International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 3 (2011) pp. 635-645

Comparison of Depth Super-Resolution Methods for 2D/3D Images

Benjamin Langmann, Klaus Hartmann and Otmar Loffeld

University of Siegen, ZESS - Center for Sensor Systems,
Paul-Bonatz-Strasse 9-11, 57076 Siegen, Germany
{*langmann, hartmann , loffeld*}*@zess.uni-siegen.de*

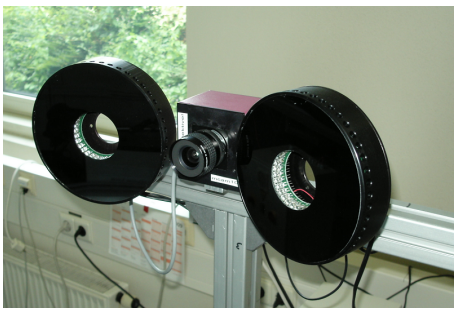
Abstract: **For a few years cameras have been available which are able to provide depth values gained from a PMD chip through the Time-of-Flight principle. Recently, cameras combining a normal color chip with such a PMD chip in a monocular setup have been developed. One drawback of these 2D/3D cameras is that the resolution of the depth images is much lower than those of the color images due to the limited resolution of current PMD chips. This holds true whenever PMD cameras and normal cameras are used together. However, for certain applications high resolution depth images are desirable. The color images can be utilized to generate high resolution depth images which are closer to the ground truth than the depth images produced with common scaling methods. A widely spread method to fuse the color and depth images is cross bilateral filtering and this rather general method is adopted in several approaches.**
In this paper different bilateral filtering strategies are compared in theory as well as in practice and especially the iterative application is addressed. Similar approaches based on a cost volume or on Markov Random Fields are addressed additionally.

Keywords: Bilateral Filter, Time-of-Flight, PMD, Super-Resolution, Sensor Data Fusion.

I. Introduction

In recent years cameras utilizing PMD technology to generate depth information through the Time-of-Flight principle have become available. They operate by emitting modulated infrared light and by measuring the phase difference to the received light. The phase difference directly corresponds to the distance of the reflecting object. PMD based cameras are often used in conjunction with normal color cameras and typically, the generated images are registered to gain pixels with color and depth information. Furthermore, cameras that combine a color and a PMD chip in a monocular setup have been developed, e.g., ZESS MultiCam see figure 1 and [1] or the ZCam (utilizing a different technology), eliminating the need for a registration of the color and depth images.
These cameras record images with high frame rates but the resolution of the depth images, typically 64×48 up to 204×204 pixels, is up until today low compared to the color images. Another problem is the relative high noise level of the depth information.

