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

Figure 4: Save of the RGB image

Figure 5: Save of the depth point cloud

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.

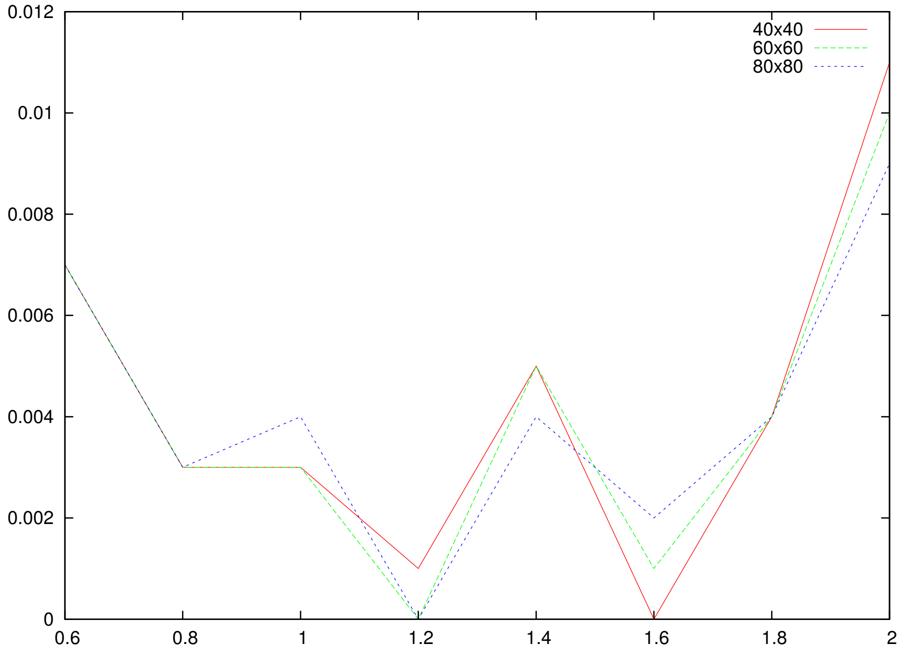

Figure 6: Mean absolute error of the mean depth in relation to the ground truth distance

Analysing the mean error (fig. 6), no conclusive pattern can be recognized in relation of the window size and the ground truth distance. The variance (fig. 7) on the other hand increases with the distance and the window size. This can be explained by the geometric characteristics of the measurement (see fig.8).

Figure 8 shows a 2D schematic view of the measurement. The camera records the depth values of an area around a maximum value of θ. The depth value at C doesn't contain the actually desired distance to the wall d, but rather

$$h = \frac{d}{\cos\theta}.$$

As the distance to the wall d increases, so does the absolute error between h and d. Increasing the observed window size increases θ, which has a similar effect.

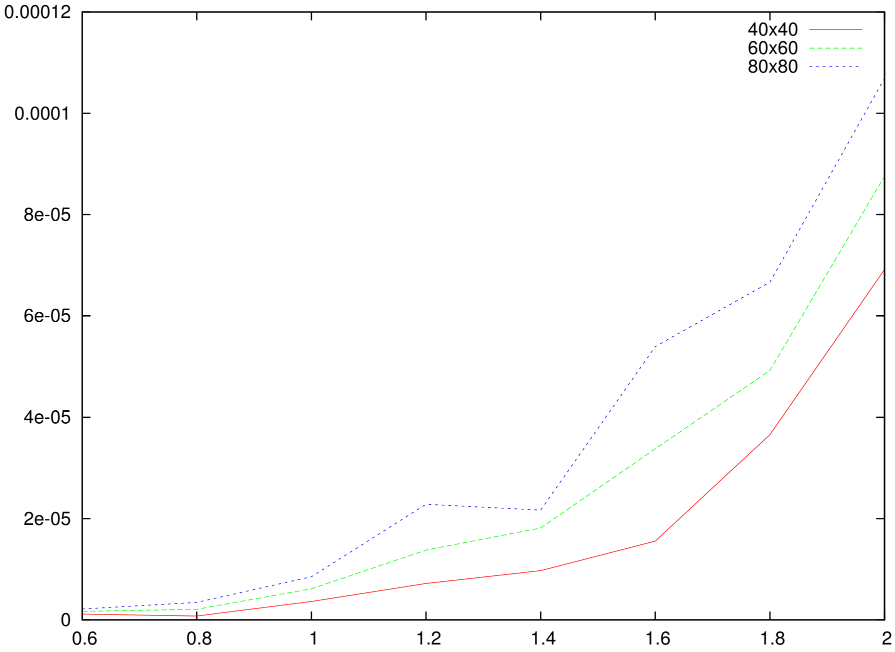

Figure 7: Mean of the Variance of the depth in relation to the ground truth distance

