Diploma thesis

Resolution Enhancement For 3D Time-of-Flight Cameras

Sebastian Schuon

March 25, 2010

Lehrstuhl für Datenverarbeitung
Technische Universität München

Sebastian Schuon. Resolution Enhancement For 3D Time-of-Flight Cameras. Diploma thesis, Technische Universität München, 2010.

Supervised by Prof. Dr.-Ing. K. Diepold and Dr. Christian Theobalt; submitted on March 25, 2010 to the Department of Electrical Engineering and Information Technology of the Technische Universität München.

© Sebastian Schuon

Abstract

This thesis investigates into a new method for creating 3D models from real world objects. First an introduction into various methods to capture 3D data is presented. The focus of this work are Time-of-Flight (ToF) sensors to capture 3D data. Therefore the principle of operation is described along with the theoretical concepts and related work with these cameras. The major obstacle towards 3D scanning with ToF cameras is the recording solution. This thesis tries to solve this issue by applying the concept of multi-frame superresolution towards depth data. For completeness other approaches to increase the resolution of ToF sensors from literature are also reviewed.

The superresolution principle is to capture multiple scans from the same (static) scene while slightly translating the camera in-between shots. Therefore more information about the scene is gathered, that can be used to compute a higher resolution scan. Two algorithms for superresolution with ToF systems have been developed within this thesis: the first applies image-based superresolution towards depth data. Here we could verify the enhancement of resolution, along with a reduction in noise. The second algorithm, LidarBoost, has been developed entirely towards depth data. This resulted in yet another quality improvement, which we show both quantitatively and qualitatively. For both algorithms we show results from synthetic and real data sets. The thesis concludes with an outlook on how LidarBoost can be used to capture 3D models with Time-of-Flight cameras.

Acknowledgment

First of all I would like to acknowledge the support of both my supervisors: Christian Theobalt and Klaus Diepold. The idea for this thesis was born when I approached Christian with a vague concept of superresolution. Back then my idea was to record a higher quality image of the Golden Gate bridge by capturing multiple shoots with the comparably bad iPhone camera. It was Christian who explained the principle of superresolution to me and encouraged me to apply it towards his new Time-of-Flight cameras. He was a superb advisor, both during my time at Stanford as well as in Germany and working with him was a pleasure throughout. It is Klaus Diepold who in large parts responsible for me to make it to Stanford where the idea for this research was born. He continued to be a great advisor, both with scientific questions as well as questions of broader scope. I am very thankful that I had the chance to work in his group.

I also want to acknowledge the support of everyone in the *3D Video and Vision-based Graphics* research group, especially Young-Min Kim, Derek Chan, Carsten Stoll and Yan Cui. Furthermore I want to thank James Davis and Sebastian Thrun for their collaboration, from which many fruitful ideas developed. A further thanks goes to everyone at Stylight for allowing me to finish this thesis. Finally I would like to thank my friends and family without whom this work and most other things I accomplish would have never been possible.

Contents

1 Introduction

For a long time, humans were interested in 3D information, being it architects that wanted to characterize a building exactly or scientists what wanted the most accurate representation of an object. Before computers were known they were limited to taking photos (perspective re-productions of the 3D world) or could take only sparse measurements.

Also in the world of movies the third dimension has been of great interest. While 3D cinema has been around for nearly hundred years (with different levels of technical sophistication) the dawn of rendered 3D movies such as Toy Story dramatically increased the need of accurate 3D representations of real-world objects. Such models are the fundamentals for realistic movies that are created artificially within the computer.

This thesis reviews a number of techniques which can be used to capture 3D data and 3D models specifically. As the majority of these are rather expensive and/or complex to operate we then focus on a relatively novel and soon-to-be-cheap device, a 3D Time-Of-Flight (ToF) camera. These devices are the size of a traditional webcam, but record 3D information instead of color images. Unfortunately the accuracy with which these cameras can represent the scene in front of them is rather low. The main contribution of this thesis is to introduce a method that significantly improves the data quality of ToF cameras. We demonstrate for the first time that the super-resolution principale (super-sampling the scene) can be used for depth data and develop a novel algorithm based on this to boost resolution. Finally using the presented method we can create models of sufficient quality for 3D movies. The presented framework is both easy to operate and financially economic.

Apart from the movie industry many other fields will profit from low priced and accurate 3D sensing. Recently novel game interfaces have been demonstrated that use 3D data as input to control a game. The potential in new games that could be developed with such interfaces are peaked on with the Nintendo Wii. Here the user has to hold a controller in his hand, that records the 3D movement of itself while being moved. Even though this 3D information is very unreliable a whole new classes of games have been developed. The hope is that if full-body motion is tracked instead of sparse movements even more exciting games can be designed. Examples include a full-body tracking for dancing games [4].

Another area that benefits strongly from high quality 3D data is robotics. In order to navigate an environment a system needs information about its surroundings and its current position. Previous techniques to gather this information could either deliver only sparse

information (such as laser scanners that only record a plane) or were prone to errors (i.e. stereo methods that face problems in environments with little texture). Here Time-of-Flight cameras can provide substantial better data to the robotic systems and thus help to create smarter machines.

This work is structured as follows: in Chapter 2 the basic principles of Time-of-Flight sensors are described along with respective error sources. Furthermore related work on improving ToF sensors as well as image sensors is given in Chapter 3. We investigate the use of algorithms designed to improve resolution for color images and their application to 3D data in Chapter 4. Also we describe the setup used to record the scenes. The novel algorithm specifically designed for 3D data and that improves their resolution and accuracy is presented in Chapter 5. We conclude with a discussion and possible directions of future work in Chapter 6.

The research described in this thesis lead to the following publications

- *High-quality Scanning using Time-Of-Flight Depth Superresolution* by Schuon, Sebastian and Theobalt, Christian and Davis, James and Thrun, Sebastian at CVPR Workshop on Time-of-Flight Computer Vision 2008 [43]
- *LidarBoost: Depth Superresolution for ToF 3D Shape Scanning* by Schuon, Sebastian and Theobalt, Christian and Davis, James and Thrun, Sebastian at CVPR 2009 [44]
- *3D Shape Scanning with a Time-of-Flight Camera* by Cui, Yan and Schuon, Sebastian and Chan, Derek and Thrun, Sebastian and Theobalt, Christian at CVPR 2010.

2 Background

To acquire 3D data, there are a plentiful of technologies around: from stereo-imaging technologies to structure-from-motion to laser scanning (see [28] for an overview).

Stereo imaging has been around for a long time. A good introduction is given in [13] and [41] provides a benchmark of state-of-the-art algorithms. In 3D-from-stereo two or more cameras record the same scene. The relation to each other, i.e. a mathematical model that transforms one of their viewpoints into the other needs to be known (this is achieved during camera calibration). Then corresponding points are found in their images, a displacement of those is computed and by triangulation the depth can be computed. This method faces challenges, if the objects in the scene have no or little texture, hence corresponding points cannot be computed. In principle this technique can produce high resolution depth data (when using cameras with a high resolution), but in practice this is hard to achieve due to hard correspondence problems and noise. Furthermore these devices need quite a large housing, due to the displacement of the two cameras. Up until recently, the computation of a 3D scene required a significant amount of time, but now is available in real time. For a well known manufacturer of such devices we want to point out PointGrey [16].

Another important technique that is based of images is structure-from-motion. Here 3D information is obtained by comparing images that have been shot at different times. First the motion of the object in question is computed. Then features on the surface of the object are identified and their displacement computed. By the amount of displacement of the features the distance to the camera can be computed (remember, the closer an object is the further it appears to be displaced in to subsequent images). While this technique requires only one camera, the camera or the object has to be moved to infer computed 3D information. Furthermore the scene has to be static and it shares the problem of finding suitable features with stereo approaches. For both methods there have been approaches to overcome this issue, such as projecting structured light on to the scene to create good features.

Laser scanning uses a laser beam that is reflected on the object's surface. By measuring the time it travels the distance can be computed (a concept we will go into detail later on). Using precise techniques to measure time high resolution depth data can be captured at low computational costs. The disadvantage of this principle is that the laser beam needs to be moved through the scene to capture it. This is normally done using stepper motors and

a rotating beam. Overall a laser scanning system has a number of mechanical parts (that are more likely to fail) as well as limited temporal resolution. In [29] a good introduction to laser scanners is given.

Here we want to focus on a recent technology that overcomes most of the disadvantages of its successors mentioned before: Time-Of-Flight (ToF) cameras. The following chapter presents the theory of operation. We will explain in detail its own limitations and address how to overcome the most important one, the limit in spatial resolution. Here we introduce methods that use additional data such as color images and lay out the basics for our very own contribution, resolution enhancement based on superresolution.

2.1 3D Time-of-Flight Cameras

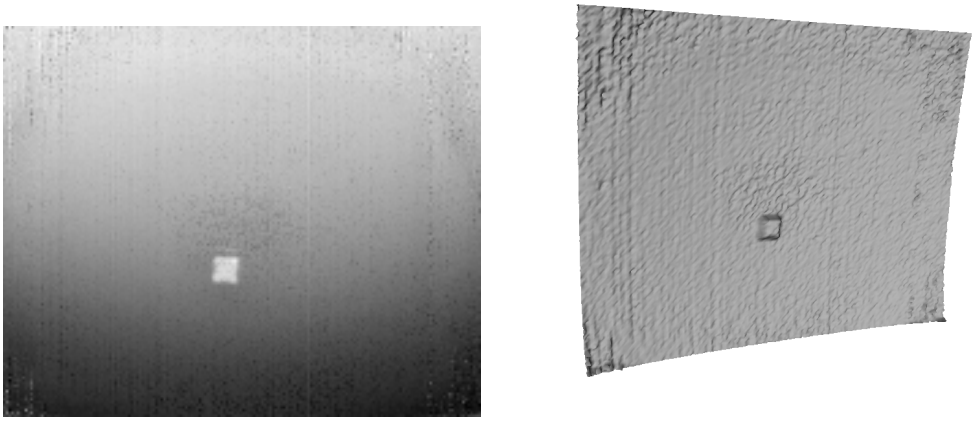
The idea of a Time-Of-Flight (ToF) camera has first been described in [27]. Here the idea of a laser scanner has been taken to the next level, where a light beam travels from the camera to the scene, is reflected and then returns to the camera. The time is taken and used to compute the distance. By illuminating the whole scene and capturing using a modified CMOS / CCD camera the whole scene can be sampled at a time. Furthermore the device can be integrated into a small housing and possibly be manufactured at quite low cost. This is a clear improvement over techniques such as stereo or laser scanning, but also introduces its own problems, which we address later in this chapter. Before that we introduce two different principles of operations and elaborate on the measuring space of the camera.

2.1.1 Principles of Operation

The principle of Time-of-Flight cameras, is to measure the duration δt it takes light (preferably at wavelengths not visible for the human observer, here mostly infrared) to travel from the camera to the scene, bounce back at the surface of an object and return to the camera. Given the speed of light c, one can easily determine the surface's distance d form the camera

$$d = \frac{1}{c \cdot \delta t} \tag{2.1}$$

All rays originate at the center of projection of the camera. Hence the distance is measured along the ray of light. This has the at first glance somewhat unusual consequence that we measure in a rayspace. Hence planar objects in the scene have different distance values on their surface. This seems reasonable, but in a common representation of depth



(a) Depth Map **(b)** Rendered 3D geometry

Figure 2.1: Depthmap (contrast enhanced) and 3D geometry
Recording of a flat wall using a SR-3000 and averaged over 100 frames

data, this often irritates the viewer. Depth maps are a form of visualization of 3D data, where the x and y coordinates resemble the pixel coordinate and depth is represented in the form of brightness at the pixel. In Figure 2.1a we have shown such a depth map with darker pixels representing further away points. The scene is a flat wall with a knob in the middle as it can be seen in the rendered geometry (Figure 2.1b). Intuitively one would imagine the depth map all to be the same color since it is contains a flat object. But since the camera records in rayspace the points on the surface of the wall are at different distances to the camera.

Refer to Figure 2.2 where we tried to depict the situation: Intuitively objects A and B are at the same depth along with all other objects on line e_1. This is true if depth would resemble to the z-coordinate in a global coordinate system, hence a orthographic projection. But the camera still resembles to the pinhole camera model, where a perspective projection takes place and all rays converge in the center of projection. Therefore objects B and C are at the same distance (to the camera). Hence they have the same brightness in the depth map and so have all other points on the circle e_2. Because of this effect it is best to consider the rendered geometry to judge a scene, since its representation is more intuitive to the human observer. This advice holds also true for situations, where subtle details are not visible in the depth map due to slight brightness variations but are clearly in the rendering. This is due to the effect that small changes in grey value are hard to spot for the human eye and the rendering can be much better inspected (i.e. the viewer's position changed).

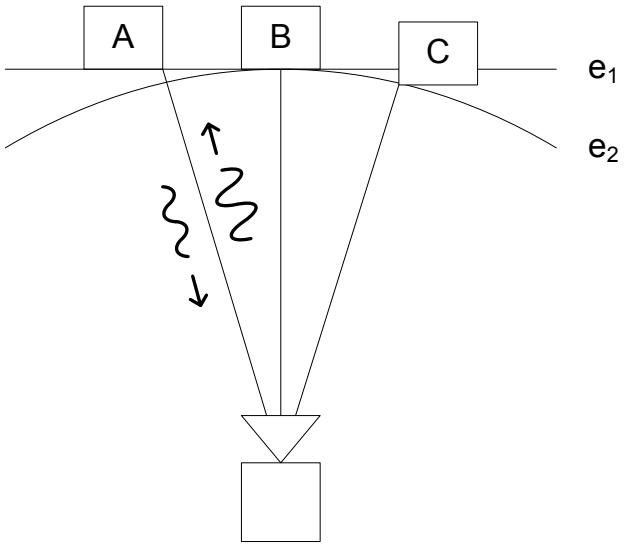


Figure 2.2: Rayspace - equidistant objects are on a circle

For reconstructing metric 3D data, one has to unproject ray space measurements according to:

$$(X,Y,Z) = D \cdot \bar{V}\,. \tag{2.2}$$

Here, $V = \frac{(x,y,f)}{\sqrt{(x^2+y^2+f^2)}}$ is the measurement ray direction (viewing vector) from the camera's center of projection through the sensor pixel at location (x,y) relative to the sensor center, and f is the camera's focal length. For metric reconstruction, x and y have to be specified in terms of metric pixel size μ, i.e. $x = i_x \cdot \mu$ with i_x being the pixel index in x-direction relative to the pixel center. Further on, $D = P_d + P_w \frac{255-g}{255}$ is the depth along the measurement ray which is computed from the distance to the frontal clipping plane P_d, the depth of the 3D view frustum P_w, and the gray value g in the depth image supposing it is quantized to eight bit. For some cameras the frontal clipping plane starts at $P_d = 0$, which depends on the very implementation of the camera. Since the quantization limits the system's resolution, it is not recommended to store measurements in depth maps but to use some sort of raw format most vendors offer.

Considering again the measurement principle for normal operating conditions, the time in flight can be as short as some ten femto seconds. Therefore innovative ways to measure the duration are needed. So far researchers have attempted two different approaches which both found their ways into shipped products. The first is based on a very fast shutter, the second being the more common one using phase correlation. We explain both in the following sections.

2.1.2 Shutter Based Operation

The Z-cam [14, 50, 12] by 3DV is the only ToF camera known to operate on a shutter based principle. The camera emits a *light wall* that is reflected by the scene. From the shape of the light wall the depth can be reconstructed. The light wall is a squared light pulse [50] at a wavelength of $\lambda = 800nm$ [14] [1]. The operation principle is outlined in Figure 2.3. Here the upper pulse is traveling from the camera to the scene and the lower two are returning pulses. The latter two pulses originate from the same objects.