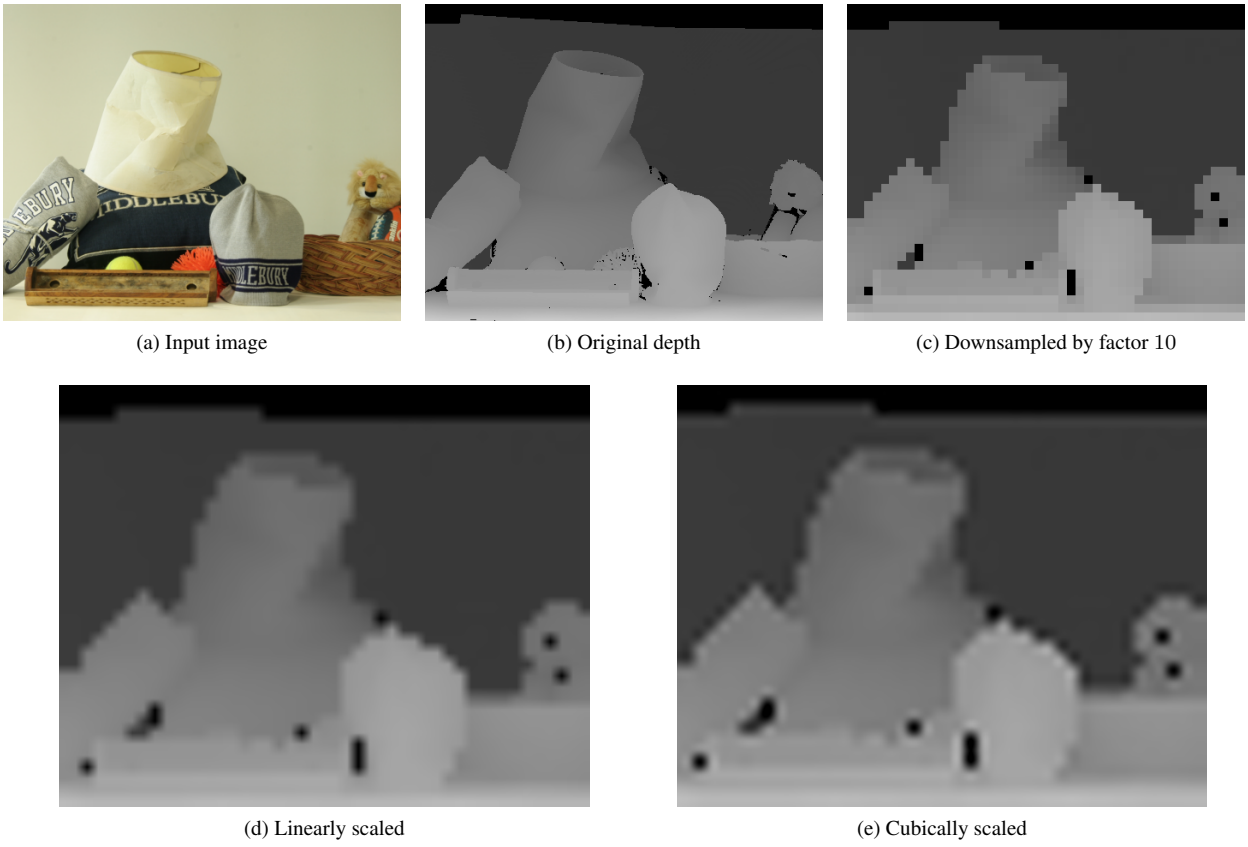
(a) 3k MultiCam F-Mount



(b) 41k MultiCam C-Mount

Figure. 1: The MultiCam was developed at the ZESS and includes a color and a PMD chip in a monocular setup. A previous version uses the 3k PMD chip from PMDTec (64×48 pixels) and a VGA color CMOS chip. The newest version incorporates a 1.3 mega pixel color CMOS chip and the 41k PMD chip (204×204 pixels).

Nevertheless, the additional dimension compared to ordinary video makes it possible to overcome ambiguities and distortions in standard image processing tasks.
For many applications high resolution depth images are desired but a simple linear or quadratic scaling of the depth images will result in invalid depth values whenever the real geometry of the scene differs from the assumed planar or spherical one respectively, see figure 2 for an illustration. The depicted color and depth images are part of the Middlebury dataset, cf. [2].
Instead of presuming certain geometry, it is possible to



(a) Input image (b) Original depth (c) Downsampled by factor 10

(d) Linearly scaled (e) Cubically scaled

Figure. 2: Illustration of the deficiencies of standard resizing techniques when resizing depth images.

assume that color and depth coincide, i.e., close pixels with similar color have also a similar depth. A common approach for this purpose is cross bilateral filtering but there are several possibilities how to apply it and how to choose the parameters. In this paper several different approaches are discussed and compared in theoretical and a practical setting. Additionally, bilateral filtering is a computationally expensive procedure and therefore, it is also addressed how to restrict the bilateral filtering to an useful region of interest that accounts for the nature of the depth values gained from a PMD-chip.
Furthermore, recent depth super-resolution methods based on Markov Random Fields (MRF) or a cost volume to judge depth assignments are discussed and evaluated.

