Université Jean Monnet

Machine Learning & Data Mining

Computer Vision Project

3D Reconstruction using Kinect Sensors

Maedeh | Afshari
Sejal | Jaiswal

UNIVERSITÉ
JEAN MONNET
SAINT-ÉTIENNE

Table of Contents

1 Introduction

In computer vision and computer graphics, 3D reconstruction is the process of capturing the shape and appearance of real objects.3D data acquisition and object reconstruction can be performed using stereo image pairs. The images acquisition can occur from a multitude of methods including 2D images, acquired sensor data and on site sensors.

3D reconstruction from multiple images is the creation of three-dimensional models from a set of images. It is the reverse process of obtaining 2D images from 3D scenes. The essence of an image is a projection from a 3D scene onto a 2D plane, during which process the depth is lost. The 3D point corresponding to a specific image point is constrained to be on the line of sight. From a single image, it is impossible to determine which point on this line corresponds to the image point. If two images are available, then the position of a 3D point can be found as the intersection of the two projection rays. This process is referred to as triangulation. The key for this process is the relations between multiple views which convey the information that corresponding sets of points must contain some structure and that this structure is related to the poses and the calibration of the camera.

Three dimensional reconstruction can be achieved either by using active or passive methods. Active methods make use of light source such as lasers or infra-red emitters for scanning a given environment and measuring the depth, to create a depth map.
In contrast in passive methods, colour images of the environment in different perspectives are used to create a three dimensional model of the environment.

For this project, we are attempting to reconstruct a 3D scene by implementing an active method using a set of depth maps of the objects of interest in a scene. The scene was captured in a lab setting using various objects as reference objects and using a Microsoft Kinect, a motion sensing input devices. It is based around a webcam-style add-on peripheral, it enables users to control and interact with their console/computer without the need for a game controller, through a natural user interface using gestures and spoken commands.

A set of depth maps of the scene is acquired. The acquired depth maps are pre-processed (enhanced). Enhanced depth maps are converted into 3D point clouds using the intrinsic parameters of the Kinect sensor.
A point cloud is a set of data points in some coordinate system.In a three-dimensional coordinate system, these points are usually defined by X, Y, and Z coordinates, and often are intended to represent the external surface of an object.

The process of 3D reconstruction has been divided into 3 main steps:

1. Image Acquisition
2. Camera Calibration
3. 3D-Reconstruction

2 3D Imaging Techniques

2.1 Triangulation

Triangulation is the process of determining the location of a point by forming triangles to it from known points.
The laser L projects a ray of light onto the object O.
The intersection point P with the object is viewed by a camera and forms the spot P' on its image plane I.
We will have to know the laser-camera configuration. We shall also need additional device (Laser).

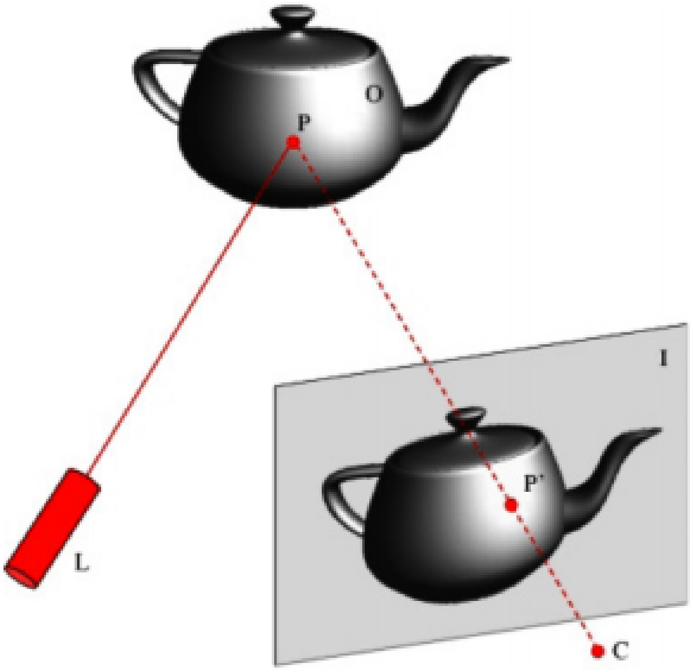

Figure 1: Triangulation Process

2.2 Structured light based Technique

To compute the 3D structure, we project ”structured” light patterns onto the object. The distortion of light pattern allows us to compute the 3D structure for which we requires a camera and a projector.
To find correspondences between the images from different cameras, we use structured light to simplify the matching by searching the pattern in the camera images (pattern decoding).

