

A WAVELET BASED DENOISING METHOD FOR TIME OF FLIGHT CAMERA
DEPTH IMAGES

Thesis

Submitted to

The School of Engineering of the
UNIVERSITY OF DAYTON

In Partial Fulfillment of the Requirements for

The Degree of

Master of Science in Electrical Engineering

By

Achour Idoughi

Dayton, Ohio

August, 2022



University of
Dayton

A WAVELET BASED METHOD FOR TOF CAMERA DEPTH IMAGES DENOISING

Name: Idoughi, Achour

APPROVED BY:

Keigo Hirakawa, Ph.D.
Advisory Committee Chairman
Professor, Electrical and Computer
Engineering

Eric Balster., Ph.D.
Committee Member
Professor and Chair, Electrical and
Computer Engineering

Vijayan K. Asari, Ph.D.
Committee Member
Professor, Electrical and Computer
Engineering

Robert J. Wilkens, Ph.D., P.E.
Associate Dean for Research and Innovation
Professor
School of Engineering

Margaret F. Pinnell, Ph.D.
Interim Dean
School of Engineering

© Copyright by

Achour Idoughi

All rights reserved

2022

ABSTRACT

A WAVELET BASED METHOD FOR TOF CAMERA DEPTH IMAGES DENOISING

Name: Idoughi, Achour
University of Dayton

Advisor: Dr. Keigo Hirakawa

This work addresses the problem of shot noise in Time-of-Flight (ToF) camera depth sensors, which is caused by the random nature of photon emission and detection. In this paper, we derive a Bayesian denoising technique based on Maximum A Posteriori (MAP) probability estimation, implemented in the wavelet domain, which denoises (2D) depth images acquired by ToF cameras. We also propose a new noise model describing the photon noise present in the raw ToF data. We demonstrate that the raw data captured by ToF camera depth sensors follows a Skellam distribution. We test the resulting denoising technique, in the millimeter level, with real sensor data and verify that it performs better than other denoising methods described in the literature.

For Dad and Mum

ACKNOWLEDGMENTS

I would like to thank Keigo, my lab mates for all of their support and my family. I could not have done this without you guys.

TABLE OF CONTENTS

ABSTRACT	2
DEDICATION	3
ACKNOWLEDGMENTS	4
LIST OF FIGURES	6
LIST OF TABLES	8
CHAPTER I. INTRODUCTION	9
CHAPTER II. PRIOR ART	11
2.0.1 Image-based methods	11
2.0.2 Machine learning methods	13
CHAPTER III. TIME OF FLIGHT CAMERAS BACKGROUND	14
3.1 Time of Flight Principle	14
3.2 Depth Image Reconstruction	18
3.3 Noise in ToF depth images	23
CHAPTER IV. STATISTICAL MODELING OF TOF SENSOR DATA	27
CHAPTER V. DEPTH IMAGE DENOISING	32
5.1 Denoising in the wavelet domain	32
5.2 Statistical modeling of ToF sensor data in the wavelet domain	33
5.3 Bayesian denoising	37
CHAPTER VI. RESULTS	42
CHAPTER VII. CONCLUSION	53
BIBLIOGRAPHY	54

LIST OF FIGURES

3.1	Operating principle of Time-of-Flight cameras: emission of a near IR signal which is then reflected and processed by the sensor to calculate the depth. . . .	15
3.2	Circuit design of ToF sensors. (a) Time of Flight detector anatomy. (b) Approximated electronic circuit of the detector.	16
3.3	Raw data given by the Helios ToF camera manufactured by Lucid Vision Labs.	20
3.4	X and Y as a function of delay.	21
3.5	X and Y L1 unit ball.	22
3.6	Reconstructed Depth.	24
4.1	Different parameters of X and Y probability mass functions as function of Δ for the four different quadrants. (a) μ_0 and μ_1 . (b) μ_0 and μ_1	29
4.2	Quantile-Quantile plot that compares X and Y ToF camera raw data distribution with a Skellam distribution. (a) Q-Q plot of X . (b) Q-Q plot of Y	31
5.1	2×2 depth image neighborhood.	33
5.2	Quantile-Quantile plot that compares W_X components distribution to a Skellam distribution.	36
5.3	Quantile-Quantile plot that compares W_Y components distribution to a Skellam distribution.	36
5.4	Prior Gaussian Mixture Model.	39
5.5	Posterior mixture probabilities. (a) Probability of a not a horizontal edge. (b) Probability of a horizontal edge. (c) Probability of a not a vertical edge. (d) Probability of a vertical edge.	40
6.1	(a)-(c) Noisy reconstructed depth images. (b)-(d) Their corresponding intensity images.	43
6.2	Denoising results. (a) Ground Truth image. (b) BM3D denoised result. (c) Total Variation denoised result. (d) Joint Bilateral denoised result. (e) Non Local Mean denoised result. (f) Proposed denoised result.	45

6.3	Finer depth range of Figure 6.2. (a) Ground Truth image. (b) Noisy image. (c) BM3D denoised result. (d) Joint Bilateral denoised result. (e) Total Variation denoised result. (f) Proposed method result.	47
6.4	Denoising results. (a) Ground Truth image. (b) BM3D denoised result. (c) Total Variation denoised result. (d) Joint Bilateral denoised result. (e) Non Local Mean denoised result. (f) Proposed denoised result.	48
6.5	Finer depth range of Figure 6.4. (a) Ground Truth image. (b) Noisy image. (c) BM3D denoised result. (d) Joint Bilateral denoised result. (e) Total Variation denoised result. (f) Proposed method result.	49
6.6	Top and side views of the 3D depth charts.	51
6.7	Denoising results. (a) Ground Truth image. (b) Noisy image. (c) Total Variation denoised result. (d) Joint Bilateral denoised result. (e) BM3D result. (f) Proposed denoised result.	52

LIST OF TABLES

6.1	Denoising performance of various methods, averaged over 15 image. Mean squared error (MSE) to the ground truth of the methods under consideration. Our proposed method gives the smallest MSE.	50
-----	--	----

CHAPTER I

INTRODUCTION

Sensors allowing 3D reconstruction have attracted a lot of attention in recent years, both for the novelty of certain technologies and for the development of related fields. In order to obtain three-dimensional information, one of the most widespread methods is certainly the use of technologies using the time-of-flight principle since these sensors are generally precise and versatile. The principle is based on the emission of a signal by a source and its reflection by a sensor, generally a CCD or CMOS, the time traveled by light during the round trip (Δ) is used to calculate the distance according to the basic equation below:

$$\Delta = \frac{2D}{C}, \quad (1.1)$$

where D is the distance and C is the speed of light.

However, these cameras have limitations that affect the accuracy of their measurements due to the type of sensor they use. The main limitation is the lack of precision and the uncertainty of the depth measurements due to statistical nature of the photons and the distortions caused by the multipath propagation of light.

This thesis focuses on the problem of random shot noise in ToF cameras. Shot noise, also called photon noise, models the statistical character of the arrival of photons in the sensor and the process of electron-hole pairs generation. We propose in this thesis a new algorithm, implemented in the wavelet domain, which uses the MAP Bayesian estimator to estimate the clean depth values of the acquired noisy 2D image-like depth maps. In Chapter II, we start by discussing approaches proposed in literature for ToF images denoising. Chapter II is an overview on ToF technology and how depth information is calculated. In Chapter III, we propose a statistical model for the noise present in ToF cameras, a model that is

used in Chapter V to derive a denoising for 2D depth images acquired by a ToF camera. Finally, we show in Chapter ?? the results of our denoising technique applied on various depth images.

CHAPTER II

PRIOR ART

The accuracy of ToF cameras suffers from the influence of many error sources, including the external environment and the internal structure of the camera itself, introducing noise into the collected depth data. Therefore, issues related to the assessment and correction of depth inaccuracies have received considerable attention from the scientific community, as many methods have been proposed to improve image accuracy.

Since we are concerned with shot noise in this article, the ToF camera noise reduction techniques that have been suggested in the literature regarding this type of error can be divided into two broad categories: image-based methods and machine learning methods. While the first techniques use only data provided by the camera to reduce noise, the latest techniques, as their name suggests, involve a machine learning phase before they are applied.

2.0.1 Image-based methods

Among the image driven techniques, the various classical filtering methods have been applied to ToF cameras. To start with, bilateral filters are a well-known technique that have been applied to ToF depth images [1–3]. The filter proposed in [2] is a regular bilateral with fixed parameters, while other versions of bilateral filter adapt the filter parameter according to the depth values of the images [3] or use additional data to better preserve the edges of the image [4]. In the case of ToF cameras, the additional data may be luminance as it is less prone to noise [5, 6], the resulting filter is referred to as cross or joint bilateral filter which shows better denoising abilities.

Another depth noise reduction method uses clustering scheme [7], a method based on dividing the noisy depth image into various parts, each part is then denoised with distinct parameters.

Both adaptive and in nonadaptive variations of normalized convolution and median filtering have been considered in [8]. Non Local Mean (NLM) approach has also been considered for ToF cameras, the algorithm uses the similarity measures of different patches in the image as weights in the filtering process based on averaging. Adaptations of the filter has been applied on ToF depth images. To remove space varying depth noise, an adaptive NLM filtering method is suggested in [9], [10] uses depth noise model to determine the parameters of the NLM filter.

Wavelet denoising has also been applied to ToF in [11, 12]. It was demonstrated in [13] that the distribution of Haar wavelet coefficients of stochastic inhomogeneous Poisson processes (Discrete nature of photon emission) is Skellam. Skellam-based denoising methods recovers clean wavelet coefficients, and a subsequent inverse transform is applied to recover the denoised pixels.

To detect and preserve important edges in the data, Total variation regularization [4] and anisotropic filtering [14] are applied to ToF cameras. The approach in [4] takes into consideration the geometric properties, namely edges and slopes, of the depth data while denoising. The approach in [14], requires some adaptation of the parameters to the characteristics of the scene, and gives better results of moving objects by avoiding long averaging times.

In some previous works, the data provided by TOF is fused with other data provided by other imaging devices, denoising algorithms are then applied to the resulting fuse images.

For instance, ToF data can be fused with RGB data [15, 16] or with stereo data [17]. Denoising techniques are applied to the fused images, examples of the filters applied are bilateral TV regularization [18], bilateral (joint) filtering [19] and adaptations of non-local means [15].

2.0.2 Machine learning methods

In recent years, Deep learning approach for denoising depth images produced by Time of Flight (ToF) cameras have attracted attention. Mac Aodha et al in [20] proposed an algorithm for single-depth image super-resolution. This approach can be applied on ToF cameras, but it requires building the corresponding noise model, which compromises the main advantage of database-based methods which is the non-need to analyze the noise characteristics Hence, a simple but efficient Convolutional Neural Network (CNN) has been proposed in [21] where both of the depth and infrared data are used to denoise ToF depth images . In addition, an end to end CCN-based method has been presented in [22], this method addresses all of the joint multipath, noise and phase unwrapping problems related to ToF cameras. The results performed well on a variety of challenging scenes.

CHAPTER III

TIME OF FLIGHT CAMERAS BACKGROUND

3.1 Time of Flight Principle

The Time of Flight camera is a category of system allowing the three-dimensional acquisition of a scene. In ToF cameras, a signal must be emitted by a source and received by a sensor, most often a CCD or CMOS, which provides real-time depth images. Generally, ToF camera can be categorized into two types. The first is based on pulses (pulsed TOF): a short light pulse is emitted towards the target scene and the camera measures the time it takes for the pulse to travel between the object and the camera. This approach is used in Lidar (acronym for light detection and ranging) imaging systems. The second is based on amplitude modulation waves, cameras using this technology are often called "coherence-based" ToF cameras. The camera measures the phase shift between the transmitted modulated signal and the reflected signal by demodulating the reflected signal. The phase shift is then used to calculate the depth. In this thesis, we work with the second type of ToF cameras, the figure 3.1 illustrates the approach. This ToF camera sends out a near-infrared (IR) optical signal. The signal reflected by the objects in the scene is then detected by the sensor which calculates the depth [23].

The ToF sensor is the main part of a ToF camera. It is similar to a standard (2D) camera sensor in that it consists of an active part that transforms incident light into an electrical signal. However, ToF sensors are much more complex because each pixel is also able to measure the time of flight that the incident light has traveled between the camera and the object. This complexity has the disadvantage of requiring a larger area per pixel, which results in a much lower resolution.

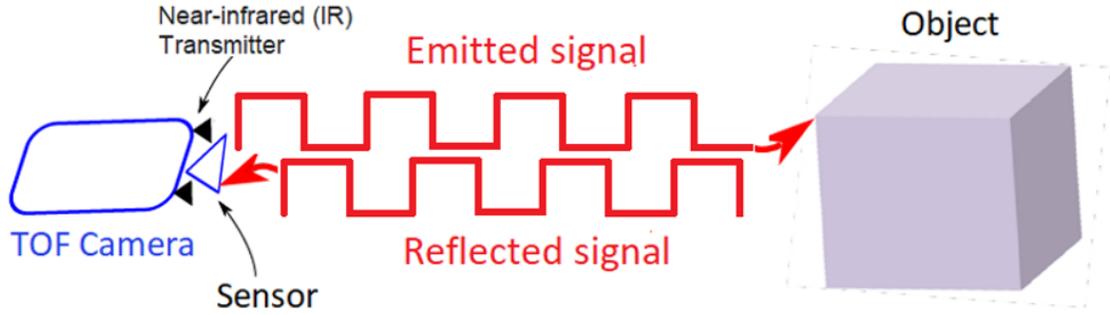


Figure 3.1: Operating principle of Time-of-Flight cameras: emission of a near IR signal which is then reflected and processed by the sensor to calculate the depth.