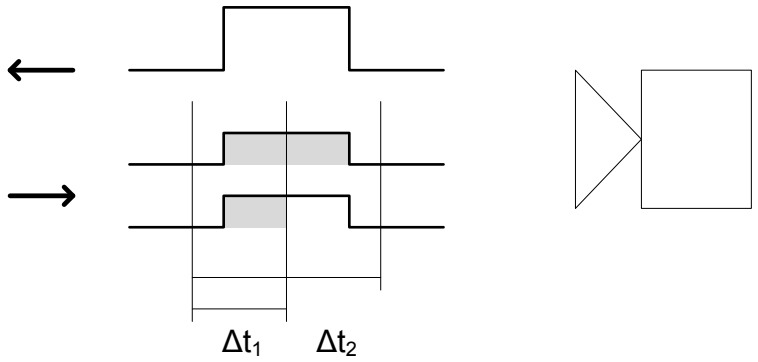


Figure 2.3: Shutter Based Time-of-Flight Measurements

Now the camera has a very fast shutter in front of the sensor. Two measurements are performed one with the shutter closing at δt_1 and one with the shutter closing after the whole pulse has been received (δt_2). From the first measurement depth can be inferred up to reflectivity of the objects scanned. This is because if the object is closer to the camera, the pulse travels shorter and more light is received while the shutter is still open. Hence the amount if light is indirect proportional to the distance of the object. This technique is called *early close* while in theory the sensor could also operate on *late open* then the amount of light would be directly proportional to depth. The second measurement is performed to identify the objects reflectivity by measuring the total amount of reflected light. Then the distance d is calculated via

$$d = \frac{I_{\text{early close}}}{I_{\text{total}}} \,. \tag{2.3}$$

With the Z-Cam the light source is a ring of LEDs around the camera that illuminate the scene (clearly visible in Figure 2.4). The Z-cam can measure full frame depth at video rate and at a resolution of 320×240 pixels. Since the sensor is simply a normal camera chip the depth resolution is limited by the dynamic range of the sensor. The Z-Cam allows

[1] In their original paper they specify "800 micron" but claim the light to be infrared. Hence we believe this is a typo and should read nanometers

to open the shutter late and close it early. Thus it can limit its input range to a certain frustum which is then quantized by the dynamic range (here eight bit). Hence in theory it could capture at a very high z-resolution but shutter speed and noise severely limit this capability. Obviously the light source needs to increase power once the shutter intervals become shorter to keep noise low. A rendered scene captured by a Z-Cam can be seen in Figure 2.5.

In contrast to competing ToF cameras, the Z-cam features a normal video camera of 640×480 pixels in the same device which enables recording of texture-mapped geometry (see Figure 2.5a for an example). Unfortunately, video and depth are not recorded through the same optics and the homographic registration data provided by the manufacturer is off by several pixels. For comparison experiments we therefore resort to our own external color camera (Chapter 3.2).

Although the Z-cam delivers scene geometry at unprecedented speed and largely independently of scene texture, the quality of recovered 3D data in a single frame is not sufficient for high-quality 3D scanning, as shown in Figure 2.5b compared with data acquired by a laser scan (Figure 2.5c). The laser scanner used has been relative simple one, that still introduces more noise than most state-of-the-art laser scanners. Even though averaging helps to decrease the noise somewhat (see Figure 2.5d), the ToF camera is still outperformed by a cheap laser scanner. This limits the use to static scenes and even than is not nearly as good as a laser scan. The depth measurements are strongly contaminated by random noise which can, at $1\ m$ average scene distance, vary by up to $5\ cm$. Depth measurements also become more unreliable towards the boundary of the field of view, since there, optical aberrations like vignetting play a stronger role, and the PSNR of the returned signal naturally decreases. Not only seems there to be a systematic bias (see Figure 2.6c): but Figure 2.6d shows variance in random noise increases strongly towards the field-of-view boundary. Noise variance is also much higher at mixed pixels that integrate over depth discontinuities in the scene. Fortunately, pixels with high measurement uncertainty typically exhibit low measurement intensity and therefore the camera's raw intensity data can be interpreted as a confidence map. Experimentally, we could verify that the depth readings at a single pixel location over time follow a slightly heavy-tailed distribution.

In addition to random noise, the camera is likely to exhibit a systematic measurement bias that may depend on reflectance, angle of incidence, and environment factors like temperature and lighting. Since focus in literature has shifted to ToF cameras based on correlation, no studies are known that research these errors in detail. In [24] it is speculated that most of the research on correlation based ToF cameras also applies to shutter based one.

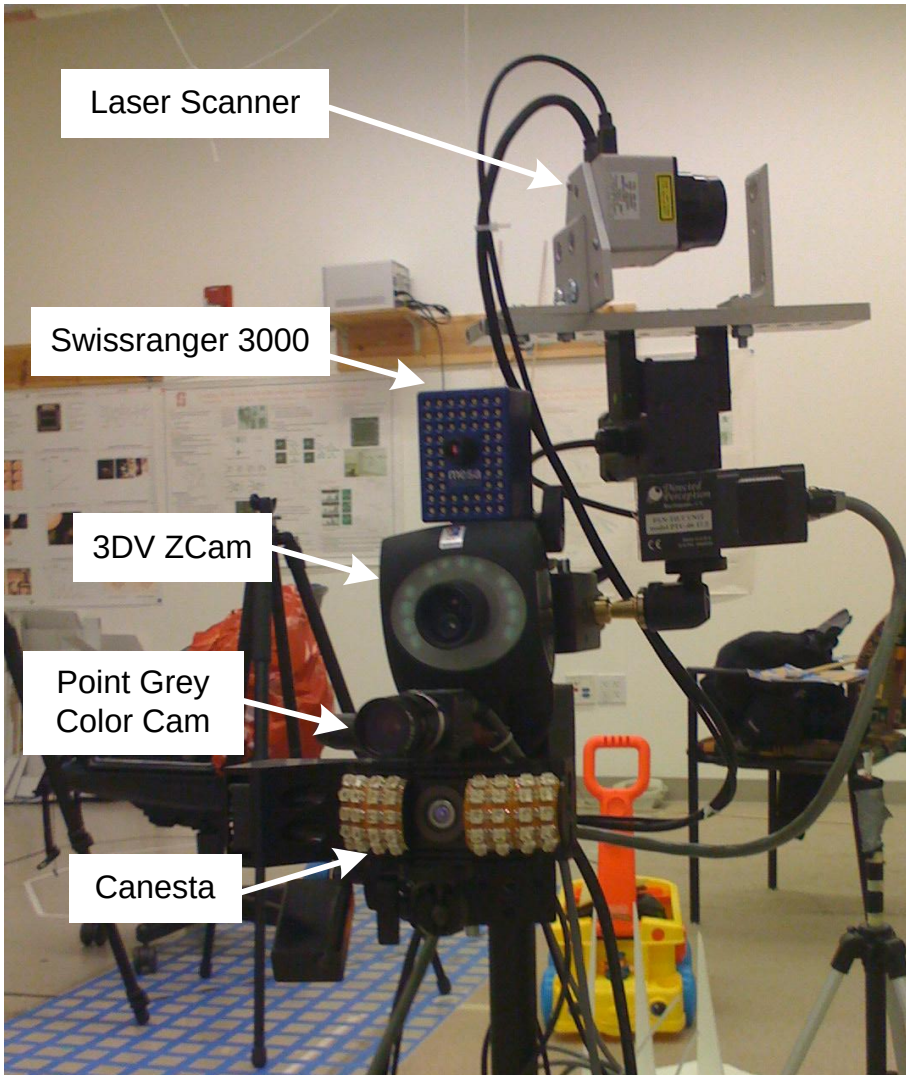
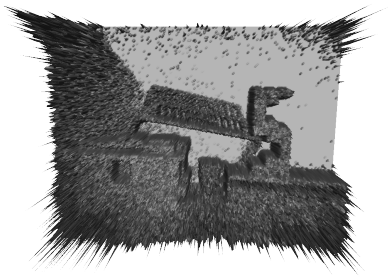
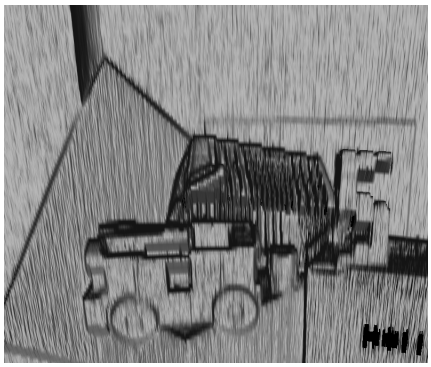


Figure 2.4: Various Time-of-Flight cameras on the Stanford ToF Camera Array

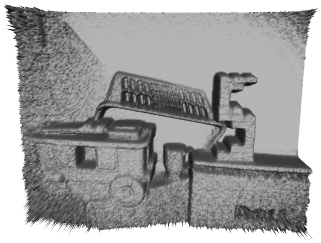
(a) Color Image



(b) Single 3D Recording

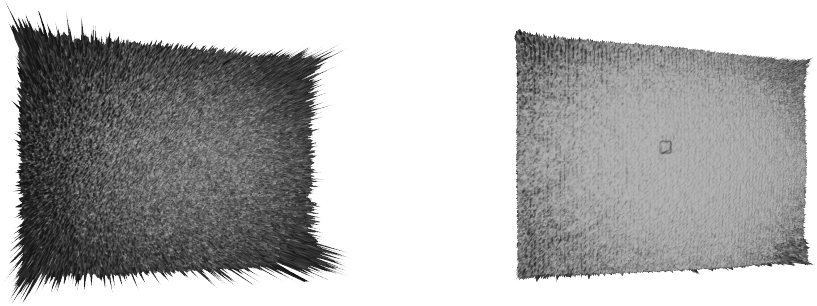


(c) Laser Scan



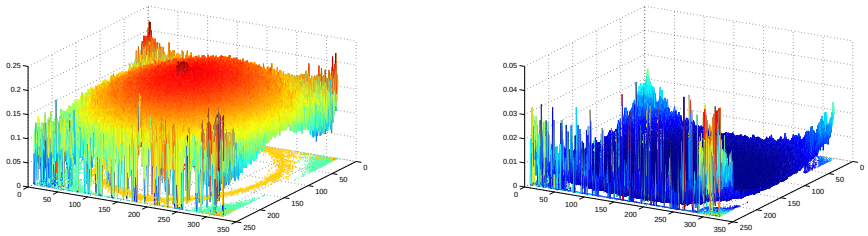
(d) Averaged 3D Recording

Figure 2.5: ZCam - Single and Averaged ($n = 50$) recordings in comparison to a laser scanned model



(a) Single 3D Recording

(b) Averaged 3D Recording



(c) Mean

(d) Variance

Figure 2.6: ZCam - Single and Averaged ($n = 100$) recordings and the resulting mean and variance plots

2.1.3 Correlation Based Operation

The most prominent approach for Time-of-Flight systems uses the phase shift that takes place when light gets reflected on an objects surface. Both the Swissranger cameras by MESA Imaging [35, 36] and the PMD cameras [49, 26, 40] build on this principle. These manufactures already do ship their cameras and they can be purchased, even though they are still way more pricy than initially promised. Canesta's older models used correlation as well [9]. Apparently they have developed a new type of sensor that is more similar to the shutter based approach [15].

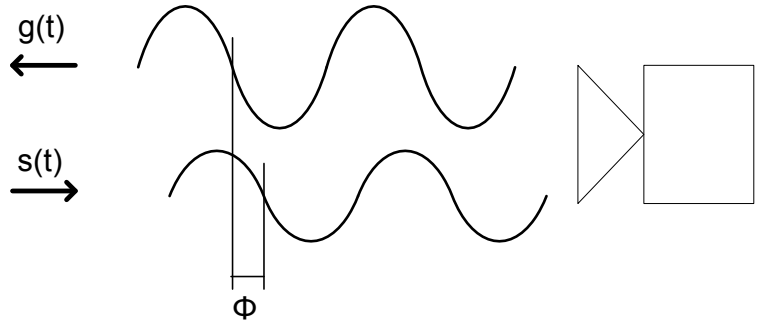


Figure 2.7: Correlation Based Time-of-Flight Measurements

Correlation based cameras have a light source that emits in theory a sinoidal waveform (see Figure 2.7). In practice the waveform is somewhat different since all sorts of physical effects do not allow for perfect waveform generation. The principles of the correlation approach have been described in [9, 26, 36] but we follow notation wise [24]: given an signal g that is emitted from a light source at the camera and a signal s that represents the reflected signal for the scene, the correlation reads

$$c(\tau) = s \otimes g = \lim_{T\to\infty} \int_{-T/2}^{T/2} s(t) \cdot g(t+\tau) dt \,. \tag{2.4}$$

Here τ is an internal phase shift that occurred during signal generation. Then again we assume sinusoidal signals modulated at frequency ω. We introduce a damping coefficient a for the incident signal (i.e. due to non perfect reflection) and a correlation bias b. With ϕ being the phase offset relating to distance the signals read:

$$g(t) = \cos(\omega t) s(t) = b + a\cos(\omega t + \phi) \,. \tag{2.5}$$

Substituting Equation 2.5 into Equation 2.4 and simplifying yields

$$c(\tau) = \frac{a}{2}\cos(\omega\tau + \phi) + b \,. \tag{2.6}$$

.

The correlation function $c(\tau)$ is then sampled to solve for ϕ. Four samples are the minimum to solve, more samples enhance the precision of the system but are expensive to realize in hardware. Therefore all known cameras sample only four times at $\tau = i \cdot \frac{\pi}{2}$: $A_i = c(i \cdot \frac{\pi}{2}), i = 0\ldots,3$. Then ϕ is readily computed as

$$\phi = \arctan(\frac{A_3 - A_1}{A_0 - A_2}) . \tag{2.7}$$

Once computed, substituting ϕ into

$$d = \frac{c}{4\pi\omega}\phi \tag{2.8}$$

determines the distance at the very pixel (with c being the speed of light).

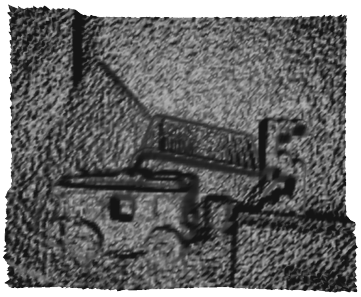
All cameras house the light source (see Figure 2.4 for an example of the Swissranger SR3000) and use infrared LEDs. The SR3000 for example uses light at $\lambda = 850$nm has approximately $1W$ illumination power. Most other components of these ToF cameras are the same as with color cameras such as the optical system. A drawback of the correlation approach as compared to the shutter based one is the limited resolution. Since custom CMOS chips are required (and they also achieve a rather low fill factor) the resolutions is limited to 176×144 (Swissranger SR3000), 204×204 (PMD's CamCube) or 64×64 (Canesta). The Z-Cam could use any CCD grayscale sensor with a possibly much larger resolution but is in fact limited by the SNR for larger resolutions. To compare both approaches Figure 2.8 has the same sample scene and Figure 2.9 shows the noise characteristic, here for a Swissranger SR3000.

2.1.4 Error Sources

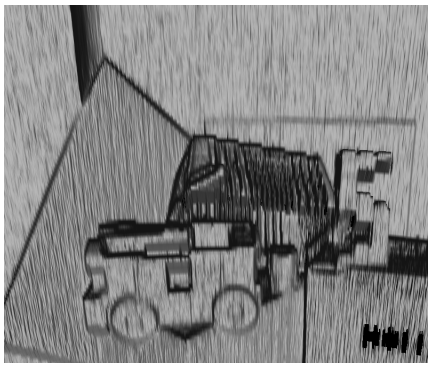
As with all technical systems Time-of-Flight cameras are not without flaws. Several sources for errors exist which degrade the quality of the recording. Some of them are unique to ToF cameras while others are known with other types of cameras. Here we will describe only the most common error sources and for completeness we refer the interested reader to extensive studies as found in [39, 26, 21, 31]. Lately even theoretical models of ToF sensors have been created that help to explain the errors [8, 42].

