B. Moser ; F. Bauer ; P. Elbau ; B. Heise and H. Schöner, "Denoising techniques for raw 3D data of TOF cameras based on clustering and wavelets", Proc. SPIE 6805, Three-Dimensional Image Capture and Applications 2008, 68050E (February 25, 2008); doi:10.1117/12.765541; http://dx.doi.org/10.1117/12.765541

Denoising techniques for raw 3D data of TOF cameras based on clustering and wavelets

B. Moser[2], F. Bauer[1], P. Elbau[3], B. Heise[1], H. Schöner[2]

[1]Department of Knowledge-Based Mathematical Systems, University of Linz, Softwarepark 21, A-4232 Hagenberg, Austria;
[2]Software Competence Center Hagenberg, Softwarepark 21, A-4232 Hagenberg, Austria;
[3]Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences, Altenbergerstraße 69, A-4040 Linz, Austria

ABSTRACT

In order to measure the 3D structure of a number of objects a comparably new technique in computer vision exists, namely time of flight (TOF) cameras. The overall principle is rather easy and has been applied using sound or light for a long time in all kind of sonar and lidar systems. However in this approach one uses modulated light waves and receives the signals by a parallel pixel array structure. Out of the travelling time at each pixel one can estimate the depth structure of a distant object. The technique requires measuring the intensity differences and ratios of several pictures with extremely high accuracy; therefore one faces in practice rather high noise levels. Object features as reflectance and roughness influence the measurement results. This leads to partly high noise levels with variances dependent on the illumination and material parameters. It can be shown that a reciprocal relation between the variance of the phase and the squared amplitude of the signals exists. On the other hand, objects can be distinguished using these dependencies on surface characteristics. It is shown that based on local variances assigned to separated objects appropriate denoising can be performed based on Wavelets and edge-preserving smoothing methods.

Keywords: TOF cameras, wavelet denoising, mean shift clustering, multisensor based smoothing

1. TOF CAMERAS

In the following, we give an introduction to the principles and issues of TOF cameras (section 1). We then give details on the two denoising approaches evaluated by us, namely Wavelet Thresholding (section 2) and Mean-Shift clustering combined with cluster boundary preserving smoothing (section 3). We give examples of applying the algorithms to real images taken by a TOF-camera (section 4), and conclude with a discussion of the approaches and possible further improvements (section 5).

1.1 Technical background

The considered camera type is a Swiss Ranger Camera SR 3000,[1,2] shortly denoted as TOF camera in this paper. The Time-of-Flight (TOF) camera is a system capable for recording 3D information of the object. It allows measuring both intensity images and depth (phase) images simultaneously. Hence by combination, 3D information on the objects can be gathered. The intensity imaging is realized by standard CCD/CMOS imaging technique, the phase information is extracted by a "time of flight" technique, which is based on measuring the time of arrival of the optical signal back reflected from the measured 3D object by a correlative approach. The optical signal is amplitude modulated with a modulation frequency $f_M = 20$ MHz. Each pixel element of the sensor is designed for independent measurement. The time of arrival t_A yields a measure for the phase ϕ, which is proportional to the distance z.

$$\phi = \frac{4\pi f_M}{c} z \tag{1}$$

Further author information: (Send correspondence to B. Moser)
F. Bauer: +43 (0)7236 3343 432, frank.bauer@jku.at; P. Elbau: +43 (0)732 2468 5244, peter.elbau@oeaw.ac.at;
B. Heise: +43 (0)7236 3343 436, bettina.heise@jku.at; B. Moser: +43 (0)7236 3343 833, bernhard.moser@scch.at;
H. Schöner: +43 (0)7236 3343 816, holger.schoener@scch.at

1.1.1 Measurement principle of the TOF camera

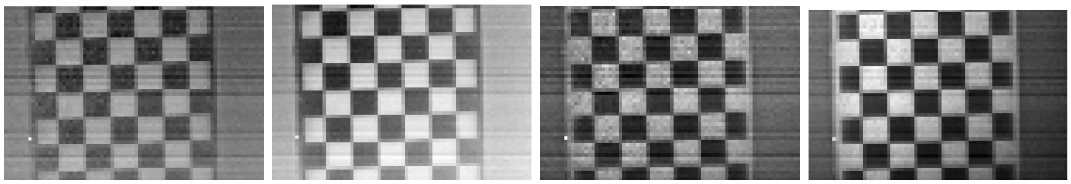

Figure 1. Raw-data images $\mathbf{r}(\tau_1), \ldots, \mathbf{r}(\tau_4)$

TOF based imaging is a correlative measuring technique evaluating the cross correlation $r(\tau)$ between the optical signal $s(t)$ and the demodulation signal $g(t)$. The camera records four phase delayed RAW-data images $\mathbf{r}(\tau_1), \ldots, \mathbf{r}(\tau_4)$. Within the 3D camera system the depth map related to the signal phase $\boldsymbol{\phi}$ and a grayscale map related to the signal amplitude $\mathbf{A}$, are determined from the raw data. Additionally, the shifted raw data images can be accessed directly from the camera. By means of this the calculations can be performed independently from the standard camera output. However, standard camera preprocessing additionally includes corrections for lens distortions and wave front corrections.

Between the pixels of the raw data images $\mathbf{r}(\tau_i)$ with $i \in \{1, \ldots, 4\}$ (figure 1), the intensity image $\mathbf{I} = \mathbf{A}^2$ (figure 2 on the left), and the depth image $\boldsymbol{\phi}$ (figure 2 in the middle) the following relations exist assuming a sinusoidal modulation for each pixel:[3]

$$r(\tau_i) = s(t) * g(t) = \frac{1}{T} \int_{-T/2}^{T/2} s(t) \cdot g(t + \tau_i) dt \tag{2}$$

$$r(\tau_i) = B + A \cos(\phi + \Delta\phi_i) \tag{3}$$

with $\Delta\phi_i = 2\pi f_M \cdot \tau_i = \frac{2\pi(i-1)}{4}$.