There are two calculation technologies for coherent Time of Flight cameras. The first is based on square wave (pulse) modulation: it measures the time it takes for the signal to travel between the object and the camera. The second is based on continuous modulation waves (Continuous-Wave (CW)): it measures the phase shift between the transmitted signal and the reflected signal by synchronous demodulation of the reflected signal. We are interested in our work in square wave modulated ToF cameras because it is more common as it is easily realized using digital circuits [24].

The measurement involves a pulsed light source ρ_τ , with a frequency of f_s and a pulse width of τ , and a CMOS imaging sensor. To be able to detect the reflected light, which is a shifted version of the emitted one then use the phase shift to compute the depth of the object, the detector has to operate synchronously with the pulsed light source; it needs to operate in the MHz levels. Achieving this frequency using mechanical shutters is nearly impossible. Instead, a the circuit design shown in Figure 3.2 is used, where each pixel of the sensor can be approximated by two doped regions, 1 and 2.

Light leaves the source, reflects off an object, and strikes the pixel creating electrons/holes pairs, the electrons are accumulated as a charge in either capacitors C_A or C_B , depending on the polarity of the voltage applied to terminals 1 and 2 .

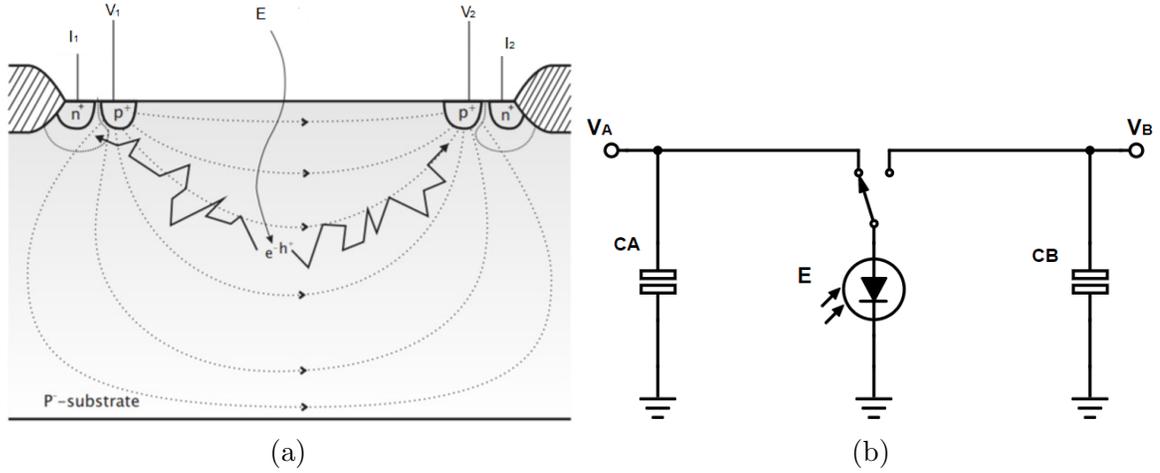


Figure 3.2: Circuit design of ToF sensors. (a) Time of Flight detector anatomy. (b) Approximated electronic circuit of the detector.

The polarities of V_1 and V_2 (Figure 3.2 (a)) are alternated between positive and negative at the same frequency as the lighting source, allowing the two capacitors to be interchangeably connected to the pixel (represented by a photo-diode) to be charged. The switch in the figure depicts the direction of electron motion. This arrangement ensures that the charge difference in the capacitors is directly related to the phase shift (between the transmitted pulse and the received pulse) which is determined by both the reflectance and the distance from the target object in the scene.

The current I_1 represented in Figure (3.2) (a), which is the current generated at terminal 1 due to photoelectric effect, can be expressed as:

$$I_1(t) = I_{sat} \left(e^{\frac{e \cdot v}{kT}} - 1 \right) + \frac{e \cdot A}{hc} \int Q(\lambda, t) E(\lambda, t) \lambda dt, \quad (3.1)$$

where Q is the accumulated charge at the node A and E , which is expressed in Equation (3.2), is the total reflected light. I_{sat} is the reverse saturation current, e is the base of the natural logarithm, q is the charge of an electron, v is the voltage across the photodiode, k is Boltzmann's constant and T is the temperature in kelvin.

$$E(\lambda_0, t) = \frac{T(\lambda_0)}{\pi} \left(E_s(\lambda_0, t - \frac{2D}{C}) + E_d(\lambda_0) \right) R(\lambda_0) + L_e(\lambda_0), \quad (3.2)$$

where E_d is the reflected atmospheric light on the sensor, L_e is the light directly coming to the sensor from the environment. λ_0 refers to the reflectance of the narrow-band filter (850nm) applied on the sensor. T is the transmittance. E_s is the light source reflected back to the sensor expressed as:

$$E_s(t) = e_s \rho_\tau \left(t - \frac{2D}{C} \right), \quad (3.3)$$

where e_s is the intensity amplitude of the reflected signal, $\rho_\tau \left(t - \frac{2D}{C} \right)$ is a delayed pulsed light source.

The circuit design illustrated in Figure 3.2 (a) can be approximated by the circuit shown on Figure 3.2 (b). The produced charges upon light strikes the sensor are accumulated in the two capacitors C_A and C_B , and their corresponding voltages V_A and V_B are read. The voltage at the node A is then calculated as follows:

$$V_A = \frac{1}{C_A} \int_0^{t_0} I_1(t) dt + V_{reset}, \quad (3.4)$$

where t_0 is the integration time.

Substituting Equation (3.3) into Equation (3.4), we obtain:

$$\begin{aligned}
V_A &= V_{reset} + \frac{t_0}{C_A} I_{sat} \left(e^{\frac{e \cdot v}{kT}} - 1 \right) + \frac{e^A}{C_A h C} \int_0^{t_0} Q(\lambda_0, t) E(\lambda_0, t) dt, \\
V_A &= V_{reset} + \frac{t_0/2}{C_A} I_{sat} \left(e^{\frac{e \cdot v}{kT}} - 1 \right) + \frac{\lambda_0 e^A q t_0}{2 C_A \cdot hc} \left(\frac{T(\lambda_0)}{\pi} E_d \lambda_0 R(\lambda_0) + L_e(\lambda) \right) \\
&\quad + \frac{e^A \lambda_0 q T(\lambda_0) e_s R(\lambda_0)}{C_A H C \pi} \int p_\tau \left(\tau - \frac{2D}{C} \right) \rho_\tau(t) dt,
\end{aligned} \tag{3.5}$$

where A is the measured detector area, C is the speed of light and h is the Planck constant.

Equation 3.5 can be simplified as follows:

$$V_A = \gamma + \beta R(\lambda_0) + \alpha R(\lambda_0) \cdot t_0 \cdot \frac{\tau - \Delta}{2\tau}, \tag{3.6}$$

where α and β are scene dependent constants, while γ is a scene independent constant.

The equation for V_B is derived using the same method. After simplification, V_B is expressed as:

$$V_B = \gamma + \beta R(\lambda_0) + \alpha R(\lambda_0) \cdot t_0 \cdot \frac{\tau}{2\tau}. \tag{3.7}$$

Because the characteristics of the two capacitors C_A and C_B may not perfectly match due to fabrication, the constants α , β and γ are no longer identical for both capacitors.

Therefore, Equations (3.6) and (3.7) become:

$$V_A = \gamma_A + \beta_A R(\lambda_0) + \alpha_A R(\lambda_0) \cdot t_0 \cdot \frac{\tau - \Delta}{2\tau}, \tag{3.8}$$

$$V_B = \gamma_B + \beta_B R(\lambda_0) + \alpha_B R(\lambda_0) \cdot t_0 \cdot \frac{\tau}{2\tau}. \tag{3.9}$$

3.2 Depth Image Reconstruction

To be able to determine the delay Δ of the emitted signal after its rebound on the detectors, we need to have access the both voltages V_A and V_B . Nevertheless, the raw data

provided by ToF cameras only gives access to their difference $V_A - V_B$, which is expressed in Equation (3.10) as:

$$V_A - V_B = (\gamma_A - \gamma_B) + (\beta_A - \beta_B)R(\lambda_0) + R(\lambda_0) \cdot t_0 \left(\frac{1}{2}\alpha_A - \frac{\Delta}{2\tau}(\alpha_A + \alpha_B) \right). \quad (3.10)$$

Furthermore, finding Δ requires the cancellation of the three parameters α , β and γ . One way to achieve this is to take multiple measurements. Four measurements are taken instead of one, a measurement refers to the difference of voltage across the two nodes, i.e. $V_A - V_B$.

Figure 3.3, shows raw data obtained by the *Helios* ToF camera manufactured by Lucid Vision Labs; the figure illustrates the four measurements taken by the camera. The four measurements are given in single raw image.

All measurements are performed using the same circuit, the sole thing that differs in each measurement is the voltage polarities pattern. For instance, during the first measurement, called the $0 - phase$ measurement, V_A is read first, then V_B is read, V_B is out of phase with respect to the V_A . During the second measurement, referred as the $180 - phase$ measurement, V_B is read first instead of V_A . To double the sensor depth range, two other measurements referred as $90 - phase$ and $270 - phase$, are taken.

To cancel the constants previously stated, the two variables X and Y , are constructed. X is the difference between the $Phase_0$ image and the $Phase_{180}$ image, while Y is the difference between the $Phase_{90}$ image and the $Phase_{270}$ image.

The variable X can be expressed as:

$$\begin{aligned} X &= Phase_0 - Phase_{180} \\ &= (V_{A_0} - V_{B_0}) - (V_{A_{180}} - V_{B_{180}}) \end{aligned}$$

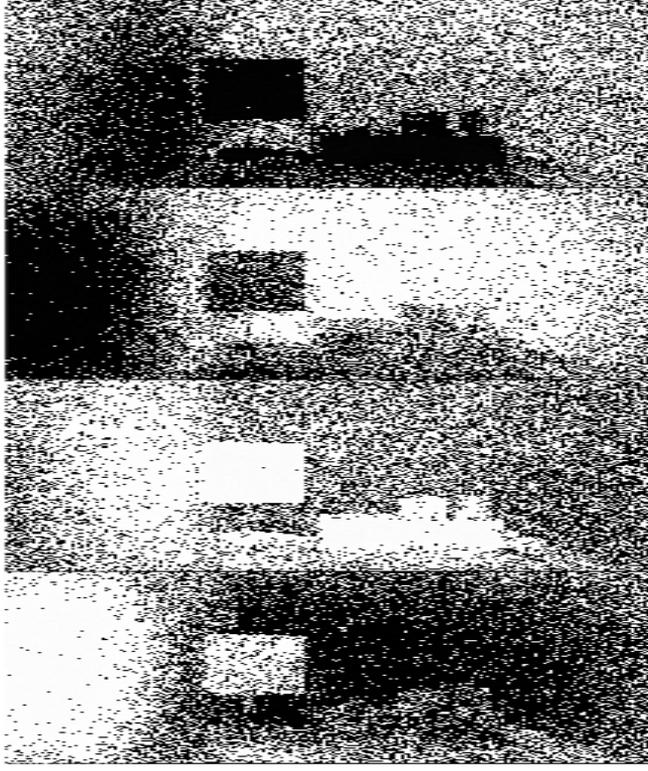


Figure 3.3: Raw data given by the Helios ToF camera manufactured by Lucid Vision Labs.

$$= \begin{cases} R(\lambda_0)t_0(\alpha_A + \alpha_B)(\frac{1}{2} - \frac{\Delta}{\tau}), & \text{for } 0 \leq \Delta < \tau \\ R(\lambda_0)t_0(\alpha_A + \alpha_B)(\frac{1}{2} - \frac{\Delta}{\tau}), & \text{for } 0 \leq \Delta < 2\tau \end{cases} \quad (3.11)$$

Similarly, Y is derived as follows:

$$\begin{aligned} Y &= Phase_{90} - Phase_{270} \\ &= (V_{A_{90}} - V_{B_{90}}) - (V_{A_{270}} - V_{B_{270}}) \\ &= \begin{cases} R(\lambda_0)t_0(\alpha_A + \alpha_B)(\frac{\Delta}{\tau/2}), & \text{for } 0 \leq \Delta < \tau \\ R(\lambda_0)t_0(\alpha_A + \alpha_B)(\frac{\tau/2 - \Delta}{\tau/2}), & \text{for } 0 \leq \Delta < 2\tau \end{cases} \end{aligned} \quad (3.12)$$

We can clearly see from Equation (3.11) and Equation (3.12) that both X and Y are a function of the delay Δ . The two variables are plotted in Figure 3.4 as a function of Δ .