2.3 Kinect

- Kinect has three infrared light sources each generating a modulated wave with different amplitude.
- In order to capture reflected waves, Kinect also has infrared camera.

- Kinect generates a static cloud of variably intense dots (rather than strips) in a pattern that appears to be random.
- Infrared laser emitter (invisible light to human eye) strikes a diffraction grating to split the beam into thousands of individual points of light.
- Number of dots: 30,000 to 300,000.
- A 3x3 grid is repeated into 211x165 spot pattern, which creates overall grid of 633x495 or 313,335 points of light.
- With Structured Light Imaging the coding has to be unique per position in order to recognize each point in the pattern.

From the emitter, a constant pattern of speckles projected. This pattern is captured by the infrared camera and is correlated against a reference pattern.When a speckle is projected on an object whose distance to the sensor is different than that of the reference plane then position of the speckle in the infrared image is shifted in the direction of the baseline between the laser projector and perspective center of the infrared camera.These shifts are measured for all speckles, which yields a disparity image.For each pixel the distance to the sensor can then be retrieved from the corresponding disparity.

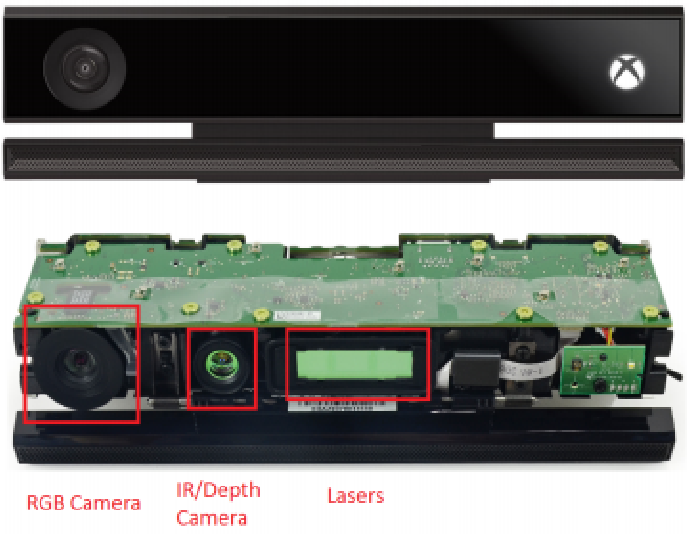

Figure 2: Kinect

3 Image Acquisition

3.1 Checker-board Images

Images were acquired keeping the object of interest as a checker-board. The checker-board images were captured with varying orientation and depth of the checker-board with respect to the Kinect sensor. These images are used for finding the intrinsic value of the camera and the extrinsic values.The dimension of the checkerboard is known and hence can be used for this purpose.

Total images captured:

- Left Camera: 14 images * 2 (Depth and RGB images) [Leftx.jpg]
- Right Camera: 14 images * 2 (Depth and RGB images) [Rightx.jpg]

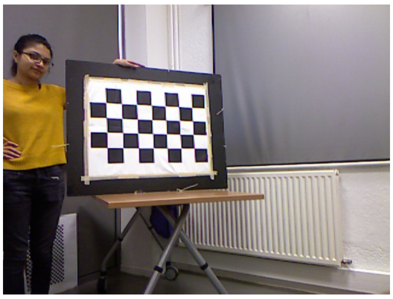

Figure 3: An image captured from the Left Camera

3.2 Scene Images

Another set of images were acquired keeping various object of interest with varying shapes and dimensions. These images are used for 3D-reconstruction later.

Total images captured:

- Left Camera: 1 image * 2 (Depth and RGB images) [L_scene.jpg]
- Right Camera: 1 image * 2 (Depth and RGB images) [R_scene.jpg]

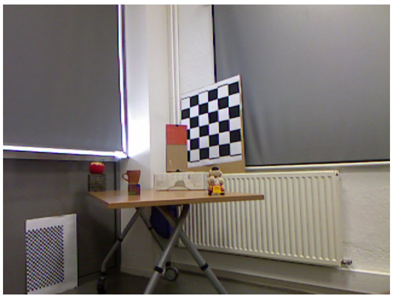

Figure 4: Image of the scene captured from the Left Camera

4 Camera Calibration

We used the toolbox provided by Caltech for calibration. We followed the Example 1 and Example 5 for the same, as we did during the Practical Lab Session for Computer Vision.