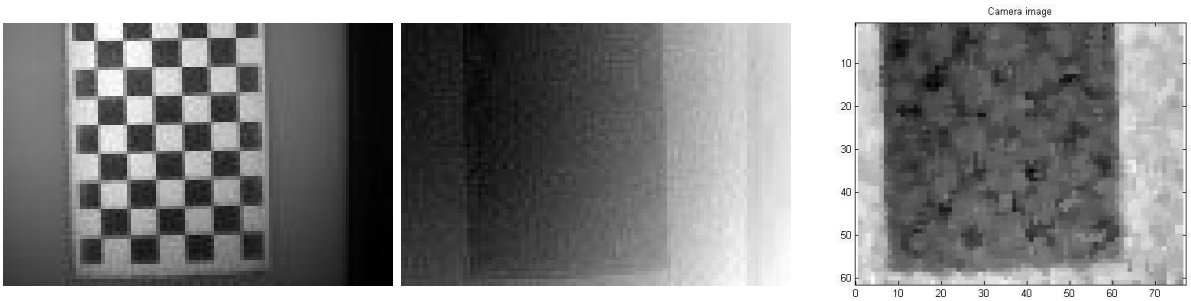

Figure 2. **Left:** intensity image $\mathbf{I}$, **middle:** depth image $\boldsymbol{\phi}$ from raw data, **right:** depth image from camera.

Recording four phase shifted images the offset B, the modulation A and the phase ϕ of each pixel can be determined by

$$A = \frac{\sqrt{(r(\tau_4) - r(\tau_2))^2 + (r(\tau_3) - r(\tau_1))^2}}{2} \tag{4}$$

$$B = \frac{r(\tau_1) + r(\tau_2) + r(\tau_3) + r(\tau_4)}{4} \tag{5}$$

$$\phi = \arctan\left[\frac{r(\tau_4) - r(\tau_2)}{r(\tau_3) - r(\tau_1)}\right] \tag{6}$$

In the depth image, computed from the raw data, no checkerboard pattern is to be seen, see figure 2 in the middle. However, a slight crosstalk between amplitude and phase can be recognized in the depth image provided directly by the camera, figure 2 on the right. For appropriate interpretation of depth information, additionally

correction for inhomogeneous illumination has to be performed. In simple cases, a slight improvement can be achieved by removing a linear trend in data due to unbalanced illumination. However, illumination correction needs deeper investigations and has to be handled carefully.

1.1.2 Comparison of TOF camera and Interferometry

The underlying principles of most of the 3D sensors used for shape or distance measurement can be divided into 1) triangulation 2) time-of-flight and 3) interferometric techniques. According to the chosen measuring principle different scales of the distance z and uncertainty ranges δz can be achieved. The interesting parameters in 3D-measurement are the depth measurement range Δz and the incremental depth resolution. The measuring uncertainty δz of the three techniques scales to[4] 1) $\delta z \propto z^2$, 2) $\delta z \propto z^0$, 3) $\delta z \propto z^{-1}$.

Classical interferometry is able to reach a resolution in the micrometer up to nanometer range with a lowest absolute measuring uncertainty between $\lambda/10$ and $\lambda/100$. On the other hand the precise interferometric setup includes higher costs and calibration efforts. TOF cameras are a price efficient solution with a depth range of several meters determined by the range of unambiguousness for the demodulation, and an incremental resolution in the mm and cm range. On the other hand in practice the resolution often does not reach the accuracy due to instabilities and high sensitivity to phase fluctuations and temperature variations. This can be slightly improved by averaging and additional image processing filters. According to the chosen 3D-application a trade-off between measuring uncertainty, range and costs has to be found.

1.2 Denoising methods

In comparison to conventional CCD cameras where the read out noise and illumination artifacts are mainly responsible for image degradation, for TOF imaging object features as reflectance and roughness influence the measurement results. The noise here can be considered as a kind of speckle noise due to phase fluctuations at each receiver element, sometimes combined with specular reflection artifacts in case of shiny materials. This leads to partly high noise levels with variances dependent on the illumination and material parameters. It can be shown that a reciprocal relation between the variance of the phase and the squared amplitude of the signals exists,[5] $\mathrm{var}(\phi) \propto 1/A^2$.

On the other hand, objects can be distinguished using these dependencies on surfaces characteristics. Based on local variances assigned to the separated objects an appropriate denoising can be performed.[6] Although it is a very rough generalization, we assume that the noise can be described as Gaussian White Noise, because for this noise type a collection of standard denoising methods exist. We focused our interest on wavelet based denoising and cluster based denoising, which are described in the following two sections, and compared them to standard camera preprocessing (using a median filter).

2. ADAPTIVE WAVELET THRESHOLDING

Some artifacts of our Swiss Ranger SR 3000 TOF camera can already be removed reliably by the standard processing done by the accompanying software. Among these are geometrical distortions (removed by calibration), and faulty depth information at single pixels (removed by median filtering). Left over is a region-dependent pixel noise, which can reach relatively high levels in border regions and for objects of certain material and color.

In the following we describe a modification of the well-known wavelet thresholding algorithm which has been especially adapted to fit into the TOF-camera regime with the space dependent change of the noise level.

2.1 Standard Algorithms

As long as each pixel of the depth image $\overline{\phi}(n)$ with image number n (out of N images altogether) can be described by

$$\overline{\phi}(n) = \phi + \delta e(n),$$

where ϕ is the true image and $\delta e(n)$ is Gaussian white noise of variance δ, one can perform a rather easy algorithm. As white noise transforms via the wavelet transform to white noise in the wavelet coefficients (actually this holds for any orthonormal basis system) one can simply do hard thresholding.