Low Resolution As previously mentioned ToF systems suffer from a rather small resolution. Even the Z-Cam has a resolution that is easily surpassed by webcams and mobile phones. This is due to the special design of ToF sensors that are relatively new compared to current CCD/CMOS technology. Hence less optimization has been applied and

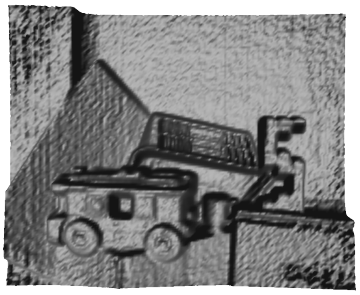
(a) Color Image (Point Grey)



(b) Single 3D Recording

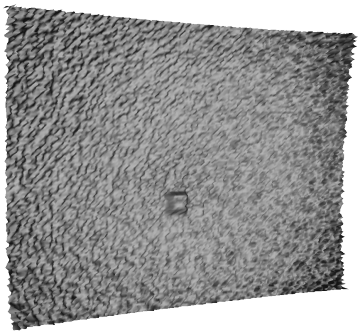


(c) Laser Scan

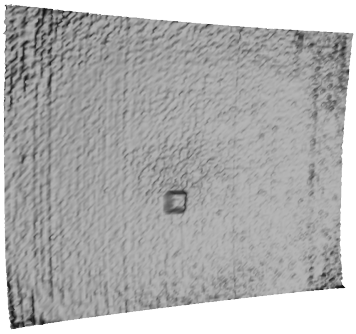


(d) Averaged 3D Recording

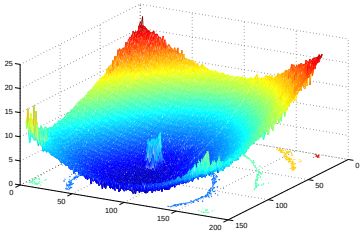
Figure 2.8: ZCam - Single and Averaged ($n = 50$) recordings in comparison to a laser scanned model



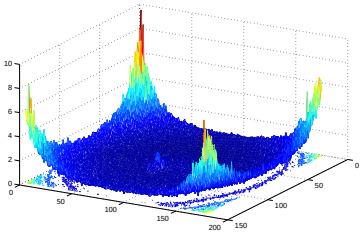
(a) Single 3D Recording



(b) Averaged 3D Recording



(c) Mean



(d) Variance

Figure 2.9: ZCam - Single and Averaged ($n = 100$) recordings and the resulting mean and variance plots

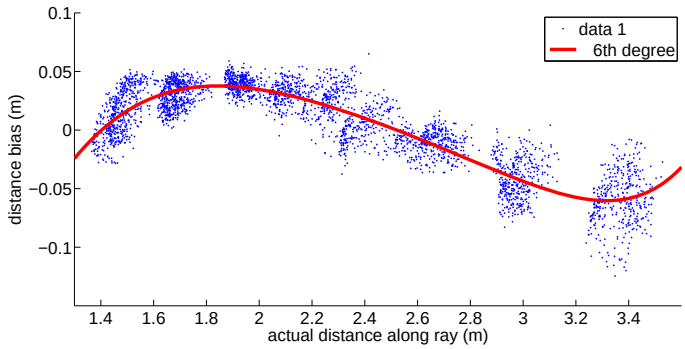


Figure 2.10: Systematic bias of a SR3000 ToF camera, Figure courtesy of Kim et al.

research. Furthermore the illumination sources are limited in their power to stay within regulations for eye safety. This in turn limits the amount of photons and relatively large pixels are required. Furthermore the low fill-factor prohibits higher resolutions. The fill factor is commonly defined as the ratio between active sensing area and the total area of a chip. With ToF cameras, especially correlation based ones; this is rather low since quite some additional electronics need to be placed on the chip.

General noise, noise not distributed equally In general the noise level tends not to be distributed equally among a recorded frame. In Figures 2.6c and 2.6d (ZCam) as well as Figures 2.9c and 2.9d (Swissranger) we have plotted the mean and variance of the noise. From our experiments we believe the noise is distributed in a radial fashion, i.e. increased with the distance from the center pixel. This could be explained well with pixels closer to the edges the beam has to travel further, when recording a flat objects (re-call ToF cameras record in the rayspace, Chapter 2.1.1). Due to the increased distance, less light is returned to the camera, hence noise increases. Also the illumination source is less powerful towards the edges, which in turn decreases the returned light towards the borders even more.

Systematic error All ToF cameras exhibit a systematic error that is dependent on the meseaured depth. Most research has been carried out for correlation based cameras, but we encountered similar effects also for the shutter based system. Most notably is that the error is not monotonic increasing with distance, but follows some sort of sinoidal pattern (see Figure 2.10). In [21] Kim et al. found a 6th degree polynomial as a good fit. This reduced the average error from approx. 5cm to nearly 1.4 cm (along with some other correction). To date the reason for that error remains unknown, but some argue it is due to imperfect waveform generation for the scene illumination.

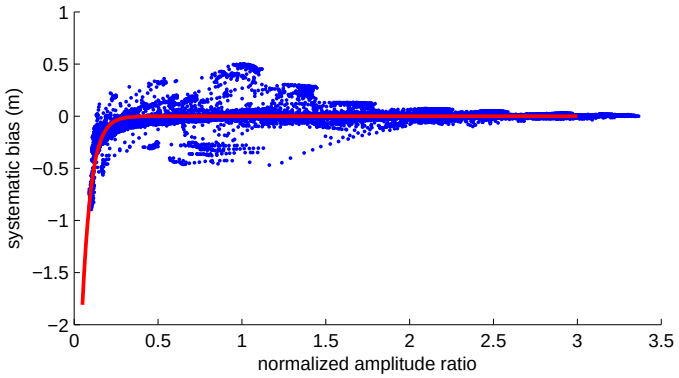


Figure 2.11: Noise is strongly related with the intensity seen at the camera, Figure courtesy of Kim et al.

Error on low reflectant objects As discussed previously if little light is returned to the camera noise increases. Another obvious reason for little light returning to the camera are objects that absorb most of the light, such as those which appear black to the human observer. Again Kim et al. studied this phenomenon in detail: they found that if the normalized intensity returned from an object is below a certain level the error increases heavily (see Figure 2.11 for details). We later make use of this fact and label measurements below that threshold as unreliable.

Measurement at borders (flying pixels) / Motion Blur When recording objects with distinct edges, one encounters points returned that represent a depth that is somewhere between the closer and the further away object. As these points have no obvious connection to their surrounding they are referred to as *flying pixels*. Their occurrence can be perfectly well explained when assuming ToF cameras operate with the same principal as color cameras. Imagine an edge between a white and a black region on an object. When recorded with an ordinary camera, pixels on the border on the edge will appear gray, as the camera integrates intensities over the area of the pixels. And here the intensities are partly black and party white. The same happens to Time-of-Flight cameras, where the camera would integrate over depth samples and return an average of the two depth levels when recording an edge. In the community there have been reports, where the flying pixels were outside of the expected range (i.e. the measured depth was larger or smaller respectively than the background or foreground). We could image such a phenomenon only with correlation based systems, since they are non-linear systems. Nevertheless during our experiments we never encountered such an error. On the contrary we performed measurements with the correlation-based Swissranger SR3000 to investigate the behavior along edges. When recording a plane that is slightly rotated with respect to the pixel grid,

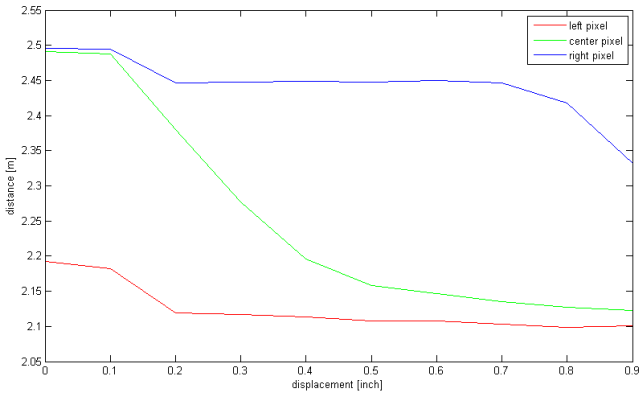


Figure 2.12: ToF cameras as a linear system towards depth discrepancies: the linear descent hints that we can treat ToF as linear systems

a linear system such as color cameras are, would show a smooth linear transition between the two levels. Now please refer to Figure 2.12 where the measured depth for the previous setup is plotted. The left pixels record an area that is completely on the near plain, where as the right pixels corresponds to the far plain. The center pixels record the mixed area, where flying pixels would be encountered. Since the plain has been rotated slightly the ratio between far and near plain constantly decreased. Clearly no spikes are included in the descent which would correspond to erroneous flying pixels. The mostly linear descent hints that we can treat a ToF cameras as a linear system, which is an important criteria for superresolution to be applicable.

Another source of error all cameras are prone to is motion blur. Due to the scene not being static, the depth changes within the integration period, which is even more severe due to multiple samples being involved in computing the correlation. We did encounter such blur also for Time-of-Flight cameras. But since superresolution at the current state requires static scenes anyways we did not investigate further. A simple measure is to keep integration times short (at the cost of higher noise obviously).

Smaller issues: Multiple Reflectances, Multiple Cameras, Outdoor usage / high contrasts / background light subtraction Multiple reflectances can also degrade the sensor's readings, as they can be interpreted as strong noise on the true signal. Similar false signals can be generated when two or more cameras are operating at the same time. Then light emitted by the first camera is recorded by the second. Simple approaches use cameras running at different frequencies (such as in [23]). More advanced techniques could include using advanced waveform patterns, such as gold codes. With these orthogonal codes multiple sensors could operate at the same frequency without interference.

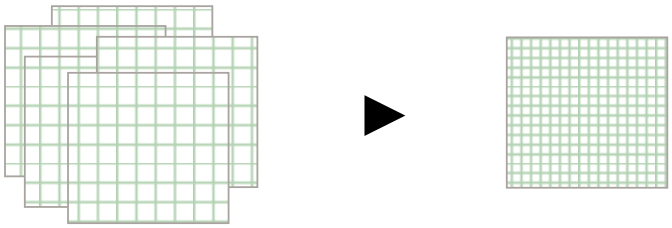


Figure 2.13: The principle of superresolution: recording the same scene multiple times and displacing the camera slightly between recordings. This allows for a reconstruction of a higher resolution image.

When operated outside of a controlled lab environment further challenges unfold: light hitting the sensor directly such as from the sun is a major problem, since they saturate the pixels quickly. Some manufactures now ship their cameras with background light subtraction support that helps in such situations. Still high contrasts, i.e. by highly reflective materials are major challenges for ToF cameras operated outdoors.

2.2 Superresolution For Images

The resolution enhancement for Time-of-Flight cameras that we developed will draw on a technique called *multi-frame superresolution* that has been known in other fields such as image processing. Here we provide a short introduction: the idea behind superresolution is, that the information that is available about a scene is increased, if multiple shots are taking from approximately the same viewpoint. The underlying assumption is, that the changes to the viewpoint are so little, that effects caused by the viewpoint shift, such as parallax effects, can be neglected. One might argue that in order to increase the resolution of an image, it could simply be upsampled. While this obviously increases the physical resolution, no information (such as fine detail, previously below the sensors resolution) that was not present in the original image, can be included in the upsampled one.

Image based superresolution targeted at standard color or intensity images has been well studied for many years [7][17][47]. Multiple low resolution images are aligned and then a high resolution image is estimated which explains the image stack. Interested readers will find a survey informative [2].

Some researchers have formulated a joint optimization of superresolution together with shape-from-X. Shape from photometric cues [18] as well as defocus [38] have both been explored.

The noise and data statistics of depth data exhibit effects which may not be found in normal color images, so it is not obvious that color based methods are applicable. Indeed, earlier work targeted at depth superresolution pursued an alternate strategy. Later we show that color methods *are* applicable in the depth domain, and that they can perform better than the specialized depth superresolution methods previously introduced.

2.3 Improving ToF sensors

The depth accuracy of Time-of-Flight sensors can be increased by a variety of methods, e.g. by accounting for ambient light [10], simulating the shape of the reflected signal [19], and performing time gated superresolution [30]. While these methods improve resolution in the depth direction, they all operate at the level of peak detection in the sensor itself and are not directly related to improving resolution in the X-Y plane as discussed in this work.

3 Related Work

Ever since cameras existed, people wanted to increase their resolution, i.e. obtain more information about the scene recorded. Camera chips are steadily increasing their capture resolution, but people wish for a higher pace. This is why researchers investigated in ideas beyond improving the manufacturing process to boost camera resolution. Nearly 25 years ago R. Tsai and T. Huang [48] introduced the idea of using many, slightly displaced recordings to reconstruct a higher resolution one. This follows a pattern well-known in signal processing, where sampling a signal at multiple sampling frequencies allows for a better reconstruction of the signal. We will use this key idea later to improve depth recordings. Since depth recordings have a comparably low resolution to current color cameras, researchers have investigated other methods to boost resolution. Here we will show smart upsampling techniques and methods that fuse color and depth information.

3.1 Filtering & Smart Upsampling

Since ToF cameras produce quite noisy data, a first approach was to filter the data to ensure the usable resolution is close to the specified resolution. For ToF cameras not only return depth data, but also an intensity image, Boehme et al. [1] pointed out that these two are related by the shading constraint, if the reflectance properties of the surfaces are known. They assumed a general reflectance model, here Lambertian reflectance, and used a probabilistic model for the image formation. Then they could compute the maximum a posteriori probability of the scene. Their results have significantly less noise that makes small features appear that previously were hidden by the noise, hence increasing the usable resolution.

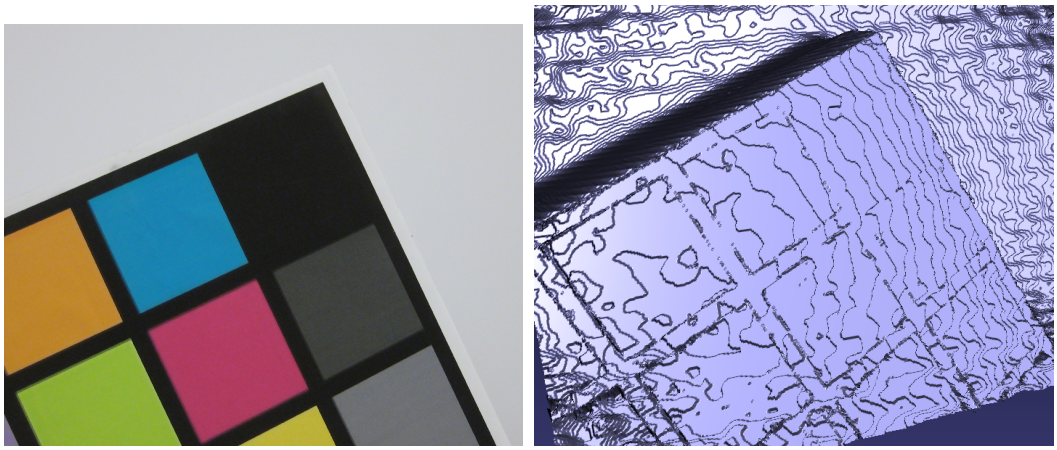
The intensity image also allows to perform a simple background / foreground separation, assuming the background is quite far away and thus returns only very little light. This fact was used by Lindner et al. [32] to perform smart upsampling. They try to estimate the edge directions and try to include that information in the upsampling process to have a depth map that has the same resolution as a color recording, allowing for easy fusion.

3.2 Depth And Color Fusion

Color cameras usually have a way higher resolution than ToF cameras, but high quality 3D information recovery using color cameras (i.e. using stero methods) has proven difficult. Therefore it seems logical to combine the strengths of color cameras and ToF cameras, in particular since edge discontinuities often correlate with color discontinuities. Hence one requires a ToF camera and a color camera closely placed to each other. Ideally they would share the same optical system, but this normally requires a beam splitter and hence is quite costly (the first ToF camera produced by 3DV featured such a system. The Z-Cam now used two different lens systems to keep manufacturing costs low). If two optical paths are used, the homography between the two has to be pre-calibrated or can be computed on-the-fly for some algorithms. Depth image superresolution then is accomplished by using a high resolution color image and upsampling the low resolution depth image. The regularizer ensures edge consistency between the color and depth image. The difference in the various methods is mostly in the formulation of the regularization term. The idea of combining a color image with a depth recording was first proposed by Diebel & Thrun [6]. Their formulation took the form of a MRF. This method proved the feasibility of the approach, but also was computationally expensive. With the dawn of bilateral filtering, Kopf et al. proposed joint bilateral upsampling in the image plane [25]. Another successful approach was by Yang et al. [51] to perform bilateral filtering on the cost volume. These methods can reproduce high frequency detail, however they assume that color is always correlated with depth. With textured objects this is often not the case, as it can be seen in Figure 3.1. Here the checker board on the right (Figure 3.1a) leads the fusion method (here a implementation of Diebel & Thrun's method) to create a checkerboard structure on the originally plain surface of the board.

Recent research has addressed the problem and is successful in diminish the texture copy effect [5] while at the same time operating in real-time. We will later propose a method, which is inherently robust against texture copying by relying on depth information only.

When using more than one camera, stereo methods can be used to compute depth information (for a overview of recent advances and benchmarks see the Middleburry Database [41]). One of the most problematic objects for stereo methods are un-textured objects. Here the algorithms fail to establish correspondences between the two camera images, which are required to compute depth. In [52] Zhu et al. combined a stereo vision system with a ToF camera to significantly improve the stereo reconstruction. Nevertheless this requires three cameras while we will propose a method that needs only one Time-of-Flight camera.