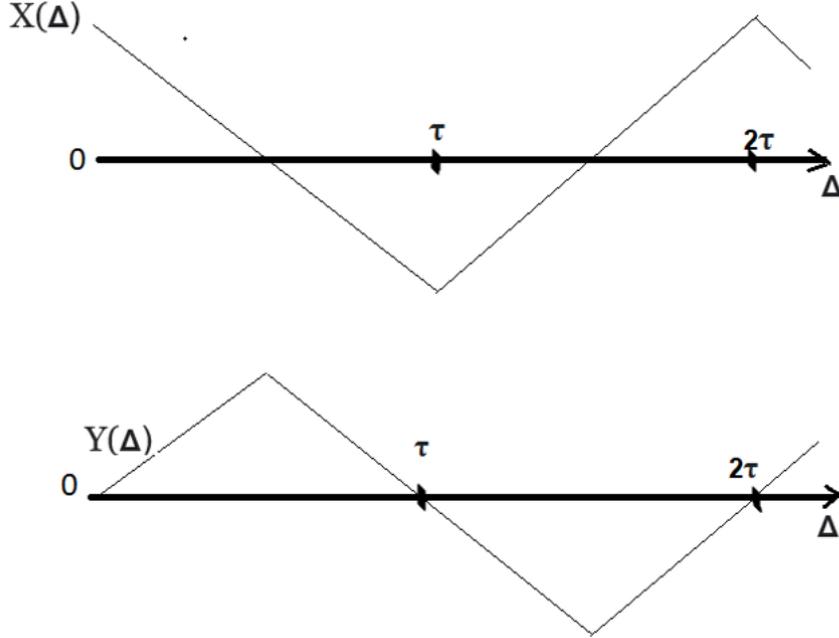


Figure 3.4: X and Y as a function of delay.

Looking at Figure 3.4, we can notice that there is a unique encoding of Δ (Phase) by the values of X and Y . In other words, there is a unique pair (X, Y) for all $\Delta \in [0 \ 2\tau]$. Just like a sine wave, another representation of X and Y in a form of a unit ball of norm L_1 in the shape of a diamond is shown in Figure 3.5.

If we divide the diamond into equally spaced intervals of Δ , this latter can be calculated. Depending on which quadrant of the diamond, the delay being the phase between X and Y , can be computed as follows:

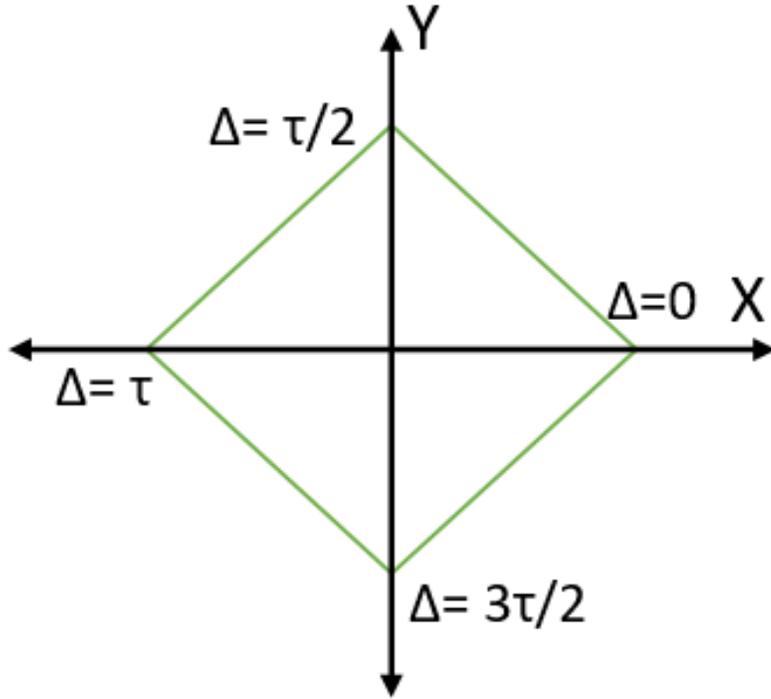


Figure 3.5: X and Y L1 unit ball.

- First quadrant: ($X > 0$ and $Y > 0$)

$$\frac{Y}{|X| + |Y|} = \frac{\Delta}{\frac{\tau}{2}} \quad (3.13)$$

Consequently, the delay can be found as:

$$\Delta = \frac{\tau}{2} \cdot \frac{Y}{|X| + |Y|} \quad (3.14)$$

- Second quadrant: ($X < 0$ and $Y > 0$)

$$\frac{-X}{|X| + |Y|} = \frac{\Delta - \tau/2}{\tau/2} \quad (3.15)$$

The delay is then:

$$\Delta = \frac{\tau}{2} \left(1 - \frac{X}{|X| + |Y|} \right) \quad (3.16)$$

- Third quadrant: ($X < 0$ and $Y < 0$)

$$\frac{-Y}{|X| + |Y|} = \frac{\Delta - \tau}{\tau/2} \quad (3.17)$$

Therefore, the is expressed as:

$$\Delta = \frac{\tau}{2} \left(\frac{Y}{|X| + |Y|} + \tau \right) \quad (3.18)$$

- Fourth quadrant: ($X > 0$ and $Y < 0$)

$$\frac{X}{|X| + |Y|} = \frac{\Delta - 3\tau/2}{\tau/2} \quad (3.19)$$

Consequently, the delay is found as:

$$\Delta = \frac{\tau}{2} \left(3 - \frac{X}{|X| + |Y|} \right) \quad (3.20)$$

The depth values can then be reconstructed by substituting equations of Δ , represented in Equations 3.14, 3.16, 3.18 and 3.20, into Equation 1.1. Depending on the quadrant, the equations for the reconstructed depth D are summarized in the following:

$$\begin{cases} D = \frac{C \cdot \tau}{4} \cdot \frac{Y}{|X| + |Y|}, \\ D = \frac{C \cdot \tau}{4} \left(1 - \frac{X}{|X| + |Y|} \right), \\ D = \frac{C \cdot \tau}{4} \left(\frac{Y}{|X| + |Y|} + \tau \right), \\ D = \frac{C \cdot \tau}{4} \left(3 - \frac{X}{|X| + |Y|} \right). \end{cases} \quad (3.21)$$

Now that the equations of the phase delay Δ are derived and the reconstructed scene depth D values are calculated using Equation 3.21, depth information of the raw sensor image represented in Figure 3.3 is reconstructed. The corresponding depth image is illustrated in Figure 3.6.

3.3 Noise in ToF depth images

The advantage of cancelling the different constants of V_A and V_B expressions can be seen in the reconstructed depth shown in Figure 3.6. The cancellation of the constants led

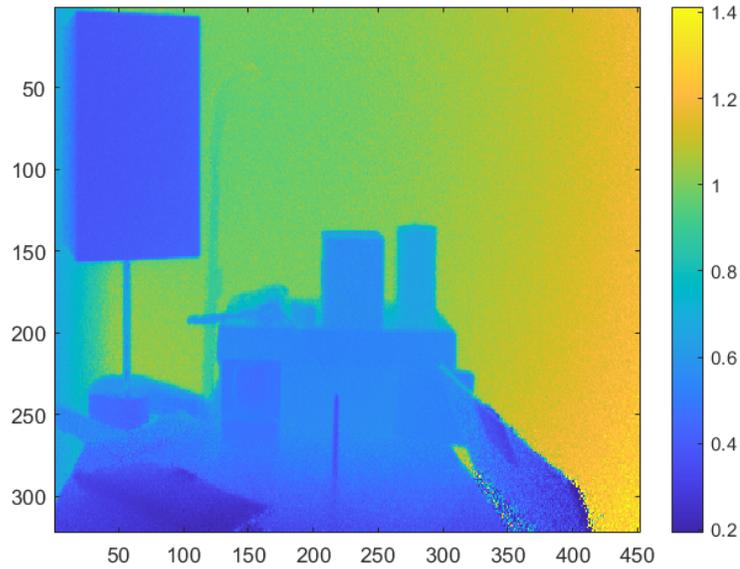


Figure 3.6: Reconstructed Depth.

to the elimination of the Fixed Pattern Noise (FPN) that is clearly apparent in the raw image given by the Helios ToF camera shown in Figure 3.3. Nevertheless, as we can see from Figure 3.6, various depth inaccuracies are still recorded. In fact, accuracy of ToF cameras is affected by numerous sources of errors which will be discussed in this section. Depth measurements with ToF cameras face the appearance of both deterministic and stochastic errors. While frame averaging can attenuate the effects of stochastic errors on the image quality, averaging multiple shots does not make any difference as there is no variation from shot to shot.

Examples of deterministic errors in ToF cameras include motion blur and multipath interference. Motion blur errors are caused by any movement of camera or objects present in the scene. In other words, the error is due to the physical movement of either the objects

or the camera during the integration time used for sampling. When occurred, blurred regions in a 2D depth image give rise to significant distortions in the ensuing 3D reconstructions. Multipath interference errors are unpredictable since they depend on the geometry of the observed scene. This error is caused by the fact that the illumination produced may be reflected many times before reaching the pixel.

The most important stochastic error ToF suffer from is quantum noise, also called shot noise or photon noise, which constitutes the theoretical limit since it is the source of noise that cannot be eliminated; only averaging can limit the effects. Since this error is amplitude dependent, anything that affects amplitude can affect sensor accuracy. For instance, an object at a greater distance will be less illuminated than a near object and the reflectivity of the surface will also influence the measurement. Therefore, it is a random error.

Existing works present solutions to reduce noise affecting depth images. These works are mainly filtering methods, applied in the spatial domain, using the traditional filters which rely mainly on averaging, examples of these are the median filter and the bilateral filter.

Generally, Time of Flight cameras have high signal-to-noise ratio. That is, signal is dominated by object boundaries which have very large depth changes. Even under noisy circumstances, the depth map clearly shows object boundaries. However, noise does affect the estimated depth accuracy. Even if the sensor can achieve millimeter-level accuracy in theory, the noise makes it impossible for the sensor to operate at this accuracy level in practice. Hence, the only available option for the sensor to operate at the millimeter resolution now is to average across frames, to average out the noise. However, frame averaging is only acceptable for stationary scenes. We, therefore, need a denoising method if we wanted to work with video or dynamic scenes.

In addition, for the moment, there has been little work characterizing the internal noise of Time-of-Flight cameras. Most of the noise models presented in the literature assume that noise follows a normal distribution without empirical validation.

It would therefore be interesting to characterize and analyze the shot noise in the Time-of-Flight sensor to be able to model and estimate the uncertainty for each pixel, then use the model for to propose a new denoising technique other than the classical filtering methods. In the next chapter, we will present our new noise model which characterises the shot noise present in the Tof raw data.

CHAPTER IV
STATISTICAL MODELING OF TOF SENSOR DATA

As mentioned in Chapter III, the Time-of-Flight sensor circuit design is a modified version of an ordinary photodetector circuit because the latter circuit includes a single doped region while the ToF sensor circuit design can be described by two doped region (See Figure (3.2).b). Knowing that the process of electron-hole pairs generation post photons arrival at the a photo detector is statistically modelled as a Poisson process, which can be modeled by the following probability:

$$P(X = k) = \frac{e^{-\lambda t} (\lambda t)^k}{k!}, \quad (4.1)$$

where X is the number of photons measured in time t and λ is the intensity of the photons [25], the heteroskedastic nature of the photon arrival process on each doped region of the ToF sensor also follows a Poisson distribution.

The raw data of the ToF cameras correspond to the output voltage of their sensors which is directly related to the difference in charge between the two capacitors of the doped regions. Therefore, the camera's raw data being proportional to the random number of photons hitting its sensor, can be modeled as a difference of two Poisson distributions.

In fact, there exists a distribution that models the difference of two Poisson's processes. The Skellam distribution, formulated by Skellam in 1946, is a function of the difference k between two Poisson random variables, expressed by the following probability mass function (PMF):

$$P(k, \mu_1, \mu_2) = e^{-(\mu_1 + \mu_2)} \left(\frac{\mu_1}{\mu_2} \right)^{\frac{k}{2}} I_k(2\sqrt{\mu_1 \mu_2}), \quad (4.2)$$

where μ_0 and μ_1 are are expected values or means of two Poisson distributions, and I_k is the modified Bessel function of the first kind.

Consequently, during measurement, raw data corresponding to the photon count hitting a ToF sensor can be modelled as a random variable following a Skellam distribution.

To retrieve scene depth information from the phase delay Δ , as described in Chapter II, variables X and Y are created. The two variables are a difference of two raw ToF camera phase measurements, where each measurement is Skellam distributed:

$$\begin{cases} Phase_0 \sim \text{Skellam}(a_0, b_0) \\ Phase_{180} \sim \text{Skellam}(a_1, b_1) \\ Phase_{90} \sim \text{Skellam}(a_2, b_2) \\ Phase_{270} \sim \text{Skellam}(a_3, b_3) \end{cases}$$

In the following, we prove that X and Y also follow a Skellam distribution, since the difference of two Skellam distributed random variables gives a new random variable which is Skellam distributed.