II. Related Work

Crabb et al. use in [3] a bilateral filter for depth augmented alpha matting, which is also the focus of the work of Wang, cf. [4]. A preliminary foreground described in terms of probabilities is generated with the help of a dividing plane in space. The closer the pixel is to the point of view the more likely it belongs to the foreground. A bilateral filter is then applied on this alpha matte fusing the depth and color information. Schuon et al. proved in [5] the ability of bilateral filtering to deal with geometric objects and in [6] a variant designed to handle noise and invalid measurements is presented.
Yang et al. define a cost function or cost volume in [7] which describes the cost of in theory all possible refinements of the depth value associated with a color pixel. Again a bilateral filter is applied on this volume and after sub-pixel refinement a proposed depth is gained. The optimization is performed iteratively to achieve the final depth map. The incorporation of a second view is discussed also. Bartczak and Koch presented a similar method using multiply views, see [8]. An approach working with multiple depth images is described in [9]. The data fusion is here formulated in a statistical manner and modeled using Markov Random Fields on which an energy minimization method is applied. Earlier Diebel and Thrun presented in [10] a similar method which operates on a color and a depth image.
Another advanced method to combine range and color information was introduced by Lindner in [11]. It is based on edge preserving biquadratic upscaling and performs a special treatment of invalid measurements.

III. Bilateral Filtering

Firstly, the method of bilateral filtering will be introduced and later it is discussed how it can be applied on 2D/3D images. Afterwards, more complex approaches are briefly explained.

A. Fundamentals

The method of bilateral filtering was introduced by Tomasi in [12] and it works by calculating for each pixel a weighted mean of all other pixels. The weight depends on the distance in space between both pixels and their photometric similarity. In practice, only the neighbors are involved in the calculation (meaning the measure is truncated in space). Let the vector $\underline{x} = [x, y]^T$ denote the position of a pixel and let a (vectorial) signal $\underline{s}(\underline{x}) = [s_1(\underline{x}), s_2(\underline{x}), ..., s_n(\underline{x})]^T$ represent some (vectorial) value at that position, e.g., the RGB color values and the (measured) distance. Then the weighted mean $\underline{\mu}_{\underline{s}(\underline{x})}$ for a pixel $\underline{x}$ is given by

$$\underline{\mu}_{\underline{s}(\underline{x})} = \frac{\sum\limits_{\underline{x}_i \in N(\underline{x})} \underline{s}(\underline{x}_i) \cdot h\left[\underline{g}(\underline{x}), \underline{g}(\underline{x}_i)\right]}{\sum\limits_{\underline{x}_i \in N(\underline{x})} h\left[\underline{g}(\underline{x}), \underline{g}(\underline{x}_i)\right]} \tag{1}$$

with $N(\underline{x})$ representing some neighborhood of the pixel $\underline{x}$ and $h[\cdot,\cdot]$ being the weighting function depending on the distance between the pixels of interest. The summation involves all pixels $\underline{x}_i$ within the specified neighborhood. $\underline{g}(\cdot)$ is some functional mapping of the pixel, depending on the pixel position and optionally depending also on the pixel value. If it depends only on the pixel position (e.g. $\underline{g}(\underline{x}) = \underline{x}$) we have a conventional space variant or space invariant filter ($h\left[\underline{g}(\underline{x}), \underline{g}(\underline{x}_i)\right] = h[\underline{x} - \underline{x}_i]$), defined in the space domain. If $\underline{g}(\cdot)$ additionally depends on the signal value itself (e.g. $\underline{g}(\underline{x}) = \left[\underline{x}^T, \underline{s}(\underline{x})^T\right]^T$), we have a filter which works in the space/similarity domain. If we restrict ourselves to scalar valued weighting functions a general quadratic form expressed in terms of the vector $\underline{g}$ is a good choice:

$$h\left[\underline{g}(\underline{x}), \underline{g}(\underline{x}_i)\right] = \exp\left\{-\left(\underline{g}(\underline{x}) - \underline{g}(\underline{x}_i)\right)^T \cdot \Pi^{-1} \cdot \left(\underline{g}(\underline{x}) - \underline{g}(\underline{x}_i)\right)\right\} \tag{2}$$

Instead of defining the neighborhood directly with the Euclidean distance in space

$$N_d^2(\underline{x}) = \{\widehat{\underline{x}} : \|\underline{x} - \widehat{\underline{x}}\|_2 < d\} \tag{3}$$

it can now also be specified using the weighting function $h[\cdot,\cdot]$

$$N_\varepsilon(\underline{x}) = \left\{\widehat{\underline{x}} : h\left[\underline{g}(\underline{x}), \underline{g}(\widehat{\underline{x}})\right] > \varepsilon\right\} \tag{4}$$