(a) Color recording in high resolution **(b)** False structure by joint bilateral upsampling

Figure 3.1: Texture Copying - Color-Depth fusion methods assume color discontinuities are correlated with depth edges. This can lead to an effect known as texture copying as seen on the checker boards surface

3.3 Oversampling

Here, the goal is to enhance the resolution by combining only depth recordings of a static scene that were taken from slightly displaced viewpoints. Kil et al. [20] were among the first to explore such an idea for laser triangulation scanners. They heavily oversample the scene by taking up to 100 scans from similar viewpoints to achieve four-times upsampled geometry. Since their data is so dense, and the random noise level of a laser scanner is significantly lower than that of a ToF camera, they can obtain good results by regular resampling from the aligned scan points with associated Gaussian location uncertainty. Reportedly, results may exhibit unnecessary blur and it is unlikely that this data fusion principle will work for highly noisy ToF data.

3.4 MRF based superresolution methods

Only recently researchers looked into performing superresolution on ToF camera data. Rajagopalan et al. [37] proposed a Markov-Random-Field based resolution enhancement method from a set of low-resolution depth recordings that formulates the upsampled 3D geometry as the most likely surface given several low resolution measurements. Their MRF uses a neighborhood system that enforces an edge-preserving smoothness prior between adjacent depth pixels. Their formulation of the problem bears two disadvantages:

first complex parameter selection and secondly the formulation of the prior renders the problem non-convex, and hence more sophisticated solvers are required.

4 Image Adapted Superresolution

It is our goal to obtain high-quality 3D measurements of a static scene despite the significant noise in the raw data (see Figure 4.1 for a visual motivation). By performing superresolution, we seek to increase X-Y measurement resolution and, at the same time, reduce the overall random noise level. We seek to apply the idea of superresolution known from color images do 3D data. To this end, several depth maps captured from minimally displaced viewpoints would be aligned, and subsequently combined into a higher resolution depth image. From this superresolved depth image, we can eventually reconstruct superresolved 3D geometry.

In this chapter we will introduce the necessary recording setup for superresolution (Chapter 4.1) and show how superresolution methods previously used for color images can be used towards depth data (Chapter 4.2). We present the results in Chapter 4.4, where we also compare the new ethod against upsampling methods that fuse color and depth information.

4.1 Recording Setup

Key for multi-frame superresolution to work with Time-of-Flight sensors is that a ToF camera has some similar properties to a regular optical camera. Most importantly, it must

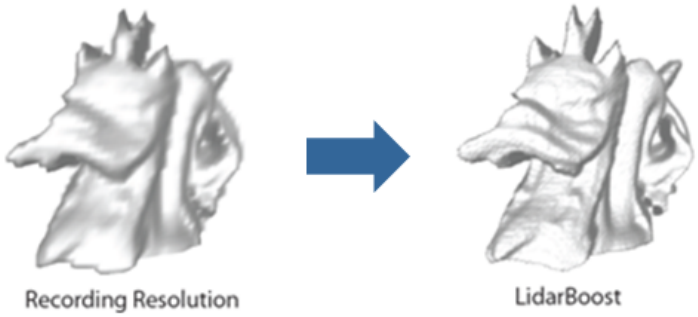


Figure 4.1: Superresolution for Time-of-Flight cameras transforms multiple raw recordings (such as on the left) into higher detail, less noisy representations (as on the right)

Figure 4.2: The suggested processing pipeline: first raw recordings are made which the superresolution process then fuses and a more detailed 3D geometry is the result.

return at a pixel position the average depth of the area covered by the pixel. Speaking mathematically, the pixel value must be the integral of all depth over the pixels coverage area. We have experimentally investigated and the previous section argues that this property indeed holds true.

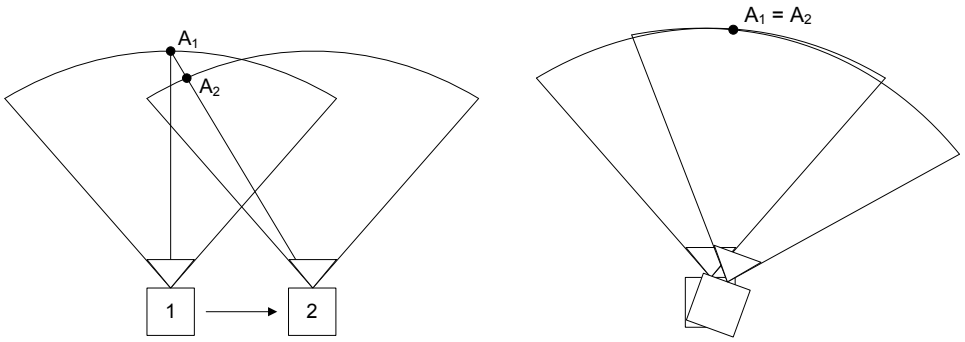
Our processing pipeline (see Figure 4.2) starts by recording the raw depth maps, performing superresolution on them and converting the result into 3D geometry. Below we will detail the recording process; this and the next chapters present two algorithms that perform superresolution, along with rendered 3D geometry.

In our measurement setup, the depth camera is located between $50\ cm$ and $150\ cm$ away from the scene. Typically, we capture $N = 15$ images by slightly translating the camera orthogonally to the viewing direction (see Figure 4.3a). Please note that the alignment of images captured by the above procedure effectively leads to the creation of a multiperspective image in which parallax effects may play a role. One way to overcome these effects would be to slightly rotate the camera around the center of projection rather than translate it. Compare with Figure 4.3b, this indeed diminishes this effect, since we are recording in the so-called rayspace. However, with as small displacements as we apply it we could experimentally not verify an increase in reconstruction quality if the camera is rotated, so the parallax effect cen be ignored. Therefore, we always record with translational offsets.

From the first to the last frame of a superresolution sequence, the camera is, in total, displaced by around $1\ cm$ to $1.5\ cm$. Here the scene distance was $1.5\ m$ on average. In order to cancel out random noise, we average over multiple depth measurements at each camera position. We also discard depth measurements on the outer boundary of the image due to the previously described higher measurement uncertainty.

4.2 Algorithm

By appropriately combining the low resolution depth images $\mathbf{Y_k}$, $k = 1,\ldots,N$ taken from slightly displaced viewpoints, we can create new depth maps at significantly higher resolution. Using reprojection, the upsampled depth maps can then be converted to high



(a) Shift **(b)** Shift

Figure 4.3: The recording setup: rotating between two captures is the theoretically correct approach. If the displace is kept small translation can also be used.

resolution 3D geometry. Our depth superresolution method is based on the approach by Farsiu et al. [7] who investigated superresolution for normal photographs.

We cast superresolution as the problem of inverting the formation process of low resolution depth images of a high resolution 3D scene. To formulate the problem, we make the simplifying assumption that the formation process of a depth image can be described in analogy to the image formation process of a normal optical camera. The quality of our final results shows that this simplification is valid. For a single depth image $\mathbf{Y_k}$, the formation process therefore looks as follows:

$$\mathbf{Y_k} = D_k H_k F_k \mathbf{X} + \mathbf{V_k},$$

where $\mathbf{X}$ is the original scene or, in other words, the superresolved image of the 3D scene from which we sample. Henceforth, we will refer to the upsampling factor between low and high resolution images in x- and y-direction as β. F_k is a translation operator representing the motion between the superresolution image and the current low resolution image. In our setting, we assume pure translational motion. H_k is a blur operator accounting for the blur introduced during the capture process (i.e. due to the optic system or motion). In our experiments we assumed no blur, hence H_k was equivalent to the unity matrix. D_k is a decimation operator modeling the downsampling from the superresolution image to the size of the low resolution image. Finally $\mathbf{V_k}$ represents additive noise inherited during the capture process. To extract the high resolution image from the set of low resolution depth maps, we need to solve the following minimization problem:

$$\widehat{\mathbf{X}} = \underset{\mathbf{X}}{argmin} \left[\sum_{k=1}^{N} \| D_k H_k F_k \mathbf{X} - \mathbf{Y_k} \|_p^p \right], \tag{4.1}$$

where [7] readily argues that $p = 1$ gives optimal results in terms of robust statistics. Since with a typical set of images this estimation problem is ill-posed, one is to add a regularization term $\Upsilon(\mathbf{X})$ with weight λ yielding

$$\widehat{\mathbf{X}} = \underset{\mathbf{X}}{argmin} \left[\sum_{k=1}^{N} \|D_k H_k F_k \mathbf{X} - \mathbf{Y_k}\|_p^p + \lambda \Upsilon(\mathbf{X}) \right] \tag{4.2}$$

Different regularization terms such as Tikhonov regularization or Total Variation could be imagined. For this work, we used bilateral regularization. This robust technique, also referred to as bilateral filtering, has the advantage of preserving edges and removing random noise in areas of slowly varying depth. Also, the computation of the regularizer is relatively cheap. The bilateral regularization is given by

$$\Upsilon(\mathbf{X})_B = \underbrace{\sum_{l=-P}^{P} \sum_{m=0}^{P}}_{l+m>=0} \alpha^{|m|+|l|} \left\| \mathbf{X} - S_x^l S_y^m \mathbf{X} \right\|_1$$

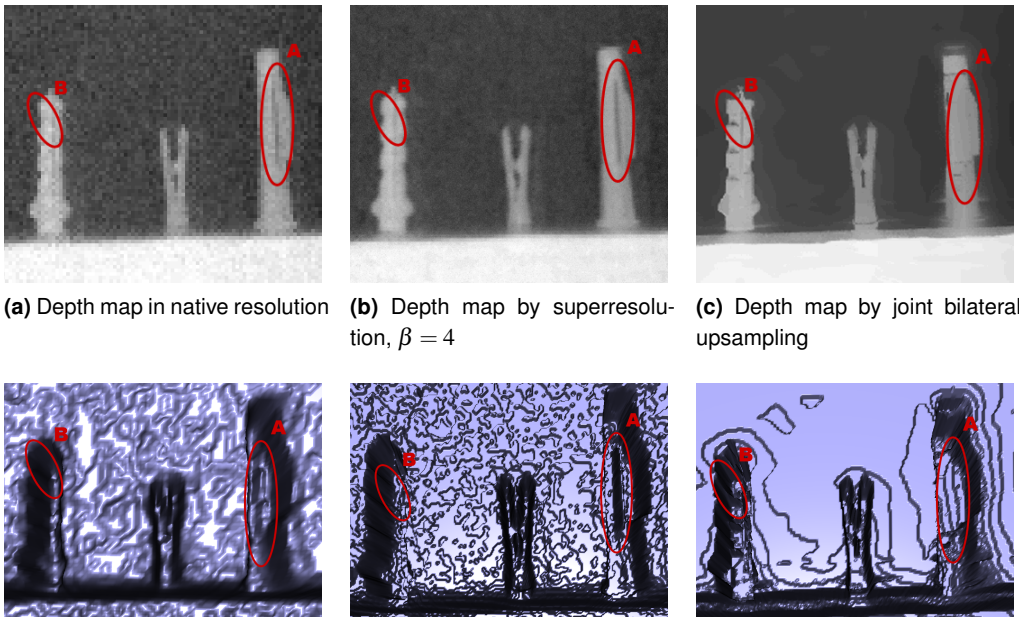
here S_x^l and S_y^m are shift operators that perform a shift in x or y direction by l or respectively m pixels. The scalar weight α, with $0 < \alpha < 1$, controls the spatial influence area of the bilateral constraint, $P \geq 1$ specifies the size of the neighborhood used for bilateral filtering. Please refer to [7] to learn about the equivalence of the above formulation to the original bilateral filter proposed in [46]. The robust bilateral formulation in Eq. (4.2) is preferable over quadratic penalization since the latter would perform worse in the presence of the heavy-tailed random noise in the raw depth data.

4.3 Implementation

Solving the optimization problem in Eq. (4.2) yields a superresolved depth image of the scene. In practice, we employ the solver implementation provided by Milanfar [34] to compute the solution. From the superresolution depth image, we reconstruct 3D geometry by reprojection. Prior to 3D reconstruction, we median filter the superresolution depth image with a kernel size of 3×3. Please remember that the effective metric pixel size in the high-resolution image is μ/β.

4.4 Results

We have tested our approach on three different scenes, all of which show geometric detail that is close to the X-Y resolution limit of the depth camera in one frame. The test scenes



(a) Depth map in native resolution

(b) Depth map by superresolution, $\beta = 4$

(c) Depth map by joint bilateral upsampling

(d) 3D model from native resolution

(e) 3D model from superresolution, $\beta = 4$

(f) 3D model from joint bilateral upsampling

Figure 4.4: Wall plug scene - superresolution (b),(e) unveils fine details, previously not visible in native resolution (a),(d). Joint bilateral upsampling (c),(f) sharpens the image, but introduces false geometry. For better visibility the contrast of depth maps was enhanced.

also feature areas that contradict the assumption color and depth discontinuities are well-aligned, which allows us to show that methods relying on this simple prior statistics will perform worse.

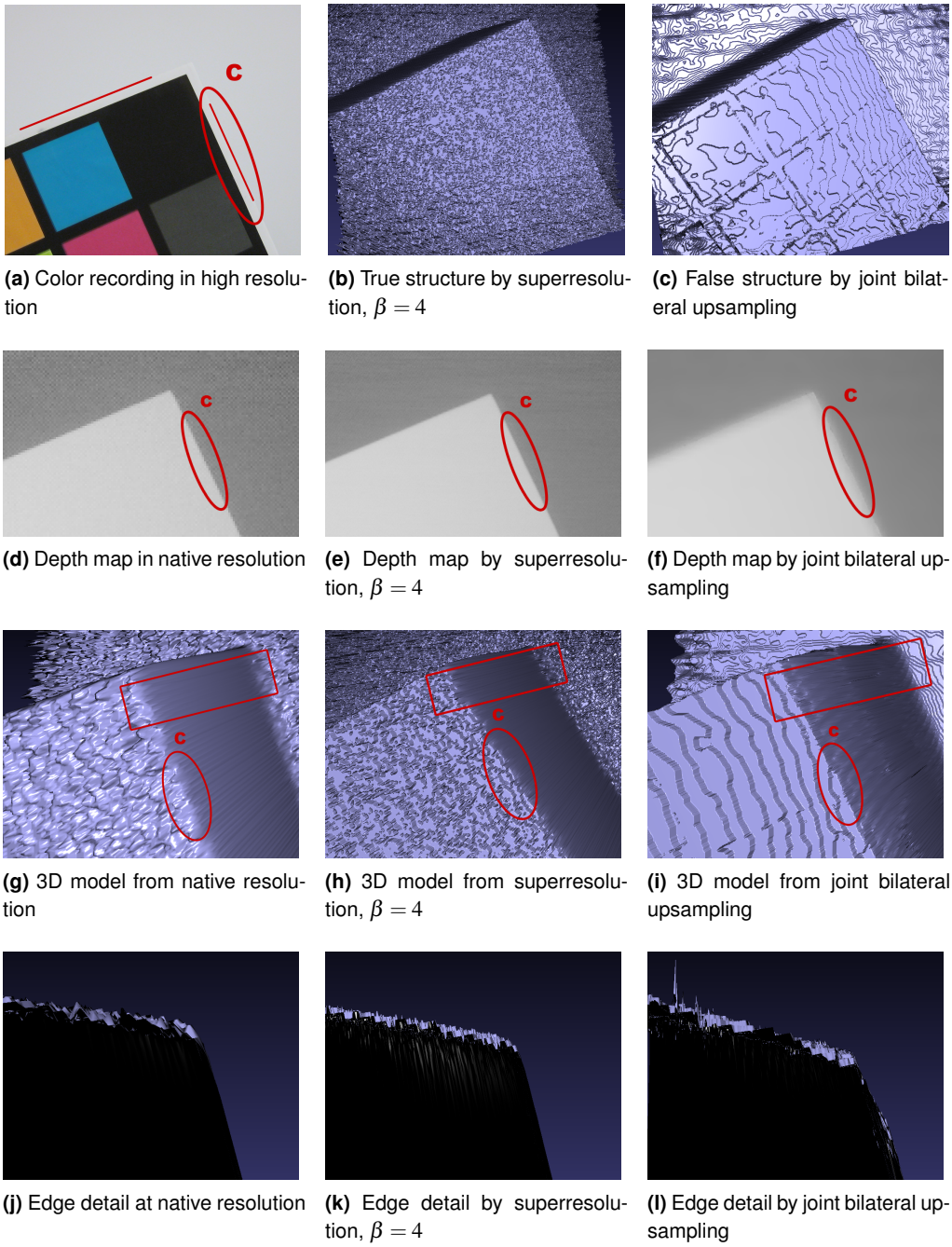
Resolving thin structure: We wanted to verify that our superresolution method can resolve thin structures. Therefore our first setup shows three wall plugs in front of a white wall, Fig. 4.4. The scene is approx. $50\ cm$ away from the camera, and was recorded from 15 displaced positions to perform superresolution. For this scene, the camera was configured to record objects from $0\ cm$ up to $100\ cm$ away. To illustrate the performance of our method, we focus on a dent and a long thin gap in the wall plugs which are marked as A and B, respectively, in Fig. 4.4. Since these features are close to the resolution limit of the Z-cam, they do not appear well in a single depth image, Fig. 4.4a, and consequently also not in the corresponding low resolution 3D reconstruction, Fig. 4.4d. In contrast, our 4-times superresolved result accurately captures these details, as visible in the depth image Fig. 4.4b, and in geometry Fig. 4.4e where they appear as true 3D structure with correct depth. To display the 3D geometry we convert the depth maps into triangulated height

fields and render them using basic Phong shading. Please note that for fair comparison we always perform superresolution at 8-bit depth precision in all tested methods, as this is the limit of the software by Milanfar et al. [34]. Therefore, discretization artifacts in the form of depth steps are visible in the renderings. To verify that our 3D reconstructions do not suffer from incorrect scaling or distortion we compared the size of several landmarks in our results to their real-world size. In all cases, this comparison showed an exact match which proves the reliability of our algorithm.