Let S_1 and S_2 be two Skellam distributed random variables, which are both a difference of two Poisson's random variables (P_1, P_2) and (P_3, P_4) respectively : We have:

$$\begin{cases} S_1 = P_1 - P_2 \\ S_2 = P_3 - P_4 \end{cases}$$

Since the sum of two Poisson random variables is also a Poisson distributed, we have:

$$S = S_1 - S_2 = \underbrace{(P_1 + P_4)}_{\text{Poisson}} - \underbrace{(P_2 + P_3)}_{\text{Poisson}} \quad (4.3)$$

A difference of two Poisson variables generates a Skellam distributed random variable. Hence, both X and Y follow a Skellam distribution, with means $(\mu_0 - \mu_1)$ and $(\mu_2 - \mu_3)$ respectively:

$$\begin{cases} X = Phase_0 - Phase_{180} \\ X \sim \text{Skellam}(a_0 + b_1, a_1 + b_0) \\ X \sim \text{Skellam}(\mu_0, \mu_1) \end{cases}$$

Similarly for Y, we have:

$$\begin{cases} Y &= Phase_{90} - Phase_{270} \\ Y &\sim \text{Skellam}(a_2 + b_3, a_3 + b_2) \\ Y &\sim \text{Skellam}(\mu_2, \mu_3) \end{cases}$$

The two random variables X and Y can then be expressed as having the following probability functions :

$$P(X, \mu_0, \mu_1) = e^{-(\mu_0 + \mu_1)} \left(\frac{\mu_0}{\mu_1} \right)^{\frac{X}{2}} I_X(2\sqrt{\mu_0\mu_1}), \quad (4.4)$$

$$P(Y, \mu_2, \mu_3) = e^{-(\mu_2 + \mu_3)} \left(\frac{\mu_2}{\mu_3} \right)^{\frac{Y}{2}} I_Y(2\sqrt{\mu_2\mu_3}). \quad (4.5)$$

To derive the equations for μ_0, μ_1, μ_2 and μ_3 , these latter are plotted in terms of normalized Δ in Figure 4.1.

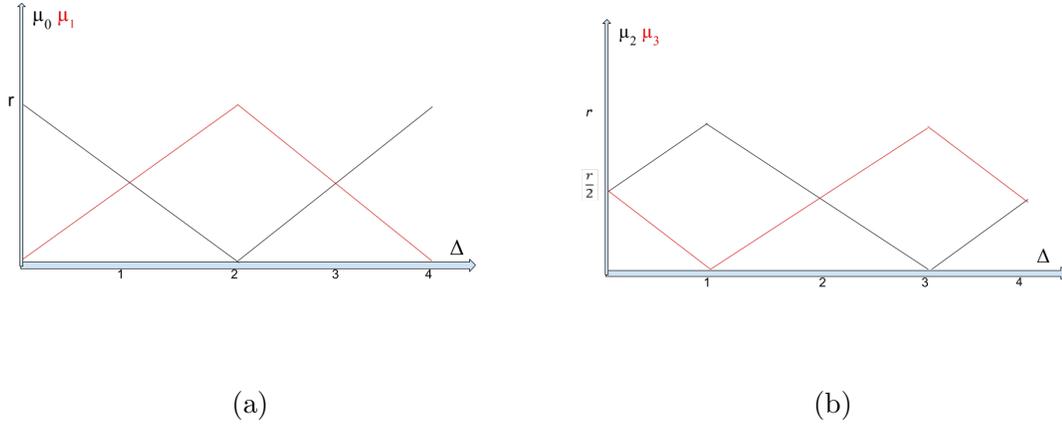


Figure 4.1: Different parameters of X and Y probability mass functions as function of Δ for the four different quadrants. **(a)** μ_0 and μ_1 . **(b)** μ_2 and μ_3 .

The numbers 1,2,3 and 4 on Figure 4.1 represent not only the normalized delay Δ but also the four different quadrants. Equation 4.6 summarize the parameters of X and Y .

$$\begin{cases} \mu_0 = r \cdot Q_x(q)(1 - \frac{1}{2}\Delta), \\ \mu_1 = r \cdot [(1 - Q_x(q)) + (\frac{Q_x(q)}{2}\Delta)], \\ \mu_2 = \frac{r}{2}[(Q_y(q) \cdot \Delta + 1) + (1 - Q_y(q))], \\ \mu_2 = \frac{r}{2} \cdot Q_y(q)(1 - \Delta). \end{cases} \quad (4.6)$$

Where:

$$Q_x = \begin{bmatrix} 1 \\ 1 \\ -1 \\ -1 \end{bmatrix} \text{ and } Q_y = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}.$$

The index $q = \{1, 2, 3, 4\}$ specifies which quadrant of the L_1 unit ball the ToF camera is operating and r is the reflected intensity value.

To validate our model, we use the Q-Q plot, also known as the Quantile-Quantile plot, which is a diagnostic plot to check how close our raw ToF camera data is to our Skellam model and its assumptions. The quantiles of our distribution are plotted against the quantiles of a theoretical Skellam distribution, as shown in Figure 4.2. A 45 degree reference line is also drawn. When the points fall approximately along the line, the two sets, our raw X and Y data and the theoretical data of a Skellam distribution, have the same distribution. We can clearly see in Figure (4.2) that the quantiles of X and Y correspond perfectly to those of a Skellam distribution, which allows us to validate our Skellam model for X and Y . The obtained probability mass functions can then be used in the next chapter to derive a denoising algorithm for 2D depth images acquired by a ToF camera.

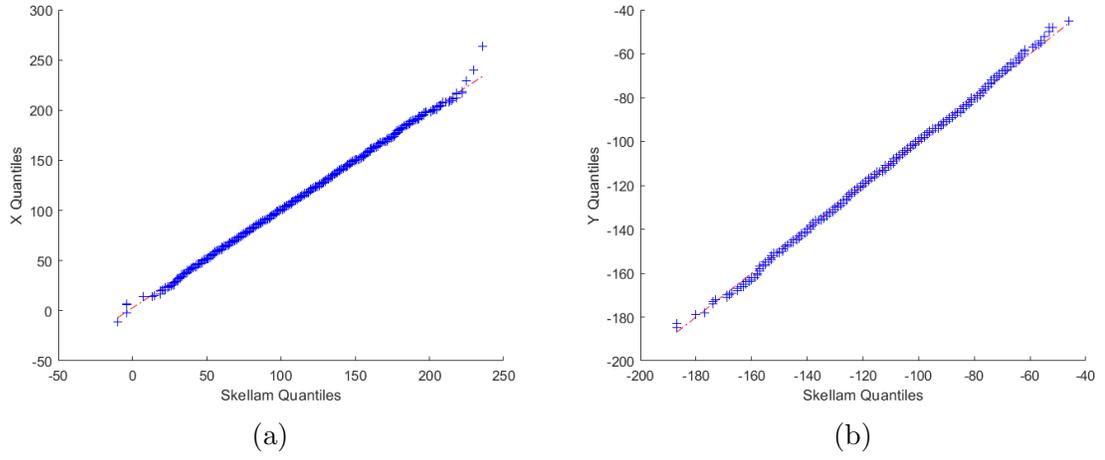


Figure 4.2: Quantile-Quantile plot that compares X and Y ToF camera raw data distribution with a Skellam distribution. **(a)** Q-Q plot of X . **(b)** Q-Q plot of Y .

CHAPTER V

DEPTH IMAGE DENOISING

In this chapter, we present our novel wavelet-based denoising method that uses Bayes theory to both estimate denoised depth information and preserve depth edges at the millimeter level.

5.1 Denoising in the wavelet domain

In general, it is possible to carry out a decomposition into wavelets of an image then to reconstruct this image from its wavelet coefficients. However, this technique would not have much interest if we did not modify these coefficients because we would obtain a final image identical to the initial image. The wavelet coefficients mark the discontinuities which occur in the image, thus they match the details. If, now, these coefficients are thresholded, this amounts to eliminating the finest details of the image.

Numerous wavelet-based thresholding methods were proposed for image denoising because of the simplicity and the effectiveness of the method. However, the method's problem lies in the optimal threshold selection, a threshold that is used to discriminate between the coefficients that are for the most part due to signal and those mostly due to noise.

In contrast, the prior probabilistic knowledge of the noisy image in wavelet domain can be exploited in the denoising process. In lieu of using a defined threshold to shrink the noisy wavelet coefficients, not taking into account previous information about signal structure and noise well, the shrinkage function is rather designed based on minimizing a Bayesian risk, which can generally be under the Minimum Mean Square Error (MMSE) cri-

terion, Maximum A Posteriori (MAP) criterion or Minimum Mean Absolute Error (MMAE) criterion.

In order to use Bayes theory to design a wavelet shrinkage function established by posterior summaries of the wavelet coefficients, the probability distributions of our sensor data X and Y must be derived in the wavelet domain. Section 5.2 below addresses this.

5.2 Statistical modeling of ToF sensor data in the wavelet domain

We chose the Haar transform in this work to address our wavelet analysis. It is one of the oldest transform functions, proposed in 1910 by the Hungarian mathematician Alfréd Haar. It is found effective in applications related to image processing as it provides a conceptually simple and computationally efficient approach for analysing the local aspects of a signal.

D_{11}	D_{12}			
D_{21}	D_{22}			

Figure 5.1: 2×2 depth image neighborhood.

Considering the four pixels neighborhood represented on Figure 5.1, the haar first level wavelets of X can be calculated using the matrix below:

$$\begin{bmatrix} W_{X_S} \\ W_{X_H} \\ W_{X_V} \\ W_{X_D} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & -1 \end{bmatrix} \begin{pmatrix} X_{11} \\ X_{12} \\ X_{21} \\ X_{22} \end{pmatrix}$$

Similarly for the wavelets of Y , we have:

$$\begin{bmatrix} W_{Y_S} \\ W_{Y_H} \\ W_{Y_V} \\ W_{Y_D} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & -1 \end{bmatrix} \begin{pmatrix} Y_{11} \\ Y_{12} \\ Y_{21} \\ Y_{22} \end{pmatrix}$$

To start with, to prove that X and Y are Skellam distributed in the wavelet domain as well, we will consider their horizontal wavelets components, W_{X_H} and W_{Y_H} respectively.

$$\begin{aligned} W_{X_H} &= (X_{11} + X_{12}) - (X_{21} + X_{22}) \\ W_{Y_H} &= (Y_{11} + Y_{12}) - (Y_{21} + Y_{22}) \end{aligned} \tag{5.1}$$

Given that both X and Y are both Skellam distributed with the parameters given in Equation 4.6, we have:

$$\begin{aligned} W_{X_H} &\sim \text{Skellam}(\mu_{0_{X_{11}}} + \mu_{0_{X_{12}}}, \mu_{1_{X_{11}}} + \mu_{1_{X_{12}}}) - \text{Skellam}(\mu_{0_{X_{21}}} + \mu_{0_{X_{22}}}, \mu_{1_{X_{21}}} + \mu_{1_{X_{22}}}) \\ &\sim \text{Skellam}[rQ_X(q)(2 - \frac{1}{2}(D_{11} + D_{12})), r((2 - 2Q_X(q)) + \frac{Q_X(q)}{2}(D_{11} + D_{12}))] \\ &\quad - \text{Skellam}[rQ_X(q)(2 - \frac{1}{2}(D_{21} + D_{22})), r((2 - 2Q_X(q)) + \frac{Q_X(q)}{2}(D_{21} + D_{22}))] \end{aligned} \tag{5.2}$$

Using the property described in Equation IV, the horizontal wavelet of X is a Skellam distributed with the following parameters:

$$\begin{aligned} W_{X_H} &\sim \text{Skellam}(2 - \frac{1}{2}Q_X(q)(\mathbf{D}_{11} + \mathbf{D}_{12} - \mathbf{D}_{21} - \mathbf{D}_{22}), 2 + \frac{1}{2}Q_X(q)(\mathbf{D}_{11} + \mathbf{D}_{12} - \mathbf{D}_{21} - \mathbf{D}_{22})) \\ &\sim \text{Skellam}(2 - \frac{1}{2}Q_X(q)\mathbf{W}_{\mathbf{D}_H}, 2 + \frac{1}{2}Q_X(q)\mathbf{W}_{\mathbf{D}_H}). \end{aligned} \tag{5.3}$$

Similarly for W_{Y_H} , we have:

$$\begin{aligned} W_{Y_H} &\sim \text{Skellam}(\mu_{2_{Y_{11}}} + \mu_{2_{Y_{12}}}, \mu_{3_{Y_{11}}} + \mu_{3_{Y_{12}}}) - \text{Skellam}(\mu_{2_{Y_{21}}} + \mu_{2_{Y_{22}}}, \mu_{3_{Y_{21}}} + \mu_{3_{Y_{22}}}) \\ &\sim \text{Skellam}[\frac{r}{2}(4 - 2Q_Y(q) + Q_Y(D_{11} + D_{12})), \frac{r}{2}(2Q_Y(q) - (D_{11} + D_{12}))] \\ &\quad - \text{Skellam}[\frac{r}{2}(4 - 2Q_Y(q) + Q_Y(D_{21} + D_{22})), \frac{r}{2}(2Q_Y(q) - (D_{21} + D_{22}))] \end{aligned} \tag{5.4}$$

Using the property described in equation IV, W_{YH} is also Skellam distributed as shown below:

$$W_{YH} \sim Skellam(r(2 - \frac{1}{2}Q_Y(q)\mathbf{W}_{D_H}), r(2 + \frac{1}{2}Q_Y(q)\mathbf{W}_{D_H})). \quad (5.5)$$

Following the same process, the probability distribution of the vertical and the diagonal wavelets of X and Y are derived. It turns out that all of the wavelets of X and Y are Skellam distributed with parameters that depend on the quadrant the depth image pixels are and the level of wavelet transform. First level W_X and W_Y equations are summarized in the two following equations:

$$\begin{cases} W_X \sim Skellam(r(2 - \frac{1}{2}Q_X(q)\mathbf{W}_D), r(2 + \frac{1}{2}Q_X(q)\mathbf{W}_D)), \\ W_Y \sim Skellam(r(2 - \frac{1}{2}Q_Y(q)\mathbf{W}_D), r(2 + \frac{1}{2}Q_Y(q)\mathbf{W}_D)). \end{cases} \quad (5.6)$$

Equation 5.6 is generalized for different levels as:

$$\begin{cases} W_X \sim Skellam(r(2^l - \frac{1}{2}Q_X(q)\mathbf{W}_D), r(2^l + \frac{1}{2}Q_X(q)\mathbf{W}_D)), \\ W_Y \sim Skellam(r(2^l - \frac{1}{2}Q_Y(q)\mathbf{W}_D), r(2^l + \frac{1}{2}Q_Y(q)\mathbf{W}_D)), \end{cases} \quad (5.7)$$

where l refers to the level of the wavelet transform. r is taken as the averaged wavelet coefficients of intensity values (intensity scaling coefficient).