4.1 Left Camera Calibration

File: `Calib_ResultsLeft.mat` `Focal length`: fc = [532.882508835719590; 531.7955263547384];
`Principal point`: cc = [291.009409618044910; 222.806112996766560];
`Skew coefficient`: alpha_c = 0.000000000000000;
`Distortion coefficients`: kc = [0.279255336452240; -0.967616941336424; -0.038740716990524; -0.017697415980067; 0.000000000000000];
`Focal length uncertainty`: fc_error = [26.240714919577247; 27.188791681602751];
`Principal point uncertainty`:cc_error = [25.451832195362091; 17.250244324461189];
`Skew coefficient uncertainty`: alpha_c_error = 0.000000000000000;
`Distortion coefficients uncertainty`: kc_error = [0.129897320544029; 1.211804978228078; 0.015802103209377; 0.020561726926303; 0.000000000000000];

4.2 Right Camera Calibration

File: `Calib_ResultsRight.mat` `Focal length`: fc = [601.730844251917690; 581.887105857937970];
`Principal point`: cc = [219.829047509894080; 254.724905544805300];
`Skew coefficient`: alpha_c = 0.000000000000000;
`Distortion coefficients`: kc = [0.251753728073581; 0.437903627546445; -0.005312312783367; -0.091730786720801; 0.000000000000000];
`Focal length uncertainty`: fc_error = [44.747175869776598; 32.270952761463199];
`Principal point uncertainty`:cc_error = [40.413434634510679; 15.768174509580676];
`Skew coefficient uncertainty`: alpha_c_error = 0.000000000000000;
`Distortion coefficients uncertainty`: kc_error = [0.128766967190221; 0.562307363123869; 0.010381112643499; 0.034758167484828; 0.000000000000000];

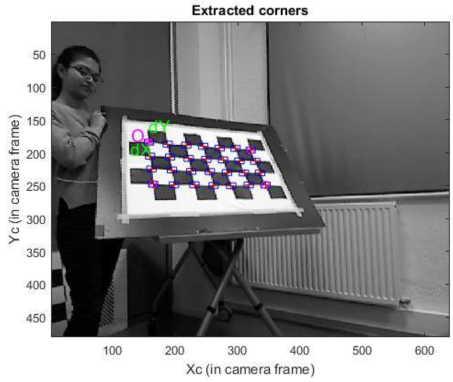

Figure 5: Calibration Process

4.3 Stereo Calibration

File: `stereoParams.mat`

After the calibration process, we are able to extract the following values for use in the 3D-reconstruction:

- Focal length (left RGB camera, right RGB camera).
- Principal Point (left RGB camera, right RGB camera)
- Rotation vector (left to right camera)
- Translation vector (left to right camera)

5 3D-Reconstruction

Steps in order to perform 3D-reconstruction:

- Transformation: Pixel frame to IR camera world coordinates
- Transformation: IR camera coordinates to RGB camera reference
- Transformation: RGB 3D to Pixel frame in RGB camera
- Transformation: One camera reference to the other
- Merge the points.

Our aim for 3D-Reconstruction is to recover the geometric structure of a scene from two images: Given the pixel coordinates (u,v) of a point m in a digital image, determine the world coordinates (X,Y,Z). This can be recovered from its projections m1 and m2 in the images by triangulation.

By using 2 images (RGB and Depth), 3D information can be (partially) recovered by solving a pixel-wise correspondence problem. Since automatic correspondence estimation is usually ambiguous and incomplete, further knowledge (prior knowledge) about the object is necessary. A typical prior is to assume the object surface to be smooth.

To align RGB and Depth image from Kinect, we use the depth camera intrinsic values.
Each pixel (x_d,y_d) of the depth camera can be projected to metric 3D space using the following formulas:

- $Pixel3D.x = (x_d - cx_d) * depth(x_d, y_d)/fx_d$
- $Pixel3D.y = (x_d - cy_d) * depth(x_d, y_d)/fy_d$
- $Pixel3D.z = depth(x_d, y_d)$

We then use R (Rotation matrix for Right camera wrt to the left camera) and T (Translation matrix) to move from IR 3D camera reference to 3D RGB camera reference, using the formula:

- $Pixel3D = R * pixel3D + T$

After reprojecting to the pixel frame of the RGB camera, we develop two point clouds for each the left and right RGB camera. We then merge the two point clouds.