with a threshold $\varepsilon \leq 1$ thus involving all dimensions of the signal in the specification of the neighborhood. This corresponds to a truncated weighting function $h[\cdot,\cdot]$. Now assuming that $\underline{s}(\underline{x}) = [\underline{x}, s_1(\underline{x}), s_2(\underline{x}), ..., s_n(\underline{x})]^T = [\underline{x}, \widehat{\underline{s}}(\underline{x})]^T$, $\underline{g}(\underline{x}) = \underline{s}(\underline{x})$ and $\Pi = \begin{bmatrix} I \cdot \sigma_{space} & 0 \\ 0 & \Pi_{signal} \end{bmatrix}$, where I is the identity matrix corresponding in rank to the dimensions of $\underline{x}$, the multivariate weighting function separates into the product of two Gaussian kernels:

$$h\left[\underline{g}(\underline{x}), \underline{g}(\underline{x}_i)\right] = \exp\left\{-\frac{(\underline{x} - \underline{x}_i)^T (\underline{x} - \underline{x}_i)}{\sigma_{space}^2}\right\} \cdot \exp\left\{-(\widehat{\underline{s}}(\underline{x}) - \widehat{\underline{s}}(\underline{x}_i))^T \cdot \Pi_{signal}^{-1} \cdot (\widehat{\underline{s}}(\underline{x}) - \widehat{\underline{s}}(\underline{x}_i))\right\}. \tag{5}$$

Thereby, it is possible to specify the neighborhood radius d such that $N_\varepsilon(\underline{x}) \subset N_d^2(\underline{x})$ by $d = \sqrt{-\sigma_{space}^2 \log \varepsilon}$, which reduces the computational complexity, since there are less than $4d^2$ pixels $\underline{x}_i \in N_d^2(\underline{x})$ and these pixels can be directly addressed.

Another typical assumption is restricting Π_{signal} to be diagonal $\Pi_{signal} = diag\left(\sigma_1^2, \sigma_2^2, ..., \sigma_n^2\right)$ resulting in an independence of the signal dimensions. The different natures of space, depth measurements and color values can be accounted for by choosing different smoothing values $\sigma_{space}^2, \sigma_1^2, \sigma_2^2, ..., \sigma_n^2$ in the weighting function.

In the examples the signals $\underline{s}(\underline{x})$ are of the form $\underline{s}(\underline{x}) = [x, y, d(\underline{x}), r(\underline{x}), g(\underline{x}), b(\underline{x})]^T$ with $d(\underline{x})$ being the distance measurement and $r(\underline{x}), g(\underline{x}), b(\underline{x})$ denote the RGB color value for a pixel $\underline{x}$. The mapping $\underline{g}(\cdot)$ is simply the complete signal value $\underline{g}(\underline{x}) = \underline{s}(\underline{x})$ and $\Pi = diag\left(\sigma_{space}^2, \sigma_{space}^2, \sigma_d^2, \sigma_{color}^2, \sigma_{color}^2, \sigma_{color}^2\right)$. It might be advantageous to weight the color components differently or to use the L*a*b or the Luv color space to make differences between color values perceptually uniform.

B. Iterative Application

It is possible to apply the bilateral filter iteratively. Different approaches to that end are possible. Firstly, only the depth values $d(\underline{x})$ can be refined by the bilateral filter. This means in practice that in each step the depth map produced in the previous step and the original image are supplied to the bilateral filter. For a pixel $\underline{x}$ and a recorded signal $\underline{s}(\underline{x}) = \underline{s}^{(0)}(\underline{x})$ this leads to the update formula for filtered signal values $\underline{s}^{(1)}(\underline{x}), \underline{s}^{(2)}(\underline{x}), ...$

$$\underline{s}^{(j+1)}(\underline{x}) = \underline{s}^{(0)}(\underline{x}) \cdot diag\left(1,1,0,1,1,1\right) + \underline{\mu}_{\underline{s}^{(j)}}(\underline{x}) \cdot diag\left(0,0,1,0,0,0\right). \tag{6}$$

It should be mentioned that the mapping $\underline{g}(\underline{x})$ in $\underline{\mu}_{\underline{s}^{(j)}}(\underline{x})$ uses the updated values of $\underline{s}^{(j)}(\underline{x})$, i.e., $\underline{g}(\underline{x}) = \underline{g}^{(j)}(\underline{x}) = \underline{s}^{(j)}(\underline{x})$.