Similar to what we did in the spatial domain, we use the Q-Q plot to check how close the ToF camera data, expressed in the wavelet domain, is to our derived wavelet Skellam model. The quantiles of our Skellam distributions W_X and W_Y are plotted against the quantiles of a theoretical Skellam distribution. Figure 5.2 shows the Q-Q plot of three wavelet components of W_X (Horizontal, Vertical and Diagonal), while Figure 5.3 illustrate the Q-Q plot of W_Y wavelet components.

We can clearly see from Figures ((5.2) - (5.3)) that the quantiles of W_X and W_Y correspond perfectly to those of a Skellam distribution, allowing us to validate our Skellam model in the wavelet domain.

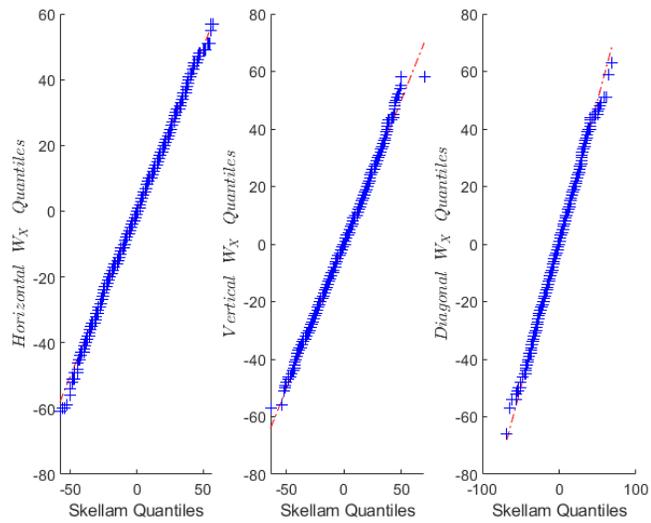


Figure 5.2: Quantile-Quantile plot that compares W_X components distribution to a Skellam distribution.

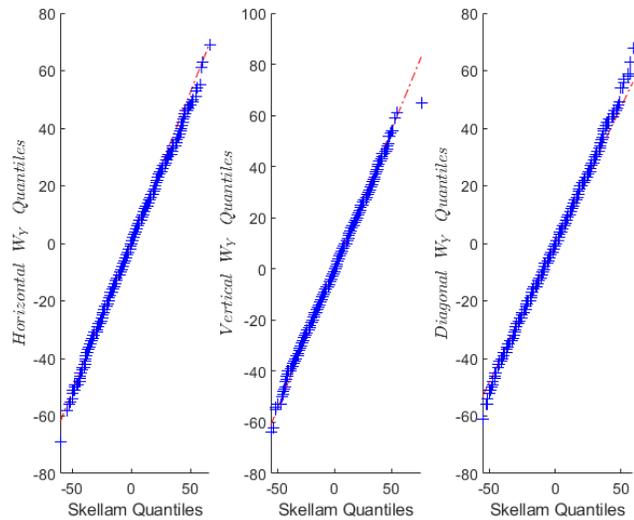


Figure 5.3: Quantile-Quantile plot that compares W_Y components distribution to a Skellam distribution.

5.3 Bayesian denoising

Bayesian statistics are based on the posterior distribution $P(\theta | T)$. The posterior distribution, which is described in Equation 5.8, can be interpreted as a summary (in a probabilistic sense) of the information available on the parameter θ , once T is observed.

$$P(\theta) = \frac{P(T | \theta)P(\theta)}{\int P(T | \theta)P(\theta)d\theta} \quad (5.8)$$

As we can see from Equation 5.8, the likelihood $P(T | \theta)$ along with prior $P(\theta)$ are used to obtain a posterior distribution, which is used to estimate clean parameters. In our case, Equation 5.8 becomes:

$$P(D, R | X, Y) = \frac{P(X, Y | D, R)P(D, R)}{\int P(X, Y | D, R)P(D, R) dD} \quad (5.9)$$

where $P(D, R | X, Y)$ is the likelihood, $P(D, R)$ is the prior and the denominator is the marginal likelihood. The prior in our case is both the depth D and the intensity R as the observations X and Y depend on both.

The most widely used Bayesian estimators in signal processing applications are the MMSE estimator and the MAP estimator. In our work, we are interested in applying the latter to estimate the "clean" depth values d of our noisy depth D . The MAP estimator is obtained by seeking the mode of the a posteriori rule of the prior parameters, namely D and R , given in Equation 5.9.

Our MAP (Maximum A Posteriori) Estimator is then defined as:

$$\begin{aligned} \tilde{d}(X, Y) &= \arg \max_D P(X, Y | D, R) = \arg \max_D \frac{P(X, Y | d, R)P(D, R)}{\int P(X, Y | D, R)P(d, R) dD} \\ &\propto P(X, Y | D, R)P(D)P(R) \\ &\propto P(X, Y | D, R)P(D) \end{aligned} \quad (5.10)$$

Since we want our new denoising method to be applied in the wavelet domain, it is no longer the depth D that is denoised but its wavelets W_D . Our MAP estimator in the wavelet domain is thus described as:

$$\begin{aligned}\tilde{W}_d &= \arg \max_{W_D} P(W_D | W_X, W_Y) \\ &\propto P(W_X, W_Y | W_D, W_R)P(W_D)\end{aligned}\tag{5.11}$$

A well known choice for prior coefficients distributions is the use of a combination of scales of two distributions. In our case, we opt for a scaling mixture of two normal distributions. While one component of the Gaussian mixture would correspond to "trivial" coefficients otherwise considered to be flat regions of the $2D$ depth image, the other corresponds to "significant" regions that are likely to be depth edges. Since we divide depth wavelets into edge and edge-less ones, we can write the complete distribution of the prior $P(W_D)$ as:

$$P(W_D) = \pi P(W_D | \sigma_1) + (1 - \pi) P(W_D | \sigma_2),\tag{5.12}$$

where $\pi = P(\text{region is a not on an edge})$ and $(1 - \pi) = P(\text{region is on an edge})$, $P(W_D | \sigma_1)$ and $P(W_D | \sigma_2)$ represent the two Gaussian distributions, σ_1 and σ_2 being their respective standard deviations. Figure 5.4 illustrates the two Gaussian distributions that constitute our Gaussian mixture model.

Consequently, using total conditional probability properties, our posterior distribution can be written as:

$$\begin{aligned}P(W_D | W_X, W_Y) &= P(W_D | W_X, W_Y, \sigma_1)P(\sigma_1 | W_X, W_Y) \\ &+ P(W_D | W_X, W_Y, \sigma_2)P(\sigma_2 | W_X, W_Y),\end{aligned}\tag{5.13}$$

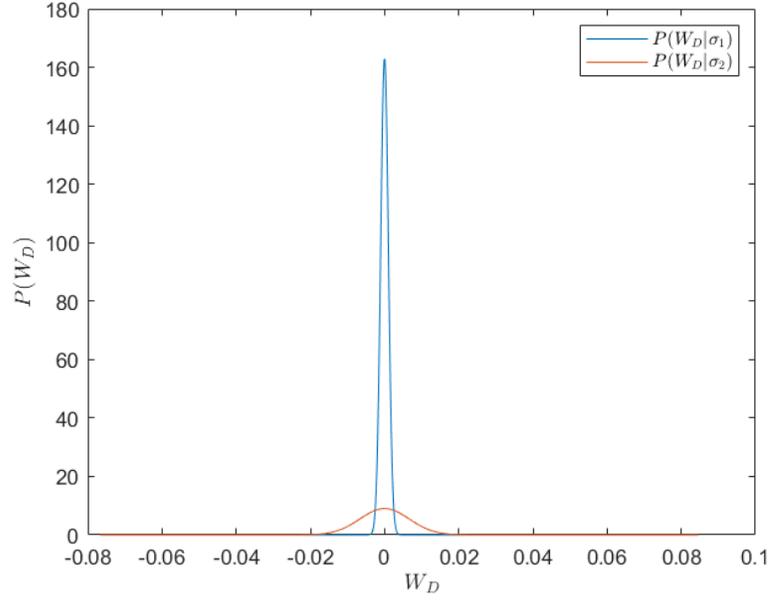


Figure 5.4: Prior Gaussian Mixture Model.

where $P(\sigma_1 | W_X, W_Y)$ and $P(\sigma_2 | W_X, W_Y)$ represent the posterior mixture weights, which can be expressed as:

$$\begin{aligned}
 P(\sigma_1 | W_X, W_Y) &= \frac{P(W_X, W_Y | \sigma_1)P(\sigma_1)}{P(W_X, W_Y | \sigma_1)P(\sigma_1) + P(W_X, W_Y | \sigma_2)P(\sigma_2)}, \\
 P(\sigma_2 | W_X, W_Y) &= \frac{P(W_X, W_Y | \sigma_2)P(\sigma_2)}{P(W_X, W_Y | \sigma_1)P(\sigma_1) + P(W_X, W_Y | \sigma_2)P(\sigma_2)}.
 \end{aligned} \tag{5.14}$$

Developing further we get:

$$\begin{aligned}
 P(\sigma_1 | W_X, W_Y) &= \frac{\pi \int P(W_X, W_Y | W_D)P(D | \sigma_1) dD}{\pi \int P(W_X, W_Y | W_D)P(D | \sigma_1) dD + (1 - \pi) \int P(W_X, W_Y | W_D)P(D | \sigma_2) dD}, \\
 P(\sigma_2 | W_X, W_Y) &= \frac{(1 - \pi) \int P(W_X, W_Y | W_D)P(D | \sigma_2) dD}{\pi \int P(W_X, W_Y | W_D)P(D | \sigma_1) dD + (1 - \pi) \int P(W_X, W_Y | W_D)P(D | \sigma_2) dD}.
 \end{aligned} \tag{5.15}$$

Now that we have a mathematical equation for our posterior distribution, we want to apply the MAP estimator to it. The estimator can be applied directly to the entire expression, however, we want to distribute the MAP estimator over the two posteriors that make up our mixture posterior, expressed in the Equation 5.13.

An assumption must be made before distributing the MAP estimator on Equation 5.13. The assumption consists on the posterior mixture weights behaving like an edge detector. In other words, when $P(\sigma_1 | W_X, W_Y)$ is high approaching the value 1, $P(\sigma_2 | W_X, W_Y)$ must be low approaching the value 0, and vice versa. Figure 5.5 shows the horizontal and the vertical components of the updated weights probabilities of the reconstructed depth image in Figure 3.6.

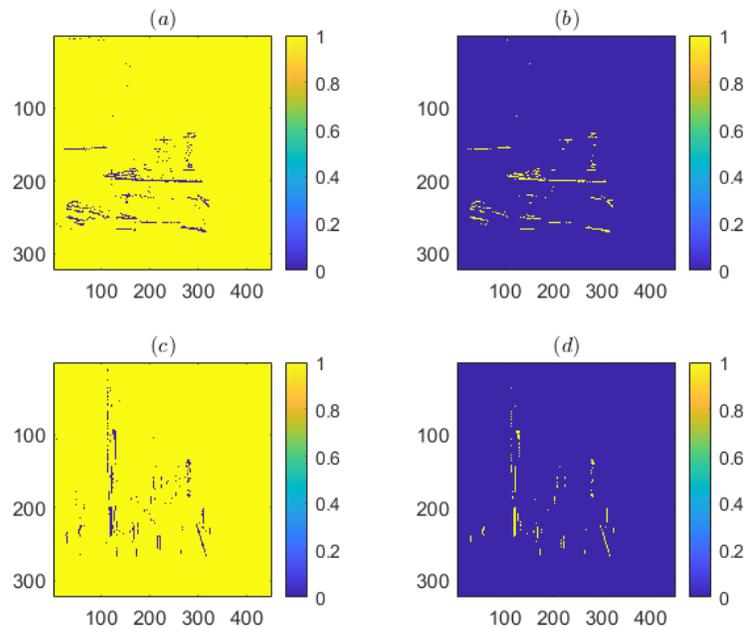


Figure 5.5: Posterior mixture probabilities. (a) Probability of a not a horizontal edge. (b) Probability of a horizontal edge. (c) Probability of a not a vertical edge. (d) Probability of a vertical edge.

