

# Laboration 2: ASUS Xtion Pro: Calibration, noise characterization and filtering

Sensors and Sensing

Michael Floßmann, Anders Wilkström

2015–12–07

## 1 Introduction: Structured light cameras

Structured light cameras are a low-cost option for depth measuring in three dimensional space. The cameras project a known light pattern to a scene and record the reflection of that light pattern. This recorded data is then used for triangulation.

For this lab, the ASUS Xtion Pro sensor was used as a structured light camera.

## 2 Task and implementation

The task at hand was to set up and calibrating the sensor, as well as to characterize the noise in the depth measurement and to set up filtering routines.

### 2.1 Basic setup

To set up the camera, the package `openni2` for `ros-indigo` was used. When launching the node `openni2.launch`, it publishes a wide range of topics from the camera.

For this laboration, only the topics which publish a viewable image were of interest. This included two main topics:

- `/camera/rgb/`  
This topic publishes data from the RGB camera on the ASUS Xtion Pro. The topic `/camera/rgb/raw` shows the unprocessed RGB image like a regular camera. A sample image from this topic is shown in figure 2.1.
- `/camera/depth/`  
This topic publishes the depth data as a 2D-array of float variables containing the depth values in meters. A sample image from this topic is shown in figure 2.
- `/camera/depth_registered`  
This topic combines the RGB and the depth image into a coloured point cloud. A visualization of this topic through the tool `rviz` is shown in figure



Figure 1: Output image of `/camera/rgb/raw`



Figure 2: Output image of `/camera/depth/`



Figure 3: Screenshot of the vizualized pointcloud of `/camera/depth_registered`

## 2.2 Basic ROS node

After the basic setup, a ROS node template was used as a base to process the images and point clouds published. The received images are shown in figure 4 and figure 5.

## 2.3 Color camera calibration

## 2.4 Noise characterization

In this part we cropped small windows of varying sizes in the center of the depth image. Then the camera was placed in several known distances from a wall and the average and standard deviation of the depth values in the cropped images were recorded.

The mean error of 10 seconds of measurement in relation to the distance is shown in figure 6. The variances are shown in figure 7.

## 2.5 Noise filtering

In this part we applied several filters to the depth images in order to remove the noise from the measurement data. These filters are:

- Gaussian blur
- Median filtering
- Bilateral filtering
- Median over several image samples



