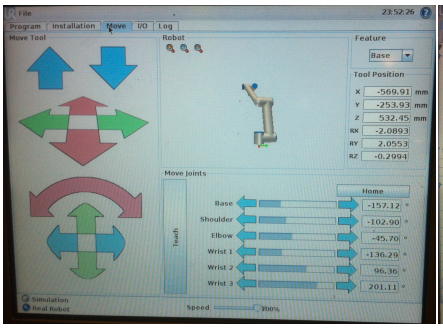
•Components
1.Virtual body grid
2.Detection
3.Processing
4.Robot

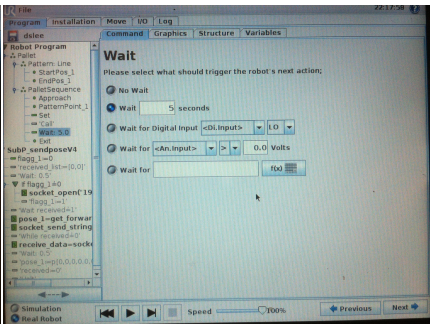
Robot component UR5 Control

JOHNS HOPKINS UNIVERSITY

- Components
  1. Virtual body grid
  2. Detection
  3. Processing
  4. Robot

- Current paradigm: GUI Controller
  - Input of desired 1) robot tool pose or 2) joint angles
  - Real-time pose and joint angle calculation
  - Programmable motion trajectories / actions
    - Very useful for Cartesian motion

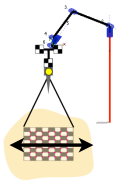
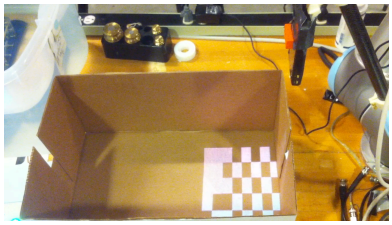
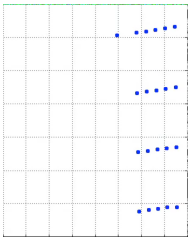


- Room for improvements : systemic, automatic control
  - TCP/IP communication with PC
  - Implementation being attempted on Matlab.

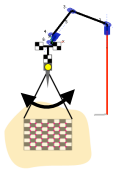
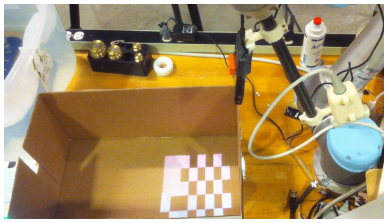
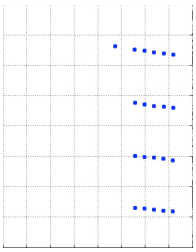
Experimental Demo Video

JOHNS HOPKINS UNIVERSITY

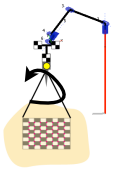
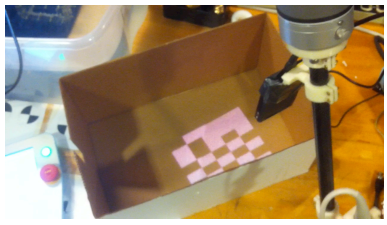
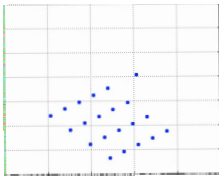
Translation

Rotation

Rotation

Dependency Check

- Hardware
  - MicronTracker
  - Universal Robots robot arm and controller
  - Robot - projector adapter ⚠ Broke, reprinting by Apr. 4th
  - Laptop
- Commercial Software
  - MicronTracker SDK
  - Universal Robots control system
- Internal algorithm and software
  - Pose estimation of the projector given the coordinates of markers
- Miscellaneous
  - Access to Hackerman hall Robotorium

Key Dates - Original

- Feb. 28th :
  - Literature study, training for UR5 control
  - Virtual marker grid development
  - Resolve dependencies
- Mar. 15th :
  - Develop and document a package of routines to acquire data from the MicronTracker and UR5 robot.
- Mar. 31st :
  - Experimental design (robot arm trajectories, marker shapes, etc.)
  - Data gathering for fixed pose and a trajectory of poses
  - *Minimum deliverables*
- Apr. 15th :
  - Analyze and determine optimal marker parameters.
  - *Expected deliverables*
- Apr. 30th :
  - Further experiments, such as on non-level surfaces, publication
  - *Maximum deliverables*
- May 9th : Poster presentation, final report