Let θ_k be a threshold possibly dependent on the number k of wavelet coefficients. If the Wavelet coefficient is bigger than $\delta\theta_k$ we can be sure with rather high probability that the coefficient actually contains information; otherwise it is very likely that it does not. Clearly one can discard coefficients which do not contain information; this algorithms is called hard thresholding. Equivalently, one can just downweight these coefficients, and gets a soft thresholding algorithm (see e.g. Donoho[7] and the references therein). A nice feature of this kind of wavelet algorithms is the good preservation of discontinuous features like boundaries of objects as desired.

2.2 Modification of the Standard Algorithms

Obviously the above algorithm is designed to work with just one known noise level. Due to the big differences of the noise level in our pictures this would mean that in some regions we would oversmooth and loosing features and in others noise would not be removed at all. On the other hand we have more than one picture at hand; out of these one can estimate the variance of each wavelet coefficient and estimate an individual threshold for every wavelet coefficient. This means in particular when we formally denote the wavelet transform by

$$\overline{\phi}(n) = \sum_{k=1}^{K} w_k(n)\psi_k$$

where ψ_k is the kth wavelet and $w_k(n)$ its corresponding coefficient. The expected image (so just the average of all input measurements) can be written by

$$\hat{\phi} = \sum_{k=1}^{K} \widehat{w_k}\psi_k$$

where the coefficients $\widehat{w_k}$ are the standard average

$$\widehat{w_k} = \mathbb{E}w_k(n) = \frac{\sum_{n=1}^{N} w_k(n)}{N}.$$

Equivalently one can get the expected variance $\widehat{v_k}$ of each coefficient by

$$\widehat{v_k}^2 = \mathbb{E}(w_k(n) - \widehat{w_k})^2 = \frac{\sum_{n=1}^{N}(w_k(n) - \widehat{w_k})^2}{(N-1)}$$

Hence the denoised signal is

$$\widetilde{\phi} = \sum_{k=1}^{K} \widetilde{w_k}\psi_k$$

where in the case of hard thresholding

$$\widetilde{w_k} = \begin{cases} \widehat{w_k} & \text{for } |\widehat{w_k}| \geq \theta_k\widehat{v_k} \\ 0 & \text{for } |\widehat{w_k}| < \theta_k\widehat{v_k} \end{cases}$$

respectively in the case of soft thresholding

$$\widetilde{w_k} = \begin{cases} \widehat{w_k} - \theta_k\widehat{v_k} & \text{for } \widehat{w_k} \geq \theta_k\widehat{v_k} \\ 0 & \text{for } |\widehat{w_k}| < \theta_k\widehat{v_k} \\ \widehat{w_k} + \theta_k\widehat{v_k} & \text{for } \widehat{w_k} \leq -\theta_k\widehat{v_k} \end{cases}$$

In the case of standard image denoising and white noise this method would exactly give the well-known (non-adaptive) wavelet thresholding algortithm with estimated δ.

For the results presented in section 4, we used Daubechies wavelets of order 3, with a decomposition level of 4. With these, we perfomed hard thresholding using a threshold of $\theta_k = 1.1\sqrt{\frac{\text{supp}(\psi_1)}{\text{supp}(\psi_k)}}$. The support $\text{supp}(\psi_k)$ of ψ_k means in this case the number of image points influenced by the wavelet, i.e. it is the lower the higher the scale of the wavelet. This correcting factor is necessary because for any sum of Gaussian random variables its standard deviation depends linearly on the square root of the number of summands.

2.3 Discussion

The advantage of the adaptive wavelet thresholding algorithm is that it is a comparably easy to analyze variant of the standard wavelet thresholding algorithm and therefore one can fall back on a very well known case. Furthermore one has a big number of different wavelets at hand which can be adapted to the specific properties of the objects one would like to consider. Another advantage is the possibility to parallelize the algorithm almost without loss of performance. However we face the standard deficiencies of wavelet thresholding algorithms, we can end up with local artifacts. This trend is enhanced because of the fact that for every wavelet coefficient the noise level is a random variable and hence there is a high probability that there are at least some outliers.

3. CLUSTERING BASED DENOISING

Another approach to deal with the noise left over by standard camera preprocessing, is the following proposed method to perform noise level dependent and object border conserving smoothing. First the regions of the present "objects" (regions with similar gray-scale, depth, and depth noise characteristics) are determined by a clustering algorithm, followed by a smoothing within these clusters and between very similar clusters, adapted to the noise level for these objects.

3.1 Clustering using multi-sensory information

The goal set for the clustering algorithm is to determine the borders between objects, most importantly in the depth, intensity, and pixel noise images. Smoothing of the depth information can then be performed inside the relatively homgeneous regions for these objects.

The shapes of objects need not follow any predefined form. This eliminates a lot of clustering techniques,* which base the cluster assignment on simple distances to a cluster prototype, or on probability distributions based on distances (like k-means clustering or Gaussian mixture models). We decided to use a variant of the Mean Shift clustering algorithm (as proposed by Comaniciu et. al,[8] based on previous publications[9–11]), which can make use of a multi-dimensional feature space, and is not restricted to preset cluster shapes or cluster numbers.

The feature space we use incorporates multimodal information we obtain from the camera. Beside the estimated pixel depth noise (which we use slightly spatially smoothed in the following), which depends on the color and reflection properties of the object material, the depth image itself and the (infrared-) gray-scale image contain information about object boundaries. The data $\mathcal{D}$ we use for clustering thus consists of 5-dimensional vectors for each pixel (x, y):

$$\mathcal{D} = \{\mathbf{d}_{x,y}\}_{x,y}, \qquad \text{with } \mathbf{d}_{x,y} = (\boldsymbol{\phi}_{x,y}, \mathrm{var}_{x,y}, \mathbf{I}_{x,y}, x, y)^T. \tag{7}$$

The different sensor images used are $\boldsymbol{\phi}$ for the depth image (computed either from the raw images, or as preprocessed by the camera, see section 1.1.1; we are using the latter in the following), "var" for the pixel dependent depth noise (variance of pixel values) estimated from successive depth images (5 in the following examples), and $\mathbf{I}$ for the gray scale intensity image.