Figure 4: Save of the RGB image



Figure 5: Save of the depth point cloud

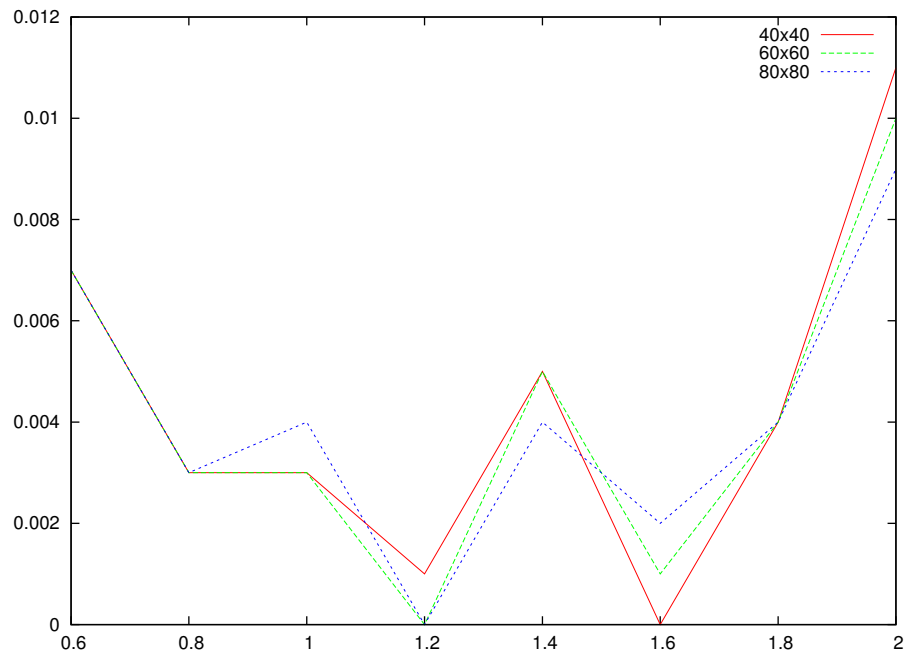


Figure 6: Mean error of the mean depth in relation to the distance

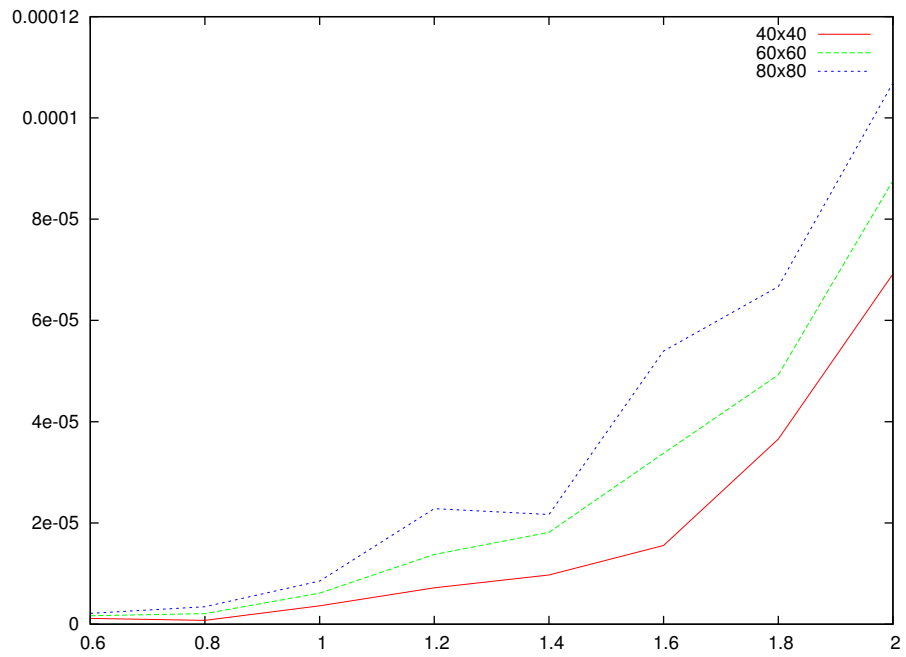


Figure 7: Mean of the Variance of the depth in relation to the ground truth distance

- Average over several image samples

The effect on the image data after the application of the filters will be discussed in the following.

### 2.5.1 NaN values

Whenever the sensor can't evaluate the IR data, the according pixel in the depth image receives a NaN value as the depth value. This can be caused by an object standing too close to the sensor or by IR rays not reaching the sensor due to shadow casting, obstructing objects or scattering. Leaving these NaN values in the image data for processing would negatively influence the filters and thus should to be replaced.

Two ways of replacing the NaN values were considered:

- Replacing the NaN values with a depth-value of 0.
- Replacing the NaN values with the last valid pixel scanned.

We chose the first method, since the replaced pixels are still identifiable as invalid. Thus, less information is lost.

