Automatic Performance Analysis in Trampoline from RGB-Depth Data

Carlos Puig Toledo

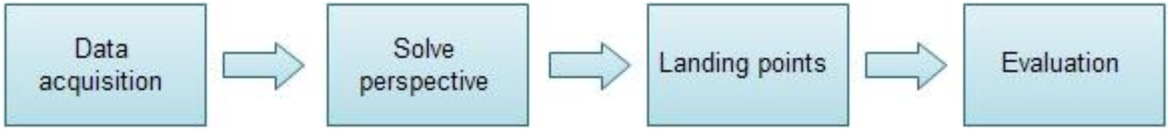
Advisors: Sergio Escalera, Jordi González, Xavier Baró and Albert Clapés

Outline

- Introduction
  - The project
  - The trampoline sport
  - Kinect
- Method
  - Data acquisition
  - The perspective problem
  - Finding the landing points
- Results
- Conclusions

Project Introduction

- In collaboration with CAR Sant Cugat and ASCAMM
- Objectives:
  - Capture multi-modal RGB-Depth data
  - Extract the landing points
    - Transform perspective view
    - Estimate jump location
  - Evaluate



The trampoline sport

The trampoline sport

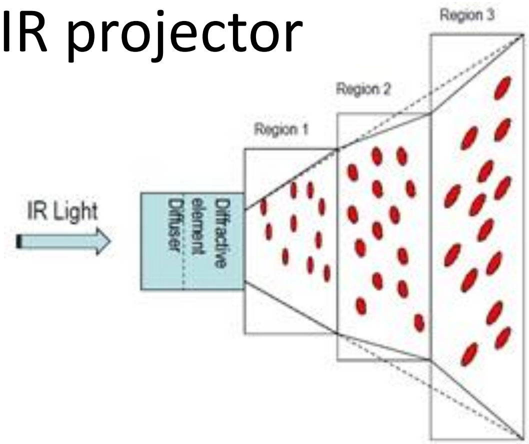
Chair of judges deductions	
Touching the bed with 1 or both hands	0.4
Touching the bed with knees or hands & knees, falling to seat, front or back	0.6
Touching the springs, pads, frame or safety platform	**0.6**
Landing/falling on the springs, pads, frame or safety platform & spotter mat	**0.8**
Landing/falling outside the area of the trampoline	**1.0**

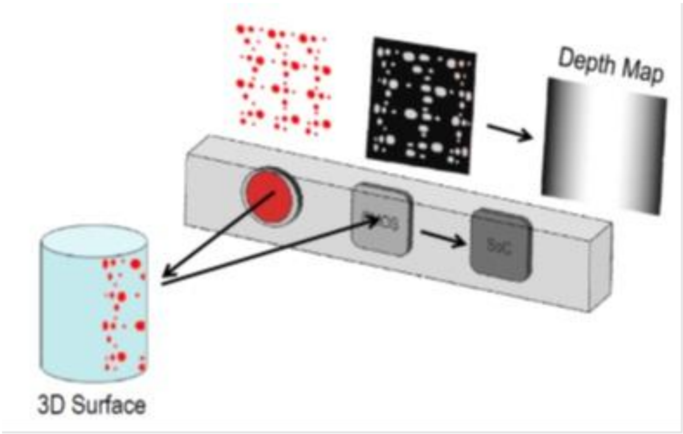
What is Microsoft Kinect?

- Structured light scanner
- Packaged into a single unit:
  1. Depth sensor:
     - Infrared laser projector with monochrome CMOS sensor
     - 11-bit VGA depth (2048 levels of sensitivity)
     - Any ambient light condition
     - 1-8 meters range
  2. RGB camera
     - 8-bit VGA resolution (640x480)
     - 30 FPS
  3. Motorized tilt
  4. Multi-array microphone

How it works?

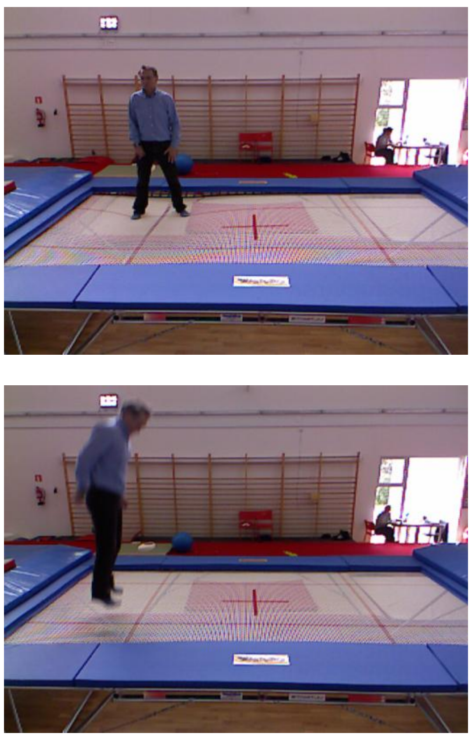
- Speckle pattern of dots is projected by IR projector
- IR camera captures the pattern
- For each dot in reference pattern:
  - Find the dot in the observed pattern
  - Compute the depth:
    - Disparity
    - Focal length
    - Baseline between projector and camera