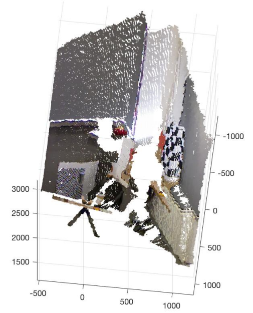

Figure 6: 3D reconstruction from the Left Camera Image - View 1

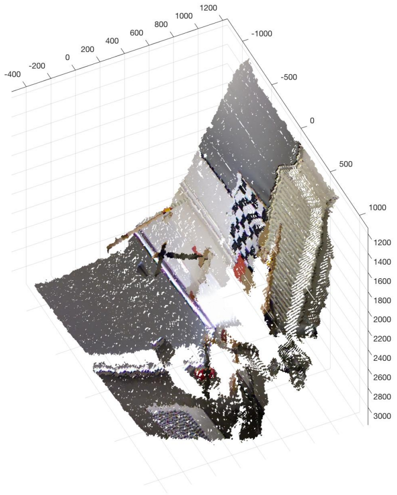

Figure 7: 3D reconstruction from the Left Camera Image - View 2

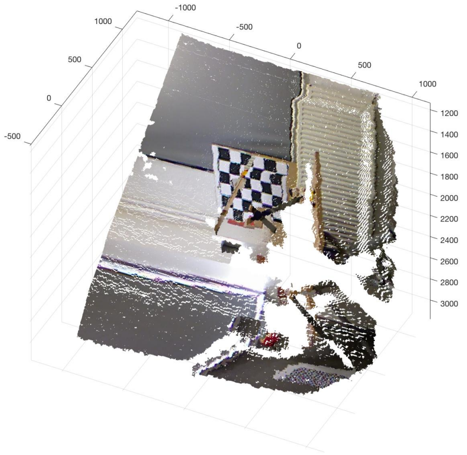

Figure 8: 3D reconstruction from the Left Camera Image - View 3

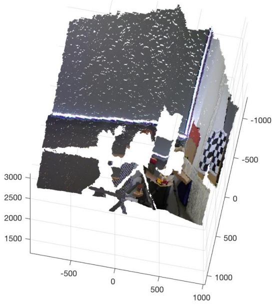

Figure 9: 3D reconstruction from the Right Camera Image - View 1

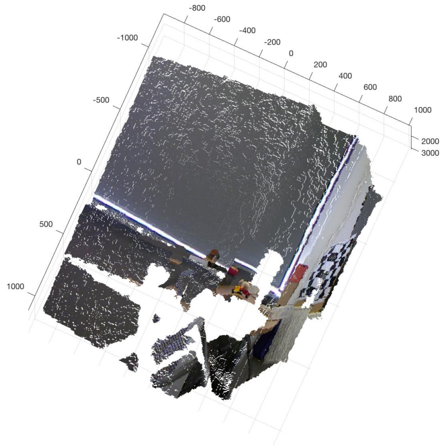

Figure 10: 3D reconstruction from the Right Camera Image - View 2

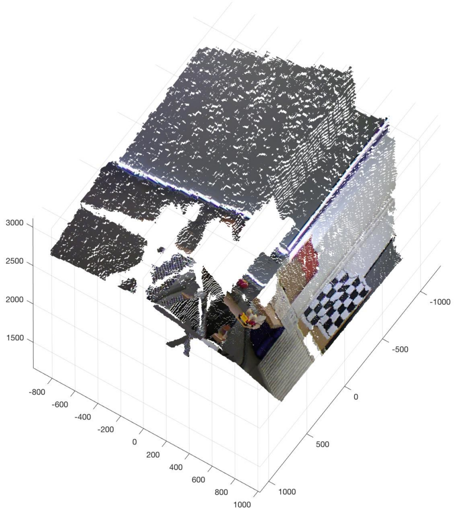

Figure 11: 3D reconstruction from the Right Camera Image - View 3

6 Conclusion

We were able to perform the 3D Reconstruction of the scene using RGB and IR images from the Kinect. The code for the same is provided in the zip file along with the images captured. Some adjustments had to be made with respect to the translation to align the images from the left and the right camera.
The performance of the same can be enhanced with usage of more images and thus leading to better calibration.