### 2.5.2 Filter effects



Figure 8: Original depth image

**Gaussian blur**

**Median filtering**

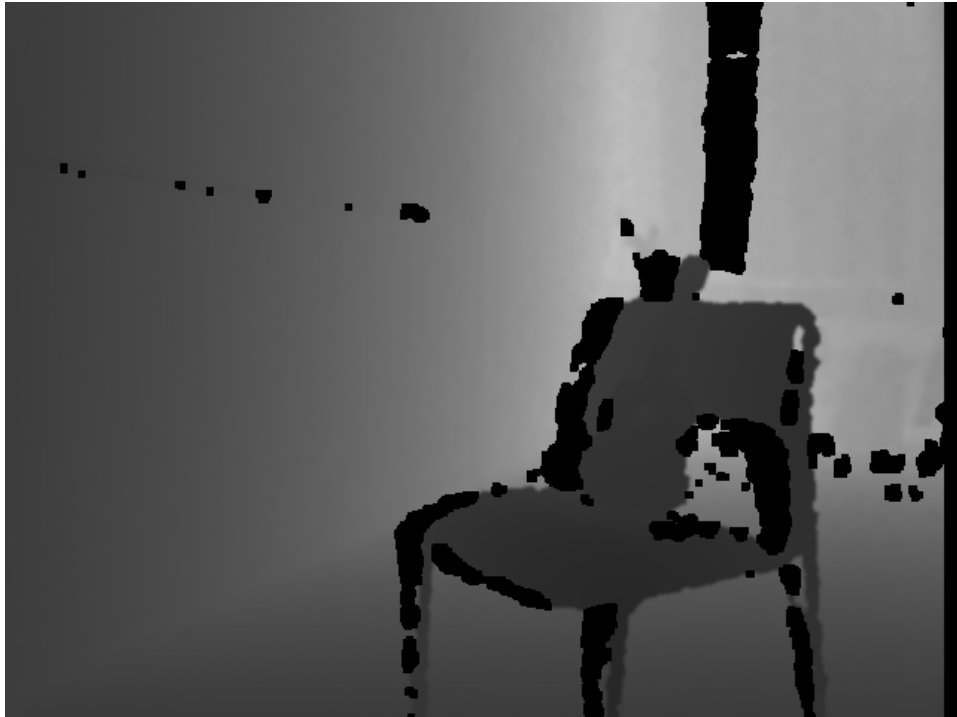


Figure 9: Gaussian blur



Figure 10: Median filtering



Figure 11: Bilateral filtering

#### **Bilateral filtering**

#### **Median over several image samples**

#### **Average over several image samples**

#### **2.5.3 Salt-and-pepper removal properties**

The depth images didn't suffer from noise visible to the eye. In order to test the noise cancelling properties of the filters, artificial salt-and-pepper noise was added to the image file. This was achieved by changing random pixels in the image to black or white in each received depth-image.