Another possibility is to refine all measured dimensions. This can be written with the recursion formula

$$\underline{s}^{(j+1)}(\underline{x}) = \underline{s}^{(0)}(\underline{x}) \cdot diag\left(1,1,0,0,0,0\right) + \underline{\mu}_{\underline{s}^{(j)}}(\underline{x}) \cdot diag\left(0,0,1,1,1,1\right). \tag{7}$$

Finally, it can be operated on all dimensions simultaneously leading to

$$\underline{s}^{(j+1)}(\underline{x}) = \underline{\mu}_{\underline{s}^{(j)}}(\underline{x}). \tag{8}$$

When processing each point independently from all others, i.e.,

$$\underline{\mu}_{\underline{s}^{(j)}}(\underline{x}) = \frac{\sum\limits_{\underline{x}_i \in N(\underline{x})} \underline{s}^{(0)}(\underline{x}_i) \cdot h\left[\underline{s}^{(j)}(\underline{x}), \underline{s}^{(0)}(\underline{x}_i)\right]}{\sum\limits_{\underline{x}_i \in N(\underline{x})} h\left[\underline{s}^{(j)}(\underline{x}), \underline{s}^{(0)}(\underline{x}_i)\right]}, \tag{9}$$

this procedure is equivalent to the Mean-Shift algorithm [13]. The coordinates of the signal are not changed in the first two possibilities (eq. 6 and 7) and therefore, it is

possible to address the neighboring pixels directly. The complexity lies then in $O(d^2)$ for each iteration with d being the radius of the neighborhood $N_d^2(\underline{x})$. In the last case (eq. 8) the neighborhood is unknown if the pixels are processed simultaneously. Therefore, the signals for all pixels have to be checked in a trivial implementation, although space partitioning techniques such as kd-tree [14] or a spatial registration of the filtered signals using dynamic arrays can be applied to reduce the complexity.

C. Cost Volume Optimization Problem

Yang in [7] and Bartczak in [8] assign each (discrete) depth change of a pixel a quadratic and truncated cost. Let $d_1, \ldots, d_m$ be all possible depth values $d(\underline{x})$ for a pixel $\underline{x}$. Then the cost function $c(\underline{x}, d_k)$ which assigns a cost to a change of the depth $d(\underline{x}) = d_k$ is given by

$$c(\underline{x}, d_k) = \min\left\{\gamma, (d(\underline{x}) - d_k)^2\right\} \tag{10}$$

where γ is a truncating threshold. This cost function spans the initial cost volume $C^{(0)} = \left(c\left([x,y]^T, d_k\right)\right)_{xyd_k}$. The costs $C^{(\cdot)}_{\cdot\cdot d_k}$ to refine all pixels to a certain depth d_k are called a slice and on each slice a bilateral filter is applied iteratively. For the slice $C^{(j)}_{\cdot\cdot d_k}$ a signal is given by

$$\underline{s}^{(j)}(\underline{x}) = \left[x, y, d(\underline{x}), r(\underline{x}), g(\underline{x}), b(\underline{x}), C^{(j)}_{\underline{x}d_k}\right]^T \tag{11}$$

and the mapping is again $\underline{g}(\underline{x}) = \underline{s}(\underline{x})$. The smoothing values are similarly $\Pi = diag\left(\sigma^2_{space}, \sigma^2_{space}, 0, \sigma^2_{color}, \sigma^2_{color}, \sigma^2_{color}, 0\right)$. The filtered cost volume in iteration $(j+1)$ is determined by

$$C^{(j+1)}_{\underline{x}d_k} = \underline{\mu}_{\underline{s}^{(j)}}(\underline{x}) \cdot [0,0,0,0,0,0,1]\,. \tag{12}$$

Now the filtered signal can be calculated with

$$\underline{s}^{(j+1)}(\underline{x}) = \left[x, y, \widehat{d}(\underline{x}), r(\underline{x}), g(\underline{x}), b(\underline{x}), C^{(j+1)}_{\underline{x}d_k}\right]^T, \tag{13}$$

$$\widehat{d}(\underline{x}) = \arg\min_{d_k}\left\{C^{(j+1)}_{\underline{x}d_k}\right\}. \tag{14}$$

Alternatively, the local minimum of a quadratic function that is fitted to the minimum $\widehat{d}(\underline{x})$ and its neighbors can be used in the refined signal, which is called sub-pixel refinement.
The cost volume method is based on the assumption that the scene is piecewise planar and therefore the cost of a pixel for a certain depth is correlated to the cost of refining the neighboring pixels to the same depth.
The minimum cost for a pixel can be directly used in the next iteration as current depth estimate.