The Mean Shift clustering algorithm is a density based method. For each point it determines the corresponding cluster by following the gradient of a kernel density estimate (without actually computing the density) to the maximum of the density estimate. Points with the same or similar maxima (for the case of ridges in the density function) are put into the same cluster. The feature space $\{\mathbf{d}_{x,y}\}_{(x,y)}$, in which the clustering is performed, is rescaled according to the standard deviations $\boldsymbol{\sigma}_k^{\text{feature}}$ of features k, and using a choosable feature specific weighting factor $\mathbf{w}_k$, according to

$$(\mathbf{d}'_{x,y})_k = \mathbf{w}_k \cdot (\mathbf{d}_{x,y})_k / \boldsymbol{\sigma}_k^{\text{feature}}. \tag{8}$$

The feature weights used for the results presented in the next section are $\mathbf{w} = (1, 1, 1, 0.2, 0.2)^T$, which turns out to be optimal for the images under consideration and was found experimentally. Finally, we merged the smallest clusters to the closest other cluster, as long as there were clusters with sizes below 15 pixels.

*At least when applied to a feature space which includes the location of pixels, which is sensible to obtain contiguous pixels as clusters.

3.2 Smoothing in and between clusters

Smoothing the depth information with standard kernel smoothing methods is not appropriate for our data, because the noise level is not constant in all regions (cf. section 1). Additionally, there are regions (mostly in the image corners) with a generally amplified noise level. Thus the strength of the smoothing should be adapted to the local noise level. Another problem of standard kernel smoothing methods is the blurring of edges, ie. depth differences between different objects.

These drawbacks of kernel smoothing can be avoided by using information about the location and noise levels of the clusters found by the described clustering method. The members of the clusters define the region, in which smoothing should be performed. Smoothing just inside each cluster allows to use that cluster's noise level for determination of the smoothing strength, and avoids any smearing of edges in the depth image. This works well as long, as the clustering detects the borders between objects as cluster boundaries. A drawback of this approach is the possibility, that one object can be segmented into more than one cluster. This can be the case especially for very large objects (e.g. the background), because the pixel location is incorporated into the feature space and thus the distance between feature vectors.[†] Another reason for finding separate clusters for one object can be, that the object is not located at a constant distance from the camera, but has a gradient in the depth image.

To address this, we do not perform smoothing just inside each cluster. Instead, we determine, how similar two clusters are (wrt. location, color, noise level, depth, and depth gradients). The smoothing algorithm then uses weights, quantifying the similarity of each two clusters, to determine the amount of smoothing to perform across the boundary between two clusters. In the following, $\boldsymbol{\omega}_{u,v}$ denotes the similarity-weight between clusters, 0 for no similarity, 1 for very similar characteristics, see below eq. (12). m denotes the number of clusters found by the clustering algorithm, $\mathbf{C}_{x,y}$ the cluster number of pixel (x, y), and $\mathbf{D}^{(u)}_{x,y}$ is 1 for those pixels (x, y) belonging to cluster u, and 0 otherwise. $\boldsymbol{\nu}^{(u)}_{x,y}$ is a cluster u specific matrix specifiying the similarity of pixel (x, y)'s cluster to cluster u. The smoothing then works as follows:

$$\boldsymbol{\nu}^{(u)}_{x,y} = \boldsymbol{\omega}_{u,\mathbf{C}_{x,y}}, \tag{9}$$

$$\boldsymbol{\mu}^{(u)}_{x,y} = \boldsymbol{\nu}^{(u)}_{x,y}\boldsymbol{\phi}_{x,y}, \tag{10}$$

$$\boldsymbol{\phi}^{*}_{x,y} = \sum_{u=1}^{m} \mathbf{D}^{(u)}_{x,y} \frac{\big(\text{smooth}(\boldsymbol{\mu}^{(u)}, u)\big)_{x,y}}{\big(\text{smooth}(\boldsymbol{\nu}^{(u)}, u)\big)_{x,y}}, \tag{11}$$

where $\boldsymbol{\phi}^*$ is the image resulting from the cluster based smoothing of $\boldsymbol{\phi}$. Normalization by $\boldsymbol{\nu}^{(u)}$ in the denominator in equation (11) is done, because pixels outside cluster u are weighted down according to $\boldsymbol{\omega}_{u,\mathbf{C}_{x,y}}$ and thus do not count completely for normalization, but only with their weight (which might be 0 for completely unrelated clusters).

The function smooth(img, u) takes an image and a cluster number, and perfoms smoothing on the given img, with a Gaussian kernel and a tenth of the image size (longer edge) as window size. The cluster is specified, because the noise level of the cluster, which is dimension 2 of the feature space and thus also of the cluster prototype $\mathbf{x}^{(u)}$, is used in determining the standard deviation of the Gaussian kernel $\boldsymbol{\sigma}^{\text{smooth}}_{u} = c_{\text{smooth}}\ \mathbf{x}^{(u)}_{2}$. The factor c_{smooth} transforms the noise levels into ranges appropriate for $\boldsymbol{\sigma}^{\text{cluster}}_{u}$, in our examples $c_{\text{smooth}} = 45\,000$ led to values between 1.5 and 95, with most clusters having between 3 and 16.

The (asymmetric) weight $\boldsymbol{\omega}_{u,v}$ between clusters u and v is calculated by determining the 10% and 90% quantiles[‡] $q_{k,u,10}$ and $q_{k,u,90}$ of the depth, noise, intensity, horizontal gradient of depth and vertical gradient of depth images (with $k = 1, \ldots, 5$ for these features). If for two clusters (ie. for the pixels belonging to these clusters) the intervals defined by these quantiles overlap for all five images, the weight is exactly 1. If they do not overlap, the distance between these intervals is set in relation to the size of the interval for cluster u, and

[†] Although the location dimensions are usually strongly scaled down by their $\mathbf{w}_k$ in comparison to the other features.