#### **Gaussian blur**

#### **Median filtering**

#### **Bilateral filtering**

#### **Median over several image samples**

#### **Average over several image samples**





Figure 12: Moving median image



Figure 13: Moving mean image



Figure 14: Original depth image with salt and pepper

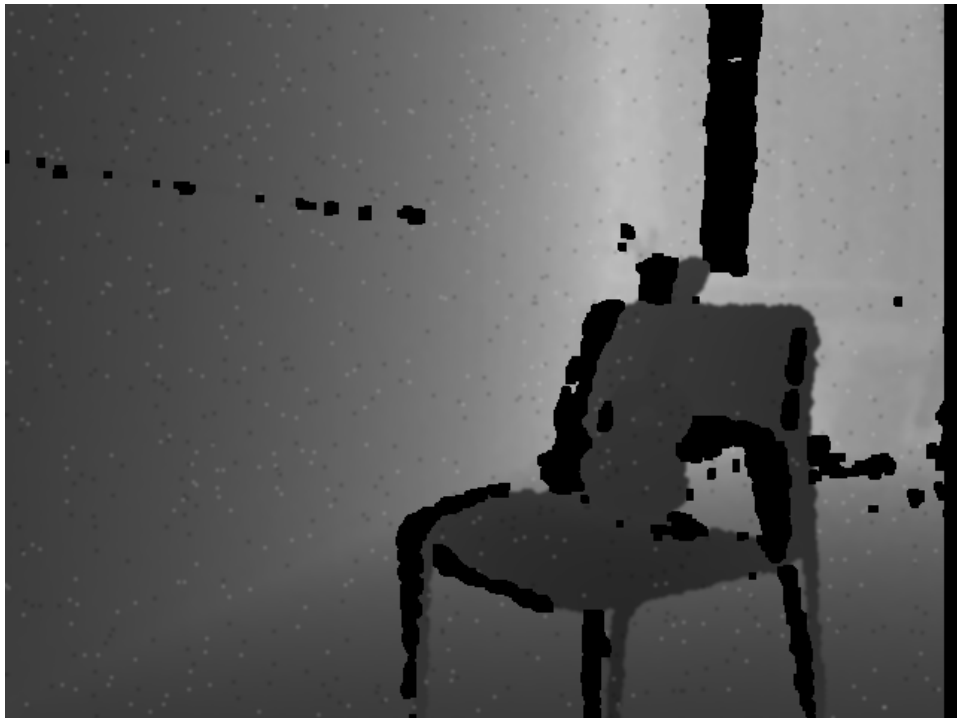