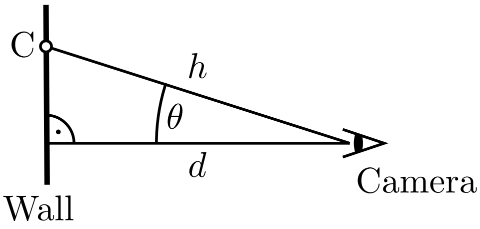

Figure 8: Schematic of the distance measurement

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 filter
- Bilateral filter
- Median over several image samples
- Average over several image samples

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

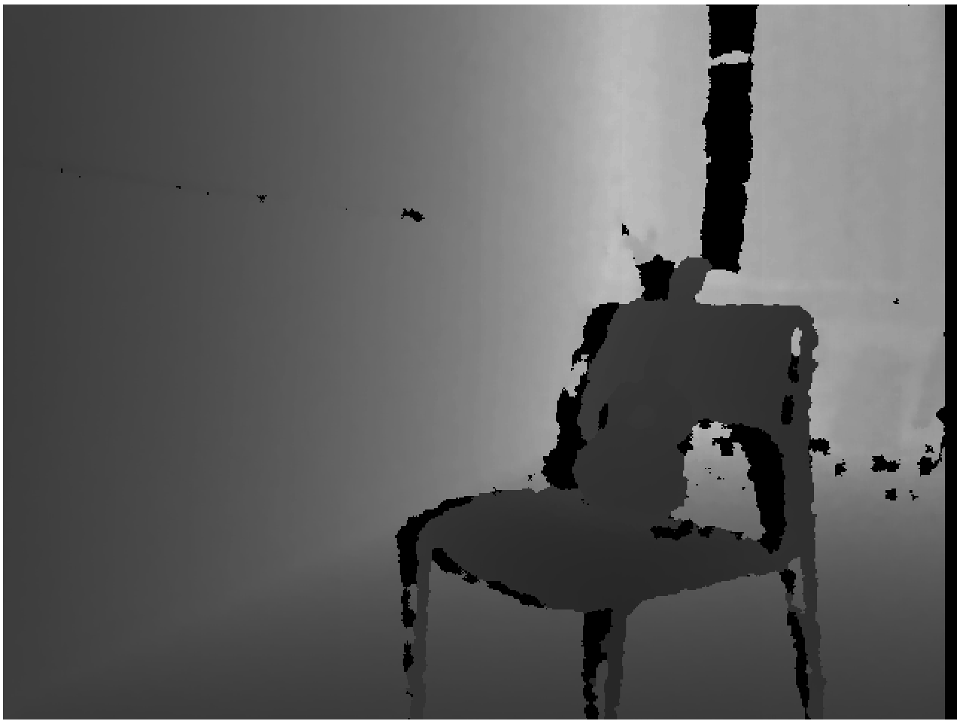

Figure 9: Original depth image

2.5.1 Filter effects

Gaussian blur The gaussian blur filter (fig. 10) blurs the contours of the observed objects. Additionally, the pixels with `NaN` values reproduce and strongly disturb the algorithm. This can be explained as every kernel containing a `NaN` value gives every pixel in the whole kernel a `NaN` value. This also explains the "blocky" artifacts in areas which originally contained small `NaN` speckles.

Median filter The median filter (fig. 11) doesn't show a big difference to the original image.

Bilateral filter The bilateral filter (fig. 12) blurs the contours of the objects. Otherwise, no remarkable differences to the original image are discernible.

In order for this filter not to produce very artifact-prone images, the `NaN`-values had to be adjusted. We chose to replace the `NaN`-values with entries of depth 0.

Median over several image samples The moving median image (fig. 13) leaves the contours visible and removes smaller spots with `NaN` values.

Mean over several image samples The moving mean image (fig. 14) blurs the contours of the objects (although not as strongly as previous contour blurring algorithms). It also removes small speckles of `NaN` values.