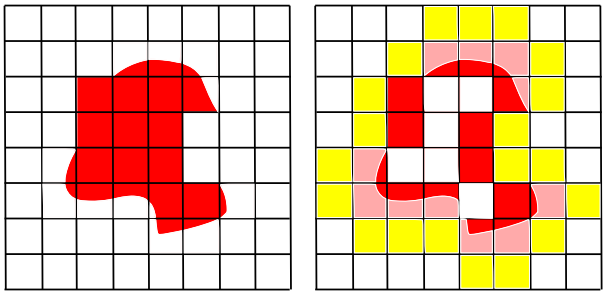
D. Markov Random Field Optimization

Another possibility to construct a high resolution depth map is to formulate this task in a Markov Random Field (MRF) and to maximize the posterior probability as is described by Diebel and Thrun in [10]. This posterior probability depends on the squared difference between the generated depth map and the available depth measurements as well as on the squared difference between the depth values of neighboring pixels weighted by their photometric similarity. This follows again the assumption that photometric and depth similarities coincide.
The logarithmic posterior is minimized by the conjugate gradient method. In each optimization step the photometric weights have to be determined multiple times (in the line search). Therefore, each iteration corresponds to multiple iterations in the bilateral filtering methods described above.

IV. Region of Interest for PMD Chips

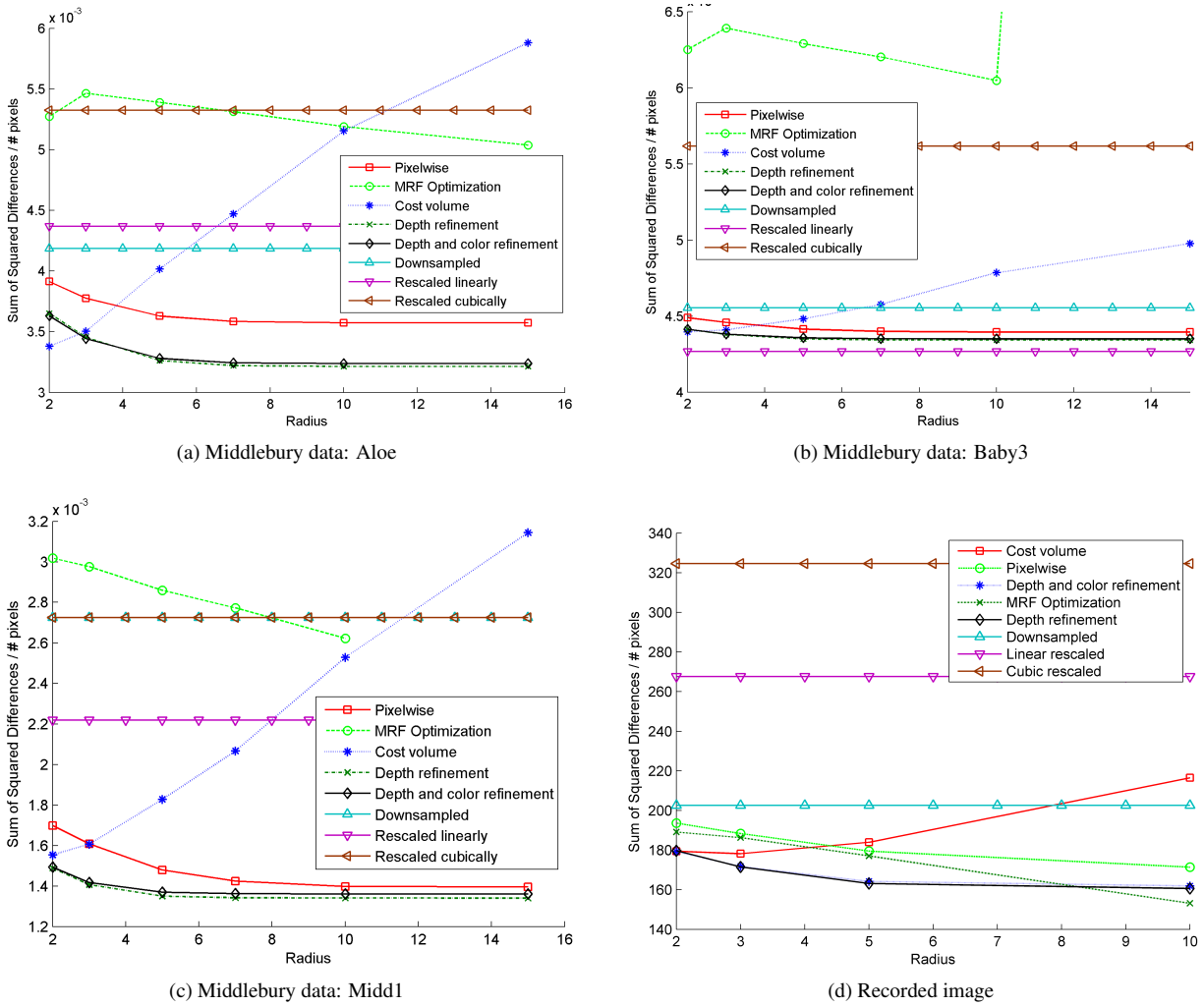
As mentioned before bilateral filtering is a computationally expensive procedure. But in many image processing tasks only certain parts of the images are actually interesting and form the so-called region of interest (ROI). Applying a bilateral filter only on the ROI care must be done cautiously. Obviously, iterative calculations depend directly on the results of all neighbors. But the information of each pixel propagates in theory throughout the whole image after a certain number of iterations. But in practice the influence of non neighboring pixels is negligible for usual values of σ_{space}. Therefore, in addition to the pixels of the ROI at least their neighbors should be processed as well.
When using a PMD-camera with a much lower resolution than the available color image, it is essential to take the characteristic behavior of the PMD-chip into account when determining which area of the image should be filtered in addition to the ROI. The PMD-data has a high noise level and is affected by reflectance, illumination (visible light as well as infrared light) and material of the objects in the scene.