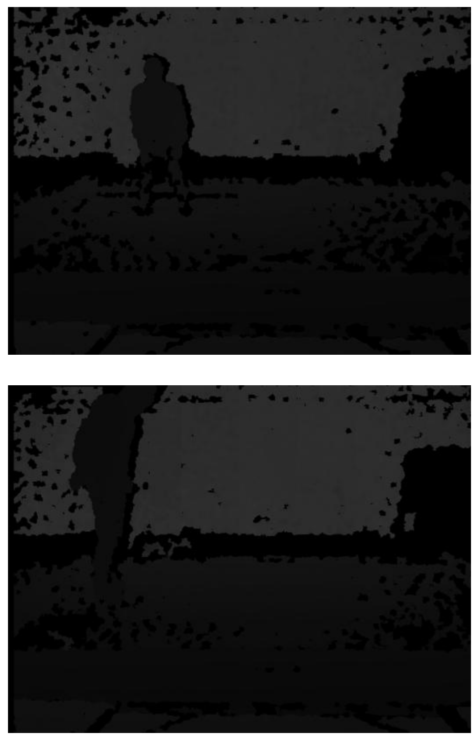


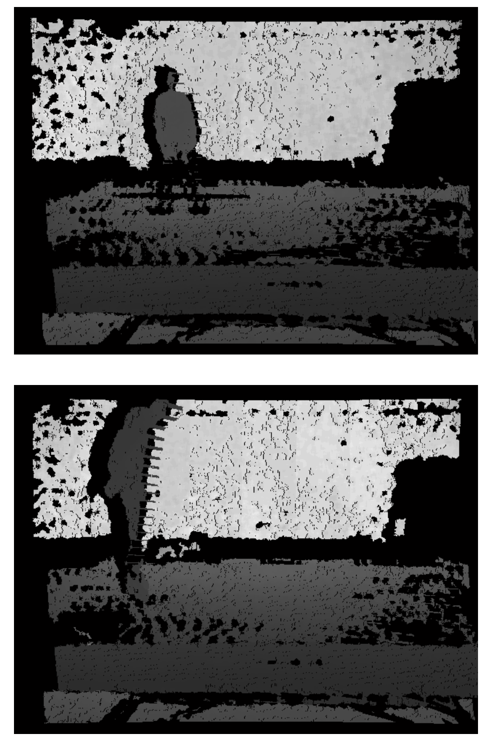


Data Acquisition

- Setup the camera
- RGB-Depth Syncronization and Registration