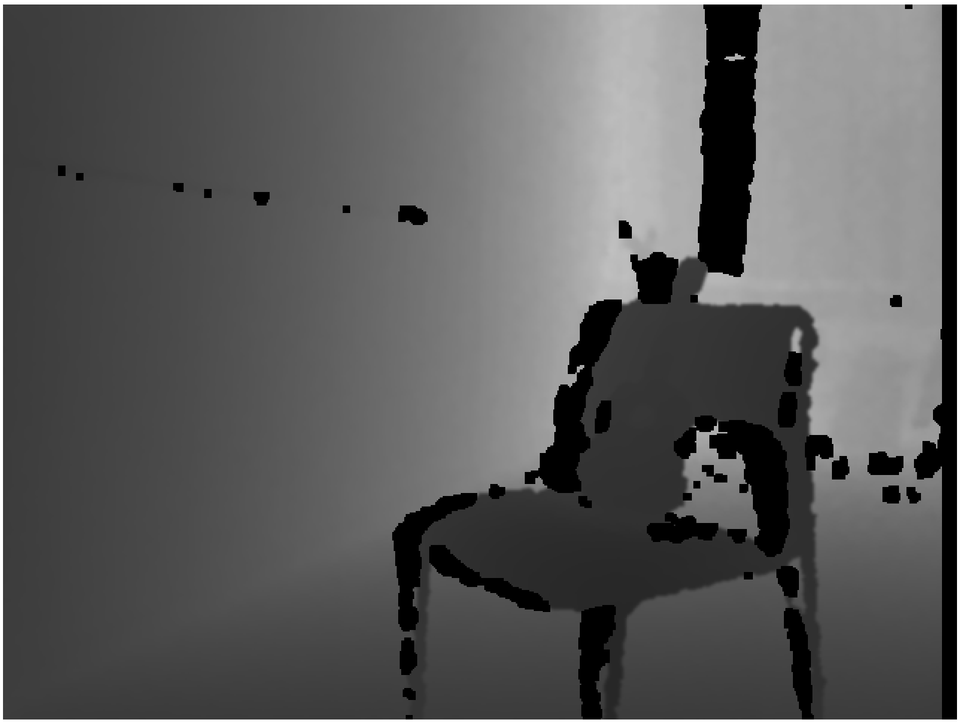

Figure 10: Gaussian blur

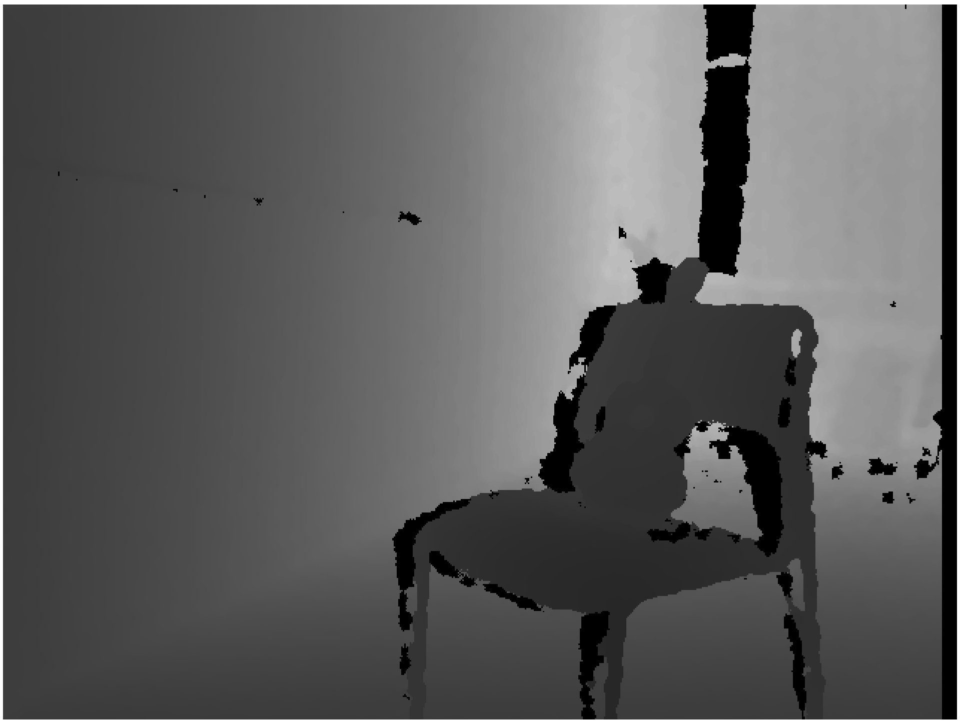

Figure 11: Median filter

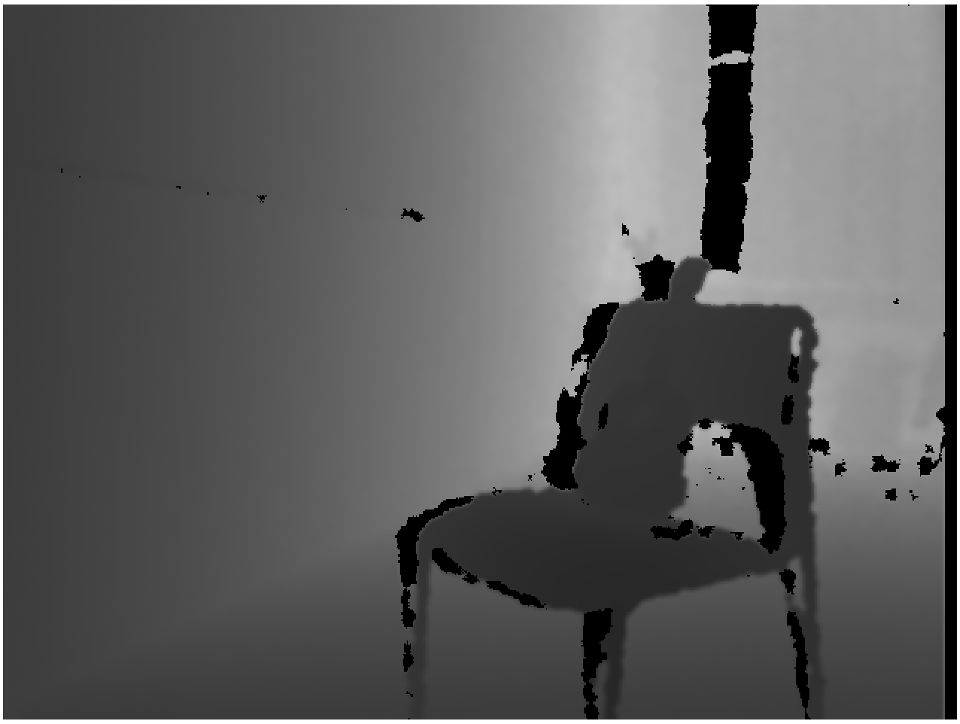

Figure 12: Bilateral filter

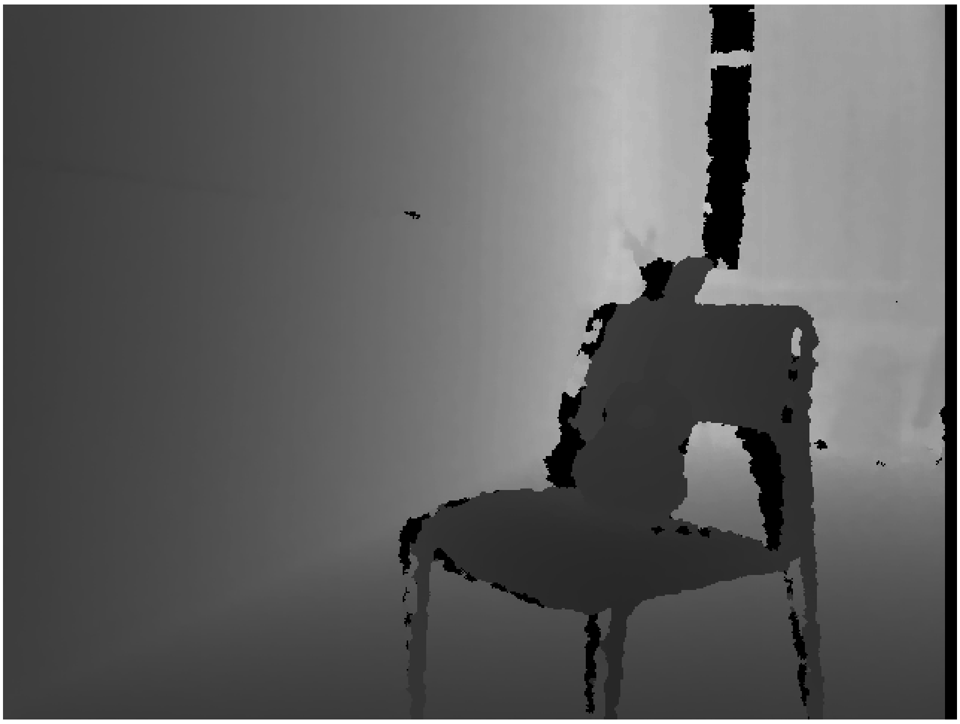

Figure 13: Moving median image

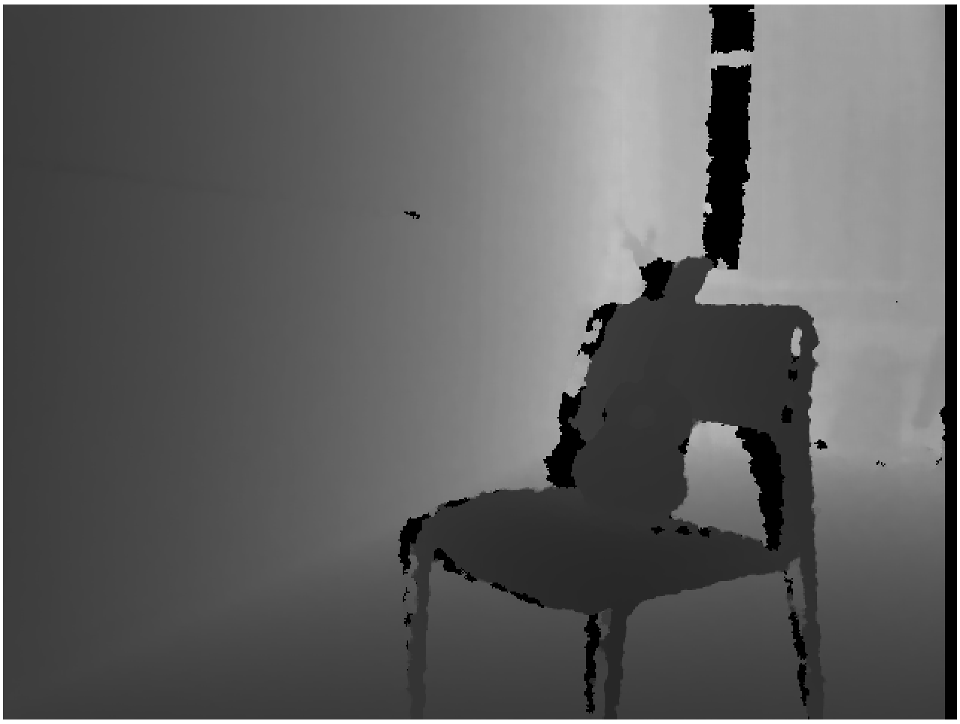

Figure 14: Moving mean image

2.5.2 Salt-and-pepper removal properties

The depth images didn't suffer from noise clearly 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 The gaussian blur (fig. 16) doesn't filter out the salt and pepper appropriately. It rather spreads the noisy data, negatively influencing the surrounding, proper data.

Median filter The median filter (fig. 17) removes the salt and pepper speckles reliably.

Bilateral filter The bilateral filter (fig. 18) decreases the influence of the salt and pepper speckles in the image, although they are still visible.

Median over several image samples The bilateral filter (fig. 19) removes the salt and pepper speckles reliably.

Average over several image samples The bilateral filter (fig. 20) accumulates the salt and pepper speckles of all recorded images. Although the speckles overall are more dull, their amount increased, still showing a considerable amount of noise.

