Journal of Electronic Imaging

SPIEDigitalLibrary.org/jei

Image processing for three-dimensional scans generated by time-of-flight range cameras

Holger Schöner
Frank Bauer
Adrian Dorrington
Bettina Heise
Volkmar Wieser
Andrew Payne
Michael J. Cree
Bernhard Moser

Copyright 2012 Society of Photo-Optical Instrumentation Engineers. One print or electronic copy may be made for personal use only. Systematic reproduction and distribution, duplication of any material in this paper for a fee or for commercial purposes, or modification of the content of the paper are prohibited

SPIE

IS&T
imaging.org

Image processing for three-dimensional scans generated by time-of-flight range cameras

Holger Schöner
Software Competence Center Hagenberg
4232 Hagenberg, Austria
E-mail: holger.schoener@scch.at

Frank Bauer
Johannes Kepler University of Linz
Department of Knowledge-based Mathematical Systems
4232 Hagenberg, Austria

Adrian Dorrington
University of Waikato
Department of Engineering
Hamilton 3240, New Zealand

Bettina Heise
Johannes Kepler University of Linz
Department of Knowledge-based Mathematical Systems
4232 Hagenberg, Austria

Volkmar Wieser
Software Competence Center Hagenberg
4232 Hagenberg, Austria

Andrew Payne
Michael J. Cree
University of Waikato
Department of Engineering
Hamilton 3240, New Zealand

Bernhard Moser
Software Competence Center Hagenberg
4232 Hagenberg, Austria

Abstract. *Time-of-flight (TOF) full-field range cameras use a correlative imaging technique to generate three-dimensional measurements of the environment. Though reliable and cheap they have the disadvantage of high measurement noise and errors that limit the practical use of these cameras in industrial applications. We show how some of these limitations can be overcome with standard image processing techniques specially adapted to TOF camera data. Additional information in the multimodal images recorded in this setting, and not available in standard image processing settings, can be used to improve reduction of measurement noise. Three extensions of standard techniques, wavelet thresholding, adaptive smoothing on a clustering based image segmentation, and an extended anisotropic diffusion filtering, make use of this information and are compared on synthetic data and on data acquired from two different off-the-shelf TOF cameras. Of these methods, the adapted anisotropic diffusion technique gives best results, and is implementable to perform in real time using current graphics processing unit (GPU) hardware. Like traditional anisotropic diffusion, it requires some parameter adaptation to the scene characteristics, but allows for low visualization delay and improved visualization of moving objects by avoiding long averaging periods when compared to traditional TOF image denoising. © 2012 SPIE and IS&T.* [DOI: 10.1117/1.JEI.21.2.023012]

1 Introduction

Three-dimensional (3-D) object recognition is a task which is becoming increasingly important in real world applications such as industrial automation and surveillance. One promising technique for acquiring such data is full-field time-of-flight (TOF) range imaging. This technique uses active illumination

Paper 11049 received Mar. 16, 2011; revised manuscript received Mar. 16, 2012; accepted for publication Mar. 28, 2012; published online May 21, 2012.

0091-3286/2012/$25.00 © 2012 SPIE and IS&T

and correlative image sensing to produce an image that contains both distance and intensity information in every pixel.

As range imaging technology advances, affordable, compact, and reliable camera systems are becoming available. However, measurements from these cameras still suffer from high noise levels and undesirable measurement artifacts, which often prohibit their use in many applications. By applying state of the art image processing to range image data, we obtain information more suitable for automatic post-processing and object recognition.

In the following we review image acquisition (Sec. 2.1), the different kinds of noise and artifacts commonly seen in range images (Sec. 2.2), and present denoising techniques especially adapted for these situations (Sec. 2.3). These techniques are first applied to synthetic data for a systematic comparison of their performance and relative strengths and weaknesses (Sec. 3). Additionally, they are applied to data acquired with two different commercially available TOF range cameras, showing significant improvements in image quality also on real recordings (Sec. 4). The conclusions (Sec. 5) wrap up our most relevant findings.

2 Methods

2.1 *Image Acquisition*

2.1.1 *General*

By its nature, TOF imaging is an active system. That is, the scene is illuminated with an intensity modulated light source, typically in the region of 10 to 100 MHz. In addition to amplitude modulated continuous wave (AMCW) TOF imaging there exist other variants as well. As most of the commercially available TOF cameras currently are of the AM-CW type, we have limited our investigations to this type. The propagation time for the illumination to travel to objects in the scene and back to the camera is determined by measuring the resultant phase change in the illumination modulation envelope. This is achieved by gain modulating, or shuttering, an image sensor with the same waveform as that of the light source modulation.[1–4] Specialized image sensors are required because traditional sensors do not have the capability to shutter the sensor at high-speed during the integration period.

Short propagation distances result in a small phase shift, meaning the illumination modulation is approximately in phase with the sensor gain modulation, resulting in a bright value recorded at a corresponding pixel. Longer propagation distances cause the illumination modulation to become out-of-phase with the sensor modulation, resulting in a darker pixel value.

Of course, a single brightness value does not provide a reliable distance estimate because factors such as object color or reflectivity, and background illumination, also affect pixel brightness. To overcome this limitation, four images are normally acquired with the relative phase of the illumination and sensor modulation signals stepped by 90 deg between each image. The phase, φ, amplitude, a, and offset, b, values are independently calculated for each pixel by

$$a = \frac{1}{2}\sqrt{(A_0 - A_2)^2 + (A_1 - A_3)^2}, \tag{1}$$

$$b = \frac{1}{4}(A_0 + A_1 + A_2 + A_3), \tag{2}$$

$$\varphi = \tan^{-1}\left(\frac{A_0 - A_2}{A_1 - A_3}\right), \tag{3}$$

where A_0, A_1, A_2, and A_3 are the pixel brightness values from the four successive images. The object distance d (now called range) is calculated from the phase, modulation frequency, f, and the speed of light, c, with

$$d = \frac{\varphi c}{4\pi f}. \tag{4}$$

2.1.2 *Generated images*

Although it is possible to produce three images containing ambient background intensity (from the offset b), active intensity (from the amplitude a), and distance (derived from the phase φ), range imaging cameras typically only output the range image and the active intensity image $I = a^2$ (simply referred to as "intensity image" henceforth). Because of the cyclic nature of the phase measurement, the distance measurements also exhibit a wrap-around nature, referred to as distance ambiguity or aliasing. This ambiguity is inversely proportional to modulation frequency, so a lower frequency provides a longer unambiguous measurement distance. However, distance measurement precision improves with increasing modulation frequency, so a trade-off must be made depending on the application. Some techniques for resolving distance ambiguity have been published.[5,6] These techniques are based on processing data from two range measurements acquired at different modulation frequencies, but are used in few commercial products.

Most cameras also have the capability to perform a perspective projection to output real-world 3-D locations and intensities for each pixel. This process also includes a lens distortion correction, using calibration data generated during manufacture. Consequently, most range imaging system are shipped with a fixed focal length and fixed focus lens, as the calibration changes if the focal length is adjusted. This also means that these cameras have a fixed field of view.