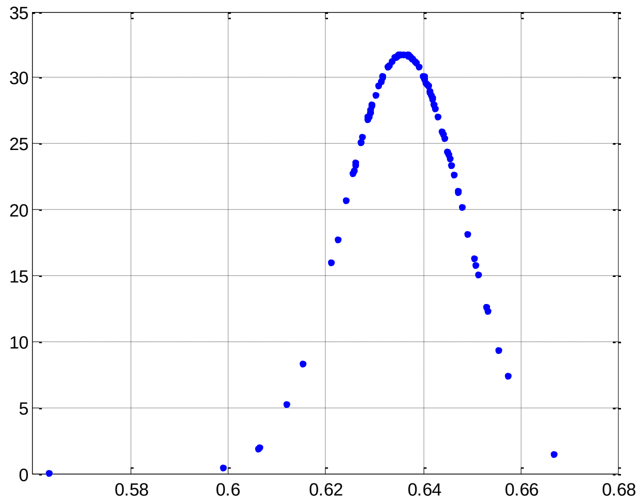

RGB

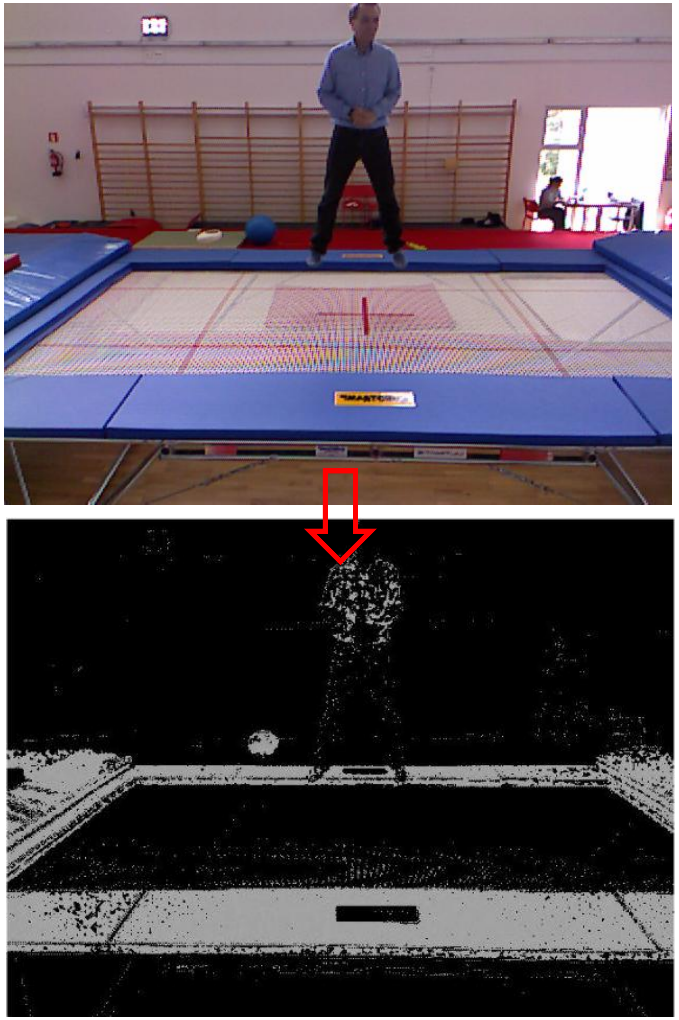

Depth


Depth registration

The perspective problem

- Mesh not occluded
- Segment blue mat