For comparison, we implemented a joint bilateral upsampling (JBU) approach [25], which uses a high-resolution color and a low resolution depth image to raise the depth resolution to the one of the color image. The color image was recorded using a standard digital camera and has been manually aligned using a homographic warp. By inspection the error was determined to be three pixels at maximum. The method's implicit assumption that color and depth edges are collocated is frequently violated in our wall plug scene causing erroneous reconstructions. Although the depth map, Fig. 4.4c, shows crisp edges which is visually pleasing if only the gray scale image is looked at, the actual reconstruction exhibits several errors. For instance, the method wrongly reconstructs the shadowed area *B* on the ripple of the left wall plug as a depth discontinuity that protrudes all the way through the scene Fig. 4.4f. Also, joint bilateral upsampling performs excessive smoothing in areas with low image gradient. Therefore, the dent in area *A* on the right wall plug, whose edges are not clear in the color image, is entirely smoothed out. Also, shadows on the back of the table appear as geometry merged to the lower part of the plugs, and the top of the right plug is cut off due color similarity to the background. We thus conclude that a slightly higher remaining level of noise, as in in our results, is preferable over such excessive smoothing since in the latter case actual shape detail is lost or incorrectly estimated.

Preserving sharp edges: Another important characteristic of superresolution is to preserve sharp edges. Hence, a second scene, with a planar checkerboard spaced approx. $50\ cm$ from a white background, was recorded to prove that our method correctly captures both sharp edges and smooth regions, Fig. 4.5a. In contrast, the joint bilateral upsampling method runs into difficulties in the presence of strong texture on actually planar geometry. Here the camera was configured for recording between $70\ cm$ and $200\ cm$. The board features a color pattern with strong intensity gradients. The pattern is slightly smaller than the actual size of the board, which has a $1\ cm$ white boundary that is visually indistinguishable from the white background.

In Fig. 4.5a, we marked the location of the actual depth edge with lines. The low resolution depth image (Fig. 4.5d) has an apparent staircase effect on the edge, while the edge appears sharp and crisp in the depth map created by the proposed superresolution method



(a) Color recording in high resolution

(b) True structure by superresolution, $\beta = 4$

(c) False structure by joint bilateral upsampling

(d) Depth map in native resolution

(e) Depth map by superresolution, $\beta = 4$

(f) Depth map by joint bilateral upsampling

(g) 3D model from native resolution

(h) 3D model from superresolution, $\beta = 4$

(i) 3D model from joint bilateral upsampling

(j) Edge detail at native resolution

(k) Edge detail by superresolution, $\beta = 4$

(l) Edge detail by joint bilateral upsampling

Figure 4.5: Board scene - The upper row shows that "phantom" geometry is introduced by joint bilateral upsampling (b), whereas superresolution retains the true geometry (c). This effect is also visible in the depth maps one row below. The two lower rows show sharp edges being preserved by superresolution, while joint bilateral upsampling yields round edges.

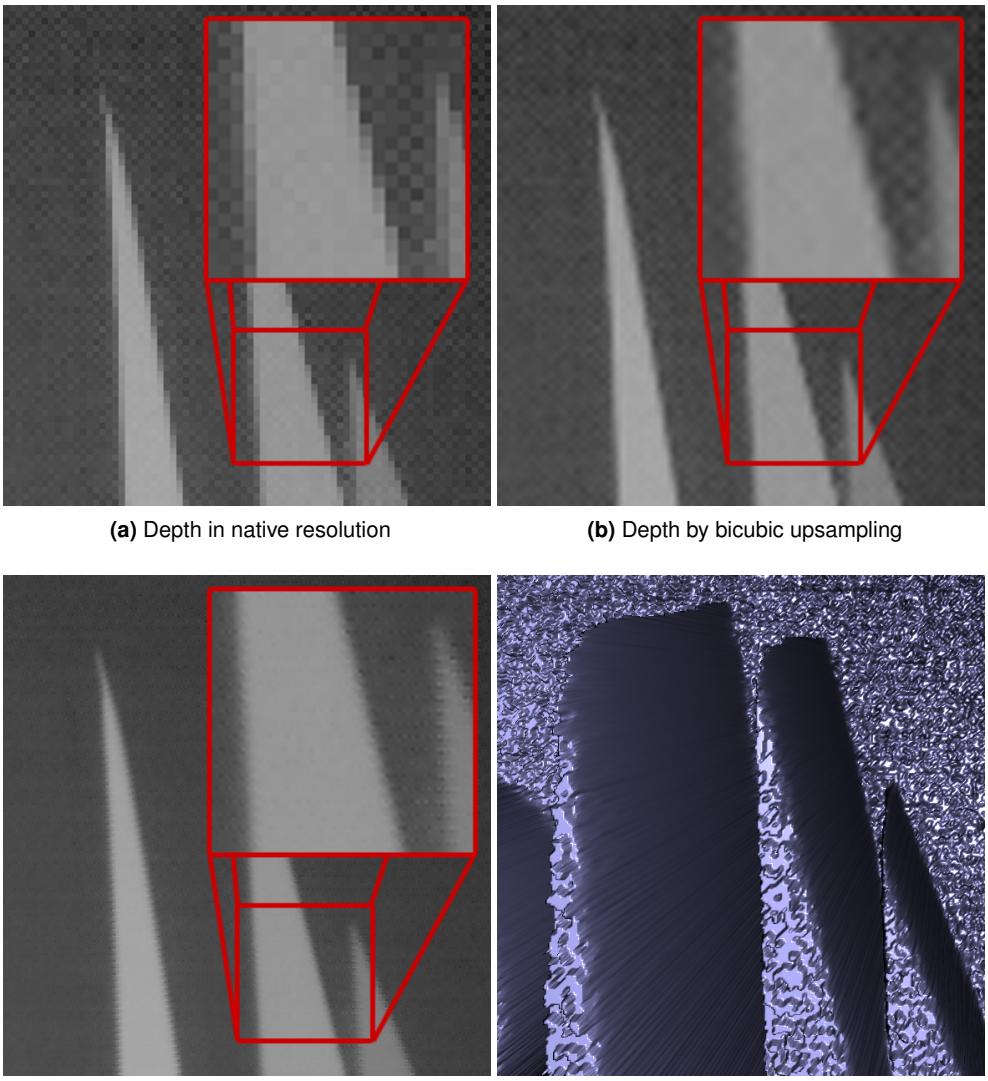
(Fig. 4.5e). The joint bilateral upsampling method is tricked by the non-collocation of the intensity gradient (black pattern boundary) and true depth discontinuity. Consequently, the true depth edge is smoothed with the background leading to a blurred edge in the JBU depth image, Fig. 4.5f. This effect can be studied best in 3D. While our superresolved geometry, Fig. 4.5h, shows a sharp edge with sharped depth discontinuity, the edge of the joint bilateral upsampling result is incorrectly shaped like a curved ramp, Fig. 4.5i. The rendering of the depth edges in a cross-sectional views, Fig. 4.5j-4.5l, makes this effect even more apparent. Or result shows a sharp corner and a straight depth edge, Fig. 4.5j, whereas the JBU result is erroneously curved, Fig. 4.5l. Another problematic region for joint bilateral upsampling is the surface of the checker board itself. Whereas it appears up to noise as a plain, the color gradients in the checker board provoke the bilateral filter to emboss this structure into the geometry (Fig. 4.5c). In contrast, our upsampling result shows a planar board, Fig. 4.5b.

Gain in resolution: To further demonstrate the true gain in resolution, we recorded three planar triangular wedges $30\ cm$ in front of a flat wall. They exhibit clear sharp depth edges and, close to the tips, fall below the resolution limit of the camera. The recording settings were $P_d = 50\ cm$ and $P_w = 100\ cm$. While the depth map at original camera resolution exhibits strong staircase aliasing at the boundaries, Fig. 4.6a, our 4-times upsampled result faithfully captures crisp depth edges, Fig. 4.6c. Consequently, the upsampled 3D geometry also shows sharp edges, Fig. 4.6d. Simple bicubic upsampling of the low resolution data cannot produce the same superresolution effect. It mainly upsamples the staircase pattern and boosts the random noise, Fig. 4.6b.

Our method is subject to a few limitations. Since several depth images have to be combined it is, in contrast to joint bilateral upsampling, only suitable for static scenes. Also, given a runtime of approximately one minute to compute a superresolved depth map, our approach is not suitable for real-time applications. Furthermore our approach relies on faithful image registration which may be difficult in scenes with few distinct depth discontinuities. In the future, we plan to capitalize on noise characteristics and known measurement uncertainty, from which we expected improved superresolution performance.

We will also perform a more detailed analysis of the range of achievable upsampling factors in dependence on scene structure and recording conditions. Currently, we did tests with β in the range of $2-6$. Overall, we found that, in our test scenes, $\beta = 4$ provides the best compromise between extracted shape detail and model size.

We would also like to remark that both tested superresolution methods rely on a bilateral constraint of some form. It is not the constraint itself that makes one method preferable over the other, but the particular way how it is enforced. Joint bilateral upsampling enforces the constraint in two different data domains, namely color and depth, and implicitly relies



(a) Depth in native resolution

(b) Depth by bicubic upsampling

(c) Depth by superresolution

(d) 3D model from superresolution

Figure 4.6: Wedge scene - superresolution ($\beta = 4$) achieves true resolution enhancement and shows straight alias-free edges at depth boundaries (c),(d). In contrast, staircasing artifacts are clearly visible at native resolution (a) and in the bicubic upsampled result (b). Additionally noise is significantly reduced by superresolution.

on the wrong prior. In contrast, we enforce the constraint on depth data only and do not enforce the same excessive smoothing as the former approach which renders advantageous in our setting.

5 LidarBoost

While the previous chapter showed that superresolution can be applied towards depth data, this chapter focuses on finding an optimal superresolution approach geared towards Time-of-Flight camera systems. Specifically this algorithm has an edge prior tailored to depth data and incorporate the additional information that is available by the intensity maps provided by the depth camera. This new algorithm called *LidarBoost* has recently published at CVPR [45].

5.1 Algorithm

Similar to Chapter 4, in our measurement setup we capture N depth images of a static scene , $Y_k \in \mathbb{R}^{n \times m}$, each having depth sensor resolution $n \times m$. Each depth image (also called depth map) is a grid of depth pixels, where each depth pixel records the distance to a 3D scene point along the measurement ray through the pixel. Given intrinsic ToF camera calibration data, a depth map can be reprojected into 3D geometry in world space. The Y_k are captured from only slightly displaced viewpoints which is why parallax effects can be neglected. Prior to superresolution, all depth images are registered against a reference frame out of Y_k. Once registered, we compute a single high resolution depth image with β times higher resolution $X \in \mathbb{R}^{\beta n \times \beta m}$ by solving an optimization problem of the form:

$$\text{minimize} \quad E_{\text{data}}(X) + E_{\text{regular}}(X)\,.$$

The first term $E_{\text{data}}(X)$ is a data term measures agreement of the reconstruction with the aligned low resolution maps. $E_{\text{regular}}(X)$ is a regularization or prior energy term that guides the optimizer towards plausible 3D reconstructions if data points are sparse, Chapter 5.1.

This formulation is common to most superresolution methods. However their data and prior terms are designed for intensity images and cause strong artifacts when applied to depth images, as shown in Fig. 5.7c for the example of our previous algorithm. In contrast, our prior and data terms explicitly take into account the specifics of the 3D reconstruction problem as well as the characteristics of the time-of-flight sensors used. In contrast to related 3D upsampling methods, our formulation yields a convex optimization problem which makes the superresolution procedure efficient and robust. Overall, our su-

perresolved depth maps therefore exhibit a much higher quality than it was achieved with previous approaches for ToF superresolution.

Data Term

The data term ensures that the final superresolved depth map is coherent with the registered low resolution measurements $Y_k \in \mathbb{R}^{n \times m}$. During preprocessing, $N-1$ frames out of the Y_k frames are aligned against a reference frame by computing for each a displacement vector. Typically, the first frame from Y_k is chosen as reference frame. Currently, we use hierarchical Lukas Kanade optical flow [33] to compute the registration but alternative registration approaches would be feasible. This process and the upsampling described below transform each original frame Y_k into an aligned frame $D_k \in \mathbb{R}^{\beta n \times \beta m}$:

It is our goal to compute a higher resolution version of a 3D depth map from aligned low resolution depth maps. When solving for the high resolution image we therefore have to resample the aligned high-resolution depth pixel grid of the target image. We performed experiments to determine the best resampling strategy. It turned out that a nearest neighbor sampling from the low resolution images is preferable over any type of interpolated sampling. Interpolation implicitly introduces unwanted blurring that leads to a less accurate reconstruction of high-frequency shape details in the superresolved result.

Our data term takes the following form:

$$E_{\text{data}}(X) = \sum_{k=1}^{N} \left\| W_k .* T_k .* (D_k - X) \right\|_2 ,$$

where .* denotes element-wise multiplication. $W_k \in \mathbb{R}^{\beta n \times \beta m}$ is a banded matrix that encodes the positions of D_k which one samples from during resampling on the high-resolution target grid. $T_k \in \mathbb{R}^{\beta n \times \beta m}$ is a diagonal matrix containing 0 entries for all samples from D_k which are unreliable according to the ToF sensor's readings, as described in the following:

Since a ToF camera relies on a sufficiently strong return of the emitted IR pulse to measure depth, certain scene characteristics lead to biased or totally wrong depth estimates. In consequence, if a surface reflects light away from the camera, or if it absorbs most of the light, depth measurements become unreliable. An example can be seen in Fig. 5.6, where the ball has problematic reflectance properties and the print on the box absorbs most of the light. Fortunately, a low amplitude of the returned light wavefront at each pixel (the SR 3000 camera we use gives access to an amplitude image) indicates the occurrence of such difficult situations and, thus amplitude serves as a form of confidence measure. We therefore use a thresholding approach, to detect and exclude low-confidence measurements with low amplitude. Technically this is implemented in the matrix T_k which multiplies unreliable samples by 0.

We would like to remark that the choice of error norm is critical to the quality of the final result. In essence, the norm decides at each high resolution depth pixel on how to choose a best target depth position given the depth values from all low resolution maps at that position. The previous depth superresolution methods as well as many image superresolution methods, employ a ℓ_1-norm. While a ℓ_1-norm forces the depth value at a certain high-resolution grid point towards the median of registered low-resolution samples, an ℓ_2-norm yields their mean. For very noisy data, the median is certainly reasonable since it rejects outliers. In contrast, the mean yields a smoother surface reconstruction, since the averaging cancels out recording noise. From our experience using ToF data and our method, it is more beneficial to capitalize from the smoothing effect of a ℓ_2-norm.

Regularization Term

The regularization or prior term guides the energy minimization to a plausible solution, and is therefore essential if data are sparse and noise-contaminated.

We seek a prior that brings out high frequency 3D shape features that were present in the original scenes in the upsampled 3D geometry. At the same time the prior shall suppress noise in those regions that correspond to actually smooth 3D geometry. Finally we seek it to be convex.

All these properties can be enforced by designing a prior that favors certain distribution of the spatial gradient in the final depth map. On the one hand we want to preserve local maxima in the spatial gradient that correspond to high frequency features, e.g. depth edges. On the other hand, we want the overall distribution of the gradient to be smooth and relatively sparse which cancels out random noise.