Two different cases can be considered: Either a high resolution ROI is available which was gained at least mainly from the color image or a low resolution ROI was determined based on the depth data. In both cases the ROI should be enlarged by all pixels that share the same depth pixel as a pixel in the ROI. This can be seen as filling the low resolution depth pixels. Additionally, all neighboring depth pixels should be included, because when a depth pixel



(a) Schematic depth image (b) Final region of interest

Figure. 3: Schematic view of a foreground object covering several depth pixel blocks and the resulting region of interest, on which the bilateral filter shall be applied.



(a) Middlebury data: Aloe

(b) Middlebury data: Baby3

(c) Middlebury data: Midd1

(d) Recorded image

Figure. 4: Comparison of the generated depth images with the original depth or the handmade ground truth respectively using different radii for $\sigma_s = 10$, $\sigma_c = 30$ and a downsampling factor of 10.

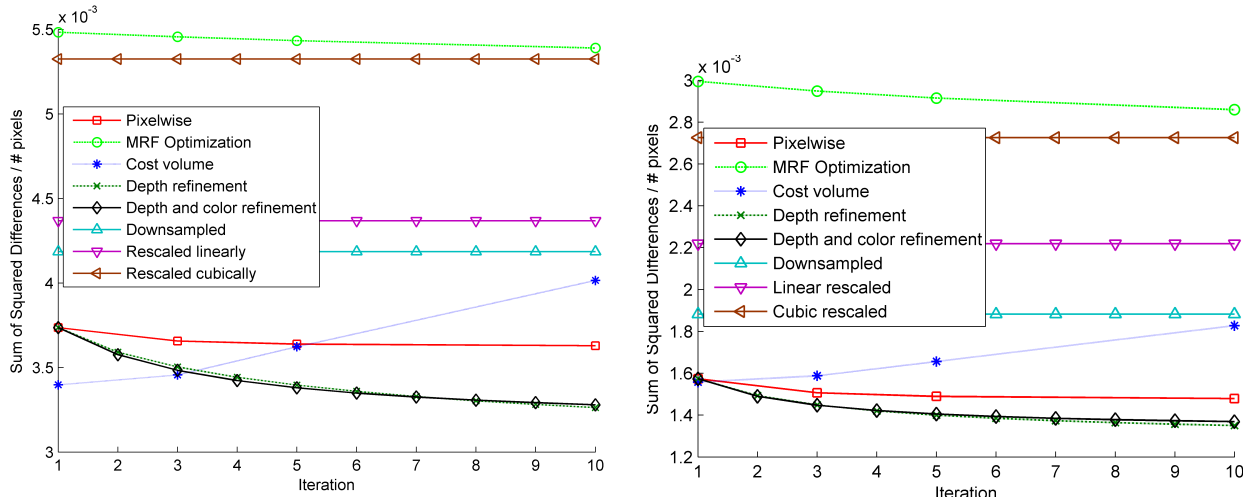
covers a border of an object it is unknown whether it is the result of measuring the background or the object. If iterative filtering is used, all pixels in the neighborhood depending on the neighborhood radius are to be processed as well. If only the border of an object is to be refined, inner pixel blocks can be neglected, see figure 3 for an illustration. When considering the computational cost of bilateral filtering, possible implementations on a GPU should be mentioned as well and will be discussed further in the next section.