  - GMM in Hue channel

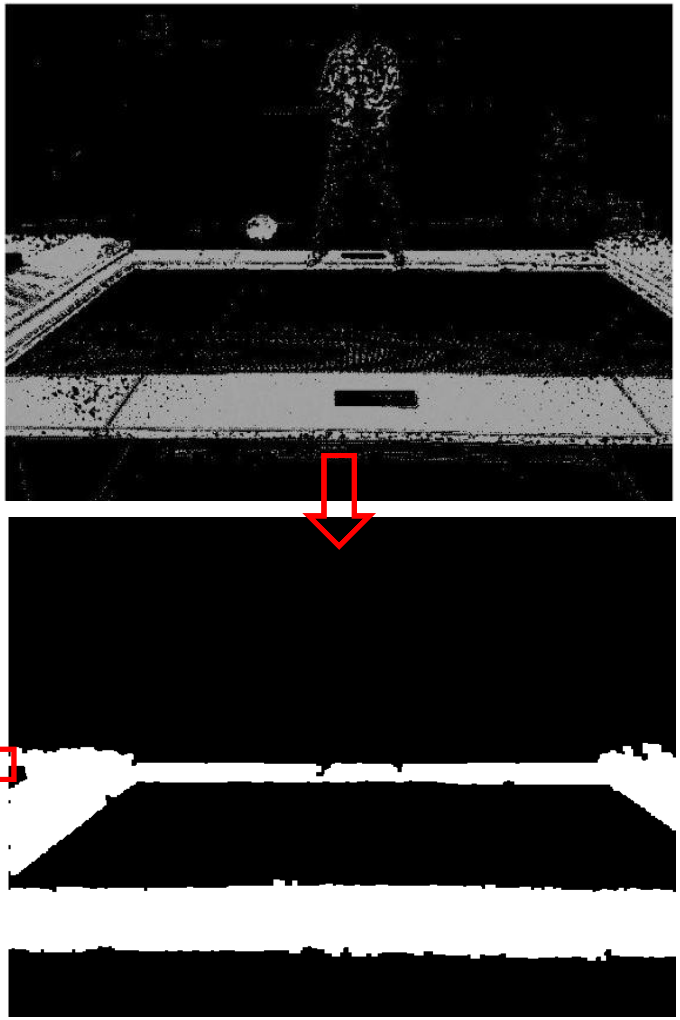




The perspective problem

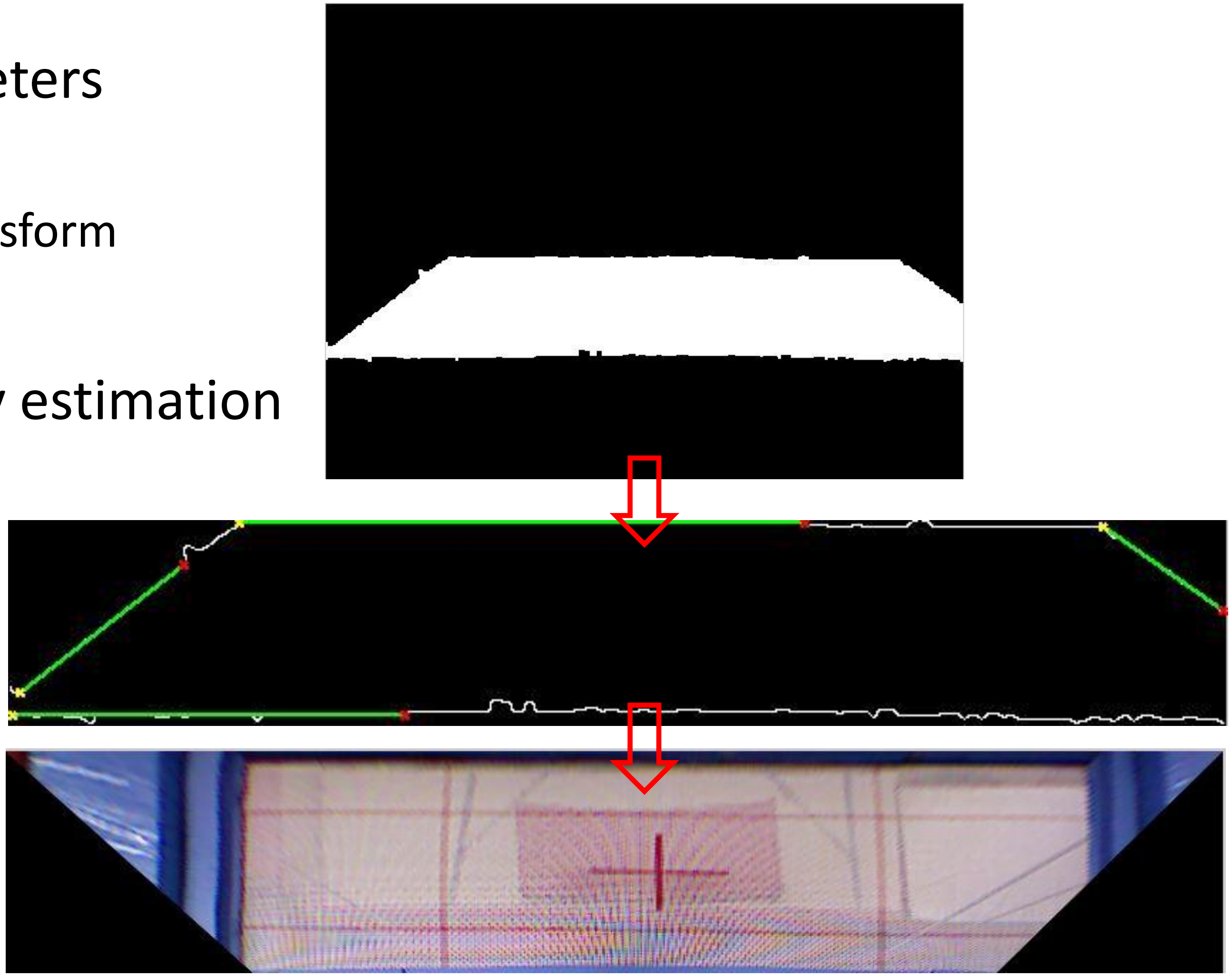
- Morphological operations
  - Opening
  - Closing

