# 3D Reconstruction with Depth Camera

Carleton University

COMP5115

Prof. Oliver Van Kaick

Faubert, Joël
jfaub011@uottawa.ca

December 9, 2019

# 1 Objective

This project investigates techniques used to produce 3d models using consumer grade depth sensors. Specifically, the ultimate goal is to use an Intel RealSense D435 camera (the RS4XX) and its accompanying Software Development Kit (the RealSense SDK) to produce high-quality 3d models.

Advanced techniques pioneered by Kinect Fusion [5] build models incrementally by merging (or fusing) a series of depth images into a common frame of reference. The registration problem – aligning two overlapping point clouds – is central to this technique. Therefore, this project studies the following:

- the stereoscopic depth camera's inner workings,
- the use of the Intel SDK to produce point clouds, and
- how to use tools such as MeshLab to align these point clouds.

# 2 Methodology

The following section describes the key concepts and algorithms for 3D reconstruction from depth images.

## 2.1 Depth Images

The RS4XX produces a four-channel image composed of the standard RGB colour channels together with a fourth depth channel.

The depth channel associates to each pixel an estimate of the distance from the camera to the surface. There are several techniques for measuring depth using light and/or lasers: most common are structured light and stereoscopic devices. The RS4XX series of devices use a stereoscopic technique inspired by human binocular vision.

The RS4XX cameras compute a pixel's depth by comparing two images produced by two cameras. After projecting the images onto a common plane (the cameras are pointed at a slight angle), the camera computes the *disparity* between the two images.

Disparity is the distance (in pixels) of a cluster of pixels from one image to the other. Objects closer to the camera will have greater disparity between the two images, distant objects less so. With the fixed distance between the two cameras, the disparity allows the camera to infer the depth using the formula:

$$d = \frac{fB}{z}$$

### 2.2 Background subtraction

### 2.3 Filters

spatial and temporal

### 2.4 Pixel to Point Cloud Transformation

## 3 Implementation

## 4 Results

## 5 Documentation

## 6 Conclusion

# References


[1] G. K. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X.-F. Sun, and P. L. Rosin, "Registration of 3d point clouds and meshes: a survey from rigid to nonrigid," *IEEE transactions on visualization and computer graphics*, vol. 19, no. 7, pp. 1199–1217, 2012.

[2] B. Bellekens, V. Spruyt, R. Berkvens, and M. Weyn, "A survey of rigid 3d pointcloud registration algorithms," in *AMBIENT 2014: the Fourth International Conference on Ambient Computing, Applications, Services and Technologies, August 24-28, 2014, Rome, Italy*, 2014, pp. 8–13.

[3] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3d reconstruction at scale using voxel hashing," *ACM Transactions on Graphics (ToG)*, vol. 32, no. 6, p. 169, 2013.

[4] L. Keselman, J. Iselin Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, "Intel realsense stereoscopic depth cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 1–10.

[5] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking." in *ISMAR*, vol. 11, no. 2011, 2011, pp. 127–136.