[‡] The x%-quantile of a list l of values is that value one gets, when taking the element round(length(l)x/100) of the sorted list.

weighted with factor $\mathbf{v}_k$, differently for different features k. The maximum of the resulting differences determines the weight between the clusters, which can range between 0 and 1. The following formula specifies this:

$$\boldsymbol{\omega}_{u,v} = \max\Big(1 - \max_k\big(\mathbf{v}_k \mathrm{dist}(k,u,v)\big), 0\Big), \text{with} \tag{12}$$

$$\mathrm{dist}(k,u,v) = \big(\max(q_{k,u,10} - q_{k,v,90}, 0) + \max(q_{k,v,10} - q_{k,u,90}, 0)\big)/(q_{k,u,90} - q_{k,u,10}). \tag{13}$$

The factors used by us in practice, and determined experimentally, were $\mathbf{v} = (15, 1, 3, 1, 1)^T$.

Because perceptually depth gradient differences are much more important around 0, than they are for large positive or negative values, and because of the scaling of the depth (phase) values delivered by the camera, we used following formula to determine the depth "gradients" $\mathrm{grad}^{(\mathrm{x})}_{x,y}$ and $\mathrm{grad}^{(\mathrm{y})}_{x,y}$ at each pixel (x, y), used for features $k = 4$ and $k = 5$ in above formula (values for pixels outside the image were taken to be that of the closest pixel inside the image):

$$\mathrm{grad}^{(\mathrm{x})\prime}_{x,y} = \tanh\big((\boldsymbol{\phi}_{x+1,y} - \boldsymbol{\phi}_{x-1,y}) c_{\mathrm{grad}}\big), \tag{14}$$

$$\mathrm{grad}^{(\mathrm{x})} = \mathrm{anisodiff}(\mathrm{grad}^{(\mathrm{x})\prime}), \tag{15}$$

and similarly for $\mathrm{grad}^{(\mathrm{y})}_{x,y}$, where c_{grad} is the constant 200, and anisodiff(img) is an edge-preservingly smoothed image img (anisotropic diffusion filter[12,13]).

3.3 Discussion

Standard denoising techniques commonly use only the information in the image to smooth itself. In contrast, the approach presented here uses information from multiple channels for image segmentation and smoothing. This allows a relatively reliable estimation of object boundaries which are used for very accurate preserving of edges during the smoothing, and might also be helpful for further analysis of the image in later stages. Another advantage is the adaptation of the smoothing strength to the noise level inside each cluster.

The current algorithm determining cluster similarity is heuristic and might be improved, by using further information about when clusters can consistently belong to one object. But our approach already yields very good results on our test scenes, as detailed in the next section.

A further possible issue, which was not a problem for our images, is the minimal cluster size, necessary for eliminating single pixels or regions with very high pixel noise, or pixel errors giving pixels with a high depth offset (which can occur, if the default median filter used by the camera is switched off). This minimal cluster size, and possibly other smoothing parameters would have to be tuned for a trade-off between removing such artefacts or preserving even small details consisting of only a few pixels. Details are also much better preserved in regions with small noise levels, because of the smaller smoothing kernel used for such clusters.

In contrast to the wavelet-based processing, the clustering based approach is not deterministic, because of the clustering initialization, and it is not easily analyzable regarding its theoretical properties. It furthermore needs considerably more computation time (in the order of tens of seconds instead of the order of one second). But in exchange for the longer computation time, the result is improved considerably wrt. its smoothness and the quality measure presented in the next section.

4. APPLICATION RESULTS

4.1 Test scene

To get an objective criterion for comparison of the performances of different image processing and smoothing algorithms on actual camera data, we designed a test scene, shown in figure 3. The scene consists of three boxes of different sizes, colors and placement. A very light and relatively small box is placed a little below and left of the image center. A somewhat larger black box is placed, slightly rotated, to the right of the center. A further, long and narrow box is placed sloping on one edge of the dark box and reaches the background close to the left of the image. The background consists of a relatively light appearing thick cardbord placed on top of the plate of a camera mount. This plate is surrounded by a darker slightly raised frame. In the top right of the images

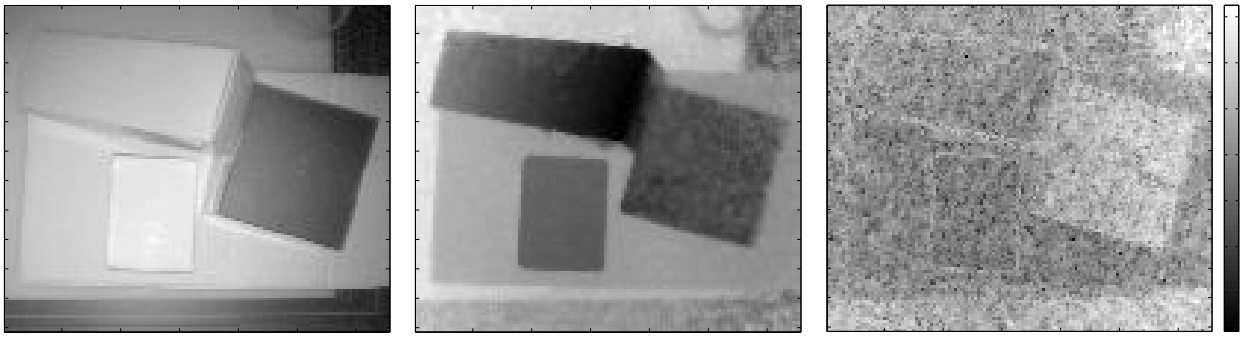

Figure 3. Test scene, for which a performance measure was defined. Left: IR-intensity image; Middle: camera depth image with standard preprocessing and smoothed over 5 frames; Right: logarithm of noise image (variance) estimated from 5 images of the scene. In the depth image, white is furthest away, while black is closest to the camera.