- Flood-fill algorithm



The perspective problem

- Edge parameters
  - Canny
  - Hough transform
- Homography estimation

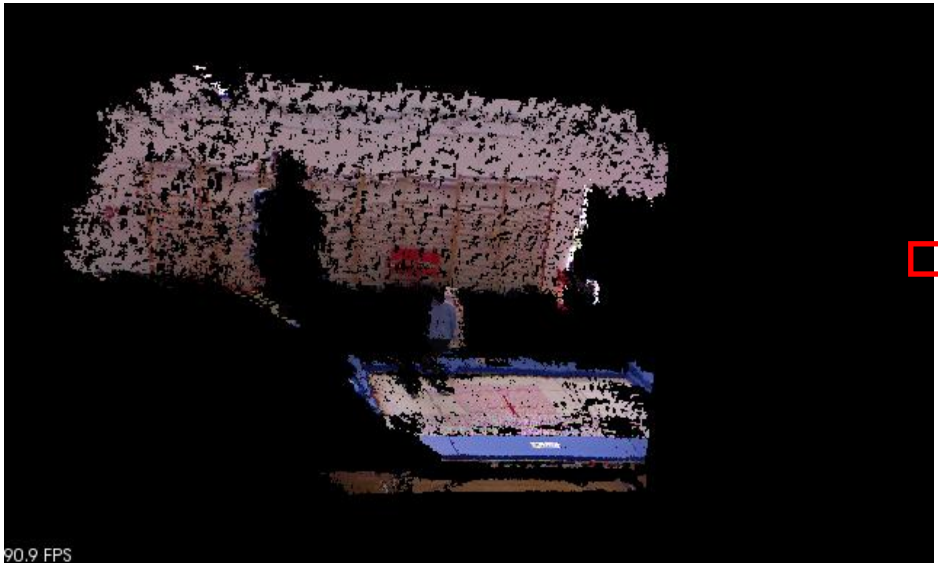


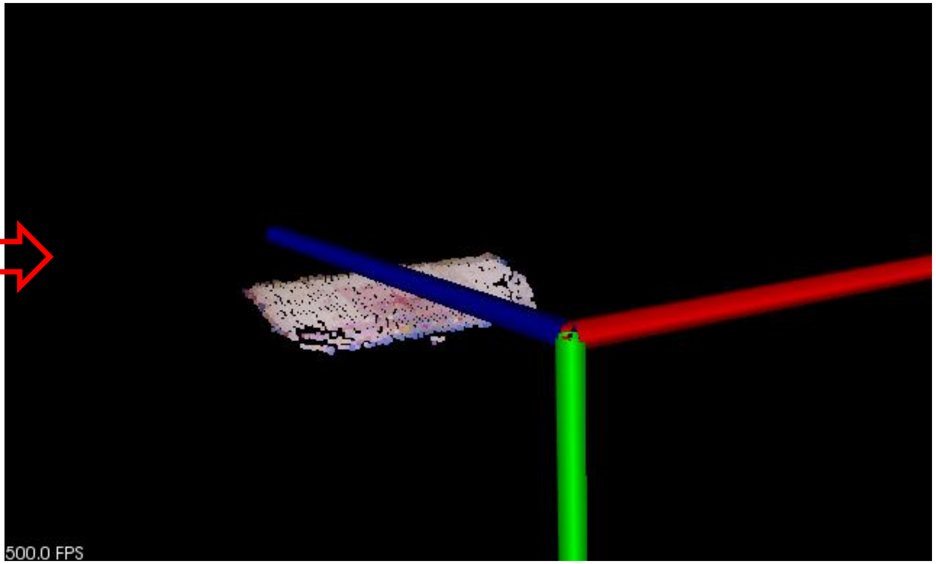
Finding the landing points

- Construct point cloud from depth map
- Segment the blue mat
- Extract the mesh

$$inverse\ focal = \frac{\frac{1}{285.63}}{\frac{image\ width}{320}}$$

$$Real\ world\, x = (x - image\ width) \cdot inv.focal \cdot z$$
$$Real\ world\, y = (y - image\ height) \cdot inv.focal \cdot z$$
$$Real\ world\, z = z$$



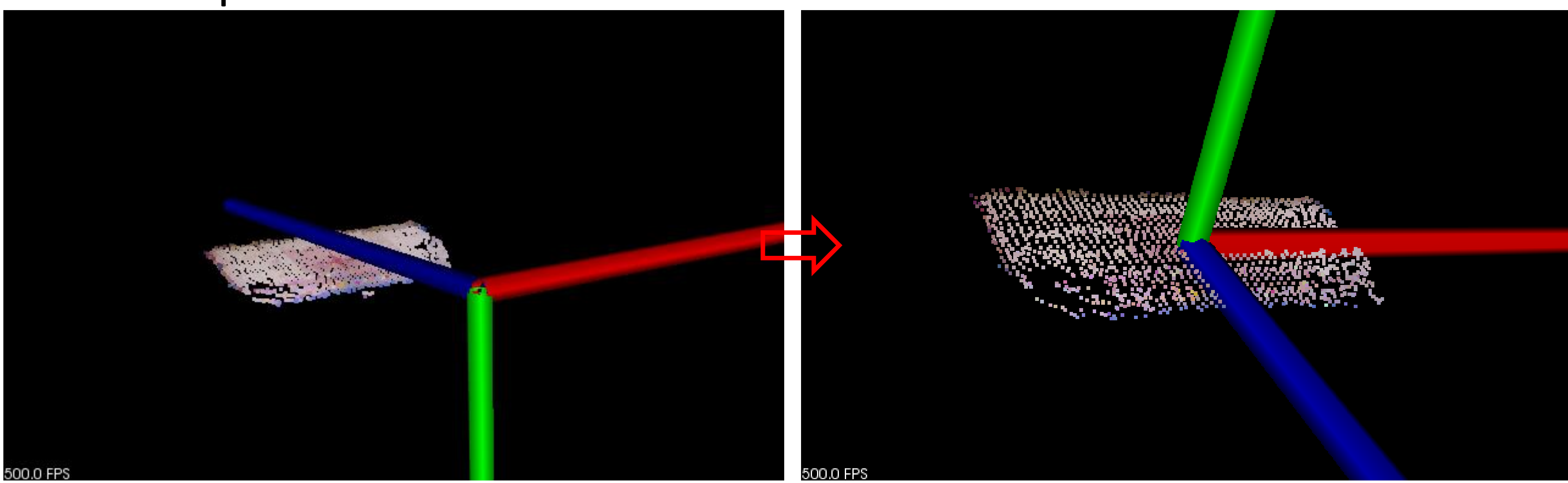


Finding the landing points

- Estimate plane parameteres with planar segmentation
- 3D transformations
  - Translate to the origin
  - Rotate to match x-z plane
- Cropbox filter