2.2 *Noise Model*

2.2.1 *Intensity based bias*

The effect of varying materials (different surface properties like color, roughness or reflectivity) at the same range resulting in different range measurements has been observed in a number of different scenes (Fig. 1). According to Eqs. (1)–(3) the brightness image A_k phase-shifted by Δ_k can be expressed as

$$A_k = b + a\cos(\varphi + \Delta_k). \tag{5}$$

As the brightness image always satisfies $A_k \geq 0$, the offset is limited to $b \geq a$, whereas the highest modulation range is achieved for $b = a$, resulting in an intensity dependent offset b proportional to a. In real world applications the background determines the offset with $b \gg a$.

The bias on the range measurement occurs when there is multi-path interference. Each pixel receives a combination of the primary desired return plus other low level interfering multi-path returns. These multi-path components arise

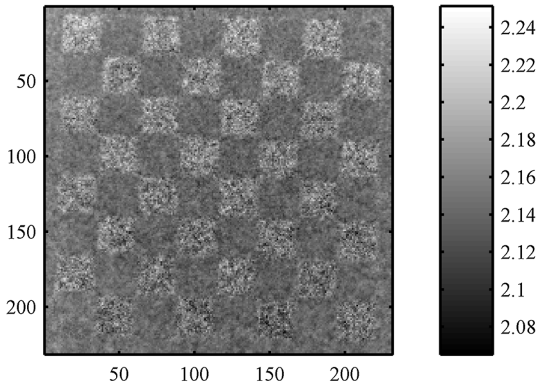


Fig. 1 Range image of a plain wooden checkerboard. Black squares are noisier and appear on average farther away from the camera than white squares. Gray level denotes distance from camera in meters.

from scattering inside the lens and/or multiple reflections from objects in the scene. If the primary return is bright, then the multi-path interference has only a small effect on the measurement. However, if the primary return is of a low intensity, the multi-path interference has a significant effect that manifests as a bias offset.

2.2.2 *Phase noise*

Both intensity and range images show strong noise on the recorded data, with the noise in the range images especially dominant. The latter is denoted phase noise, and assuming approximately Gaussian noise and sufficient amplitude then $n_\varphi \sim \mathcal{N}(\mu_\varphi, \sigma_\varphi^2)$, i.e., Gaussian white noise with mean μ_φ and variance σ_φ^2. The phase noise is influenced by the statistical surface structure (roughness) and brightness (color). Noise in the intensity image, denoted intensity noise, and given by $n_a \sim \mathcal{N}(\mu_a, \sigma_a^2)$, is of higher relevance for shorter illumination times.

Assuming the same statistical distribution for each recorded $A_k(b, \varphi, a)$ with variance $\sigma_{A_k}^2 = \sigma^2$, it can be shown for the variance of a, φ, b, that this statistical relation holds:[7]

$$\mathrm{var}(a, \varphi, b) = (\sigma_a^2, \sigma_\varphi^2, \sigma_b^2) = \left(\frac{1}{2}, \frac{1}{2a^2}, \frac{1}{4}\right)\sigma^2 \qquad (6)$$

Hence, darker regions in the intensity image cause higher phase noise in corresponding regions in the range image (cf. Fig. 1). As σ_φ^2 changes with the reciprocal of the squared amplitude it leads to phase noise varying over several orders of magnitude. However, utilizing the known dependence from the amplitude, the local variance can be estimated and used for error decorrelation and denoising. This estimation is possible because the error on the phase image decorrelates very quickly in time, hence subsequent measurements can be used to compute this quantity.

2.2.3 *Object border noise*

Pixels that image the edge of objects often contain errors in the distance information, termed flying pixel effect. Anomalous distance measurements arise because a particular pixel may receive light from two (or more) objects at different distances. Intuitively, one could expect the distance measurement resulting from such flying pixels to lie somewhere between the actual distances, but this is not always the case. As the modulation envelope of the illumination can be described by a phasor, light collected in one pixel from multiple objects or surfaces in the scene is the sum of the phasors for each return. Consequently, the combined phase (and the resulting anomalous distance measurement) falls between the multiple individual return values only if the phase difference is less than π radians. If the phase difference is larger, then the anomalous distance measurement falls outside the actual objects, and visually appears somewhat random.

Methods for identifying flying pixels that have been reported typically use processed point-cloud data. Relatively simple algorithms of identifying isolated points and median filtering are described by.[8] More sophisticated approaches such as normal-angle filtering, edge-length filtering and the cone algorithm were investigated by,[9] but it was found that generally these algorithms did not perform to a high standard. More advanced methods of detecting and even correcting flying pixels have also been reported. These include decomposing the flying pixels into their distinct components,[10] detecting discontinuities in the returned signal amplitude,[11] and deconvolving the returned signal.[12]

2.3 *Noise Removal*

2.3.1 *Intensity based bias*

We have tried three different approaches for removing the range bias:

- Naïvely estimating the multi-path interference as a homogeneous value across the image and simply subtracting a scaled phasor value prior to calculating distance.
- Separating the primary and multi-path returns by processing two acquisitions of the same scene at different modulation frequencies.
- Adding a bias linearly related to intensity.

However, because the multi-path effect changes when the scene changes we have not been able to automate the process of removing the range bias so far. Therefore we accept in Sec. 4 that there is some bias and just reconstruct this biased range image as well as possible.

2.3.2 *Phase noise*

The ideas for removing noise from images are manifold and include:

- pure isotropic filtering such as Wiener filters[13] or Gaussian filters,[14]
- anisotropic filtering such as Perona-Malik[15] and successors, see e.g.,[14]
- wavelets such as wavelet shrinkage,[16]
- stochastic methods such as median filters,[13]

and many more.

These methods mainly work in a context with noise being rather similar over the whole image. Our case with noise varying over several orders of magnitude is problematic to handle using these methods, especially as all methods above are steered by a global noise dependent parameter that controls the balance between noise removal and image preservation.

Due to this very special structure of the noise some different or specially adapted methods are required. For the stochastically motivated context we refer to Ref. 17. In the next three paragraphs we will concentrate on more numerically motivated methods, namely locally adapted wavelet shrinkage, clustering and adapted anisotropic filtering.

Wavelets. A standard tool for noise removal is wavelet thresholding (also called wavelet shrinkage)[16,18–20] where the smoothed phase is

$$\varphi_s = \sum_{i,j} c_{i,j} W_{i,j}, \tag{7}$$

with $W_{i,j}$ an appropriate wavelet basis,

$$c_{i,j} = \begin{cases} \langle \varphi, W_{i,j} \rangle & \text{for } |\langle \varphi, W_{i,j} \rangle| > \delta \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

and with δ depending on the noise level in the data. For the ease of presentation we restrict ourselves to hard thresholding. Obviously standard wavelet thresholding is not appropriate for largely varying local noise levels.

Reference 21 presents an approach using joint range and intensity images to cluster the scene, for each wavelet scale and orientation. They exploit the fact, that the intensity image is less noisy than the range image, while features (like edges) in one image are usually present in the other as well. This approach improves on standard wavelet thresholding, but suffers from estimation of a uniform noise covariance matrix across the whole image, which is not appropriate for range images (cf. Sec. 2.2.2).