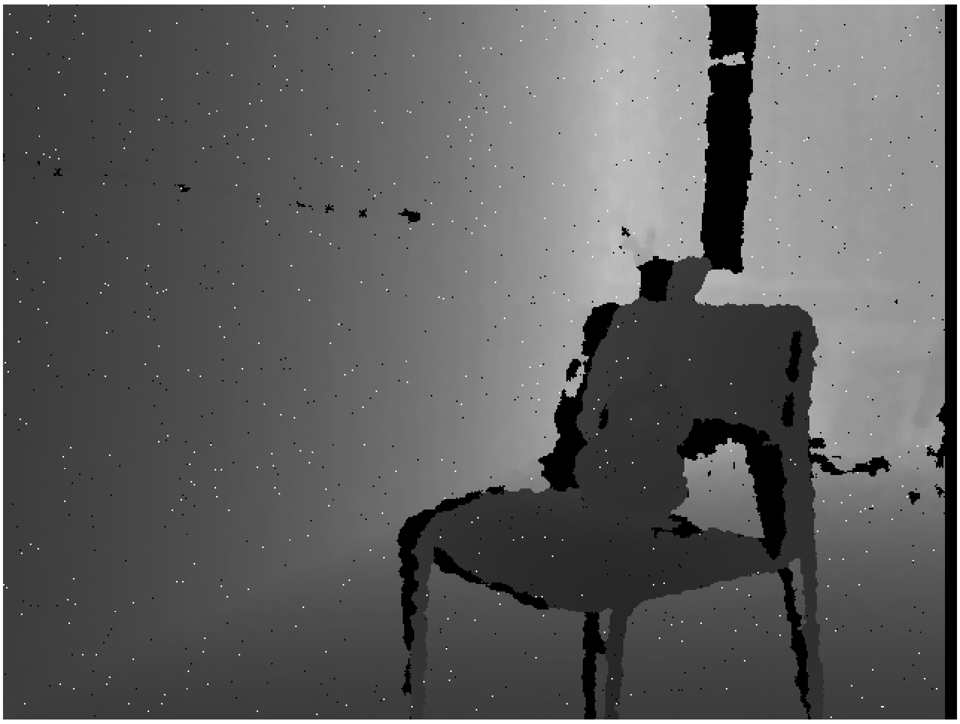

Figure 15: Original depth image with salt and pepper

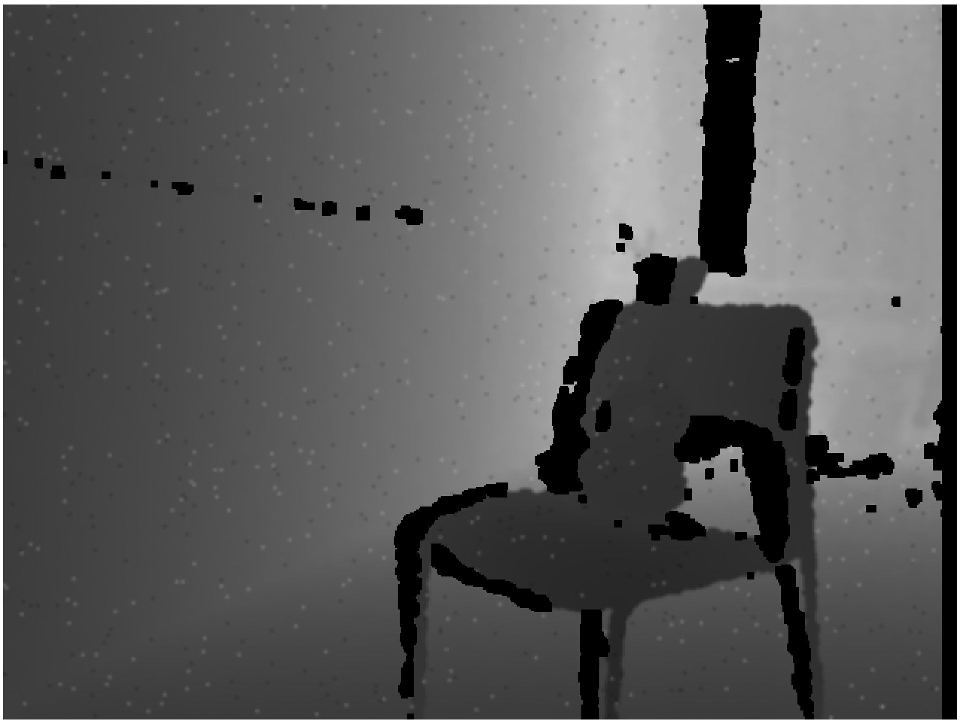

Figure 16: Gaussian blur with salt and pepper