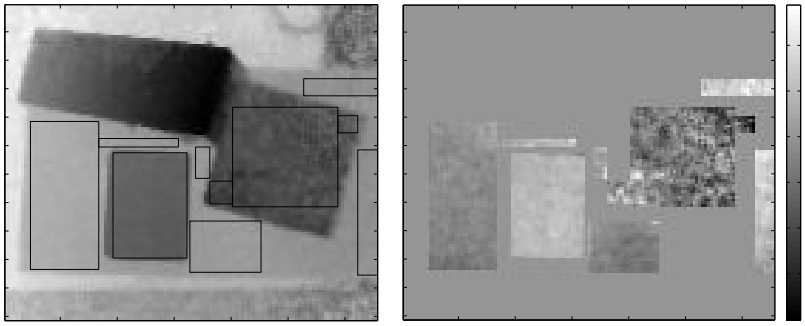

Figure 4. Left: The rectangles show the regions of the image with known height, which are used for computation of the quality measure. Right: The difference between the real heights and the optimally fitted heigths just for the relevant regions. Colorbar units are cm.

a tape roll is placed as an example of a relatively fine object in a region, where the camera produces relatively high levels of noise.

Some general deficiencies this TOF-image are already visible from the source images. Even though most surfaces are flat, considerable noise lets some surfaces appear very jagged, even tough the images are already preprocessed by the camera (geometry correction, median filter) and by an additional averaging over 5 frames. The high noise is characteristic for two conditions: Firstly, the right side of of the depth image has a generally elevated noise level. Secondly, the noise level is much higher for dark surfaces than for light ones (as was already mentioned in section 1). This noise is what we want to alleviate by the proposed smoothing methods.

It is also obvious from these images, that the color does not only influence the noise level, but it also places a bias on the average of the estimated depth. As described, the cardboard was placed on top of the camera mount plate and its frame, but according to the darker average appearance in the right of the image, the frame (which has black color, as opposed to the light brown of the cardboard) should be closer to the camera than the cardboard. So darker objects seem to appear closer than they are. This is an effect, we are not dealing with when applying our smoothing methods. It might be handled separately by an analysis, which influence different materials have on the estimated depth.

A third (here only slight) deficiency is the sub-optimal geometrical and illumination correction. Even though the plate and cardboard are almost perfectly planar, they appear slightly curved, which is especially visible in the 3D-plot (see figure 8).

Table 1. Quality of different depth images. Remaining error is also due to imperfect geometry and illumination correction.

Camera processed (original)	Anisotropic Diffusion	Wavelet smoothed	Clustering smoothed
0.2825 / 100%	0.2549 / 90.2%	0.2510 / 88.8%	0.2375 / 84.1%

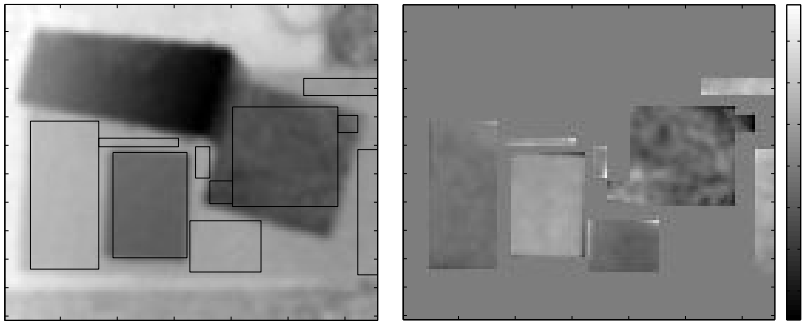

Figure 5. Image and quality of the anisotropic diffusion filtered depth image.

4.2 Quality measure

The quality criterion we defined on this test scene is based on the actual dimensions of some easily identifiable and measurable parts of the scene. These are shown as rectangles in figure 4, overlaid on top of the depth predicted by the camera. These regions were selected for their well known depth (the cardboard background, and the heights of the two boxes placed level on the cardboard), for presence of different noise levels (the light and the dark box representing the extremes), and for their closeness to edges in the depth image, to punish smearing of edges when compared to the camera image. Because the latter already introduced some smearing by the use of a median filter, the regions do not actually touch each other, but have a small gap in between, where the smearing in the camera image is located.

The image returned by the camera is not in the scale used for measuring the box dimensions, and might contain geometrical distortions, due to inappropriate correction of not-parallel rays to the camera, and due to illumination influences. Thus, before comparing the depth images with the evaluation regions, we optimally fitted the depth image to the known depths using a depth offset, a linear depth-dependent scaling, and a location-dependent scale (linear and quadratic terms for horizontal and vertical dimensions). The remaining differences are geometric distortions not covered by the linear depth scale and horizontal and vertical quadratic fit terms, slight deformations of the original boxes, and the noise in the image. Because the first two influences should be the same for all compared images, an analysis of the remaining differences between the differnt depth images allows a comparison of the remaining noise in the depth images.

We condense the remaining differences into one value per depth image by taking the root mean squared error of these, which are the standard deviation of the errors, measured in the unit cm. Table 1 gives details about the quality values for the different analyzed depth images. The Anisotrop Diffusion column gives the value for edge preserving smoothing using the anisotropic diffusion filter with the optimal (wrt. to our quality measure) iteration number and the same parameters as specified in section 3.2, see also figure 5.

4.3 Smoothing results

Figures 6 and 8 show the smoothing result for the cluster based smoothing approach, which gives the best result according to above table. Figure 6 also shows the clusters found by the mean shift algorithm, and which lead to this smoothing result. Obviously, these images indeed contain the least noise; the plane surfaces almost look so, except for a slight curvature because of imperfect geometric correction. So in one sense this is the optimal approach, at least for our test scene. Especially the smoothing for the dark box visually gives the best results.