We can alternatively make use of knowledge about the variance of wavelet coefficients obtained from a few successive images. This is possible because the noise decorrelates very quickly in time, cf. Sec. 2.2.2. By application of a small modification to Eq. (9), adapting the threshold level for each wavelet coefficient with the variance (i.e., a local threshold δ) of this particular coefficient, we arrive at the following thresholding for $c_{i,j}$, first shown in Ref. 22:

$$c_{i,j} = \begin{cases} \langle \varphi, W_{i,j} \rangle & \text{for } |\langle \varphi, W_{i,j} \rangle| > \tau \operatorname{var}\{\langle \varphi, W_{i,j} \rangle\} \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

where $\tau \geq 1$ controls the balance between accepted noise and accuracy in the reconstruction. As for all stochastic inspired methods this is prone to outliers, which result in visible artifacts. Although another problem might be the dependence of the method on parameterization (e.g., wavelet basis) to be appropriate for a specific scene, the method is quite robust regarding this according to our experiences. A large advantage is that the method is very fast compared to the methods shown in the next paragraphs. Its denoising quality might be further improved by also using information from intensity images (and thus the local noise level), instead of estimating noise from consecutive range images. The methods presented in the following two paragraphs try to explicitly exploit the dependencies between intensity and range images.

Clustering. Regions of the image with similar range and intensity values usually share the same noise level. We elaborated on this idea,[22] using a clustering approach to first identify regions which are are most probably belonging to the same object and hence have similar noise properties. Then we apply standard Gaussian smoothing inside these regions with parameters adapted to their noise characteristics.

The feature space for the clustering algorithm is 5-dimensional, with one vector

$$\mathbf{d}(x, y) = [\varphi(x, y), I(x, y), \log \hat{\sigma}_\varphi^2(x, y), x, y] \tag{10}$$

per pixel. The noise variance image $\hat{\sigma}_\varphi^2(x, y)$ is estimated from temporally adjacent range images (five in our examples) and slightly smoothed. The pixel position (x, y) is included to obtain spatially relatively compact clusters. The individual components of this vector can be scaled to adjust for characteristics of camera, illumination and scene; in the following we divide the components by their standard deviation and then usually scale by $\mathbf{w} = (0.7, 1, 1, 0.4, 0.4)$.

Several clustering algorithms are available to find regions with similar properties. We use a Mean Shift variant,[23,24] because of its flexibility regarding the number and shape of clusters. Further details of the approach are provided in Ref. 22.

Due to the properties of the Mean Shift algorithm local outliers are automatically put to the surrounding cluster and so do not spoil the results. Although the clustering results seem to be comparably good, there might be clustering algorithms that result in an even better performance.

This algorithm works very well for scenes with relatively homogeneous surfaces and extreme noise levels, but can be adapted to also preserve fine details if the noise level is not too high. It has the additional advantage of providing an initial image segmentation, but is usually too slow for real time application (about 20 s with non-optimized implementation on the stairs scene presented in Sec. 4).

It should be noted that other much faster algorithms which search for similarities in the surrounding like[25,26] would run into severe problems due to the extremely varying noise levels within the range image which results in the fact that the same difference to a neighboring point needs to be interpreted very differently at different locations of the image.

Extended anisotropic diffusion. The idea of clustering in the prior paragraph is to apply a Gaussian kernel with an adapted amount of smoothing on patches which are identified as belonging together. Isotropic Gaussian smoothing can be rewritten in terms of a differential equation, Eq. (11) with constant coefficient D.[14] Increasing amounts of smoothing are realized by propagating Eq. (11) over longer periods (i.e., larger ranges for t).

The idea due to Perona-Malik[15] is to steer the diffusion locally in a specific direction by introducing $D(\cdot, \cdot)$

$$(\partial/\partial t)\varphi = \operatorname{div}(D \nabla \varphi), \tag{11}$$

with boundary condition

$$\varphi(\cdot, \cdot, 0) = \varphi, \tag{12}$$

where D depends on the local derivatives. Popular choices are

$$D = 1/(1 + c\|\nabla\varphi\|^2) \tag{13}$$

and

$$D = \exp(-\|\nabla\varphi\|/c). \tag{14}$$

In order to stabilize the derivatives one normally operates on a slightly smoothed version of the image at this point.

Modifications of this general idea are manifold, the examples[27–31] are by no means complete or representative. Some of these also discuss ideas about how to use vector valued data and how to do some local adaption. Proofs of convergence and stability exist for easy standard cases; however it does not seem to be likely that they carry over easily to our special situation and the extensions outlined below.

Obviously, modifying D in Eq. (13), Eq. (14) by a constant factor in front does nothing but change the speed of the diffusion. Formulated differently, one can steer the amount of smoothing as well in the PDE itself and do not necessarily need to rely just on the amount of time it is performed.

Hence what is common to all variants of this PDE anisotropic diffusion approach is

- D is chosen proportional to the noise level, or
- D is chosen proportional to a “non-edge”-detector, i.e., the less likely we have an edge the bigger is D.

TOF-cameras provide a quite comfortable situation that is different to normal image processing tasks, namely

- the local noise level, that is σ_φ^2, is known *a priori*; as this quantity is very noisy itself we use a smoothed version $S(\sigma_\varphi^2)$. Alternatively one can use $1/a^2$ as remarked above in the discussion of Eq. (6).
- In many real images, material and/or color changes often coincide with range differences; i.e., edges in the intensity image I and the variance σ_φ^2 can be used as additional information to limit range smoothing across different regions of the scene.
- Furthermore one can use an edge detector on $\varphi(\cdot,\cdot,t)$.

In our example we chose the modulus of the gradient as edge detector (we denote this approach as extended anisotropic diffusion, EAD, in the following):

$$D = \frac{1 + c_1 S(\sigma_\varphi^2)}{1 + c_2\|\nabla\varphi\|^2 + c_3\|\nabla I\|^2 + c_4\|\nabla S(\sigma_\varphi^2)\|^2}, \tag{15}$$

where the quantities $1 + c_3\|\nabla I\|^2 + c_4\|\nabla S(\sigma_\varphi^2)\|^2$ and $1 + c_1 S(\sigma_\varphi)$ can be precomputed. The PDE is solved iteratively with a stepsize tuned to maximize speed without running into instabilities. The choice of the constants c_1, c_2, c_3 and c_4 is discussed below. In the case when one also identifies the flying pixels at object borders this information can be easily incorporated in the above calculations as a further term in the denominator.

It is important to note that the values of the parameters c and others greatly affect the performance and time consumption of the method. As we see the same effects as in the standard image denoising case (which is proven and well studied) we think that this new variant performs similarly and the same measures to optimize the parameter setting are successful. Furthermore, the experiments on synthetic data in Sec. 3 show, that an optimization of these parameters for general scene characteristics is feasible.

With an implementation of this method on a GPU, we are able to process a 256×256 pixel image in 30 to 40 ms, allowing application in real time even for larger images (cf. Sec. 4.4).