Distributing now the MAP estimator on the posterior described by Equation 5.13, we obtain:

$$\begin{aligned} \tilde{W}_d \approx P(\sigma_1 | W_X, W_Y) \arg \max_{W_d} P(W_D | W_X, W_Y, \sigma_1) \\ + P(\sigma_2 | W_X, W_Y) \arg \max_{W_d} P(W_D | W_X, W_Y, \sigma_2), \end{aligned} \quad (5.16)$$

where the two MAP estimators are given as:

$$\begin{aligned} \arg \max_{W_D} P(W_D | W_X, W_{Y,1}) &\propto P(W_X, W_Y | W_D, W_R, \sigma_1)P(W_D | \sigma_1), \\ \arg \max_{W_D} P(W_D | W_X, W_{Y,2}) &\propto P(W_X, W_Y | W_D, W_R, \sigma_2)P(W_D | \sigma_2). \end{aligned} \quad (5.17)$$

Therefore, the process of denoising the ToF depth images can be performed by applying the Discrete Haar Wavelet Transform on the depth images, the estimated clean wavelet distance values are obtained by applying the MAP estimator on our mixture posterior resulting in an optimization problem where the denoised distance wavelets are those that maximize the posteriors probability functions. An Inverse Discrete Wavelet Transform (IDWT) is then applied to the estimated depth image in order to recover the clean depth values in the spacial domain. Our algorithm is tested and evaluated on different ToF depth images in the next chapter.

CHAPTER VI

RESULTS

Raw images used in this thesis are taken by the Helios camera manufactured by Lucid Vision Labs. The camera integrates Sony’s new DepthSense IMX556PLR back-illuminated ToF image sensor. All images are taken in a raw sensor format, depth information of the acquired scenes are reconstructed using Equation 3.21. Camera settings were manually set to maintain consistency between shots. To evaluate our denoising method in extreme conditions, we also made sure to opt for camera settings that produce the noisiest raw images.

For the purpose of evaluating our method, a ground truth is necessary. For each scene captured, 200 raw images were taken with the same camera settings. Their corresponding reconstructed depth images are averaged and the resulting depth image is used as a ground truth.

It should be emphasized that the depth errors present in the recorded depth images arise not only from ToF camera noise, but also from all other types of errors such as multipath or intensity-dependent errors. In some cases, these types of errors even dominate errors due to photonic noise. Additionally, it can be noted that some surfaces appear curved in the 2D reconstructed depth map, this is due to the fact that ToF cameras provide the radial depth to the center of the camera.

The performance of the proposed ToF depth image denoising method is compared to state-of-the-art and standard benchmark ToF denoising methods. These methods are BM3D [26], Adaptive Total Variation [4], Adaptive Joint-Bilateral Filter [27] and Non-Local-Mean [28]. For a fair comparison, care must be taken to select the optimal parameters for each

method. For each of the methods, we search adaptively for optimal parameters such that the root mean square error (MSE) with respect to the ground truth is minimized and the depth edges are well preserved.

The depth images represented in Figure 6.1 (a) and (c) are reconstructed, the figure also illustrates their intensity images which give more information about the scene, in particular the contours. Three coins are purposely placed on the wall in the scene of Figure 6.1 (a) to test the ability of our denoising method to preserve small depth edge details.

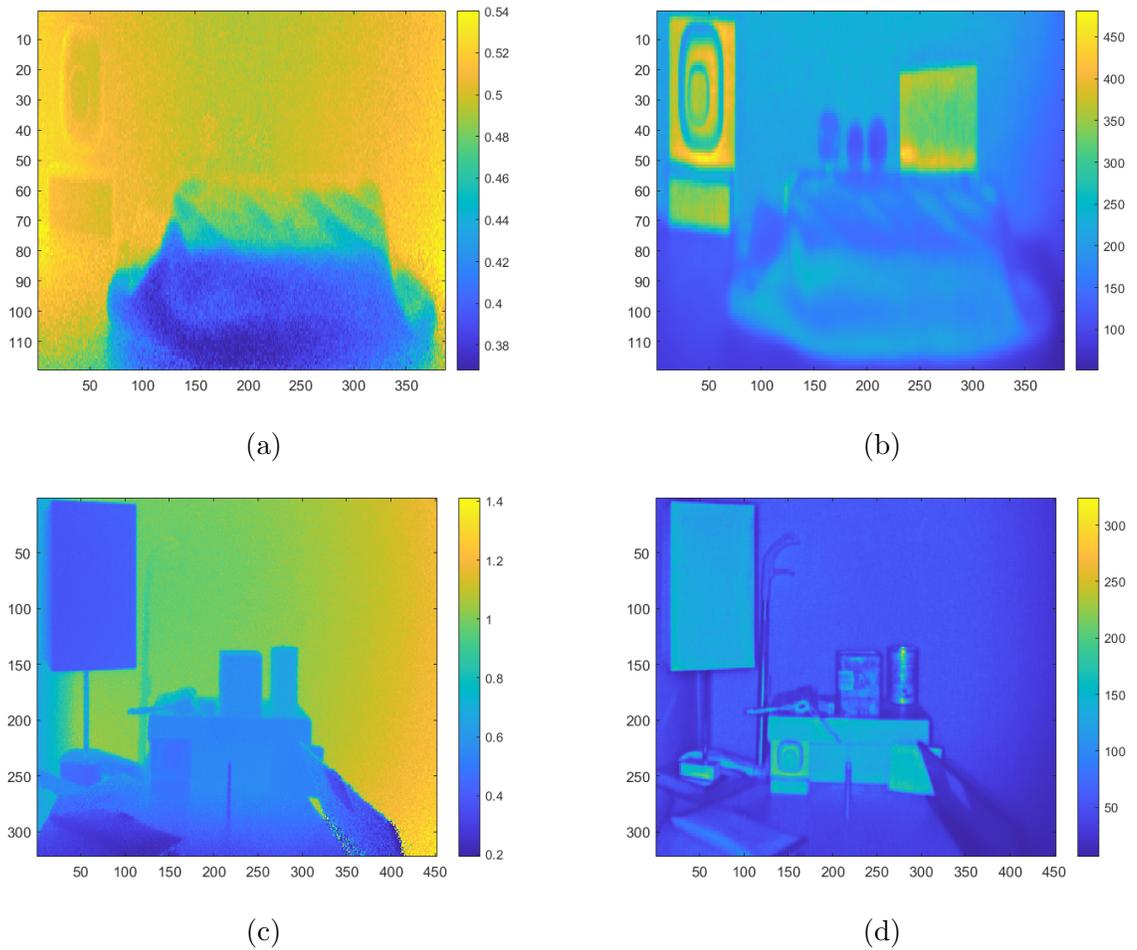
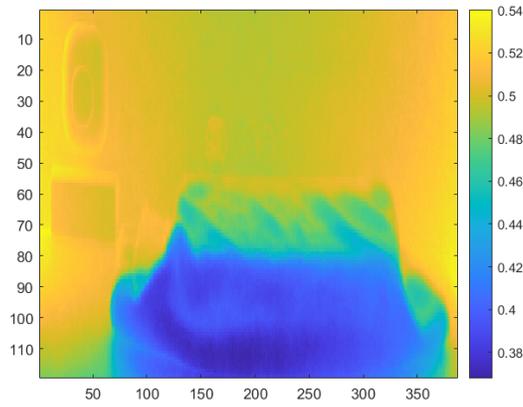


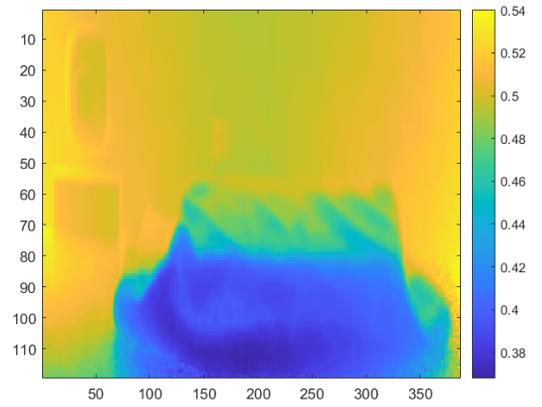
Figure 6.1: (a)-(c) Noisy reconstructed depth images. (b)-(d) Their corresponding intensity images.

Our denoising method and the previously stated other methods are applied to the noisy depth images in Figure 6.1. The results of all individual methods applied are depicted in Figure 6.2 and Figure 6.4.

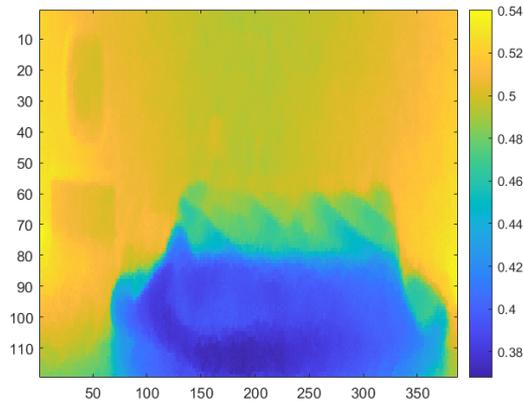
As shown in Figures 6.2-6.3, the proposed denoising method successfully reduces the photon noise while preserving fine details that enhance the overall viewing experience. Specifically, the denoising of smooth regions of all images have flat appearance with little residual noise. For each depth image acquired, we compare our denoising performance with that obtained by the BM3D, Non Local Means (NLM), Joint Bilateral and Total Variation filters. BM3D and Non Linear Mean are able to denoise regions of the scene well but not others. Some of the fine edges are over-smoothed using these filters, as we can see in Figure 6.3. Compared to other filters, the Total Variation filter was not able to smooth regions as effectively. The NLM filter was able to denoise some regions of the scenes well but not others. The Joint Bilateral together with our proposed denoising methods yielded the best results in terms of MSE and edge preserving. However, our method gave the best results, specifically with regard to edge preservation (see Figure 6.3).



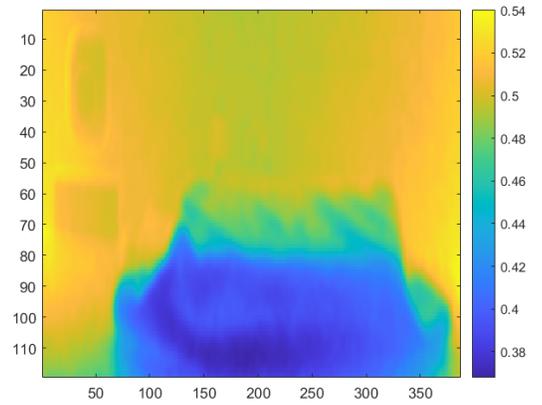
(a)



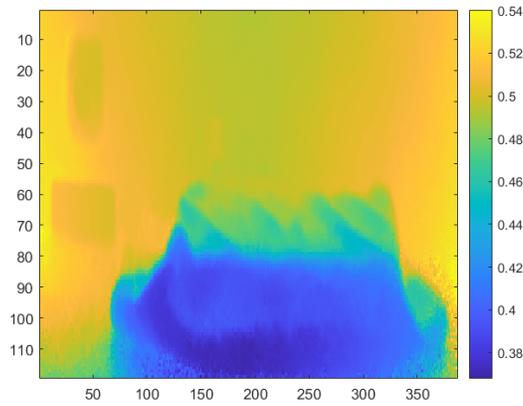
(b)



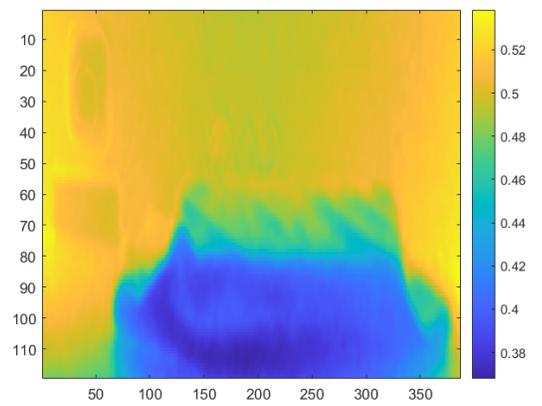
(c)



(d)



(e)



(f)

Figure 6.2: Denoising results. (a) Ground Truth image. (b) BM3D denoised result. (c) Total Variation denoised result. (d) Joint Bilateral denoised result. (e) Non Local Mean denoised result. (f) Proposed denoised result.

To better assess our denoising method at the millimeter level, the depth range of the scene presented in Figure 6.2 is reduced enough to visualize only the three coins. As we can see from Figure 6.3, the performance of our method is the best.