On the other hand, the clustering based smoothing can introduce some artifacts: edges of objects may become slightly enlarged, because slopes at previous edges can be included in the cluster for an object, and lifted or

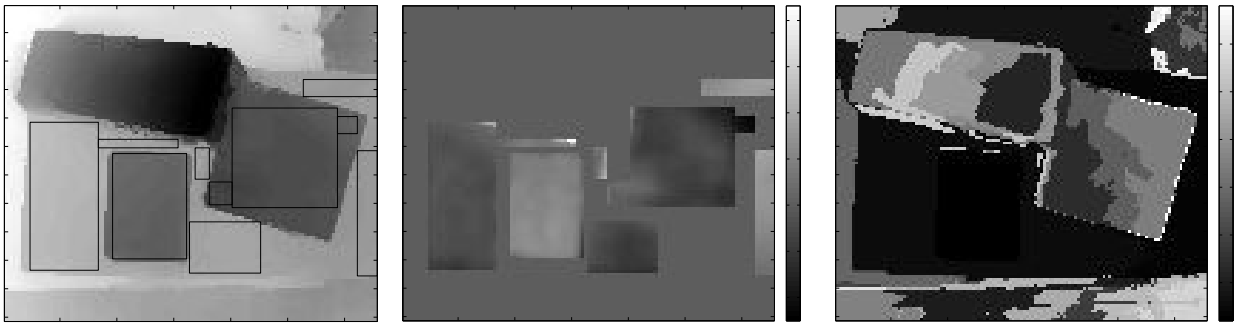

Figure 6. **Left:** Depth-image and **middle:** quality of the depth image obtained by cluster based smoothing. **Right:** clusters found by the clustering based smoothing; each gray level corresponds to one cluster.

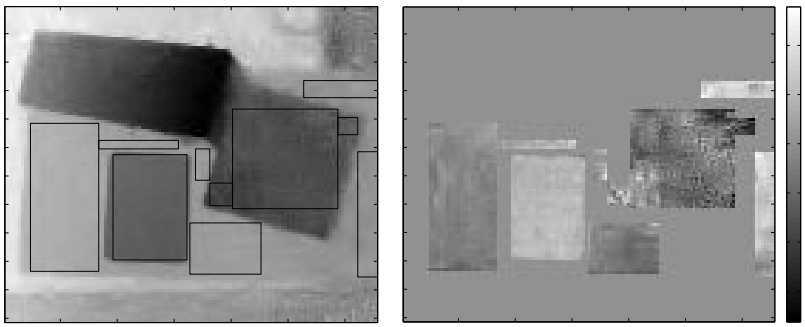

Figure 7. **Left:** Depth-image and **right:** quality of the depth image obtained by wavelet filtering.

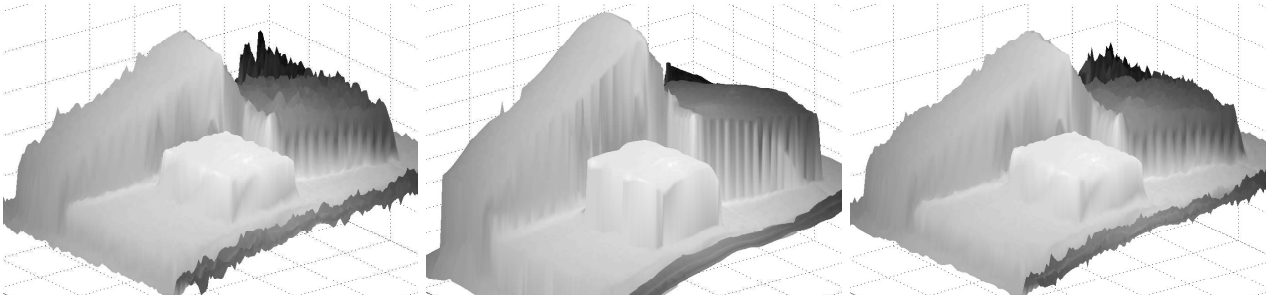

Figure 8. 3D visualization of the test scene. **Left:** The camera processed depth; **middle:** the same, smoothed with the clustering based method; **right:** smoothed by wavelet thresholding method. The depth is shown as the pixel height, while the intensity gray value is shown as its color.

lowerd to the level of the surface of that object. Furthermore, our weighting scheme can account for some slopes inside clusters (see e.g. the long side of the long, narrow box), but it still has problems with very steep slopes (e.g. the front face of that same box, which currently becomes a plateau). On the other hand, this is not a fundamental problem, as the weighting method might be further refined, and the smoothing radius could be adapted to be smaller or ellipsoidal for steep slopes.

Finally, figures 7 and 8 give results for the wavelet filtering approach. Although they are not as smooth as the clustering based result, and are only marginally better than the anisotropic diffusion smoothed image when considering the quality measure, it certainly conserves edges better than the latter, and is much faster and easier to analyze than the former. So depending on the application, this might be a good alternative to using these or just the standard camera depth, whose noise peaks are noticeably higher than those of the wavelet filtered image.

