cisis

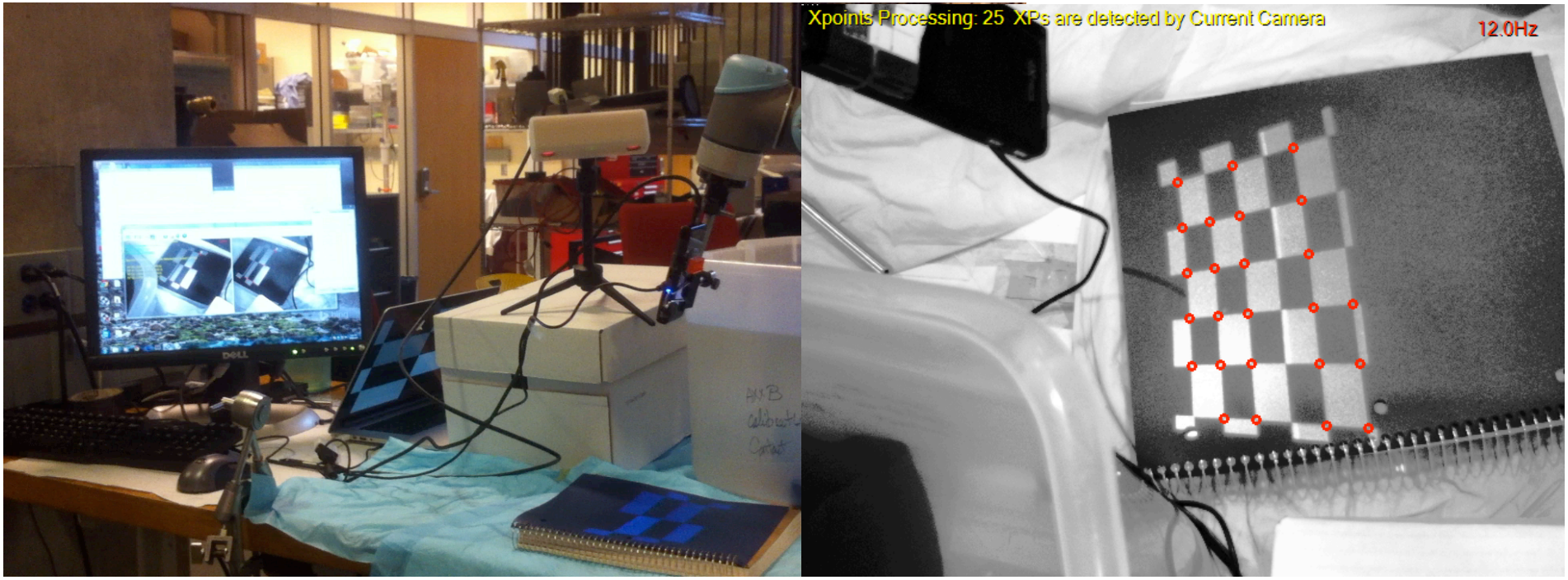
Virtual Rigid Body

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

Mar. 12th, 2014

JOHNS HOPKINS
UNIVERSITY

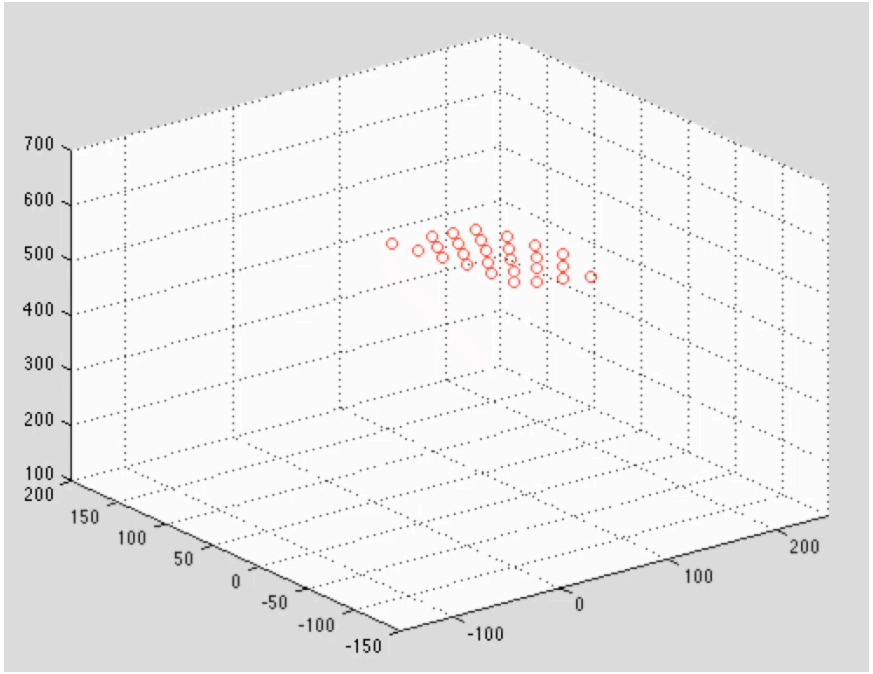
Workflow



Experimental setup and marker detection

Workflow

```
29
94.581570, -103.950056, 586.838261
76.260752, -97.641019, 587.473167
59.306907, -92.036152, 590.983363
43.350701, -86.590952, 590.826642
119.145527, -77.119750, 574.692356
99.167042, -71.791788, 577.039642
80.919145, -66.922308, 580.424736
63.419072, -62.476997, 581.353793
47.601265, -58.428383, 582.936474
123.281246, -44.821844, 565.032086
103.012918, -41.071661, 568.493978
84.395644, -37.318786, 570.333761
67.010890, -34.073111, 572.765094
50.867571, -31.057794, 576.016991
126.168403, -13.087981, 553.560718
105.443632, -10.487923, 555.144112
87.141610, -8.283836, 559.298869
69.747931, -6.062896, 562.619896
53.483385, -3.856182, 565.169732
108.537607, 20.006199, 546.073199
128.673360, 19.046584, 540.443705
89.440753, 21.115167, 548.192208
55.541320, 23.338747, 554.666139
72.024008, 22.394325, 552.363469
57.242045, 51.648171, 544.190459
73.817301, 51.604382, 540.177736
91.437573, 51.496203, 536.186557
131.523785, 52.233584, 528.259340
110.506676, 51.738047, 531.990615
28
50.312652, -94.219807, 592.272077
31.577420, -88.454151, 595.798855
14.223390, -82.550305, 596.545417
-2.199017, -76.989156, 597.264512
54.872189, -61.504416, 583.655818
35.975557, -56.841753, 585.262976
18.292301, -52.184265, 586.881452
```



Output: (x,y,z) coordinate of detected points

The points parsed and read into MATLAB

Next step

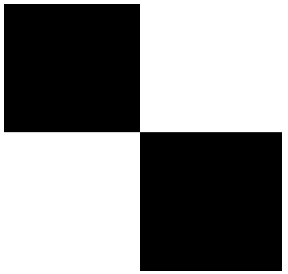