3 Performance on Synthetic Data

For a systematic comparison of the aforementioned denoising methods we used synthetically generated images. The background is a simple plane with constant distance, possibly overlaid by a pattern of bumps of configurable amplitude, by adding a low frequency 2-D sine wave. Five further planes are then placed at random positions and in random distance with given variance before or behind the background. They are also overlaid by bumps of configurable size, and can have a random tilt (slope) of a given variance and random direction. This is to simulate more realistic nonplanar object surfaces. The intensity of background is set to a relatively low level, while that of the objects is chosen randomly, and noise is added to both range and to a lesser degree to intensity images. Variance of noise is scaled according to the inverse intensity of the respective pixel, appropriate for TOF data [cf. Eq. (6)]. We defined some test scenarios where we varied the presence and strength of bumps, of tilts, and the noise level. Figure 2 shows an example for such a generated scene (range image), with medium bumps (only slightly visible because of the noise) and tilts of the object planes. Table 1 gives a short description for generated scenarios, the achieved denoising quality for each denoising algorithm (measured as the root mean squared error RMSE between the denoised and the noiseless original images), and the average signal to noise ratio (SNR), together with standard deviation. These results were obtained by first optimizing the parameters of each algorithm for five random realizations of the corresponding scenario, and then applying the method with fixed parameters to 50 other random realizations of that scenario (the standard deviations are taken from these 50 test samples). The averaging (and all other denoising methods) had four range and intensity images with independent noise (for each time point and pixel) from each scene realization available. The best

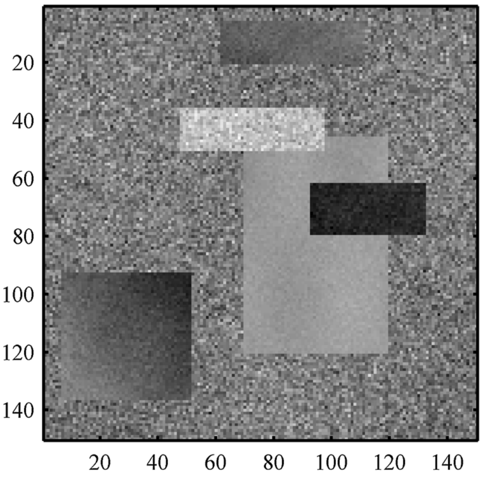


Fig. 2 Example for a range image of a synthetic data example from scene “medium bumps/tilts.”

Table 1 Performance (RMSE and Std. Dev. for 50 realizations) of denoising using standard averaging and the presented EAD, clustering, wavelet methods on synthetic data. Significantly best performing methods for each scene type are highlighted in bold.

Scene characteristics	Averaging	EAD	Clustering	Wavlet	SNR [dB]
No bumps/tilts	0.52 ± 0.061	0.12 ± 0.033	**0.08 ± 0.065**	0.22 ± 0.037	0.11 ± 4.070
Medium bumps/tilts	0.54 ± 0.061	**0.12 ± 0.022**	0.18 ± 0.077	0.24 ± 0.034	2.05 ± 3.066
Medium bumps	0.54 ± 0.076	**0.12 ± 0.024**	**0.12 ± 0.052**	0.24 ± 0.052	1.83 ± 2.686
Strong bumps	0.54 ± 0.074	**0.12 ± 0.037**	0.21 ± 0.069	0.24 ± 0.045	2.02 ± 2.781
Medium tilts	0.54 ± 0.064	**0.12 ± 0.024**	0.19 ± 0.115	0.24 ± 0.036	1.76 ± 3.294
Strong tilts	0.54 ± 0.070	**0.08 ± 0.016**	0.39 ± 0.159	0.26 ± 0.037	4.67 ± 2.455
Very low noise	0.15 ± 0.020	**0.03 ± 0.005**	0.09 ± 0.039	0.07 ± 0.011	11.69 ± 2.336
Low noise	0.32 ± 0.036	**0.05 ± 0.009**	0.10 ± 0.053	0.14 ± 0.022	6.85 ± 3.336
Strong noise	0.78 ± 0.097	**0.18 ± 0.037**	**0.19 ± 0.116**	0.33 ± 0.047	−1.28 ± 3.508
Very strong noise	1.17 ± 0.137	**0.28 ± 0.041**	**0.26 ± 0.164**	0.46 ± 0.098	−4.66 ± 3.219

performing method is printed in bold; statistical significance of performance difference was tested pairwise between algorithms using matched pairs Wilcoxon signed-ranks test[32] with a significance level of 0.05, with Bonferroni correction to account for multiple testing.

It was found, that for such scenes with widely varying levels of noise and other image characteristics, the EAD method overall performed best. It significantly outperforms the other methods, or is among the best for all but one of the considered scene variants. The clustering approach also performs well, as long as the scene does not contain strong slopes. The EAD and wavelet methods are relatively robust, while the clustering approach has a higher variation in denoising quality.

4 Results on Real Data

Images were captured using two different commercially available TOF range cameras. The first camera is a SR4000 manufactured by MESA Imaging. It produces range images with 176 × 144 pixel resolution, and uses 24 near infrared LEDs to illuminate the scene with a typical modulation frequency of 30 MHz. The second camera, a Canesta XZ422, utilizes 20 LEDs operating at 44 MHz, and provides 160 × 120 pixels over a slightly narrower field of view compared to that of the SR4000. Both cameras are capable of producing range images at video (30 Hz) frame rates, making them suitable for a wide variety of applications.

For comparison of the presented noise removal methods, we use two example scenes. The "Stairs" scene in Fig. 3 is recorded with a Canesta XZ422 camera with 0.5 ms integration time per image frame, and shows a scene that consists of differently colored cardboard pieces arranged in a stair-like fashion (row 1: black cardboard; row 2: plain paper black, brown, purple, white; row 3: white glossy paper, plain yellow paper, sand paper, black felt; row 4: plain paper green, orange, blue, red). The dark square with high noise level in the middle is caused by the low IR reflections of sand paper. This scene was designed to test influences of differently reflecting surfaces on noise level and bias in range measurements.

Figure 3 shows the "Lab" scene, which was recorded with a SwissRanger SR4000 camera with 1 ms integration time per frame, as an example of a realistic scene with objects of varying complexity and orientations. A denoising by 3 × 3 adaptive neighborhood and 3 × 3 median filtering was turned on in the camera hardware. The resultant fine structures and low noise levels in one part of the picture in contrast to the very high noise levels in dark parts of the scene pose a significant challenge to denoising algorithms.

We applied the wavelet, the clustering, and the anisotropic diffusion denoising methods to each of these scenes. The clustering method uses the same parameters for both scenes (although these settings need not be appropriate in general for all scenes), while THE WAVELET AND anisotropic diffusion methods had to be adapted to the different camera and illumination characteristics. For wavelets, we use Daubechies wavelets of order 3 with a decomposition level of 3, and thresholds $\tau = 3.0$ ("Stairs" scene), $\tau = 1.8$ ("Lab" scene). For clustering, we use a weighting of the dimensions of $\mathbf{w} = (0.7, 1, 1, 0.4, 0.4)$, together with the approach of smoothing between clusters described by Moser et al.[22]

The presented anisotropic diffusion results use the following constants for weighting edges in the different images (visually optimized): $c_1 = 1000$, $c_2 = 0.002$, $c_3 = 0.0005$, $c_4 = 0.005$ ("Stairs" scene), and $c_1 = 500$, $c_2 = 0.1$, $c_3 = 0.05$, $c_4 = 0.1$ ("Lab" scene). The step size is chosen appropriately to avoid divergences but still sufficient smoothing in 20 iterations. Results improve further with more iterations, but we imposed this constraint to enable realtime application.