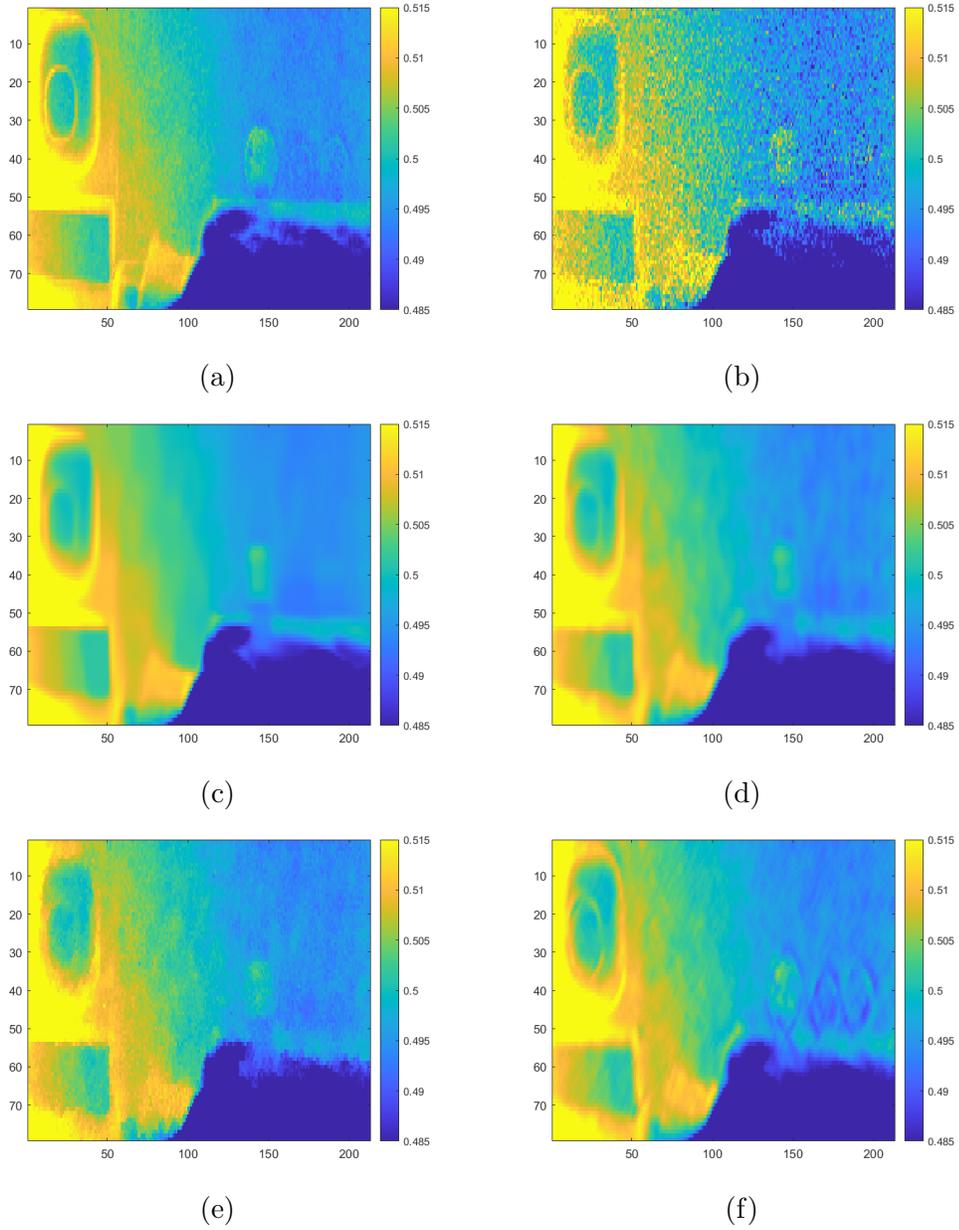
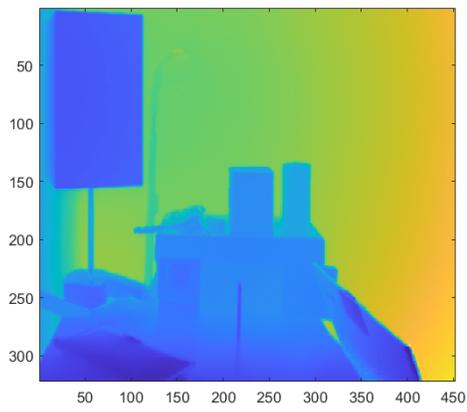
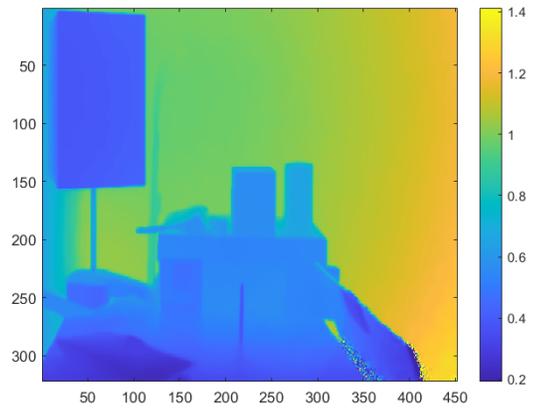


Figure 6.3: Finer depth range of Figure 6.2. (a) Ground Truth image. (b) Noisy image. (c) BM3D denoised result. (d) Joint Bilateral denoised result. (e) Total Variation denoised result. (f) Proposed method result.

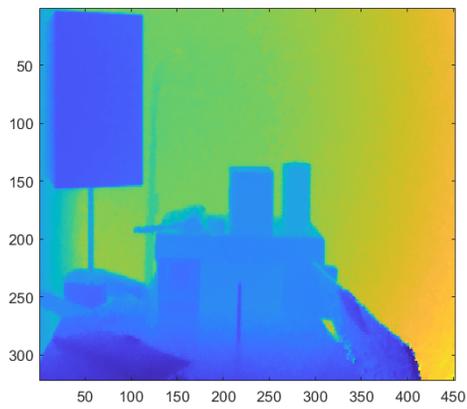
Figure 6.4 depicts the obtained results of different denoising methods of another image.



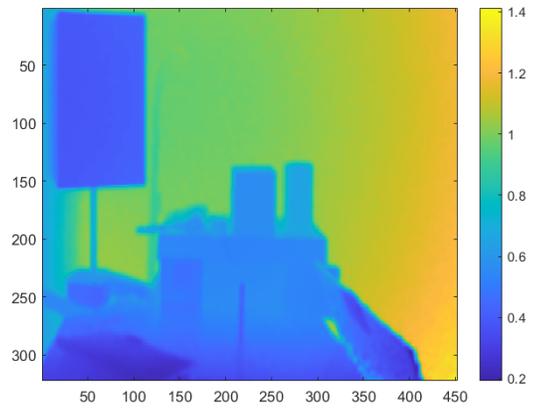
(a)



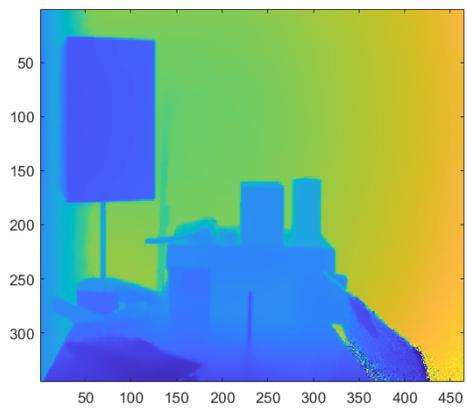
(b)



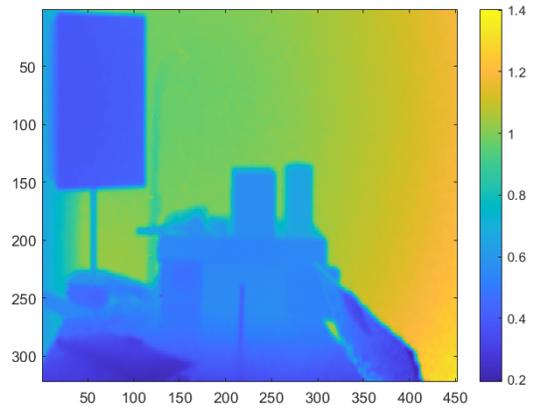
(c)



(d)



(e)



(f)

Figure 6.4: Denoising results. (a) Ground Truth image. (b) BM3D denoised result. (c) Total Variation denoised result. (d) Joint Bilateral denoised result. (e) Non Local Mean denoised result. (f) Proposed denoised result.

Figure 6.5 is a zoomed (zoomed depth) version of the previous figure.

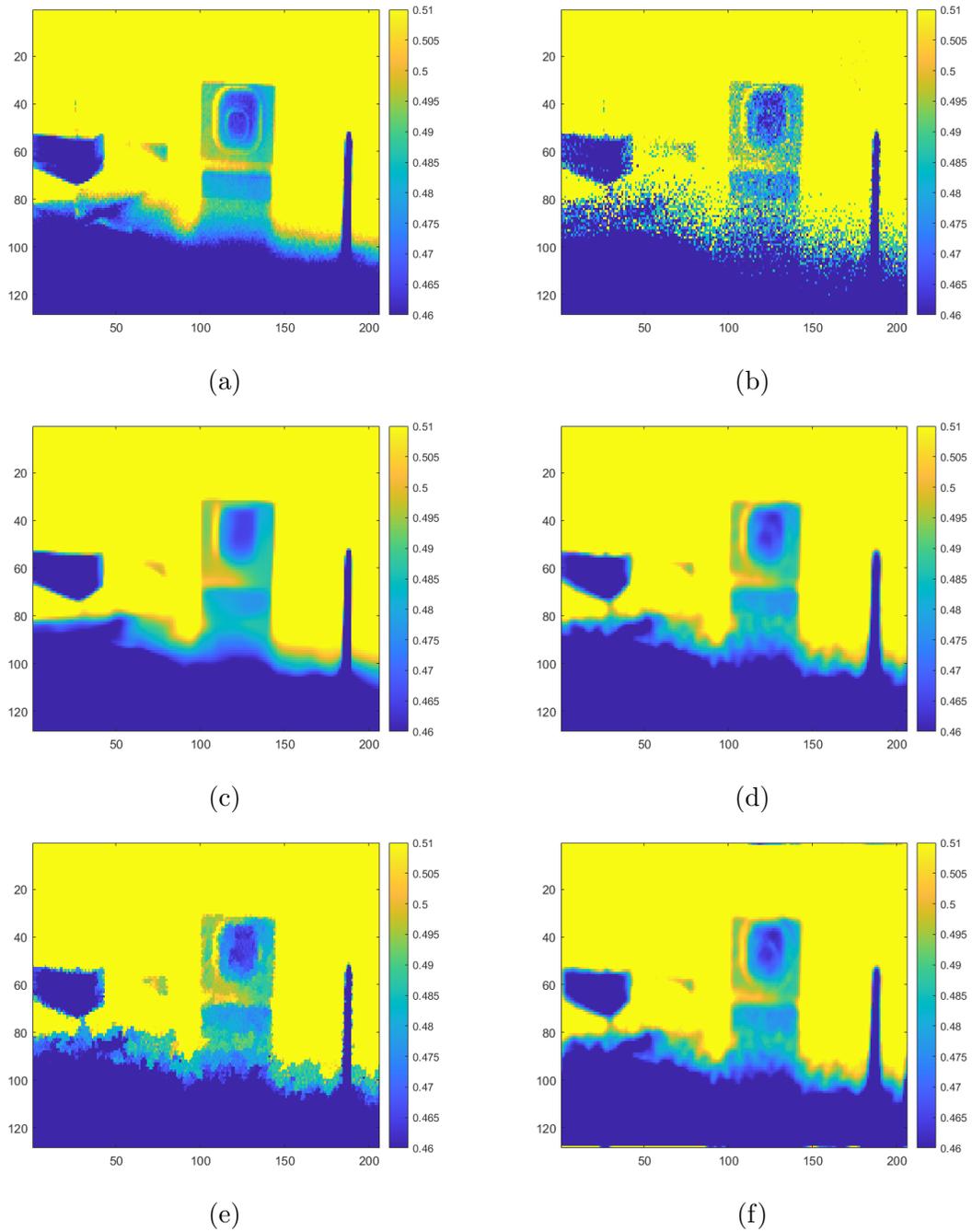


Figure 6.5: Finer depth range of Figure 6.4. (a) Ground Truth image. (b) Noisy image. (c) BM3D denoised result. (d) Joint Bilateral denoised result. (e) Total Variation denoised result. (f) Proposed method result.

Similarly, we can see that the results of our denoising algorithm in terms of edge preserving at the millimeter level are superior. The algorithm performs well in flats areas as well.

To evaluate our denoising algorithm in terms of depth accuracy and compare its result with those of the state-of-the-art methods stated previously, we provide in Table 6.1 the mean square error (MSE) to the ground truth of the different methods. The MSE is calculated for various depth image contents, including edges, reflectivity, and depth range. The results show that our denoising algorithm gave the lowest mean squared error.

Table 6.1: Denoising performance of various methods, averaged over 15 image. Mean squared error (MSE) to the ground truth of the methods under consideration. Our proposed method gives the smallest MSE.

Algorithm	Mean Squared Error (MSE)
Joint Bilateral	$1.3521 \cdot 10^{-5}$
BM3D	$1.9138 \cdot 10^{-5}$
Total Variation	$1.8122 \cdot 10^{-5}$
Non-Local Mean	$3.6715 \cdot 10^{-5}$
Proposed Method	$1.3521 \cdot 10^{-5}$

In denoising tasks, one of the major problems is to find a trade-off between noise removal and preservation of the detailed structures (edges) of the denoised image. To evaluate both clean depth reconstruction of our denoising method and its ability to preserve fine details, depth charts shown in Figure 6.6 are designed and 3D printed. The width of the triangles ranges from 1 millimeter to 10 millimeters, the triangular shapes are chosen on purpose because they taper towards their ends, which decreases the spatial resolution, making it harder for the denoising algorithm to recover the edges. Two versions of the depth maps

are 3D-printed: white and dark. The latter produces extremely noisy depth images because the pulsed light emitted is almost entirely absorbed by it.