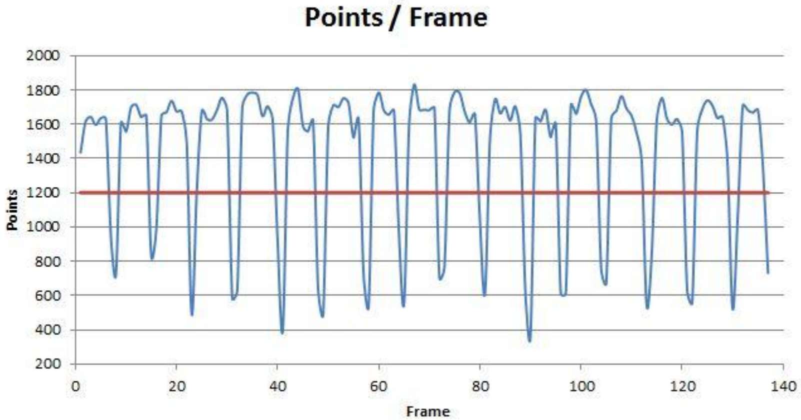
$$ax + by + cz + d = 0$$

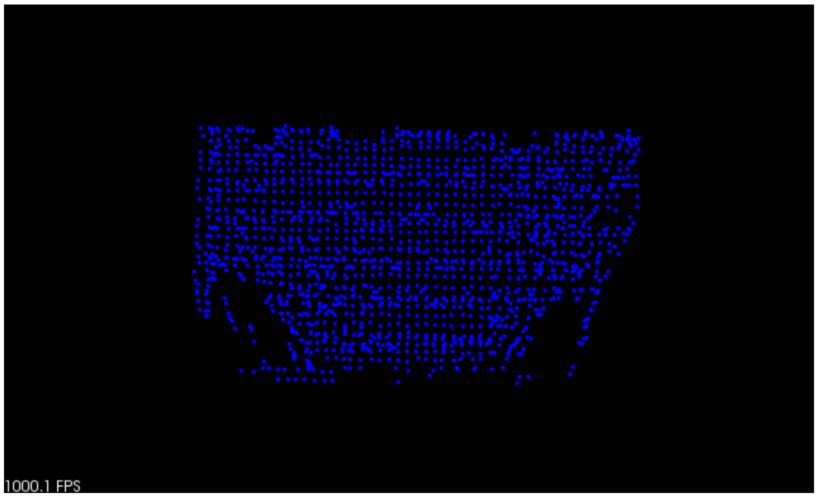
$$\boldsymbol{\theta} = arccos\left(\frac{x \cdot y}{|x| \cdot |y|}\right)$$

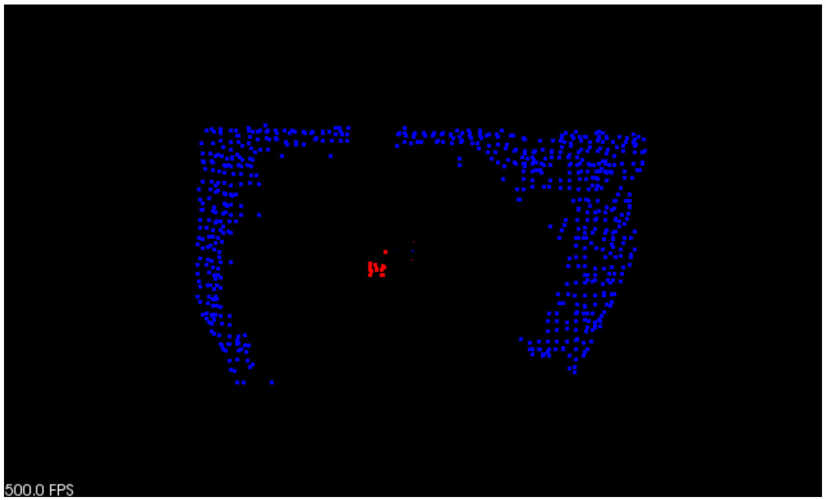


Finding the landing points

- Estimate the landing point
  - When?
  - Where?
    - Euclidean cluster extraction