For each of the methods and scenes, we show the denoised range images in Fig. 4. Additionally, in Fig. 5 we provide a plot of range versus position along a vertical cut through both scenes, at column 100 for the "Stairs" scene, and at column 110 for the "Lab" scene. These plots show the 5-frame-averaged ranges, the clustering denoised ranges, and the anisotropic diffusion denoised ranges, for direct comparison.

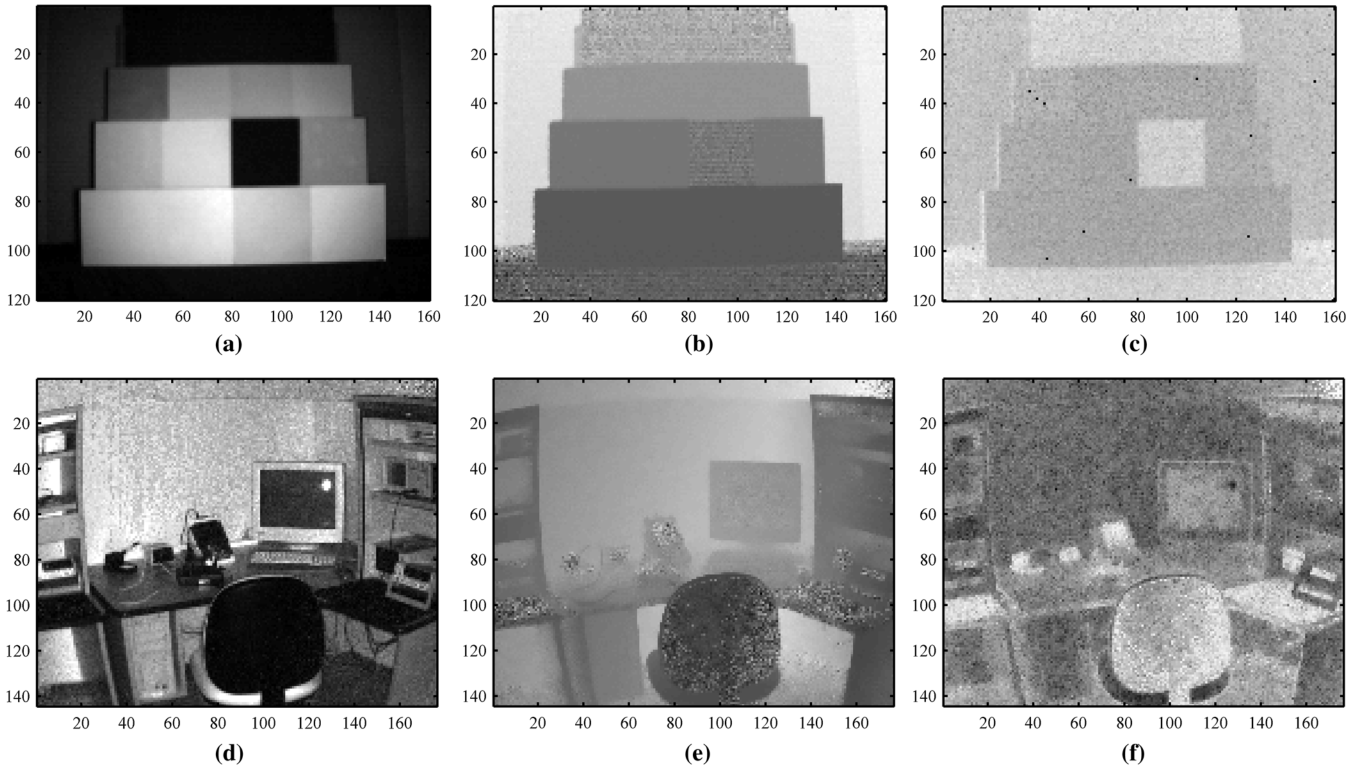


Fig. 3 Left: "Stairs" scene recorded with Canesta XZ422 camera with 0.5 ms recording time per frame; (a) IR-intensity image averaged over five consecutively recorded frames; (b) range image averaged over five frames; (c) logarithm of noise image (variance) estimated from five frames of the scene. Right: "Lab" scene recorded with SR4000 camera with 1.0 ms.

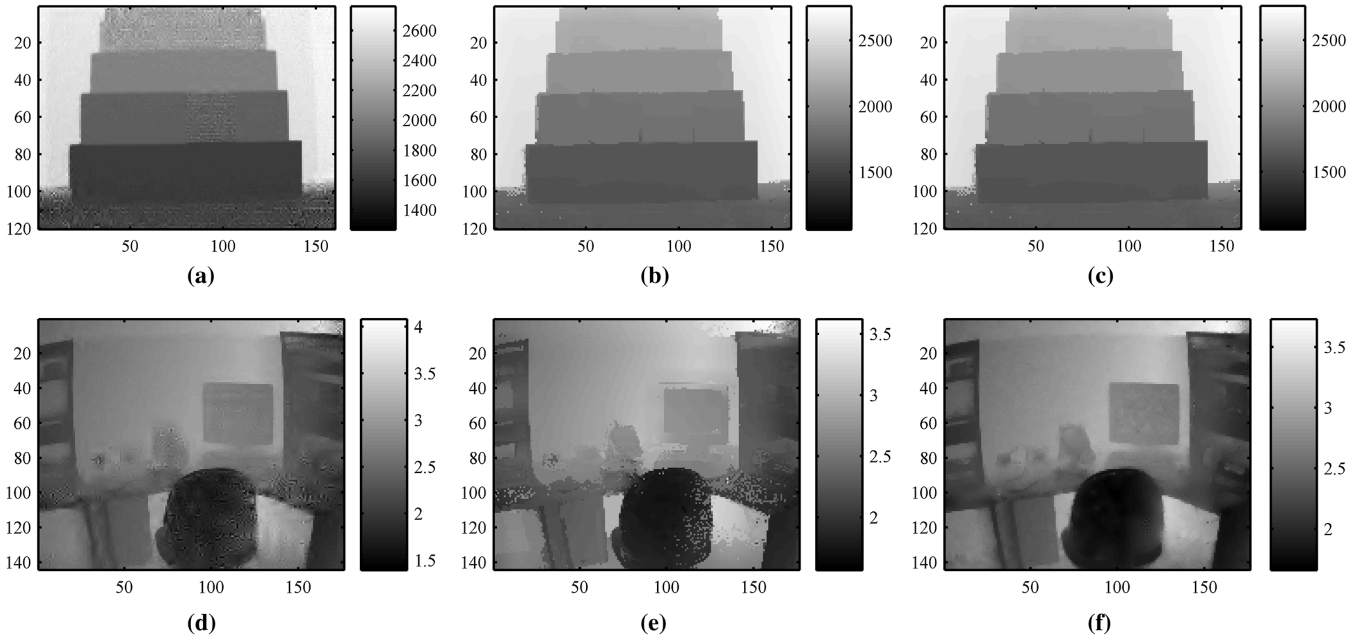


Fig. 4 Left: denoising results for "Stairs" scene, using (a) wavelet, (b) clustering and (c) ext. anisotropic diffusion methods. Gray levels are according to range, colorbar units are in mm. Right: denoising results for "Lab" scene, (d) using wavelet, (e) clustering and (f) ext. anisotropic diffusion methods. Gray levels are according to range, colorbar units are in m.