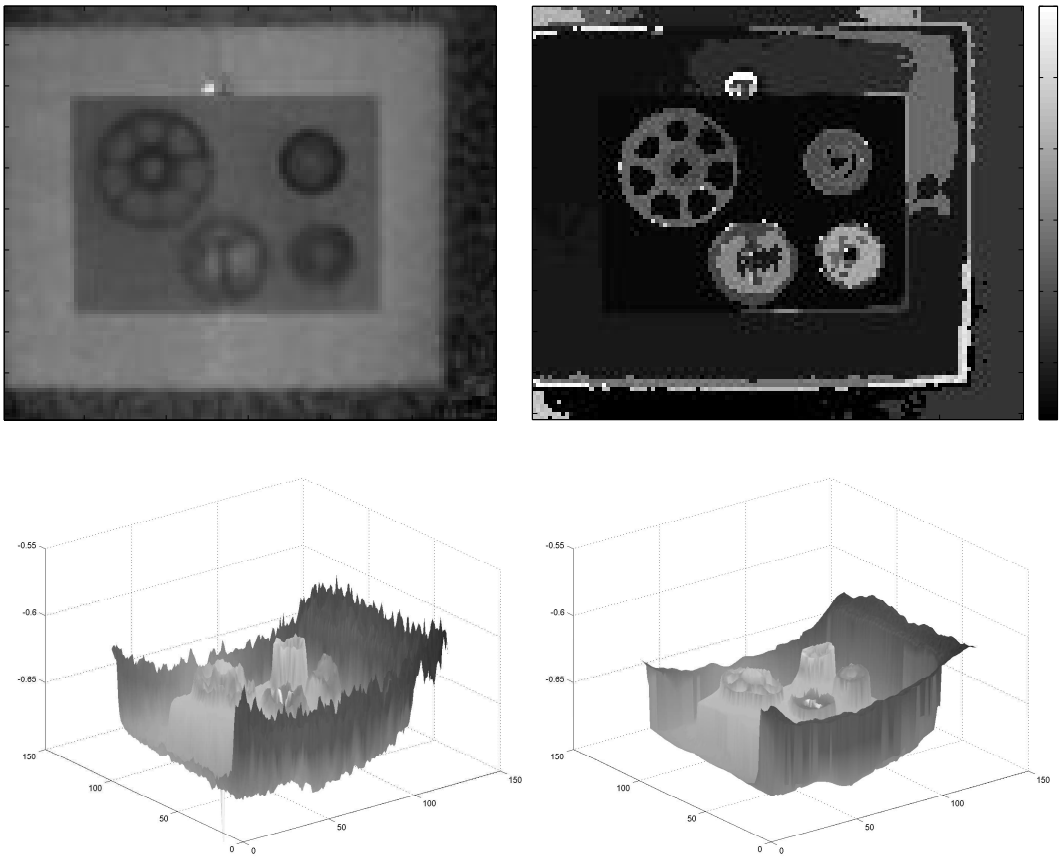

Figure 9. A more complex scenario. **Top left:** Depth Image (5 frames averaged), the white spot is an irregularity caused by reflection of the light source; **top right:** Clusters found by the Mean Shift algorithm; **bottom left:** 3d visualization of top left image (pixel height) and gray level image (pixel color); **bottom right:** 3d visualization of cluster based smoothing result.

4.4 Further results

Figure 9 shows another, more complex and realistic scenario. Although the details of the tooth wheels are not perfect, the 3d visualization appears clearer for the clustering smoothed image than for the original.

5. CONCLUSION

We presented two methods for adaptively smoothing TOF-camera depth images. One is applicable for real time improvement (adaptive wavelet method), and the other for applications allowing more processing time, but needing smoother results or initial data for segmentation (clustering based smoothing method). Additionally, we demonstrated their advantage regarding visual and measured smoothness of the resulting depth images. These results show, that, although images from TOF cameras still have quality deficiencies, these can be strongly alleviated by image processing, especially when including information from physical models of the recording process, and from additional channels available from the camera.

A further topic worthy of an investigation is the material and geometry dependent bias in the estimated depth, which is present in the recorded images. Using the available multi-channel information (and possibly also the initial segmentation results of the clustering approach), it might be feasible to alleviate these effects as well.

Applications, for which TOF cameras could already be considered, include observation of obstacles, surveillance of persons (e.g. in medical care), rough 3d position estimation of work pieces, and supplementary combination with stereoscopic imaging systems.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of the Upper Austrian Technology and Research Promotion, and of the Austrian COMET program.

REFERENCES

1. T. Oggier, M. Lehmann, R. Kaufmannn, M. Schweizer, M. Richter, P. Metzler, G. Lang, F. Lustenberger, and N. Blanc, "An all-solid-state optical range camera for 3D-real-time imaging with sub-centimeter depth-resolution (SwissRanger)," in *Proceedings of the SPIE*, **5249**(63), 2003.
2. T. Oggier, R. Kaufmann, M. Lehmann, B. Büttgen, S. Neukom, M. Richter, M. Schweizer, P. Metzler, F. Lustenberger, and N. Blanc, "Novel pixel architecture with inherent background suppression for 3D time-of-flight imaging," in *SPIE Electronic Imaging*, (San Jose), 2005.
3. R. Lange, P. Seitz, A. Biber, and S. Lauxtermann, "Demodulation pixels in CCD and CMOS technologies for time-of-flight ranging," in *Sensors, Cameras, and Systems for Scientific Industrial Applications 2*, **3965**, SPIE, 2000.
4. B. Jaehne and H. Haussecker, *Computer Vision and Applications*, Academic Press, 2000.
5. H. Rapp, "Experimental an theoretical investigation of correlating TOF camera systems," Master's thesis, Universität Heidelberg, 2007.
6. D. Donoho and I. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *J. American Statistical Association, Theory and Methods* **90**, December 1995.
7. D. Donoho, "De-noising by soft-thresholding," *IEEE Trans. on Inf. Theory* **41**(3), pp. 613–627, 1995.
8. D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**, May 2002.
9. K. Fukunaga and L. Hostetler, "The estimation of the gradient of a density function with applications in pattern recognition," *IEEE Transactions on Information Theory* **21**, pp. 32–40, 1975.
10. Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **17**, pp. 790–799, August 1995.
11. E. Choi and P. Hall, "Data sharpening as a prelude to density estimation," *Biometrika* **86**, pp. 941–947, 1999.
12. P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **12**, pp. 629–639, July 1990.
13. G. Grieg, O. Kubler, R. Kikinis, and F. Jolesz, "Nonlinear anisotropic filtering of MRI data," *IEEE Transactions on Medical Imaging* **11**, pp. 221–232, June 1992.