# Volume estimation via integrating on a curve fitted point cloud

**HiPEDS 2018 Cohort:** G. Bisbas, L. Castiglione, D. Grumberg, S. Karolčík, L. Keeble, D. Kulon, B. Kwan, C. McMeel, R. Miles, J. Ortiz, N. Perez-Nieves, V. Pham Ngoc, J. Vandebon, D. Vink

Imperial College London



Imperial College London

October 29, 2018

# Overall structure

**HiPEDS Group workflow**

- Cohort meetings on a regular basis
- Identify our goals and split into subgroups
- Integrate our progress
- Redefine goals

**Point cloud integration team overall checkpoints**

- Capture images
- Extract point cloud
- Fit a curve
- Find the volume inside

*More details in the next slides...*

# The problem and the goal

Figure: Royal vans. *Courtesy of https://www.pressandjournal.co.uk*

- **Problem:** Packaging in vans is not optimal → lots of empty space
- **Goal:** Fast estimation of available volume to ensure optimal packaging

# The hardware - Choosing the camera fitting our needs best

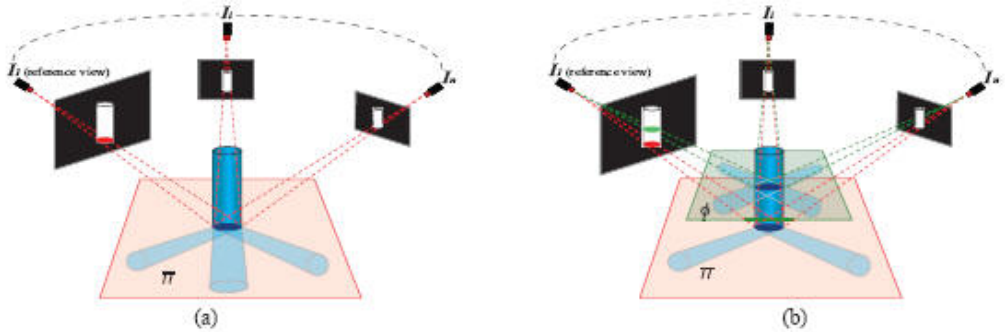


The Realsense *D*435 has been proved to be the best choice.

- \+ Relatively cheap.
- \+ Independent from lighting conditions.
- \- Not providing Real Time Information (External setup). ?????
- \- Still expensive++++.

# Extracting the point cloud: PLYExtractor

Using the RealSense SDK, a piece of software has been written to automatise the process of extracting pointclouds from different points of view



# PLYExtractor: Activity Flow

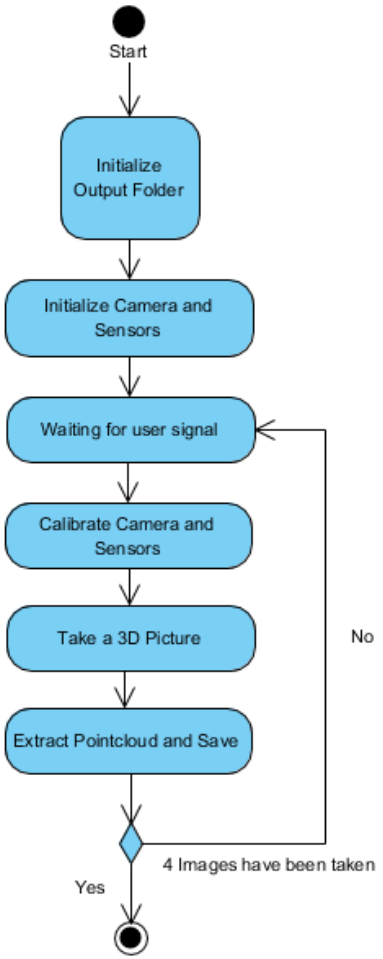



Figure: Software Activity Flow

# PLYExtractor: in Action

(a)

(b)

# Denoising the point cloud

Extracted point clouds are often very noisy, so, a need for removing outliers is needed.

- Step 1: Hard-cut denoising of points that are very far away of the mass centre.
- Step 2: Use a nearest neighbours approach with a predefined threshold to remove more outliers.

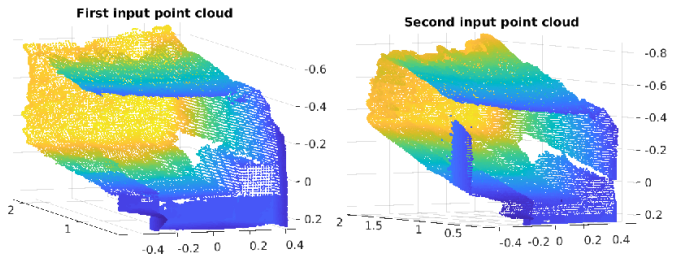
Resulted point clouds were found to be **95%** correctly denoised via simple parameter tuning.