4.1 *Wavelets*

Wavelet denoising as previously described [22] can be a simple to use and robust method to reduce the level of noise in a range image. It does not completely smooth flat surfaces, and when used aggressively it can introduce some smaller artifacts (as shown in the results for the "Stairs" and "Lab" scenes, Fig. 4). But, contrary to the other two presented methods, wavelet smoothing (as used by us) does not yet make use of any additional information present in the intensity image.

4.2 *Clustering*

In our experience, this method works very well for scenes with relatively smooth surfaces, even with very high levels of noise. This can be seen in the "Stairs" scene (Fig. 4), where the noise in the sandpaper square in the middle is leveled out very well. It can also preserve fine details very well, if the noise levels are not too high.

Figure 4 shows that it is hard to adapt the method (in its current state) to scenes which contain fine details with low noise levels and smooth surfaces with high levels of noise at the same time. Here we could only find a suboptimal compromise for each of these scene parts. This is mainly due to the use of the noisy range images for determining the clusters. In regions with low noise, this information is important and reliable, while in regions of high noise, the importance of the range dimension in the clustering feature space has to be lowered. The scaling of that dimension thus would have to depend on the local noise level, which is not the case, yet.

For many applications, an initial segmentation of the image into objects or parts of objects can be useful. Figure 6 on the left shows the clusters found by the clustering method on the original five raw frames. These are hardly usable as a starting point for segmentation or object recognition. On the other hand, if the clustering is applied to the anisotropic diffusion smoothed range image, as shown in Fig. 6 on the right, then this segmentation starts to be useful, even for previously rather highly noisy regions (or images).

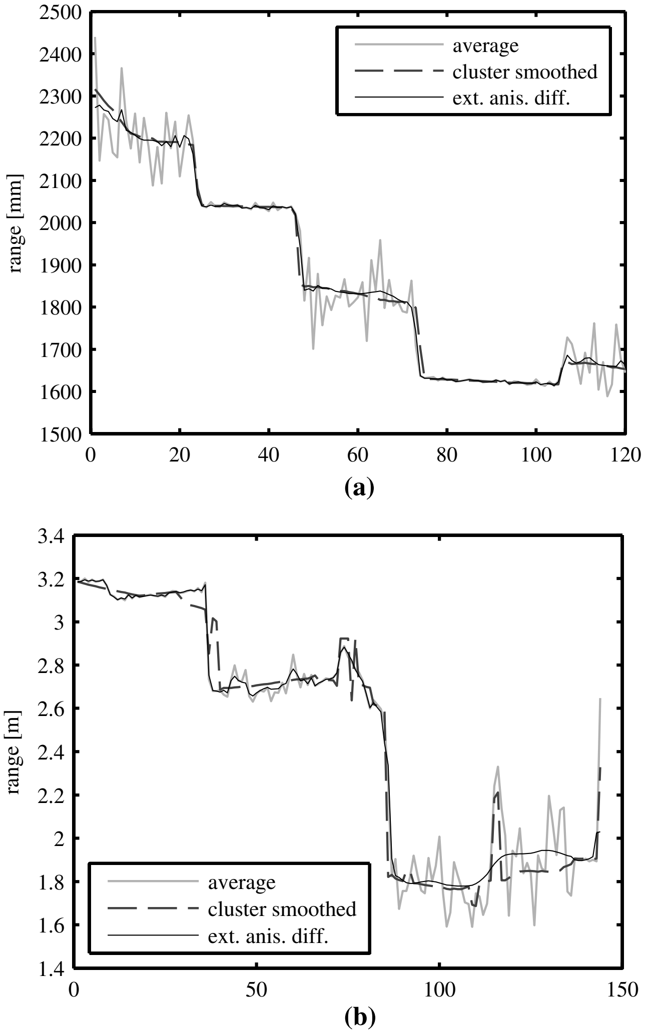


Fig. 5 Plots of cuts through the range images at (a) column 100 ("Stairs" scene) and (b) column 110 ("Lab" scene), for ranges estimated by 5-frame-averaging, cluster smoothing, and ext. anisotropic diffusion. The horizontal axis is along the row indices of the range images.

4.3 *Anisotropic Diffusion*

The results in Fig. 4 are tuned for speed, using only 20 iterations. But already in the presented state they appear to be the best in several respects. Edges are preserved almost as well as by the clustering method (cf. Fig. 5), while at the same time it is able to preserve details in low noise regions (e.g., frame around monitor, shelves on the sides in "Lab" scene; borders of cardboard background on side of stairs in "Stairs" scene), and providing good and realistic smoothing results in high noise regions (back rest of chair). With the few iterations, some blotches can remain in high noise regions (uppermost black cardboard stripe in "Stairs" scene). Also, round or tilted surfaces are better preserved with this method than with the clustering method. An example is the tilted table/keyboard surface between the chair and the monitor in the "Lab" scene, about rows 75 to 85 in the cut in Fig. 5.

The main issue remaining for practical application seems to be computation speed and the adaptation of parameters to camera, illumination and scenes. It remains to be evaluated,

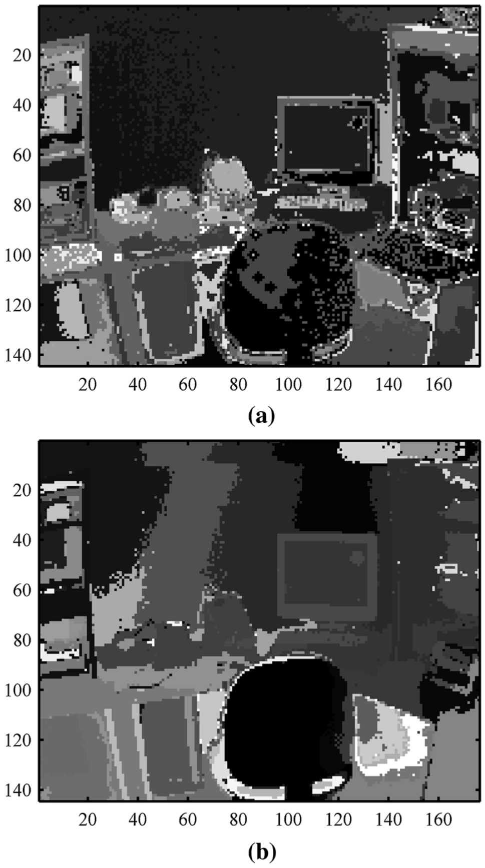


Fig. 6 Clusters found by the clustering method (a) on the original 5 raw frames, and (b) on the smoothed range image returned by the anisotropic diffusion method. For the latter, the scaling of the dimensions for clustering was adapted to $\mathbf{w} = (1.5, 0.5, 0.5, 0.4, 0.4)$.