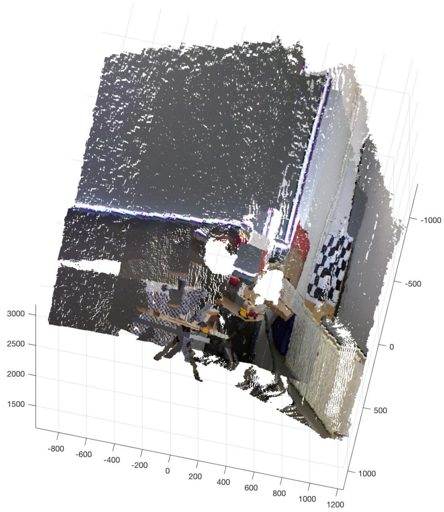

Figure 12: 3D reconstruction using images from Kinect - View 1

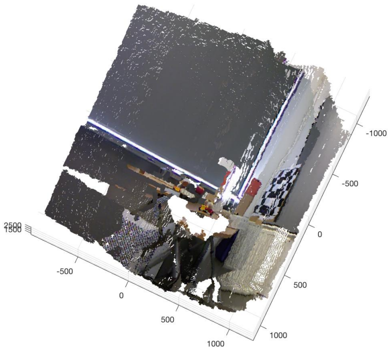

Figure 13: 3D reconstruction using images from Kinect - View 2

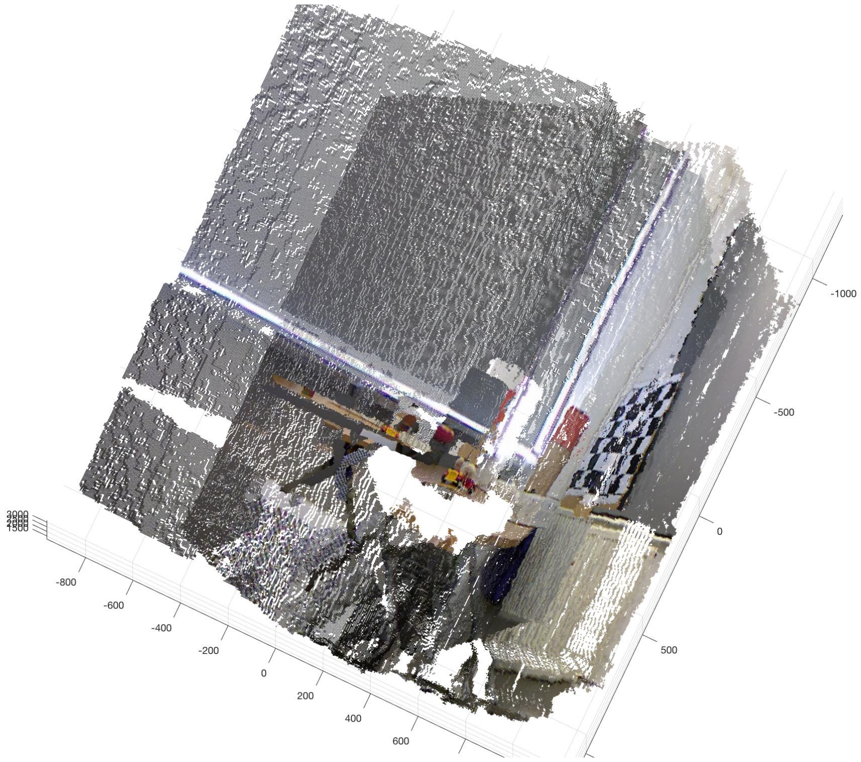

Figure 14: 3D reconstruction using images from Kinect - View 3

References

1. Camera Calibration Toolbox (http://www.vision.caltech.edu/bouguetj/calib doc/)

2. Kinect Calibration Example (http://burrus.name/index.php/Research/KinectCalibration#tocLink5

3. 3D Reconstruction from Multiple Images Part 1: Principles (Theo Moons, Luc Van Gool and Maarten Vergauwen)

4. 3D Object Reconstruction Using Kinect v2.0 (S.Varanasi, S.K.Devu)