Key Dates - Modified

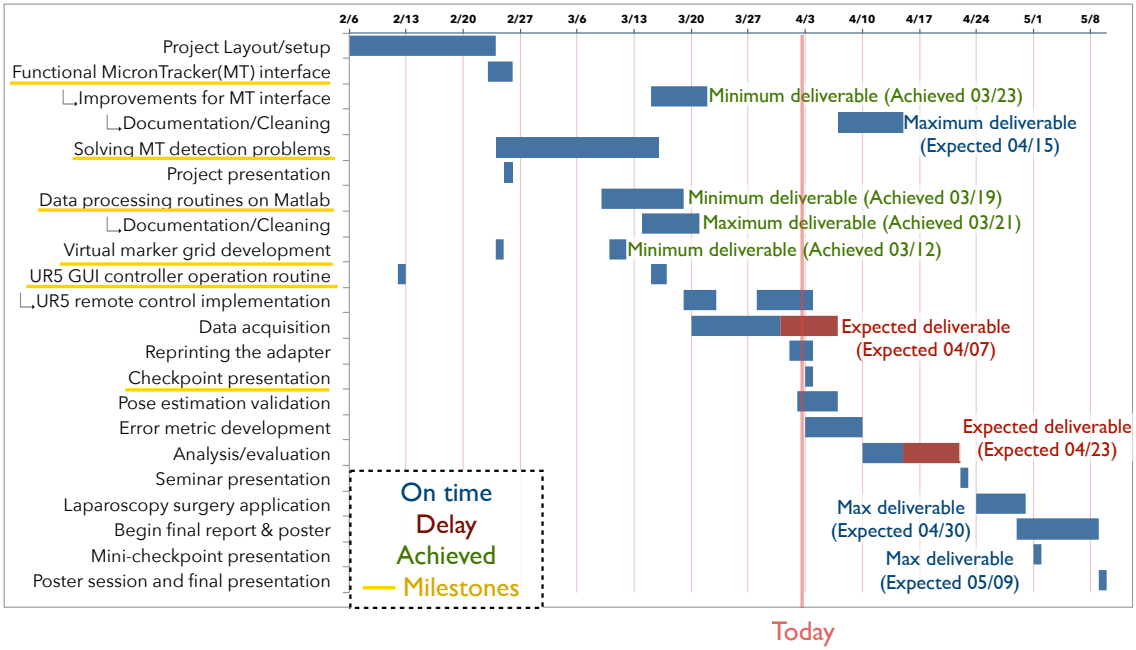
JOHNS HOPKINS UNIVERSITY

- Feb. 28th :
  - Literature study, training for UR5 control ✔
  - Virtual rigid body grid development ✔
  - Resolve dependencies ⚠ Re-print adapter (by 04/04).
- Mar. 15th :
  - Develop experimental pipeline ⚠ (by 03/23) ✔
- Mar. 31st :
  - Experimental design (robot arm trajectories, marker shapes, etc.) ✔
  - Data gathering for fixed pose and a trajectory of poses ⚠ (by 04/07) ←
  - *Minimum deliverables* ✔
- Apr. 15th : ⚠ (by 04/23)
  - Analyze and determine virtual rigid body design conditions. ←
  - Compare virtual and physical rigid body accuracy
  - *Expected deliverables*
- Apr. 30th :
  - Demonstration package for application to laparoscopy, documentation
  - *Maximum deliverables*
- May 9th : Poster presentation, final report

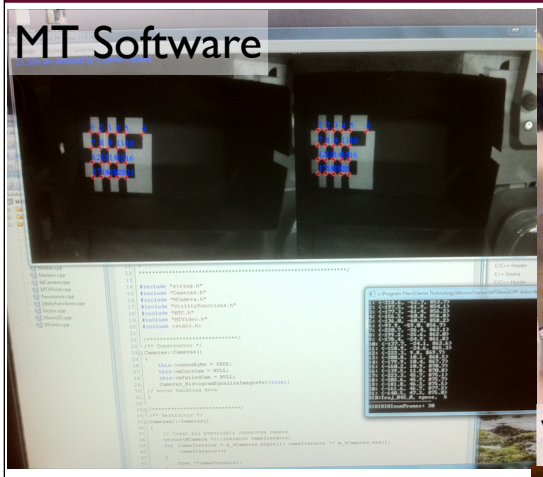
✔ complete
⚠ delayed
← Right up next

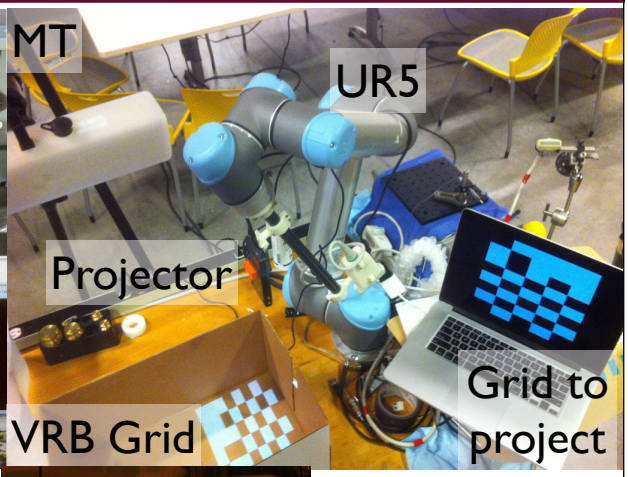
Gantt Chart

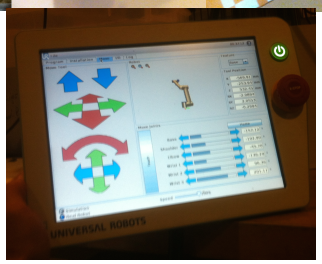
JOHNS HOPKINS UNIVERSITY



Questions?







GUI Controller