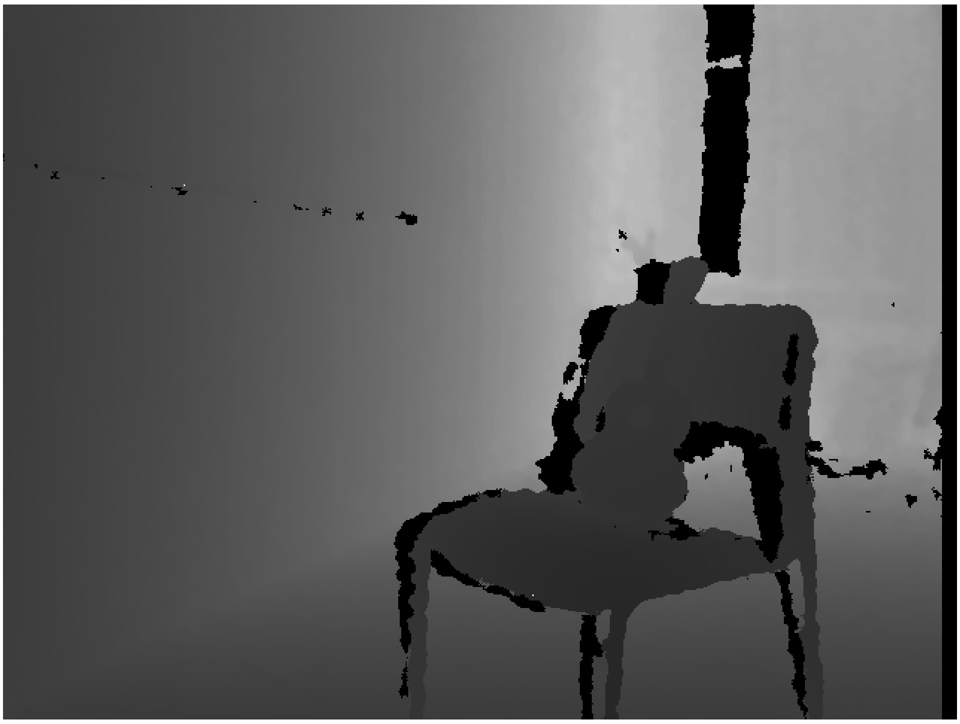

Figure 17: Median filter with salt and pepper

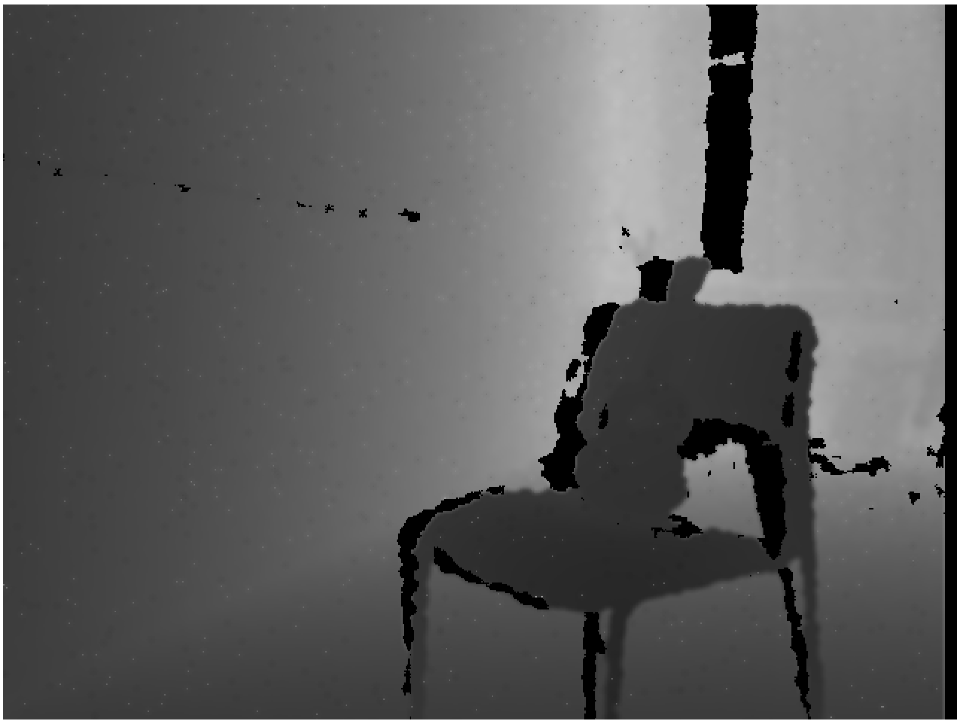

Figure 18: Bilateral filter with salt and pepper