- Point collection and assigning correspondence
  - Flickering poses some difficulties.
    - Collect multiple frames and locate as many points as possible.
    - This implies "static" trajectories, i.e., moving, stoping, detecting.

- Experimental design
  - Marker shape, size, number
  - Poses, Motion trajectories

- Flexibility and integration of the code, and automation
  - Current workflow is usable but very fragmented.
    - Robot : Python script + UR5 script
    - Optical tracker : C++
    - Parsing, pose estimation : MATLAB
  - Plan to at least partially integrate and automate them.

Correspondence

- A note: requirement for detection
  - Contrast between two groups
  - Similarity between each member of the group



Detected

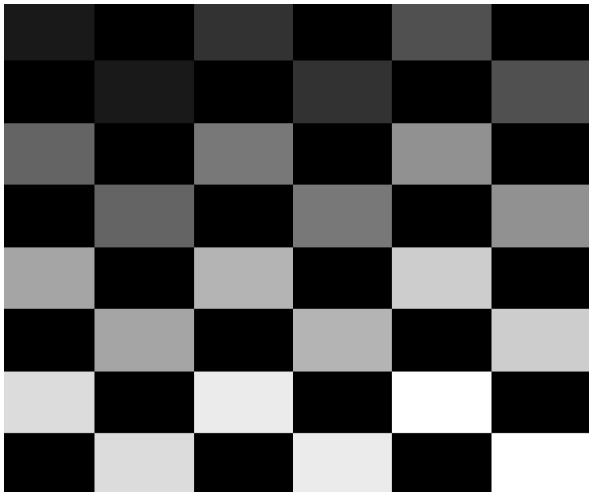
Undetected

- Intensity?
  - Sample intensity in the neighborhood.
  - The camera does not seem too sensitive to these differences.
  - Perhaps controlling the camera parameters such as refresh rate and exposure is needed.

Detected

Undetected

- Color?
  - MicronTracker seems to support RGB somewhat, but to what extent is unknown.



- Distance?
  - The first option considered but this metric is not robust for non-smooth surfaces.

Experimental Design

- Marker triangle
  - Shape
    - Obtuse? acute? right angle? equilateral?
  - Size

- Robot
  - Stationary pose
    - Compare recovered poses
  - Trajectories of poses
    - Compare the recovered trajectories by curve fitting
    - Two types:
      - Static : move, stop, record
        - This will be the first approach, considering the flickering problem
      - Dynamic : real-time recording of poses