# The ICP algorithm

The Iterative Closest Points algorithm was used to merge the different point clouds extracted from the camera.
For each point in the source point cloud, match the closest point in the reference point cloud (or a selected set).

1. Estimate the combination of rotation and translation using a root mean square point to point distance metric minimization technique which will best align each source point to its match found in the previous step. This step may also involve weighting points and rejecting outliers prior to alignment.
2. Transform the source points using the obtained transformation.
3. Iterate (re-associate the points, and so on).

# Merging point clouds




**Merging procedure:** 2 steps
1st step: 4 to 2
2nd step: 2 to 1

- aaaaa

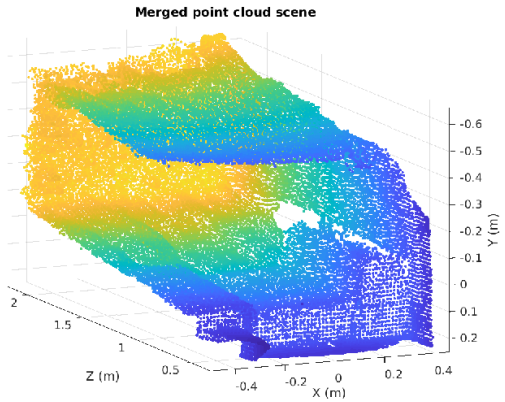



Figure: Intermediate merged pointclouds (left) and final point cloud (right). Denoised and downsampled version of final point cloud (bottom).

# Curve fitting with Linear Interpolation

Using MATLAB's Curve Fitting Toolbox, we managed to

- Fit a polynomial surface
- Step 2: Use a nearest neighbours approach with a predefined threshold to remove more outliers.

Resulted point clouds were found to be **95%** correctly denoised via simple parameter tuning.

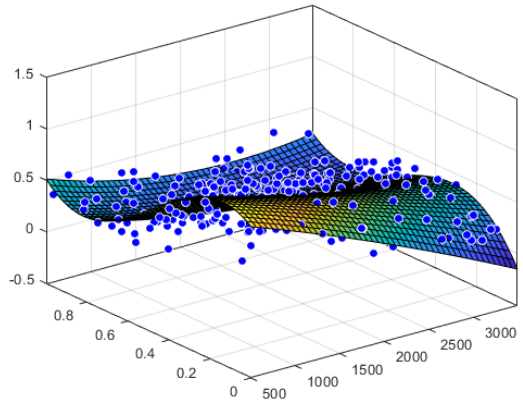



Figure: A simple curve fitting

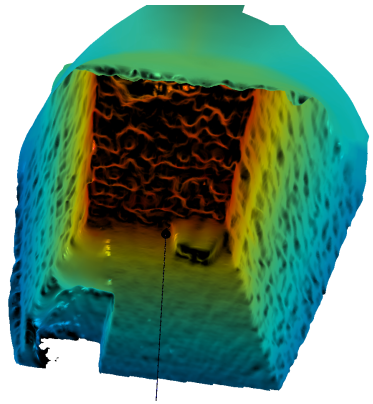


Figure: A full fitted surface

# Integration - Keep it simple scientist

$$\iiint_V f(x, y, z)\, dx\, dy\, dz \qquad (1)$$

$$\iiint_V f(x, y, z)\, dx\, dy\, dz \qquad (2)$$

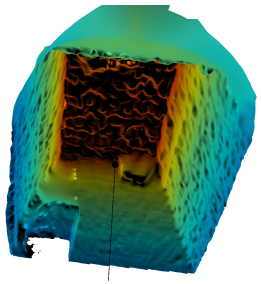


Figure: A full fitted surface

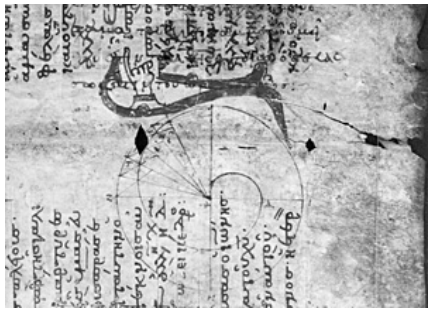


Figure: Pre-Calculus of Archimedes

# Results

# References

# Acknowledgements

This project was proposed and supported by Royal Mail, EPSRC and Imperial College London. A special thanks to Jeremy Bradley and Ben Glocker for their support and advice throughout.