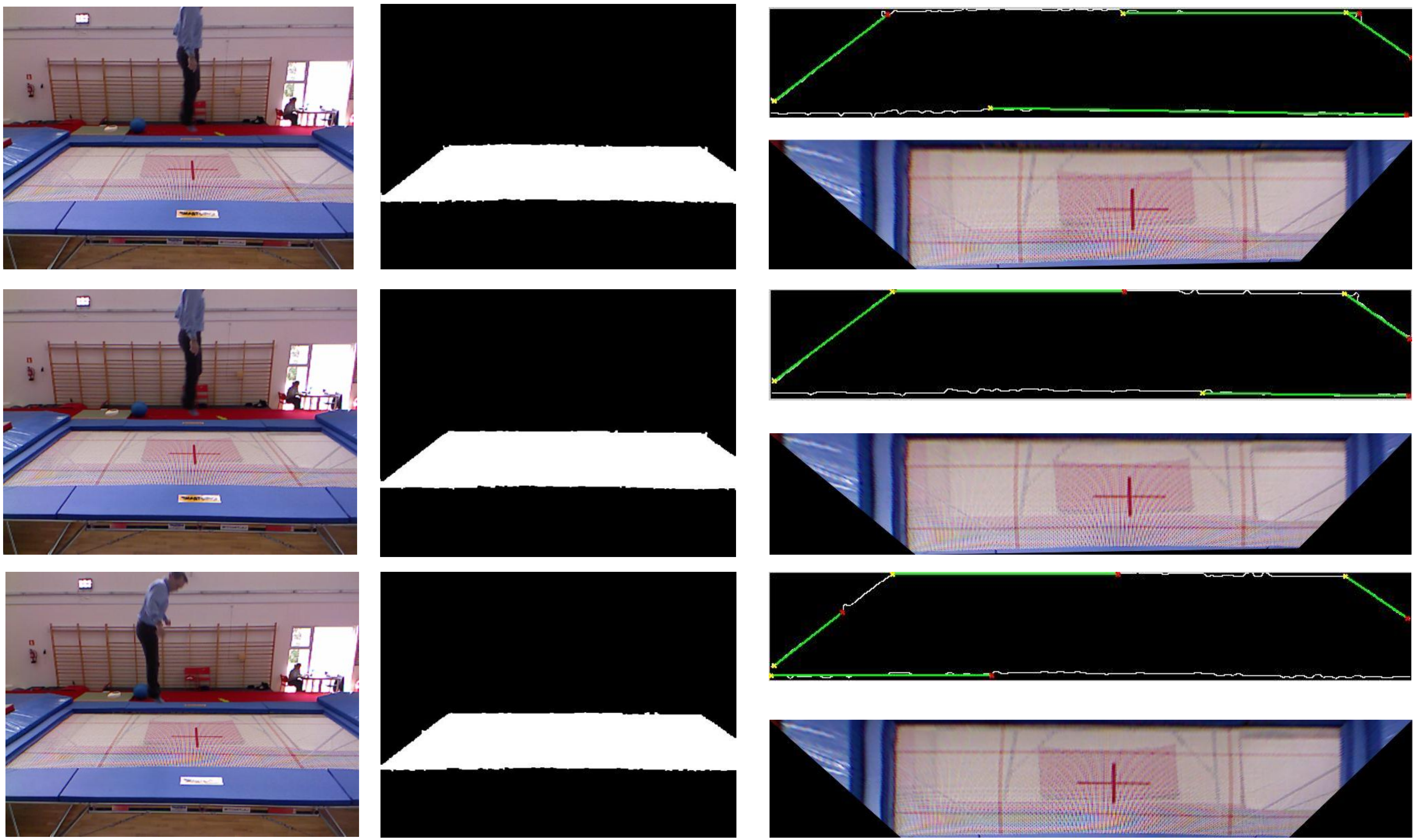




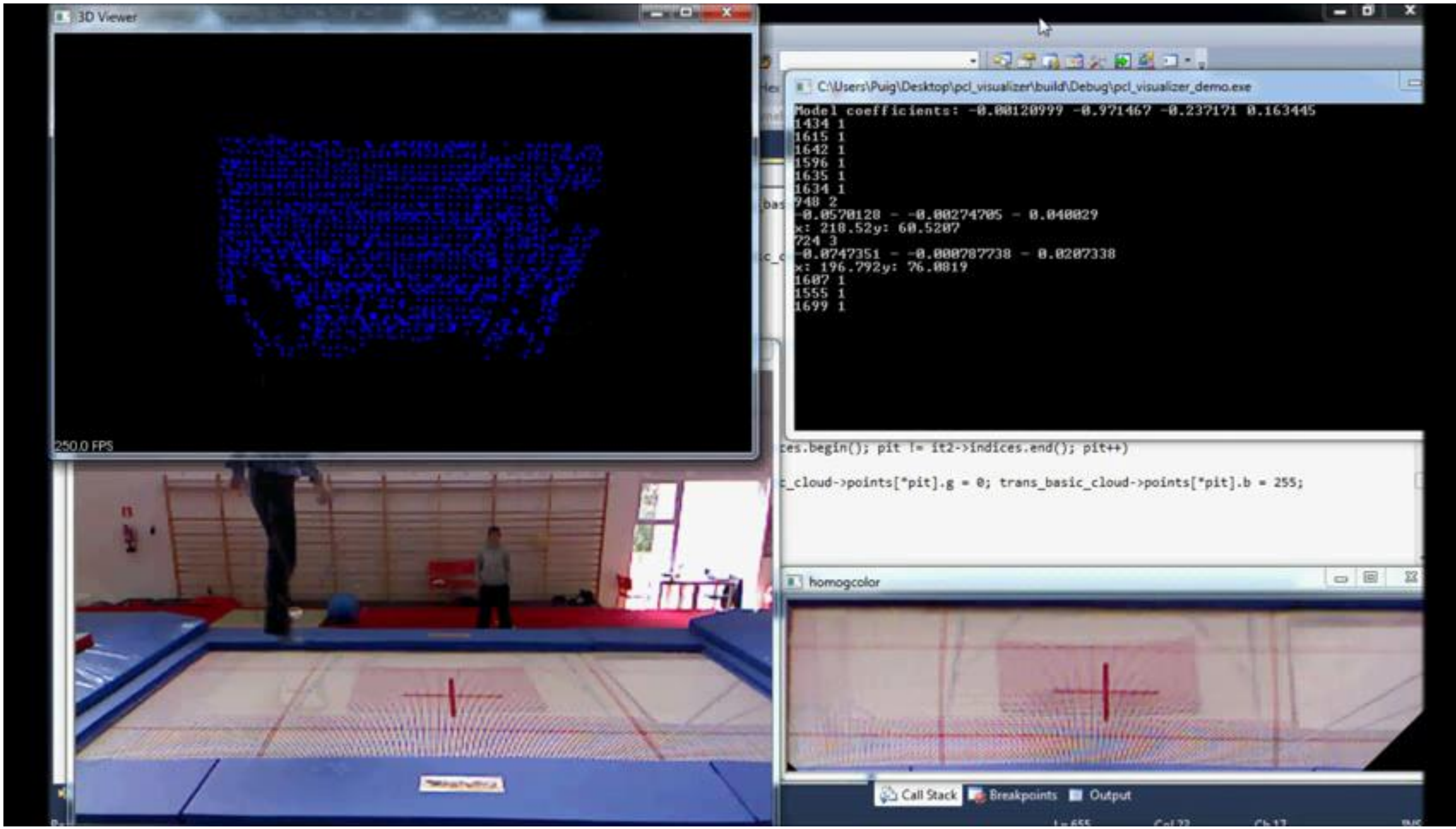


Results

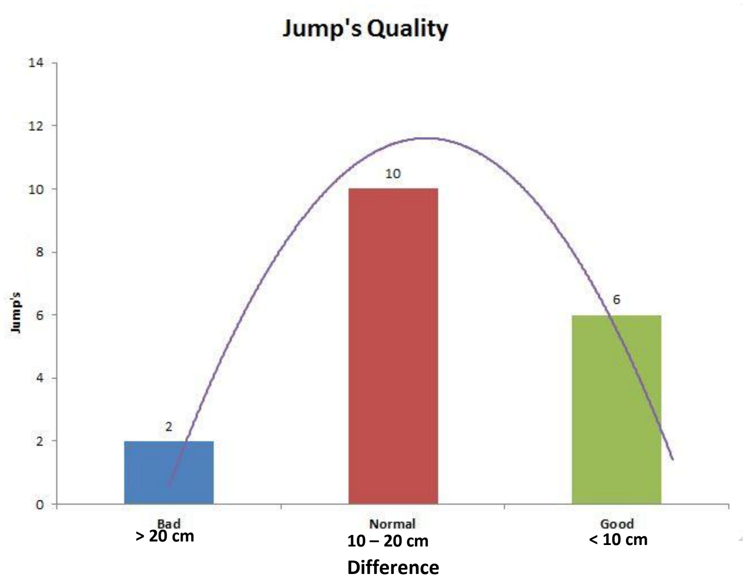
Perspective

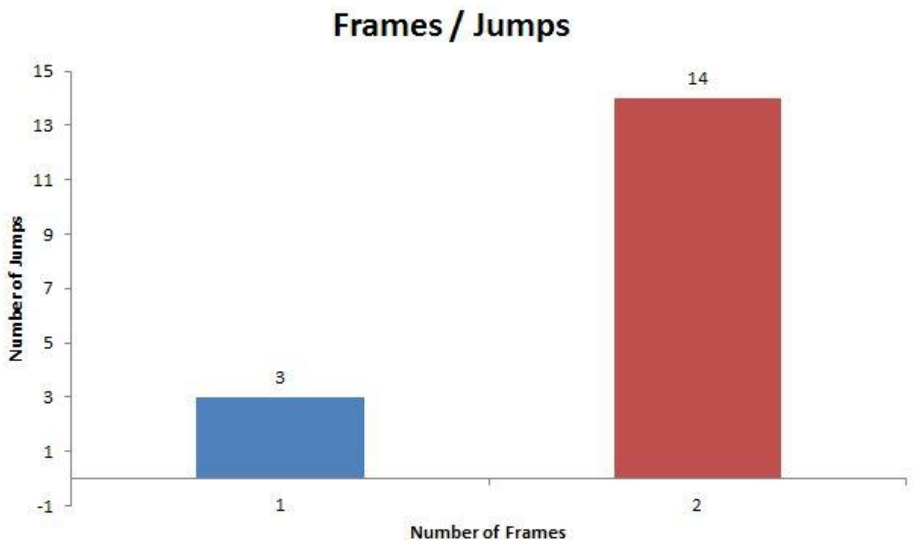


Landing points



Landing points





Conclusions

- Landing points are estimated from multi-modal RGB-Depth data from a Kinect camera.
- Athletes can automatically know their performance.
- Robust system that accomplishes the requirements defined by athletes, trainers and judges.
- Cheap and easy to setup.
- Volunteer performance vs. Athlete performance.

Future work

- Perform more field tests.
- Detail some requirements.
- Make the system more robust to changes in the scenario.
- Increase the frame rate.
- More specifications could be added:
  - Athlete position at landing.
  - History of jumps with each landing location.


Thanks for your attention!