One way to enforce this property is to resort to a sum-of-gradient-norms regularization term that can be computed efficiently, and that has also been used by previous image superresolution methods. However, the implementation of this regularizer for image superresolution often enforces sparseness on individual differences contributing to an overall finite difference approximation of the spatial gradient. For instance, the regularizer employed at our previous algorithm essentially enforces sparseness on the elements of the approximated vector (i.e. sparseness on the individual finite differences). Although this prior manages to preserve high frequency detail to a certain extent, it completely fails in areas of smooth geometry where it creates a severe staircasing pattern (e.g. Fig. 5.1f). While small staircasing artifacts may not be visible if one works with intensity data, 3D reconstructions are severely affected.

We have therefore designed a new sum-of-norms prior that can be efficiently computed and that is tailored to produce high-quality 3D reconstructions. Let $\nabla X_{x,y}$ be a combined vector of finite difference spatial gradient approximations at different scales at depth pixel

position (x,y). Then our regularization term reads:

$$E_{\text{regular}}(X) = \sum_{x,y} \|\nabla X_{x,y}\|_2 = \sum_{x,y} \left\| \begin{pmatrix} G_{x,y}(0,1) \\ G_{x,y}(1,0) \\ \vdots \\ G_{x,y}(l,m) \end{pmatrix} \right\|_2 ,$$

where each $G_{x,y}(l,m)$ is a finite difference defined as follows

$$G_{x,y}(l,m) = \frac{X(x,y) - X(x+l,y+m)}{\sqrt{l^2+m^2}} .$$

In our regularizer, we approximate the gradient with finite differences, but weight the various differences by the inverse Euclidean distances, yielding a rotation invariant approximation. Secondly we compute local gradient approximations at different scales and weight gradient approximations at lower levels of hierarchy (i.e. computed with a higher pixel position difference) lower. An important insight is that it is essential to compute the norm on all differences contributing to a local gradient approximation at different scales *simultaneously* and not on individual finite differences.

Since the (ℓ_2-) norms of all combined gradient vectors in the above sum are positive, the sum has the effect of a ℓ_1-regularization [3] on the entire set of gradient magnitudes: enforcing sparseness, i.e. drive most gradients to zero and hence smooth the result in noisy regions, but allow high-frequency detail to prevail. By combining distance-weighted gradient approximations at different scales we thus implicitly achieve feature preserving smoothing in a computationally efficient and convex way.

Given the data and regularization terms defined in the previous sections, we can now formulate the complete LidarBoost energy function as

$$\sum_{k=1}^{K} \|T_k .* W_k .* (D_k - X)\|_2 + \lambda \sum_{x,y} \|\nabla X_{x,y}\|_2 ,$$

where λ is the trade-off parameter between enforcement of data similarity and smoothness. As one can see in Fig. 5.1g, our approach produces high quality superresolved geometry which exhibits clear 3D features and only marginal noise in smooth areas.

5.2 Implementation

LidarBoost was implemented in MATLAB. All data conversion as well as image alignment took place in pure MATLAB. For the optimization problem we build on the cvx modeling

framework for disciplined convex optimization [11]. This framework transforms the problem into Second-Order-Cone-Program (SOCP) and solves it using a generic solver. Due to the size of the transformed problem, which easily exceeded a million variables, we subsequently compute solutions for images patches of 20×20 low-resolution pixels and stitch the results using two-pixel overlap (similar to primal decomposition with one iteration). Computation time for the synthetic scenes (9 patches) was about five minutes and for the real scenes (28 - 48 patches) up to two hours.

Steepest Descent While the use of a modeling framework was important to develop the algorithm, the processing time using this approach was beyond practical use. Since a rough estimate of the minimizer of our optimization problem, namely one of the input depth maps is known, the use of incremental optimization techniques seems to be justified. Here we shall use one of the simplest algorithms, namely steepest descent. Hence we first need the gradient, which we derive here. We consider the gradient for the data term and the regularization term separately, since they are joint by a linear operator). Furthermore, we assume the optimization variable to be a vector as length n instead of a matrix. This is valid, since the conversion is a simple stacking of the matrix and we never use any matrix-specific attribute such as an Eigenvalues.

$$\nabla E_{\text{data}}(X) = \nabla \sum_{k=1}^{N} \|W_k .* T_k .* (D_k - X)\|_2 = \tag{5.1}$$

$$= \sum_{k=1}^{N} -\frac{1}{\|W_k .* T_k .* (D_k - X)\|_2} W_k .* T_k .* (D_k - X) \tag{5.2}$$

. For the regularization term, we consider first the derivative in a single point, namely at $P(u,v)$. Then the regularizer for that point reads:

$$\left\| \begin{pmatrix} G(0,1) \\ G(1,0) \\ \vdots \\ G(l,m) \end{pmatrix} \right\|_2 = \left\| \begin{pmatrix} \frac{X - X(l,0)}{\sqrt{1^2+0^2}} \\ \frac{X - X(0,1)}{\sqrt{0^2+1^2}} \\ \vdots \\ \frac{X - X(l,m)}{\sqrt{l^2+m^2}} \end{pmatrix} \right\|_2 \tag{5.3}$$

Hence for that point $P(u,v)$ the gradient vector will be sparse and will have non-zero elements only for indices $X(s,t)|u-l \leq s \leq u+l, v-m \leq t \leq v+m$. Let's consider the partial derivative for $X(s,t)$:

$$\frac{\partial}{\partial X(s,t)} \left\| \begin{pmatrix} G(0,1) \\ G(1,0) \\ \vdots \\ G(l,m) \end{pmatrix} \right\|_2 \tag{5.4}$$

is by the chain rule

$$\frac{\frac{\partial}{\partial X(s,t)} G(0,1)^2 + G(1,0)^2 + \cdots + G(l,m)^2}{2 \left\| \begin{pmatrix} G(0,1) \\ G(1,0) \\ \vdots \\ G(l,m) \end{pmatrix} \right\|_2} \tag{5.5}$$

distinguishing the cases, where (s,t) is the center point, i.e. $s = u, t = v$ and where surounding pixels are considered:

$$= \begin{cases} \dfrac{\left(\frac{1}{\sqrt{0^2+1^2}} + \frac{1}{\sqrt{1^2+0^2}} + \cdots + \frac{1}{\sqrt{l^2+m^2}} \right) X(s,t)}{\left\| \begin{pmatrix} G(0,1) \\ G(1,0) \\ \vdots \\ G(l,m) \end{pmatrix} \right\|_2} & s = u, t = v \\ \dfrac{-\frac{1}{\sqrt{(u-s)^2+(v-t)^2}} X(s,t)}{\left\| \begin{pmatrix} G(0,1) \\ G(1,0) \\ \vdots \\ G(l,m) \end{pmatrix} \right\|_2} & \textit{else} \end{cases} \tag{5.6}$$

using the substitution $l = u - s$ and $m = v - t$. Rearranging this, into matrix form we have

$$\frac{1}{\left\| \begin{pmatrix} G(0,1) \\ G(1,0) \\ \vdots \\ G(l,m) \end{pmatrix} \right\|_2} \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \begin{bmatrix} \ddots & \cdots & \ddots \\ \vdots & \left(\frac{1}{\sqrt{0^2+1^2}} + \frac{1}{\sqrt{1^2+0^2}} + \cdots + \frac{1}{\sqrt{l^2+m^2}} \right) X(s,t) & \vdots \\ \ddots & \cdots & -\frac{1}{\sqrt{(u-s)^2+(v-t)^2}} X(s,t) \end{bmatrix} & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \tag{5.7}$$

We see, this gradient is basically a filter with a local kernel. The kernel itself has some notable properties. The overall sum of its elements is zero; hence it is an energy preserving

kernel. The kernel elements are weighted with the inverse distance to the filter center. Since a Euclidian norm is used (instead i.e. a taxi-cab norm) this makes the filter rotation-invariant. Furthermore the inverse distance weight attenuates large gradients, i.e. acts as an edge filter (which in turn of the optimization scheme adds cost to them and will smooth the result). Furthermore note that the total weight of the filter is furthermore multiplied by the inverse magnitude of the gradient approximation, hence favors small gradients. Also we see that most of the filter can be pre-computed, enabling an efficient implementation.

Using this gradient, a steepest-descent implementation is straight forward. We leave the implementation of a more efficient solver based on this theory for future work

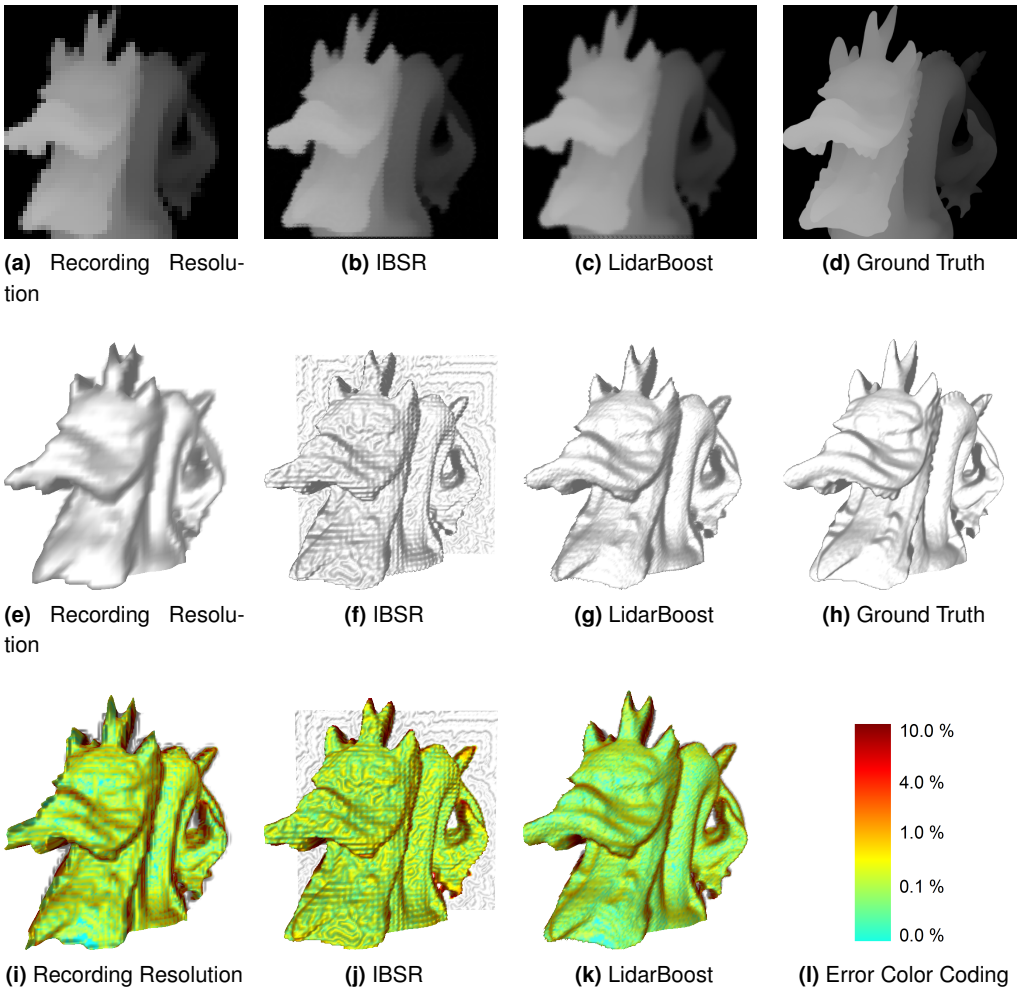
5.3 Results

To explore the capabilities of the new approach, we tested it on synthetic and real sequences captured with a Swissranger SR3000 camera (176×144 depth pixel resolution). We also compared LidarBoost to two alternative approaches from the literature. First we compare against an image-based superresolution method applied to depth data (IBSR), in particular we used our previous algorithm. We apply the publicly available implementation of Farsiu's approach and choose the following parameters: $\lambda = 0.04, N = 50, \alpha = 0.7, \beta = 1, P = 5$, and a Gaussian 3×3 PSF with standard variance (see original paper for details). The computation time was below two minutes for all scenes.

Second, on the real scenes only, we compare against color and depth fusion method, namely the method by Diebel and Thrun [6]. We ran all method with several parameterizations and show only the best results for each method in the respective scenes.

Synthetic Scene - No Noise Added A first comparison is performed on synthetic images of the Stanford Graphics Lab's dragon model created with 3D Studio Max. Synthetic ground truth depth maps of resolution 400×400 were rendered and downsampled by factor 8 (using a uniform 8×8 kernel) to simulate low resolution input depth maps. In total, $N = 10$ low resolution input images from slightly displaced viewpoints were created. One such input depth maps is shown in Fig. 5.1a, compared to the ground truth shown in Fig. 5.1d. Figs. 5.1b and 5.1c show the four times superresolved results computed by applying IBSR and LidarBoost. Below each depth map, we show renderings of the corresponding 3D geometry (obtained by reprojection into 3D) since depth maps only do not properly visualize the true gain in quality and tend to mask unwanted artifacts.

Our previous algorithm successfully brings out the outline of certain shape detail that was not visible in individual input frames, Fig. 5.1f, such as individual toes and sharp boundaries. However, the results are clearly contaminated by the previously discussed



(a) Recording Resolution **(b)** IBSR **(c)** LidarBoost **(d)** Ground Truth

(e) Recording Resolution **(f)** IBSR **(g)** LidarBoost **(h)** Ground Truth

(i) Recording Resolution **(j)** IBSR **(k)** LidarBoost **(l)** Error Color Coding

Figure 5.1: Synthetic test set without noise ($4\times$ upsampling): The first row depicts the depth maps, from which a 3D geometry has been rendered as shown in the second row. The third row shows a rendering, with color coded rMSE. IBSR recovers the overall structure, but exhibits a noise pattern. LidarBoost recovers the structure almost perfectly and yields a smooth surface.

staircase pattern (Sect. 5.1). In comparison, LidarBoost (Fig. 5.1g, with $\lambda = 0.04$) extracts more detail (e.g. the eye holes and the third small horn of the dragon) and at the same time successfully eradicates measurement noise without introducing a disturbing pattern.

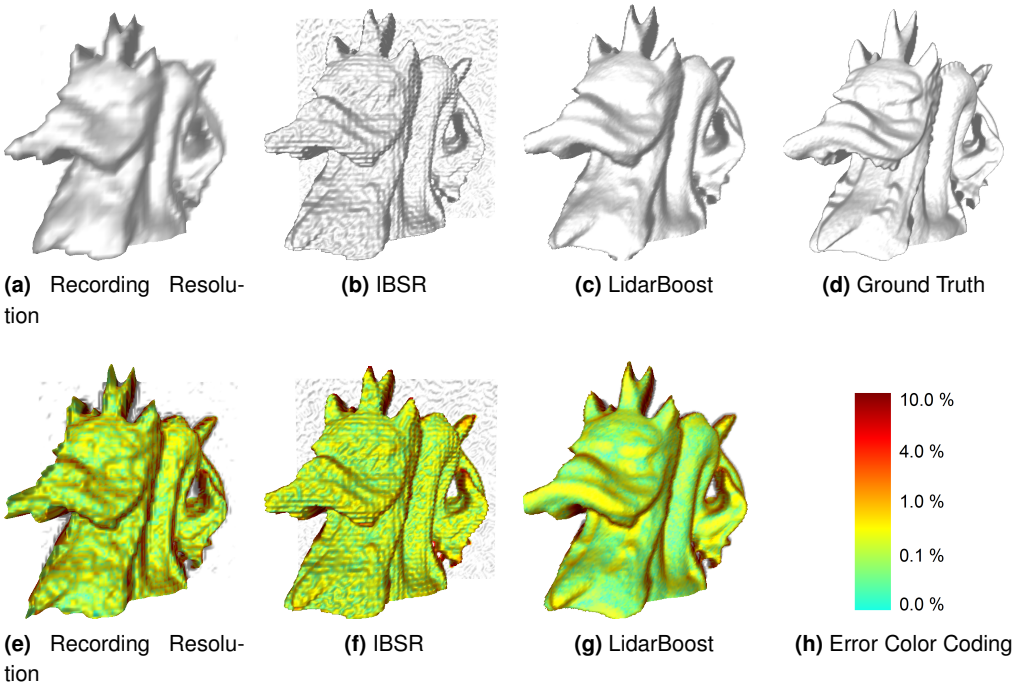
On synthetic data we can also perform quantitative comparisons against ground truth and compute the relative mean square error. It is relative, because the MSE result was divided by the number of pixels considered to keep numbers reasonable. A two times downsampled version of a reference 400×400 depth depth map forms the ground truth

- to make resolutions match. One low resolution depth map has been upsampled four times using a nearest neighbor approach to establish a baseline comparison. LidarBoost clearly outperforms the IBSR method and leads to significant improvements over a single low-resolution depth map. Figs. 5.1i - 5.1k show a color-coded rendering of the error distribution (rMSE in percent of longest bounding box dimension of synthetic object) over the model surface using the color scheme shown in Fig. 5.1l (green=low error, red=large error). Both methods struggle on edges, which comes to no surprise, as the sub-pixel exact location for a steep edge is hard to guess. Despite a potentially small mis-localisation, LidarBoost still recovers depth edges more reliably than the comparison method. Also, the pattern introduced by IBSR leads to much stronger errors in the interior regions than with LidarBoost.