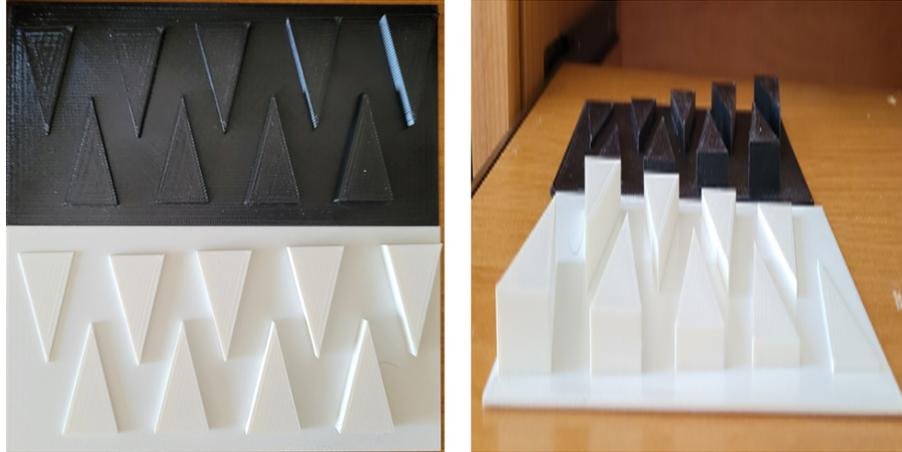
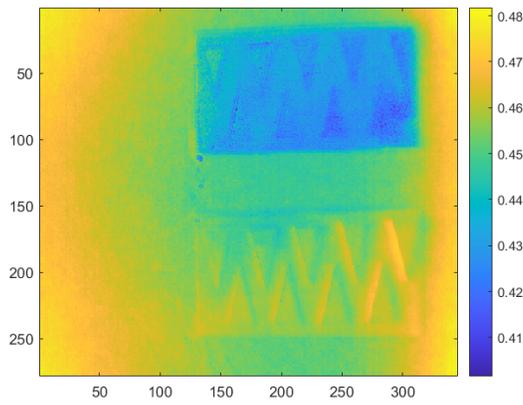
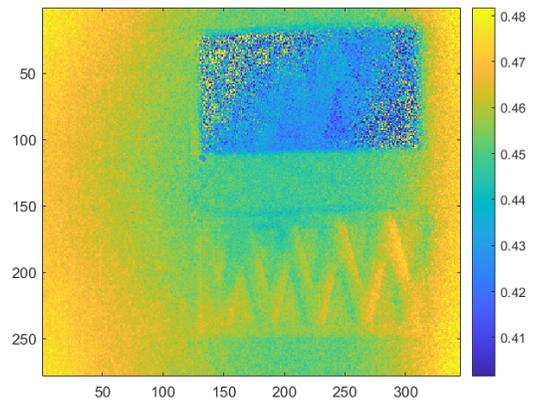


Figure 6.6: Top and side views of the 3D depth charts.

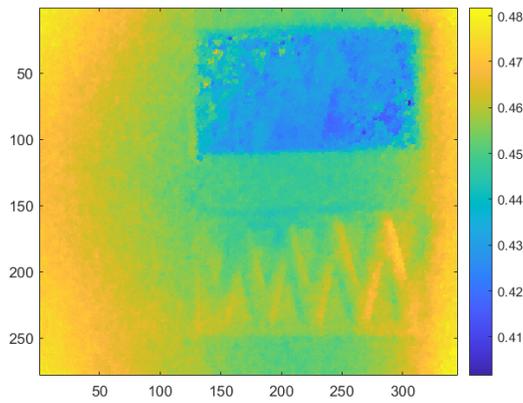
Figure 6.7 compares the denoising results of other denoising methods with the result of our method. The black chart is shown at the top of the image, the white at the bottom. We can see that our method was successful in both denoising the depth image and preserving its details. The left top both the depth charts have finer depths of 1-2 millimeters, this explains the inability of all methods, including ours, to retrieve edges in that area. However, looking at the noisy image, we can see that the image is extremely noisy in that part of the image that no edge is apparent, specifically the dark depth chart.



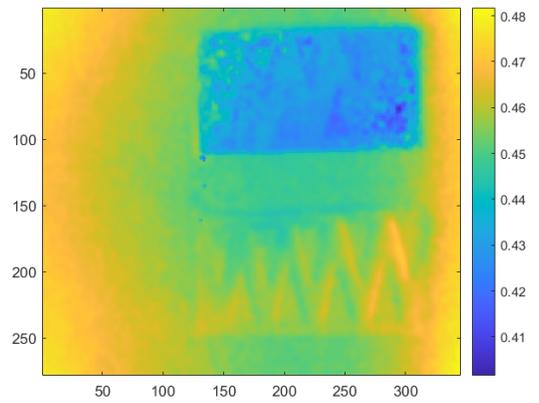
(a)



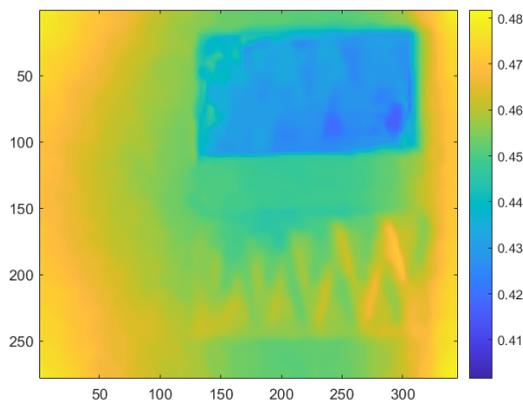
(b)



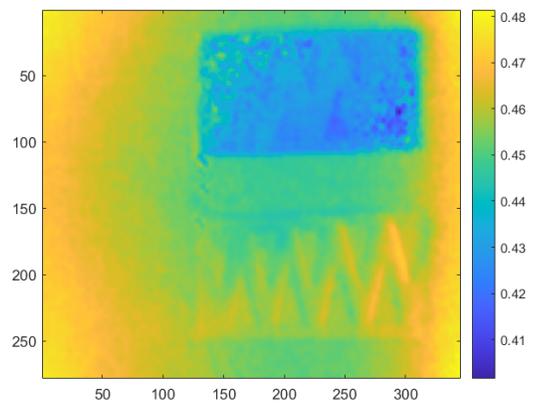
(c)



(d)



(e)



(f)

Figure 6.7: Denoising results. (a) Ground Truth image. (b) Noisy image. (c) Total Variation denoised result. (d) Joint Bilateral denoised result. (e) BM3D result. (f) Proposed denoised result.

CHAPTER VII

CONCLUSION

In this thesis, we presented a novel wavelet-based depth image denoising method based on Bayesian estimation for depth images acquired by Time of Flight cameras. We were able to reconstruct depth information from noisy raw data for which we also proposed a new statistical model and we were able to validate it both in the spatial domain and in the wavelet domain. We explicitly derived the wavelet likelihood function of the ToF sensor data and used a Gaussian's mixture prior. We developed a maximum a posteriori (MAP) error estimation of the wavelet variables using a mixture of priors in an effort to preserve image depth detail. Simulation experiments confirm that the proposed wavelet-based denoising is competitive or better than state-of-the-art methods in depth information retrieval. When denoising performance is measured by the accuracy of true depth information reconstruction, the result in terms of MSE show that our method is superior. Visual evaluation of the denoising results shows that although no state-of-the-art method, as well as ours, is perfect, our method results were the most consistent across various depth image content, including edges, reflectivity and depth range.

BIBLIOGRAPHY

- [1] A. Seitel, T. R. dos Santos, S. Mersmann, J. Penne, A. Groch, K. Yung, R. Tetzlaff, H.-P. Meinzer, and L. Maier-Hein, “Adaptive bilateral filter for image denoising and its application to in-vitro Time-of-Flight data,” in *Medical Imaging 2011: Visualization, Image-Guided Procedures, and Modeling*, ser. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, K. H. Wong and I. Holmes, David R., Eds., vol. 7964, Mar. 2011, p. 796423.
- [2] S.-H. Kim and M.-H. Kim, “Enhancement on time-of-flight camera images,” *HCI* : , pp. 708–711, 2008.
- [3] S. Lee, K. Bae, K.-m. Kyung, and T.-C. Kim, “Adaptive switching filter for noise removal in highly corrupted depth maps from time-of-flight image sensors,” in *Three-Dimensional Image Processing (3DIP) and Applications II*, vol. 8290. International Society for Optics and Photonics, 2012, p. 829009.
- [4] F. Lenzen, H. Schäfer, and C. Garbe, “Denoising time-of-flight data with adaptive total variation,” in *Proceedings of the 7th International Conference on Advances in Visual Computing - Volume Part I*, ser. ISVC’11. Berlin, Heidelberg: Springer-Verlag, 2011, p. 337–346.
- [5] E. Eisemann and F. Durand, “Flash photography enhancement via intrinsic relighting,” *ACM Trans. Graph.*, vol. 23, no. 3, p. 673–678, aug 2004. [Online]. Available: <https://doi.org/10.1145/1015706.1015778>
- [6] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama, “Digital photography with flash and no-flash image pairs,” *ACM Trans. Graph.*, vol. 23, no. 3, p. 664–672, aug 2004. [Online]. Available: <https://doi.org/10.1145/1015706.1015777>
- [7] H. Schoner, B. Moser, A. A. Dorrington, A. D. Payne, M. J. Cree, B. Heise, and F. Bauer, “A clustering based denoising technique for range images of time of flight cameras,” in *2008 International Conference on Computational Intelligence for Modelling Control Automation*, 2008, pp. 999–1004.
- [8] M. Frank, M. Plaue, and F. Hamprecht, “Denoising of continuous-wave time-of-flight depth images using confidence measures,” *Optical Engineering - OPT ENG*, vol. 48, 07 2009.
- [9] M. Georgiev, A. Gotchev, and M. Hannuksela, “Real-time denoising of tof measurements by spatio-temporal non-local mean filtering,” 07 2013, pp. 1–6.
- [10] Y. Kim, B. Kang, H. Lim, O. Choi, K. Lee, J. Kim, and C. Y. Kim, “Parametric model-based noise reduction for tof depth sensors,” *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 8290, pp. 9–, 02 2012.

- [11] B. Moser, F. Bauer, P. Elbau, B. Heise, and H. Schöner, “Denoising techniques for raw 3d data of tof cameras based on clustering and wavelets,” *Proc SPIE*, vol. 6805, 02 2008.
- [12] T. Edeler, K. Ohliger, S. Hussmann, and A. Mertins, “Time-of-flight depth image denoising using prior noise information,” in *IEEE 10th INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING PROCEEDINGS*, 2010, pp. 119–122.
- [13] W. Cheng and K. Hirakawa, “Towards optimal denoising of image contrast,” *IEEE Transactions on Image Processing*, vol. 27, no. 7, pp. 3446–3458, jul 2018.
- [14] H. Schöner, F. Bauer, A. Dorrington, B. Heise, V. Wieser, A. Payne, M. Cree, and B. Moser, “Image processing for three-dimensional scans generated by time-of-flight range cameras,” *Journal of Electronic Imaging*, vol. 21, p. 023012, 04 2012.
- [15] B. Huhle, T. Schairer, P. Jenke, and W. Strasser, “Robust non-local denoising of colored depth data,” in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2008, pp. 1–7.
- [16] J. Park, H. Kim, Y.-W. Tai, M. S. Brown, and I. Kweon, “High quality depth map upsampling for 3d-tof cameras,” in *2011 International Conference on Computer Vision*, 2011, pp. 1623–1630.
- [17] R. Nair, F. Lenzen, S. Meister, H. Schäfer, C. Garbe, and D. Kondermann, “High accuracy tof and stereo sensor fusion at interactive rates,” vol. 7584, 10 2012, pp. 1–11.
- [18] T. Edeler, K. Ohliger, S. Hussmann, and A. Mertins, “Super resolution of time-of-flight depth images under consideration of spatially varying noise variance,” in *2009 16th IEEE International Conference on Image Processing (ICIP)*, 2009, pp. 1185–1188.
- [19] D. Yeo, E. ul haq, J. Kim, M. W. Baig, and H. Shin, “Adaptive bilateral filtering for noise removal in depth upsampling,” in *2010 International SoC Design Conference*, 2010, pp. 36–39.
- [20] O. Aodha, N. Campbell, A. Nair, and G. Brostow, “Patch based synthesis for single depth image super-resolution,” 10 2012, pp. 71–84.
- [21] Q. Bolsee and A. Munteanu, “Cnn-based denoising of time-of-flight depth images,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, 2018, pp. 510–514.
- [22] S. Su, F. Heide, G. Wetzstein, and W. Heidrich, “Deep end-to-end time-of-flight imaging,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6383–6392.
- [23] R. Lange, “3d time-of-flight distance measurement with custom solid-state image sensors in cmos/ccd-technology,” Ph.D. dissertation, Universität Siegen, 2000. [Online]. Available: <https://dspace.ub.uni-siegen.de/handle/ubsi/178>

- [24] M. Hansard, S. Lee, O. Choi, and R. Horaud, *Time-of-Flight Cameras: Principles, Methods and Applications*. Springer Publishing Company, Incorporated, 2012.
- [25] S. W. Hasinoff, “Photon, poisson noise,” in *Computer Vision*. Springer US, 2014, pp. 608–610.
- [26] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, “Image denoising by sparse 3-d transform-domain collaborative filtering,” *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, aug 2007.
- [27] D. Yeo, E. Ul Haq, J. Kim, M. W. Baig, and H. Shin, “Adaptive bilateral filtering for noise removal in depth upsampling,” 11 2010.
- [28] M. Georgiev, A. Gotchev, and M. Hannuksela, “Real-time denoising of tof measurements by spatio-temporal non-local mean filtering,” 07 2013, pp. 1–6.