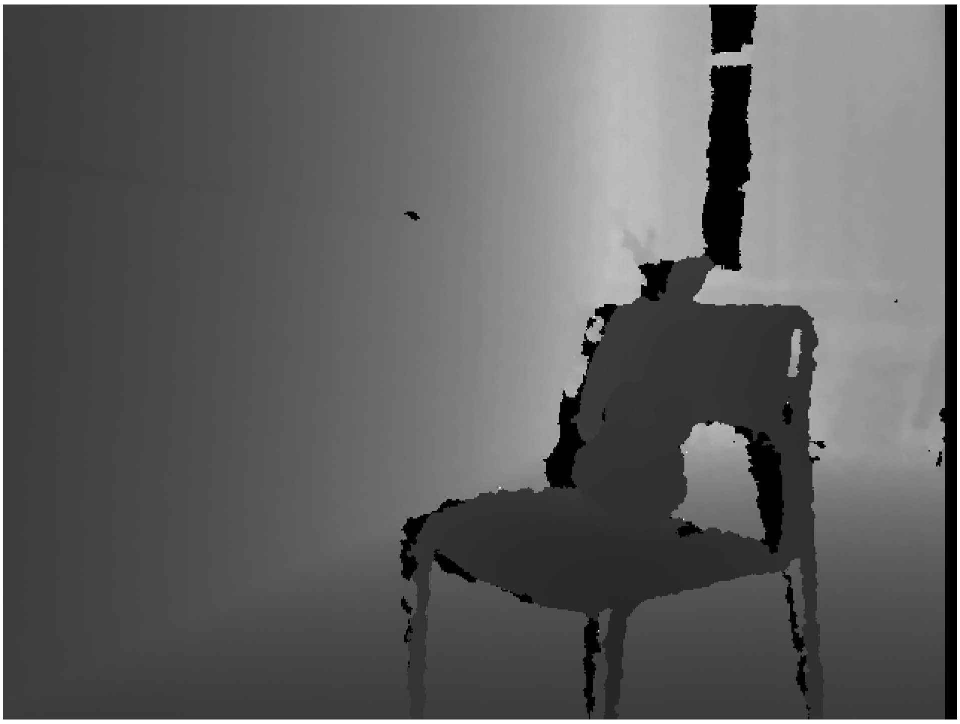

Figure 19: Moving median image with salt and pepper

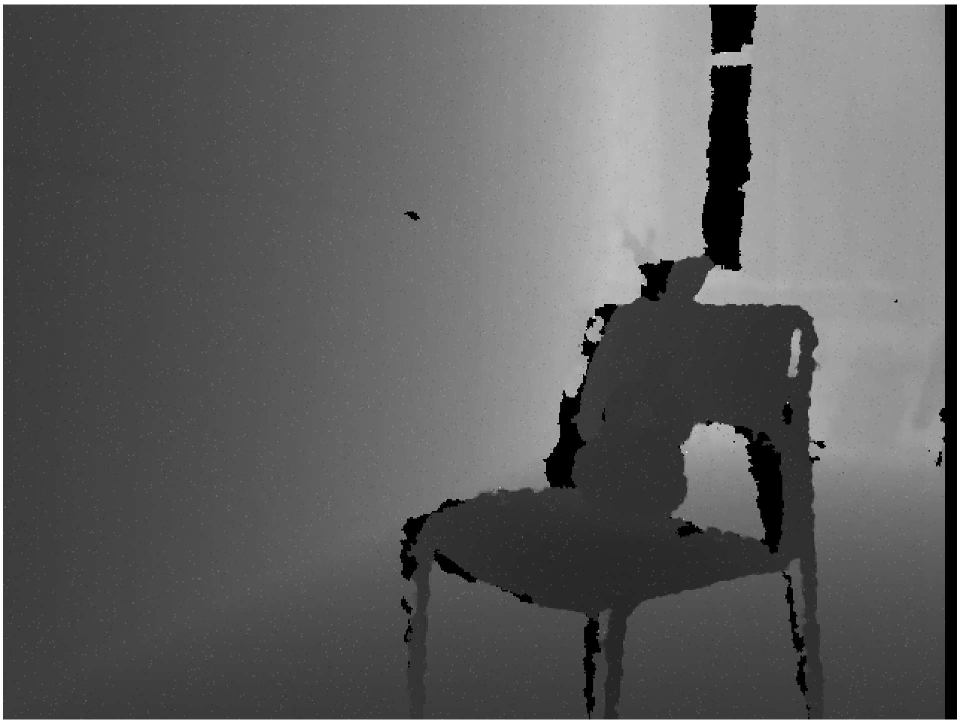

Figure 20: Moving mean image with salt and pepper