Table 2 Elapsed time for different EAD implementations.

Imagesize		256 × 256	512 × 512	1024 × 1024
DELL/8800 Ultra	OpenCV [ms]	344	1359	6453
	GPU/CUDA [ms]	32	37	50
	GPU/SAC [ms]	34	116	389
SONY/GT 425M	GPU/CUDA [ms]	77	195	2567
	GPU/SAC [ms]	64	217	2942

whether they can be set in advance using only general knowledge about these characteristics. The results on synthetic data and our experience indicate, that the denoising performance is very stable for a given parameter setting in all scenes with similar noise properties, as defined in Sec. 3, or during one recording of a real scene. In practice, we expect that a few optimized settings for a given camera are sufficient. These should depend on easily detectable scene properties like intensity image magnitude (i.e., range image noise level), and its variation across the image (ratio between minimal and maximal intensity). Keeping this in mind, the same techniques as used for optimizing parameters for standard anisotropic diffusion should be successful, as the problem structure is very similar.

4.4 *Acceleration by GPU Programming*

The anisotropic diffusion algorithm makes heavy use of basic algebraic computations, which are well optimizable using parallel computing. The runtime depends heavily on the hardware platform used, and on the framework/language. We have evaluated two approaches for runtime optimization: manual optimization using code written directly for Nvidia's CUDA framework, and auto-parallelization tool support using the language "Single Assignment C" (SAC).[33] When using CUDA, the developer has full control of the used parallelization concepts but needs special expertise in this domain. The language SAC on the other hand offers a competitive alternative for GPU programming by means of understandable syntax (thus enabling rapid prototyping) and flexible application on heterogeneous hardware platforms.

To demonstrate the feasibility of real-time use of the anisotropic diffusion variant, we use two different hardware environments for our test scenario, (1) a DELL Precision™ 690 and (2) a SONY VAIO™ PCG-81112M laptop. The former uses a NVIDIA GeForce 8800 Ultra graphic card, whereas the latter uses the newer NVIDIA GeForce GT 425M. The NVIDIA GeForce 8800 Ultra has 128 streaming processors with a core frequency of 612 MHz, memory frequency of 1080 MHz, 786 MB memory and a memory bandwidth of 103.7 GB/s, whereas the NVIDIA GeForce GTX 425M has 96 streaming processors with a core frequency of 1120 MHz, memory frequency 800 MHz, up to 1024 MB memory and memory bandwidth of 25.6 GB/s. For comparison, these are compared to an implementation using OpenCV (on the DELL machine). The average runtime is shown in Table 2.

Summarizing, GPU acceleration can give a speedup of factors between 10 and about 1000. Depending on the architecture, SAC can compare favorably with manually optimized code, while allowing quicker implementation. SAC is optimized for the newer Fermi architecture used in the GT 425M. Regardless of the type of GPU implementation, the execution times allow real time application of the algorithm for images with realistic image resolutions.

5 Conclusions

We have demonstrated that it is possible to significantly improve denoising of TOF images using adapted standard denoising techniques. Especially the use of additional information in images accompanying the range image, namely the intensity and noise level images, is very helpful. Significance was shown for synthetic scenes of differing characteristics, while the improvement for realistic scenes has been discussed and shown visually. The presented new extended anisotropic diffusion method has generally been shown to perform best, regarding robustness over different types of scenes, regarding RMSE error wrt. the noiseless image, and regarding visual appearance. It is implementable for real time processing. The choice of parameters can remain constant for given characteristics of scenes. A priori choice (instead of an optimization) was not considered in this paper, but the same problems and approaches as for traditional anisotropic diffusion should apply. The clustering based approach has been shown to be very useful for highly noisy and/or relatively smooth scene types. The wavelet method is not as powerful, but easily tuneable and very fast. Our version does not yet use the additional information from the intensity or noise level images, but see Ref. 21

We have limited the investigations to TOF images of AMCW type, because of the commercial availability of such cameras. We do expect an improvement over traditional denoising methods also for other TOF imaging types, and also for other imaging domains, as long as there is multimodal information with intermodal correlations available, which can be used to determine regions of similar noise properties in the image to denoise.

Manifold applications for denoised TOF images can be found in robotics, surveillance and other industrial applications. Based on denoised range images we can easily and reliably segment the scenes; using GPU hardware we are able to speed up the process for real time computation and inline process control.

TOF as a cheap and robust range imaging technique can be used as an alternative or to supplement stereo imaging in 3-D object recognition tasks. The presented error and artifact removal leads to a major improvement in accuracy and interpretability.

Acknowledgments

The authors from FLLL gratefully acknowledge the financial support by the Upper Austrian Technology and Research Promotion. The SCCH gratefully acknowledges funding of part of this work by the Austrian COMET program.