Synthetic Scene - Medium Noise Added Depth images are inherently noisy, therefore the algorithms need to be evaluated on such data. To simulate the effect of measurement noise introduced by real ToF cameras, we repeated the experiment from the previous section, but added Gaussian noise with a variance of 0.7 along the measurement ray directions following the sensor characterization proposed by Kim et al. [22]. In the simulated data, depth values range from 0 to 182. Although in scenes with a larger depth range a depth-dependency in noise can be expected, for our test scene with limited range we use a constant variance.

One of the low resolution inputs is depicted in Figure 5.2a, while Figure 5.2b and 5.2c show the superresolved geometry. Here, the advantage of LidarBoost over IBSR is even more apparent. Not only is the visual reconstruction quality under these more challenging circumstances clearly better, but also does the color-coded error rendering Figure 5.2e - 5.2g clearly show the superior reconstruction quality of LidarBoost. This can be seen not only on the surface, but for details such as the eye hole and the two upper horns.

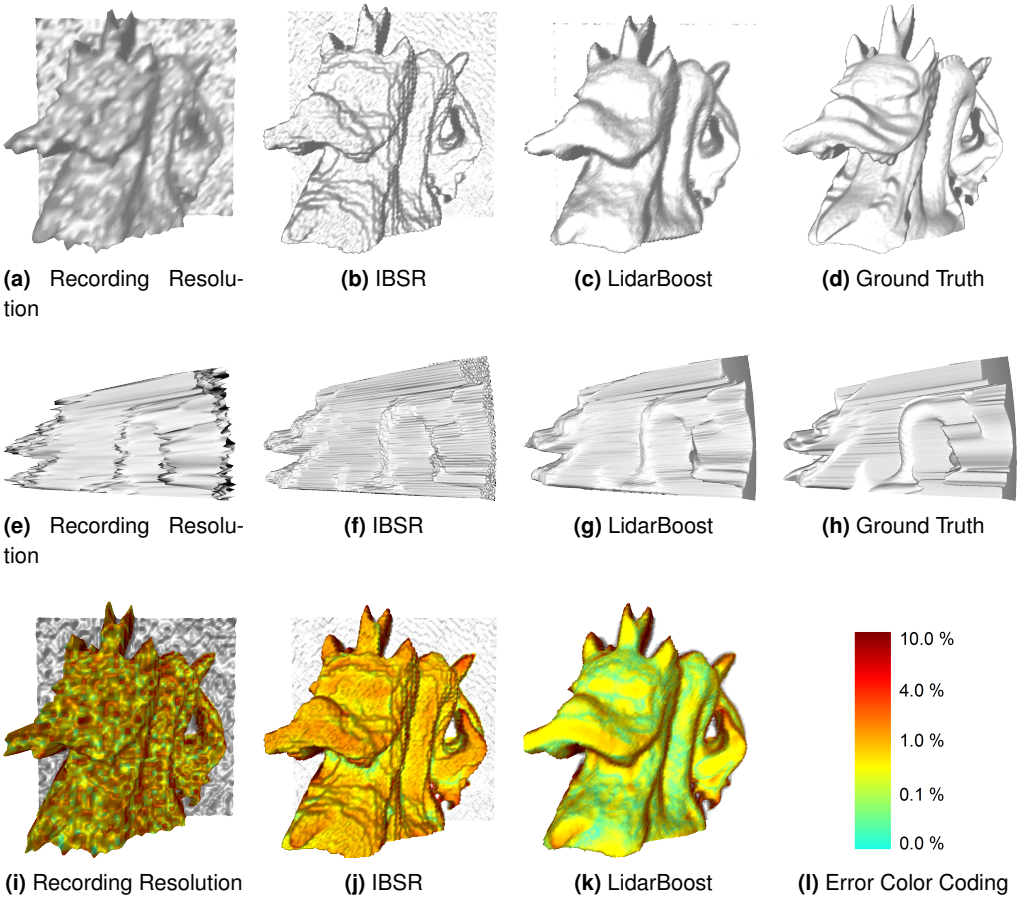
Synthetic Scene - Stark Noise Added We performed another test on data with even more noise. Here the added noise had a variance of 5.0. For the stark noise case, in a single low resolution input frame (Fig. 5.3a) all fine surface detail vanished and it is even hard to recognize the object's shape as a whole. While IBSR recovers a decent level of shape detail (Fig. 5.3b), severe staircasing becomes visible on the geometry and the result is distorted by the random pattern discussed before. In contrast, in particular under these extreme conditions, LidarBoost recovers clearly more detail (even traces of the dragon's pattern on the back, as well as the dragon's teeth) and maintains truly smooth geometry in actually smooth areas. The color-coded error rendering confirms that under these challenging conditions the advantage of using LidarBoost relative to IBSR is even stronger, Figs. 5.3i - 5.3k.



(a) Recording Resolution **(b)** IBSR **(c)** LidarBoost **(d)** Ground Truth

(e) Recording Resolution **(f)** IBSR **(g)** LidarBoost **(h)** Error Color Coding

Figure 5.2: Synthetic test set with medium noise (Variance of 0.7, $4\times$ upsampling): The first row shows 3D renderings of one input depth map (a), upsampled results (b),(c), and ground truth (d). While IBSR improves the resolution, a severe pattern is produced. In contrast, LidarBoost reproduces the overall geometry much more reliably as a comparison to the ground truth shows. the color-coded error rendering in the second row also shows quantitatively that LidarBoost yields more detailed and more accurate surfaces.



(a) Recording Resolution **(b)** IBSR **(c)** LidarBoost **(d)** Ground Truth

(e) Recording Resolution **(f)** IBSR **(g)** LidarBoost **(h)** Ground Truth

(i) Recording Resolution **(j)** IBSR **(k)** LidarBoost **(l)** Error Color Coding

Figure 5.3: Synthetic test set with stark noise (Variance of 5.0, $4\times$ upsampling) - First row: rendered 3D geometry in frontal view, LidarBoost shows shows best upsampling result. Middle row: Also in a lateral view it is apparent that LidarBoost's reconstruction is closest to ground truth. Bottom row: LidarBoost clearly produces the lowest reconstruction error.

Synthetic Scenes - Quantitative Comparison Looking at all synthetic test data sets, the overall trend in rMSE error confirms the visual observations (Table 5.1). In all noise cases our algorithm performs clearly better than the reference approach and clearly improves over the quality of a single low resolution frame. Overall, with increasing noise the performance of IBSR worsens more drastically than our method's results.

	No Noise $var = 0$	Medium Noise $var = 0.7$	Stark Noise $var = 5$
LR	157.6	161.7	203.9
IBSR	83.8	89.9	127.0
LidarBoost	70.6	72.5	82.9

Table 5.1: Relative MSE comparison on synthetic data: LidarBoost throughout outperforms all other methods and shows less sensitivity towards noise then IBSR

Parameter Selection Both LidarBoost and IBSR use a regularization term with a tunable trade-off parameter λ. Fig. 5.4 plots λ against the rMSE obtained with both IBSR and LidarBoost, as evaluated on the dragon data set with stark noise. The reconstruction quality of the former shows a strong dependency on λ, and the rMSE is in general much higher that for LidarBoost. In contrast, the rMSE of LidarBoost is consistently lower and rather stable. Therefore λ requires less tweaking which renders LidarBoost highly applicable. The same observation was made for data sets with no noise and stark noise (Figure 5.5).

Real Scene - Collection of Objects Two real scenes were recorded using a Swissranger SR 3000 depth camera. We recorded $N = 15$ frames each with 30 ms integration

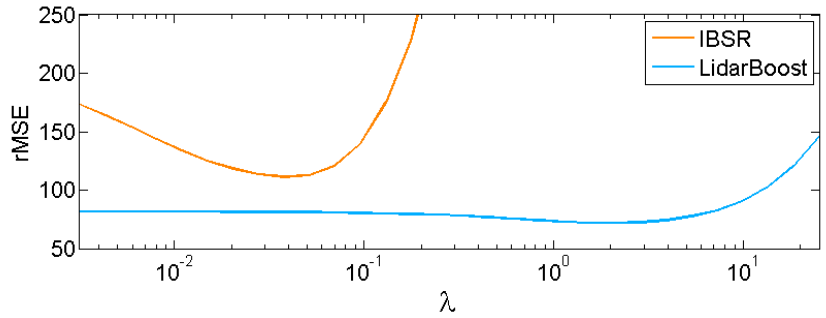
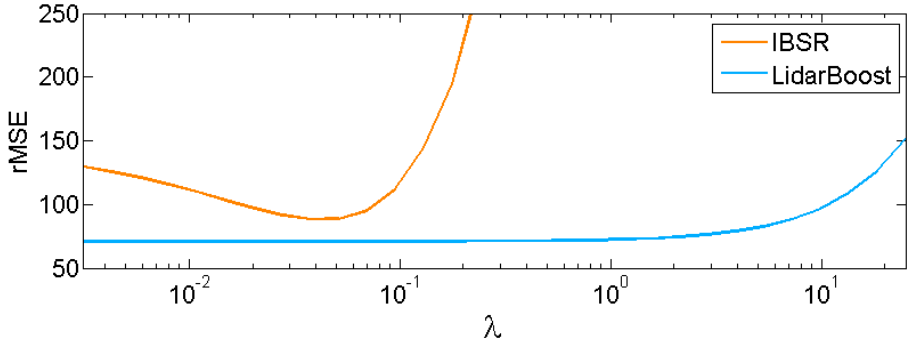
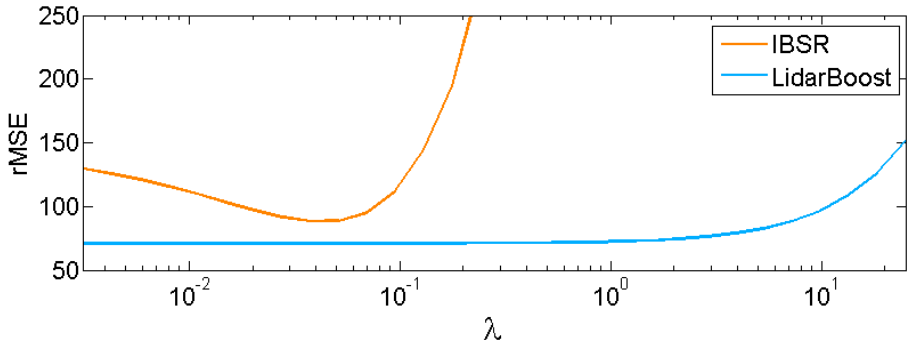


Figure 5.4: Optimal choice of regularization trade-off parameter λ: For the noisy test sets the resulting rMSE has been plotted against varying λ. IBSR is sensitive towards λ with a constant optimum at 0.04. In contrast LidarBoost is robust on a wide range of choices.



(a) No Noise



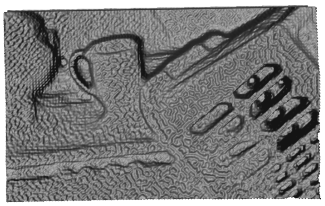
(b) Medium Noise

Figure 5.5: Optimal choice of the trade-off parameter λ: Also in the no noise (a) and medium noise (b) case, one can see that the overall rMSE error of LidarBoost is significantly below the IBSR error. In addition, the choice of λ is much more critical for IBSR which reduces its applicability.

(a) Color Image

(b) Recording Resolution

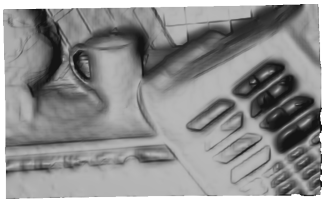
(c) Amplitude image with cuToFf area red



(d) IBSR

(e) LidarBoost

(f) LidarBoost with Confidence Weighting



(g) Diebel's MRF

Figure 5.6: Real scene - collection of objects (a): One of several low-resolution depth maps with an SR3000 ToF cam is shown in (b). IBSR (d) produces an erroneous pattern, whereas LidarBoost (e) correctly recovers high-frequency detail and smooth geometry. When the reflectivity of the materials is really low, the low resolution recordings may contain errors (such as in the red areas in (c)). LidarBoost with activated confidence weighting (f) can correct for such reconstruction errors. Diebel's MRF method (g) yields oversmoothing on many depth edges and transforms intensity patterns into geometry patterns (e.g. checkerboard).

time. The camera was displaced in-between shots using rotation only, where the maximum displacement from end to end was below 30 pixels for the first and below 15 pixels for the second scene. The SR 3000 records at 176×144 pixel resolution, but we cropped the frames in either case to the region of interest, which for the collection of objects scene (Fig. 5.6a) resulted in a 106×64 frame size, and for the second scene (Fig. 5.7a) in a 126×89 frame size.

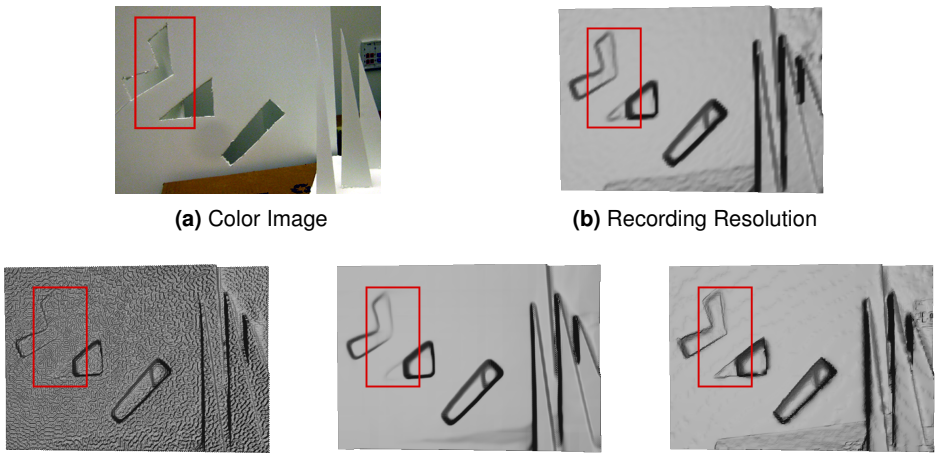
For this scene, the low resolution input (one being shown in Fig. 5.6b) conveys the overall geometry, but fine details such as the small holes of the laundry basket and the cup's

handle are hard to tell. Also, the occlusion edges are very rough and aliased. Furthermore smooth surfaces, such as the ball's or basket's surface are perturbed by noise.

IBSR's reconstruction enhances the fine details, but also introduces the previously discussed staircase pattern. In contrast, LidarBoost (running with $\lambda = 7$) also does feature these details, while yielding a noise free, smooth surface. This result also shows the effectiveness of our amplitude thresholding approach. Parts of the cardboard are painted in black, leading to low reflectivity. Fig. 5.6c shows the amplitude image with measurements below the experimentally determined thresholds being color coded in red. By assigning such pixels a weight of 0 via T_k, LidarBoost reconstructs the true surface (5.6f) of the box. Please also note that two stripes of reflective material on the soccer ball caused slight reconstruction errors since almost no light was reflected to the camera. In this particular case our confidence weighting could not fill the holes since the tiny area of correctly capture depth on the rim pulls the final surface slightly inward. Since we also took a photograph of the real scene, we can also compare to the method by Diebel et al. (Fig. 5.6g) which yields a smooth reconstruction, but struggles with fine details such as the basket's bars, and oversmooths depth edges that don't coincide with intensity edges. Furthermore the method erroneously transforms intensity texture into geometric patterns, in particular in the checkerboard structure on the background and in the pattern on the ball's surface.