Figure 15: Gaussian blur with salt and pepper

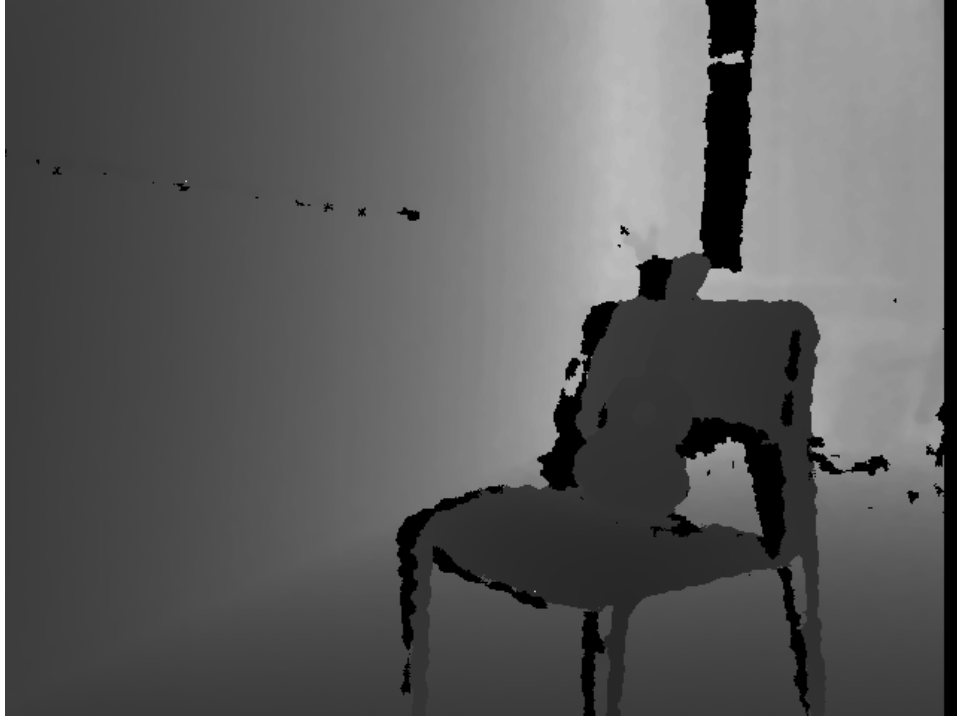


Figure 16: Median filtering with salt and pepper

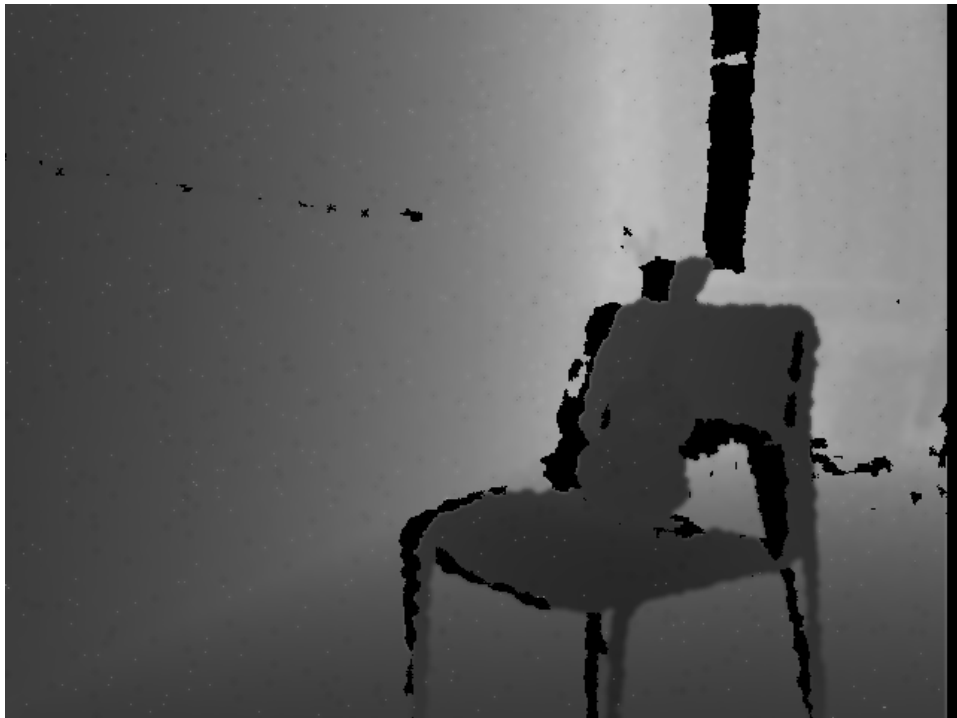


Figure 17: Bilateral filtering with salt and pepper



Figure 18: Moving median image with salt and pepper

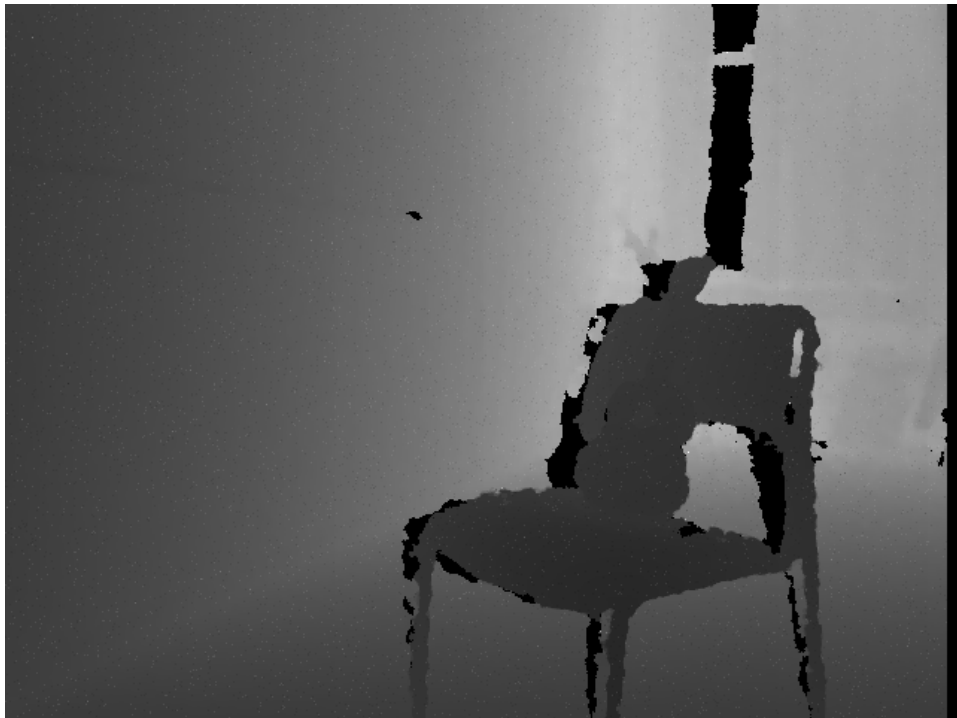


Figure 19: Moving mean image with salt and pepper