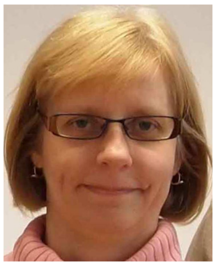
References

1. D. Carnegie, M. Cree, and A. Dorrington, "A high-resolution full-field imaging system," *Rev. Sci. Instrum.* **76**(8), 083702 (2005).
2. A. Dorrington et al., "Achieving sub-millimetre precision with a solid-state full-field heterodyning range imaging camera," *Meas. Sci. Technol.* **18**(9), 2809–2816 (2007).
3. J. P. Anthes et al., "Nonscanned ladar imaging and applications," G. W. Kamerman and W. E. Keicher, Eds., *Proc. SPIE*, **1936**(1), 11–22 (1993).
4. R. Lange and P. Seitz, "Solid-state time-of-flight range camera," *IEEE J. Quant. Electron.* **37**(3), 390–397 (2001).
5. B. Buttgen and P. Seitz, "Robust optical time-of-flight range imaging based on smart pixel structures," *IEEE Trans. Circuits Syst. I Reg. Papers* **55**(6), 1512–1525 (2008).
6. A. Payne et al., "Multiple frequency range imaging to remove measurement ambiguity," in *Optical 3-D Measurement Techniques IX* **2**, 139–148 (2009).
7. M. Frank et al., "Theoretical and experimental error analysis of continuous-wave time-of-flight range cameras," *Opt. Eng.* **48**(1), (2009).
8. M. Hebert and E. Krotkov, "3d measurements from imaging laser radars: how good are they?" in *IEEE/RSJ International Workshop on Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proc. IROS'91*, pp. 359–364 (1991).
9. P. Tang, D. Huber, and B. Akinci, "A comparative analysis of depth-discontinuity and mixed-pixel detection algorithms," in *3-D Digital Imaging and Modeling, 2007. 3DIM'07. Sixth International Conf. on*, pp. 29–38 (2007).
10. M. D. Adams and P. J. Probert, "The interpretation of phase and intensity data from AMCW light detection sensors for reliable ranging," *Int. J. Robot. Res.* **15**(5), 441 (1996).
11. M. S. K. Gebbinck and T. E. Schouten, "Decomposition of mixed pixels," J. Desachy, Ed., *Proc. SPIE*, **2579**, 104–115 (1995).
12. J. P. Godbaz, M. J. Cree, and A. A. Dorrington, "Mixed pixel return separation for a full-field ranger," in *Image and Vision Computing New Zealand. IVCNZ 2008. 23rd International Conf.*, pp. 1–6 (2008).
13. J. S. Lim, *Two-Dimensional Signal and Image Processing*, Prentice Hall, Englewood Cliffs (1990).
14. G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing*, 2nd ed., ser. Applied Mathematical Sciences, **147** Springer, New York (2006).
15. P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.* **12**(7), 629–639 (1990).
16. S. Mallat, *A Wavelet Tour of Signal Processing*, 3rd ed., Elsevier/Academic Press, Amsterdam, the sparse way, With contributions from Gabriel Peyré (2009).
17. M. Frank, M. Plaue, and F. A. Hamprecht, "Denoising of continuous-wave time-of-flight depth images using confidence measures," *Opt. Eng.* **48**(7), (2009).
18. D. L. Donoho and I. M. Johnstone, "Ideal spatial adaptation by wavelet shrinkage." *Biometrika* **81**(3), 425–455 (1994).
19. D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage." *J. Am. Stat. Assoc.* **90**(432), 1200–1224 (1995).
20. I. Daubechies and G. Teschke, "Variational image restoration by means of wavelets: simultaneous decomposition, deblurring, and denoising," *Applied and Computational Harmonic Analysis. Time-Frequency and Time-Scale Analysis, Wavelets, Numerical Algorithms, and Applications* **19**(1), 1–16 (2005).
21. L. Jovanov et al., "Content adaptive wavelet based method for joint denoising of depth and luminance images," *Proc. SPIE*, **6763**, 67630D (2007).
22. B. Moser et al., "Denoising techniques for raw 3D data of tof cameras based on clustering and wavelets," B. D. Corner, M. Mochimaru, and R. Sitnik, Eds., *Proc. SPIE*, **6805**, 68050E (2008).
23. D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(5), (2002).
24. K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function with applications in pattern recognition," *IEEE Trans. Inform. Theor.* **21**(1), 32–40 (1975).
25. A. Buades, B. Coll, and J. M. Morel, "On image denoising methods," CMLA, Tech. Rep. 2004-15 (2004).
26. K. Tabelow et al., "Diffusion tensor imaging: structural adaptive smoothing," *NeuroImage* **39**(4), 1763–1773 (2008).
27. Y. Wang, L. Zhang, and P. Li, "Local variance-controlled forward-and-backward diffusion for image enhancement and noise reduction," *IEEE Trans. Image Process.* **16**(7), 1854–1864 (2007).
28. H. Luo, L. Zhu, and H. Ding, "Coupled anisotropic diffusion for image selective smoothing," *Signal Process.*, **86**(7), 1728–1736 (2006).
29. A. Roussos and P. Maragos, "Vector-valued image interpolation by an anisotropic diffusion-projection PDE," in *Proc. 1st Intl Conf. in Scale Space and Variational Methods in Computer Vision (SSVM-2007), Ischia, Italy, May–June 2007; Lecture Notes in Computer Science 4485*, pp. 104–115, Springer, Berlin, Heidelberg (2007).
30. Y.-C. Song and D.-H. Choi, "Scale-based image enhancement using modified anisotropic diffusion filter," *Opt. Eng.* **43**(9), 2094–2099 (2004).
31. Y. Dong and M. Hintermüller, "Multi-scale total variation with automated regularization parameter selection for color image restoration," in *SSVM '09: Proc. of the Second International Conf. on Scale Space and Variational Methods in Computer Vision*, Springer-Verlag, Berlin, Heidelberg, pp. 271–281 (2009).
32. D. Sheskin, *Parametric and Nonparametric Statistical Procedures*. Chapman & Hall/CRC, Boca Raton, London, New York, Washington, D.C.(2004).
33. C. Grelck and S.-B. Scholz, "SAC: a functional array language for efficient multithreaded execution," *Int. J. Parallel Program.* **34**(4), 383–427 (2006).

Holger Schöner received MS degree in computer science from University of Colorado at Boulder in 1998 and from the Technical University of Berlin in 1999. In 2005, he finished his PhD in computer science at Technical University of Berlin. He is working on and researching applied problems in data mining, machine learning, and image processing at the Software Competence Center, Hagenberg.

Frank Bauer received his MS degree in mathematics from the University of Warwick, United Kingdom in 2001; in 2004 he received his PhD in mathematics from the Technical University of Kaiserslautern, Germany. After two years as a post-doctoral researcher at the University of Göttingen he worked at the Johannes Kepler University Linz, in the Department of Knowledge-based Mathematical Systems (FLLL), on inverse problems and image processing. He has been working in the financial industry for the past two years.

Adrian Dorrington was awarded his PhD in 2001 from the University of Waikato, Hamilton, New Zealand. He has held post-doctoral fellowships from the National Research Council at the NASA Langley Research Center, Hampton, Virginia, and from the Foundation for Research Science and Technology (New Zealand) at the University of Waikato. Currently, he is a senior lecturer in the Department of Engineering at the University of Waikato. His research interests include optoelectronics and optical measurement technologies, and in particular, three-dimensional imaging techniques.



Bettina Heise obtained her Dipl Phys degree at the Technical University of Chemnitz, Germany and her PhD at the Johannes Kepler University. She continued her research in the field of image processing and optics in Linz and is currently a member both at the Department of Knowledge-based Mathematical Systems (FLLL) and at CDL MS-MACH at Johannes Kepler University. Her main research interests are related to 3D-imaging techniques and signal processing.

Volkmar Wieser obtained his MS degree in computer science from Johannes Kepler University of Linz in 2006. He is now working on his PhD in computer science at the Software Competence Center, Hagenberg, and researching on applied problems in image processing, texture analyzing, and parallelizing such methods on graphic cards.

Andrew Payne received his MS degrees in 2004 and PhD in physics in 2009 from The University of Waikato, Hamilton, New Zealand. After a three year post-doctoral position in the Department of Engineering at The University of Waikato, he is currently working at Microsoft California, developing natural user interfaces. His research interests include optoelectronics, signal processing, and working with embedded controllers and sensors.

Michael Cree received his PhD in electrical and electronic engineering at the University of Canterbury, New Zealand in 1994. After almost three years as a researcher at the University of Aberdeen, Scotland, he took up a position at the University of Waikato, New Zealand, where he is now senior lecturer in the Department of Engineering. His research interests include medical imaging, range imaging, and image processing.

Bernhard Moser obtained his MS degree at the University of Salzburg and his PhD in mathematics at the University of Linz, Austria. His main interests are in theory and application of similarity measures, machine learning and computer vision. Currently, he is head of the Knowledge-Based Technology area at the Software Competence Center Hagenberg.