Real Scene - Wedges and Panels The second real scene recorded with the Swissranger was purposefully designed to contain wedges with thin fine edges, and many sharp occlusion boundaries (Fig. 5.7a). The same camera settings as in the previous test were used and $N = 15$ low resolution frames were captured. This scene nicely demonstrates the effectiveness of superresolution. While in the low resolution image (Fig. 5.7b), occlusion edges clearly show a staircasing aliasing pattern, both IBSR and LidarBoost recover sharper edges. However, in our previous algorithm's result there is still a little bit of jagginess around the occlusion edges and, as in previous results, there is a strong aliasing pattern in regions of smooth geometry (Fig. 5.7c). In contrast, LidarBoost (with $\lambda = 6$) creates crisp edges with no aliasing, and faithfully recovers smooth areas (Fig. 5.7d). In addition, LidarBoost does a much better job in recovering different depth layers that are visible through the small holes in the left panel (marked in red in Fig. 5.7d).

Diebel et al.'s method does well in recovering the layers, but in contrast to our method exhibits problems on several edges. Many edges on the wedges appear rounded or are still aliased (particularly on the right most wedge).



(a) Color Image **(b)** Recording Resolution

(c) IBSR **(d)** LidarBoost **(e)** Diebel's MRF

Figure 5.7: Real scene - wedges and panels (a): This scene with many depth edges (b) demonstrates the true resolution gain. IBSR (c) demonstrates increased resolution at the edges, but some aliasing remains and the strong pattern in the interior persists. LidarBoost (d) reconstructs the edges much more clearly and there is hardly a trace of aliasing, also the depth layers visible in the red encircled area are better captured. MRF upsampling (e) oversmooths the depth edges and in some places allows the low resolution aliasing to persist.

6 Discussion and Future Work

This work showed that the concept of superresolution can indeed be transferred to 3D data recorded by Time-of-Flight cameras. While the first algorithm proved this is possible (Chapter 4), our second algorithm Lidarboost tried to exploit the specifics of ToF cameras to produce even better results (Chapter 5). Both qualitative and quantitative comparisons demonstrated the gain in resolution as well as overall improved depth data. The core contribution of this work was to the first to demonstrate superresolution on 3D data and develop an algorithm tailored to Time-of-Flight cameras.

While all components are at hand to apply these algorithms, especially for Lidarboost a faster implementation would be advantageous. Since the algorithm was explicitly posed as a convex problem, this should indeed be possible. A first step is the gradient derived which can be used to implement a steepest descent or conjugate gradient method. Apart from favorable convergence properties the computation should be possible with few steps since an initial estimate in the form of the aligned raw inputs is available.

Also the algorithm offers some interesting areas for improvement. Time-of-Flight cameras over a plentiful of raw data in addition to the depth map. These data is automatically available and could be included into the algorithm to better characterize noisy pixels as well as edge pixels (these have a high variance). Possibly the reflectance image could yield such data but also looking at the different phase images with correlation based cameras could be of interest.

Another interesting direction of research would be to include a more mathematical model of the ToF camera errors. It is known that the measurement error is not uniformly distributed around a measurement point (i.e. a sphere) but rather elliptical along the measurement ray. This seems logical as the the viewing direction is fixed but the distance is determined by the measurement process. When intersecting the probability distributions of corresponding points in subsequent recording, the measurement accuracy could be drastically improved.

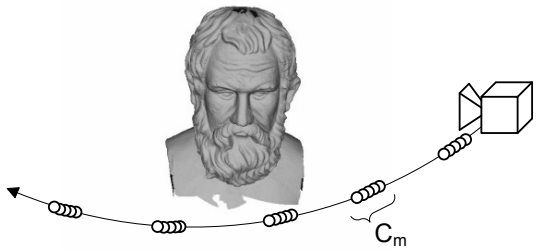


Figure 6.1: Using superresolution to scan 3D models: the dotted segments are the frame chunks C_m from which superresolved depth scans are computed.

One of the motivations we mentioned in the introduction was having a method to create high quality 3D models with a low-cost scanner. Given the improvements superresolution did towards the quality of Time-of-Flight recordings, the following concept is worth investigating: the camera is moved around the object of desire (Figure 6.1 while continuously recording. Then chunks of subsequent frames are used to compute a superresolved recording. If the movement around the object is not too fast (or a sufficiently high frame rate has been chosen) the underlying assumption of small viewpoint changes needed for superresolution is still met. The resulting superresolved scans then need to be aligned and can be fused into a full model of the scanned object. We have outlined this approach in Figure 6.2.

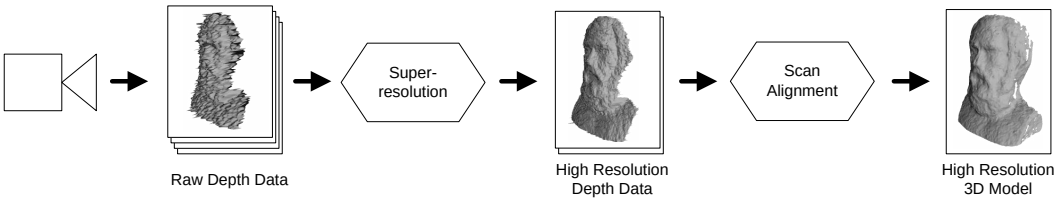


Figure 6.2: Scanning quality 3D models with ToF cameras: superresolution is the key building block

This setup is on-going research, but initial results look very promising (Figure 6.3). An antique head (Figure 6.3a was scanned by moving a Swissranger SR3000 camera around it. The camera was continuously recording scans at approximately $25fps$. A single raw scan is shown in Figure 6.3b. Then ten raw frames each were combined into a single superresolved scan using LidarBoost. These output scans were then aligned using a novel probabilistic alignment algorithm that incorporates ToF specific error sources. In Figure 6.3c we depict the result of the algorithm. In 6.3d we show the laser scanned ground truth. The RMSE comparison with the ground truth (Figure 6.3e) shows the quality of the reconstruction. This furthermore demonstrates the feasibility of our superresolution approach under real world conditions.

(a) Color Image **(b)** Raw Scan **(c)** Proposed Method **(d)** Laser Scan **(e)** Error Plot **(f)** Color Coding Legend

Figure 6.3: Scanning 3D models: antique head (a); computes a 3D model of reasonable quality (c) despite severe errors in the raw ToF data (b). RMSE comparison (e) to a laser scan (d) shows that no circumstance the error was larger than 2.5 cm, while for most of the surface it was below 1.0 cm. (Note: raw aligned scans, no hole filling done)

Bibliography

[1] BOEHME, M., HAKER, M., MARTINETZ, T., AND BARTH, E. Shading constraint improves accuracy of time-of-flight measurements. In *IEEE Conf. on Computer Vision & Pattern Recogn.; Workshop on ToF-Camera based Computer Vision* (2008).

[2] BORMAN, S., AND STEVENSON, R. L. Super-resolution from image sequences - a review. *Proc. Midwest Symp. Circuits and Systems 5* (1998).

[3] BOYD, S., AND VANDENBERGHE, L. *Convex Optimization*. Cambridge University Press, 2004.

[4] BRANT, T., CHAN, S., AND HUANG, R. Developing next generation of 3D game interface. In *Stanford CS223b Class Project* (2008).

[5] CHAN, D., BUISMAN, H., THEOBALT, C., AND THRUN, S. A Noise-Aware Filter for Real-Time Depth Upsampling. In *Proc. of ECCV Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications* (2008), pp. 1–12.

[6] DIEBEL, J., AND THRUN, S. An application of markov random fields to range sensing. In *Advances in Neural Information Processing Systems 18*. MIT Press, 2006, pp. 291–298.

[7] FARSIU, S., ROBINSON, M., ELAD, M., AND MILANFAR, P. Fast and robust multiframe super resolution. *IEEE Transactions on Image Processing 13*, 10 (Oct. 2004), 1327–1344.

[8] FRANK, M., PLAUE, M., RAPP, H., K
"OTHE, U., J
"AHNE, B., AND HAMPRECHT, F. Theoretical and experimental error analysis of continuous-wave time-of-flight range cameras. *Optical Engineering 48* (2009), 013602.

[9] GOKTURK, S., YALCIN, H., BAMJI, C., AND INC, C. A time-of-flight depth sensor-system description, issues and solutions. In *Computer Vision and Pattern Recognition Workshop, 2004 Conference on* (2004), pp. 35–35.

[10] GONZALEZ-BANOS, H., AND DAVIS, J. Computing depth under ambient illumination using multi-shuttered light. *IEEE CVPR 2* (2004).

[11] GRANT, M. C., BOYD, S. P., AND YE, Y. CVX: Matlab Software for Disciplined Convex Programming. *http://stanford.edu/boyd/cvx* (2008).

[12] GVILI, R., KAPLAN, A., OFEK, E., AND YAHAV, G. Depth keying. *SPIE Elec. Imaging 5006* (2003), 564–574.

[13] HARTLEY, R., AND ZISSERMAN, A. *Multiple view geometry in computer vision*. Cambridge Univ Pr, 2003.

[14] IDDAN, G., AND YAHAV, G. 3D imaging in the studio (and elsewhere...). In *Proc. SPIE* (2001), vol. 4298, pp. 48–55.

[15] INC., C. Canesta 101 Introduction to 3D Vision in CMOS. *http://www.canesta.com/assets/pdf/technicalpapers/Canesta101.pdf* (2008).

[16] INC., P. G. R. Bumblebee Product Series. *http://www.ptgrey.com/products/stereo.asp* (2010).

[17] IRANI, M., AND PELEG, S. Improving resolution by image registration. *CVGIP: Graph. Models Image Process. 53*, 3 (1991), 231–239.

[18] JOSHI, M. V., AND CHAUDHURI, S. Simultaneous estimation of super-resolved depth map and intensity field using photometric cue. *CVIU 101*, 1 (2006), 31–44.

[19] JUTZI, B., AND STILLA, U. Precise range estimation on known surfaces by analysis of full-waveform laser. *Proceedings of Phtogrammetric Computer Vision PCV* (2006).

[20] KIL, Y., MEDEROS, B., AND AMENTA, N. Laser scanner super-resolution. *Eurographics Symposium on Point-Based Graphics* (2006).

[21] KIM, Y., CHAN, D., THEOBALT, C., AND THRUN, S. Design and calibration of a multi-view TOF sensor fusion system. In *IEEE Conf. on Computer Vision & Pattern Recogn.; Workshop on ToF-Camera based Computer Vision* (2008).

[22] KIM, Y., CHAN, D., THEOBALT, C., AND THRUN, S. Design and calibration of a multi-view tof sensor fusion system. In *Proc. CVPR Worksh. TOF-CV* (2008), pp. 1–7.

[23] KIM, Y., THEOBALT, C., DIEBEL, J., KOSECKA, J., MISCUSIK, B., AND THRUN, S. Multi-view Image and ToF Sensor Fusion for Dense 3D Reconstruction. *IEEE Workshop on 3-D Digital Imaging and Modeling (3DIM)* (2009).

[24] KOLB, A., BARTH, E., KOCH, R., AND LARSEN, R. Time-of-Flight Sensors in Computer Graphics. *Proceedings of Eurographics (State-of-the-Art Report), Munich, Germany (March 2009)* (2009).

[25] KOPF, J., COHEN, M., LISCHINSKI, D., AND UYTTENDAELE, M. Joint bilateral upsampling. *ACM TOG 26*, 3 (2007).

[26] LANGE, R. 3D time-of-flight distance measurement with custom solid-state image sensors in CMOS/CCD-technology. *Diss., Department of Electrical Engineering and Computer Science, University of Siegen* (2000).

[27] LANGE, R., AND SEITZ, P. Solid-state time-of-flight range camera. *IEEE Journal of Quantum Electronics 37*, 3 (2001), 390–397.

[28] LANMAN, D., AND TAUBIN, G. Build your own 3d scanner: 3d photograhy for beginners. In *SIGGRAPH courses* (2009), ACM, pp. 1–87.

[29] LANMAN, D., AND TAUBIN, G. Build your own 3D scanner: 3D photography for beginners. In *ACM SIGGRAPH 2009 Courses* (2009), ACM, pp. 1–94.

[30] LAURENZIS, M., CHRISTNACHER, F., AND MONNIN, D. Long-range three-dimensional active imaging with superresolution depth mapping. *Opt. Lett. 32*, 21 (2007), 3146–3148.

[31] LINDNER, M., KOLB, A., AND RINGBECK, T. New insights into the calibration of tofsensors. In *CVPR Workshop On Time of Flight Camera based Computer Vision (TOF-CV)* (2008).

[32] LINDNER, M., LAMBERS, M., AND KOLB, A. Sub-pixel data fusion and edge-enhanced distance refinement for 2D/3D images. *International Journal of Intelligent Systems Technologies and Applications 5*, 3 (2008), 344–354.

[33] LUCAS, B., AND KANADE, T. An iterative image registration technique with an application to stereo vision. In *International Joint Conference on Artificial Intelligence* (1981), vol. 3.

[34] MILANFAR, P. MDSP resolution enhancement software. http://www.soe.ucsc.edu/~milanfar/software/superresolution.html, 2004.

[35] OGGIER, T., B
"UTTGEN, B., LUSTENBERGER, F., BECKER, G., R
"UEGG, B., AND HODAC, A. Swissranger SR3000 and first experiences based on miniaturized 3D-TOF cameras. *Proc. of the First Range Imaging Research Day at ETH Zurich* (2005).

[36] OGGIER, T., LEHMANN, M., KAUFMANN, R., SCHWEIZER, M., RICHTER, M., METZLER, P., LANG, G., LUSTENBERGER, F., AND BLANC, N. An all-solid-state optical range camera for 3D real-time imaging with sub-centimeter depth resolution (SwissRanger). In *Proc. SPIE* (2004), vol. 5249, pp. 534–545.

[37] RAJAGOPALAN, A., BHAVSAR, A., WALLHOFF, F., AND RIGOLL, G. Resolution Enhancement of PMD Range Maps. *Lecture Notes in Computer Science 5096* (2008), 304–313.

[38] RAJAN, D., AND CHAUDHURI, S. Simultaneous estimation of super-resolved scene and depth map from low resolution defocused observations. *PAMI 25*, 9 (2003), 1102–1117.

[39] RAPP, H., AND AUSGABE, G. Experimental and theoretical investigation of correlating TOF-camera systems. *University of Heidelberg* (2007).

[40] RINGBECK, T. A 3D TIME OF FLIGHT CAMERA FOR OBJECT DETECTION. *Optical 3-D Measurement Techniques, ETH Zuerich* (2007).

[41] SCHARSTEIN, D., AND SZELISKI, R. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV 47*, 1 (2002), 7–42.

[42] SCHMIDT, M., AND JAHNE, B. A Physical Model of Time-of-Flight 3D Imaging Systems, Including Suppression of Ambient Light. In *Dynamic 3D Imaging: DAGM 2009 Workshop, Dyn3D 2009, Jena, Germany, September 9, 2009, Proceedings* (2009), Springer-Verlag New York Inc, p. 1.

[43] SCHUON, S., THEOBALT, C., DAVIS, J., AND THRUN, S. High-quality scanning using time-of-flight depth superresolution. *CVPR Workshop on Time-of-Flight Computer Vision 2008* (2008).

[44] SCHUON, S., THEOBALT, C., DAVIS, J., AND THRUN, S. Lidarboost: Depth superresolution for tof 3d shape scanning. *In Proc. of IEEE CVPR 2009* (2009).

[45] SCHUON, S., THEOBALT, C., DAVIS, J., AND THRUN, S. Lidarboost: Depth superresolution for tof 3d shape scanning. *Proc. CVPR* (2009).

[46] TOMASI, C., AND MANDUCHI, R. Bilateral filtering for gray and color images. In *ICCV* (1998), pp. 839–846.

[47] TSAI, R., AND HUANG, T. Multiframe image restoration and registration. *Advances in Computer Vision and Image Processing* (1984), 317–339.

[48] TSAI, R., AND HUANG, T. Multiframe image restoration and registration. *Advances in Computer Vision and Image Processing 1*, 2 (1984), 317–339.

[49] XU, Z., SCHWARTE, R., HEINOL, H., BUXBAUM, B., AND RINGBECK, T. Smart pixel-photonic mixer device (pmd) new system concept of a 3d-imaging camera-on-a-chip. In *International Conference on Mechatronics and Machine Vision in Practice, Nanjing, China* (1998), pp. 259–264.

[50] YAHAV, G., IDDAN, G., AND MANDELBOUM, D. 3D imaging camera for gaming application. *Consumer Electronics 2007* (2007), 1–2.

[51] YANG, Q., YANG, R., DAVIS, J., AND NISTÉR, D. Spatial-depth super resolution for range images. In *IEEE Computer Vision and Pattern Recognition* (2007), IEEE Computer Society.

[52] ZHU, J., WANG, L., YANG, R., AND DAVIS, J. Fusion of time-of-flight depth and stereo for high accuracy depth maps. In *Proceedings of CVPR* (2008), Citeseer.