V. Experiments

The different methods described in section 3 to scale the low resolution depth images gained from a PMD based camera are compared in this section qualitatively and quantitatively. Since the ground truth, i.e., a high resolution depth image, is normally not available or can only be derived per hand, the Middlebury dataset, cf. [2], was used in the experiments in addition to real camera images with handmade ground truth. The following methods are being evaluated: Depth refinement (eq. 6), depth and color refinement (eq. 7), pixelwise (eq. 8, 9), cost volume (eq. 11,12) and MRF optimization.

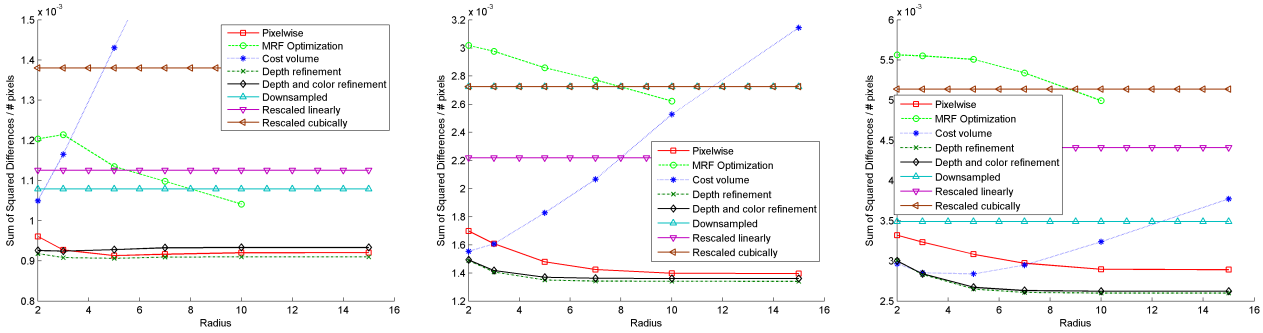
In table 1 the iterative development of the refined depth images for a Middlebury image using the different methods and common parameters is shown. The available ground truth depth image was downsampled by a factor of 10 to gain a low resolution depth image to use as the starting point for the super-resolution methods. Here $\sigma_{space} = 10$, $\sigma_{color} = 30$, $\sigma_d^{-1} = 0$ and 10 iterations were used. The resulting depth maps are then compared to the original depth image. The results for different images are shown in figure 4. Values for the downsampled depth image and linearly as well as cubically scaled images are given for comparison. A wide variety of parameters were used on several images from the dataset to ensure that the given results are representative. In figure 5 the iterative development for the different methods is illustrated and in figure 6 different downscaling factors are compared.

Afterwards, the described methods are used on images recorded with the already mentioned MultiCam. An example is shown in figure 7 using a radius of 10 and again, the iterative development is shown in figure 2. A ground truth



(a) Middlebury data: Aloe (b) Middlebury data: Baby3

Figure. 5: The discussed methods in different iteration steps for $\sigma_s = 10$, $\sigma_c = 30$, a radius of 5 and a downsampling factor of 10.

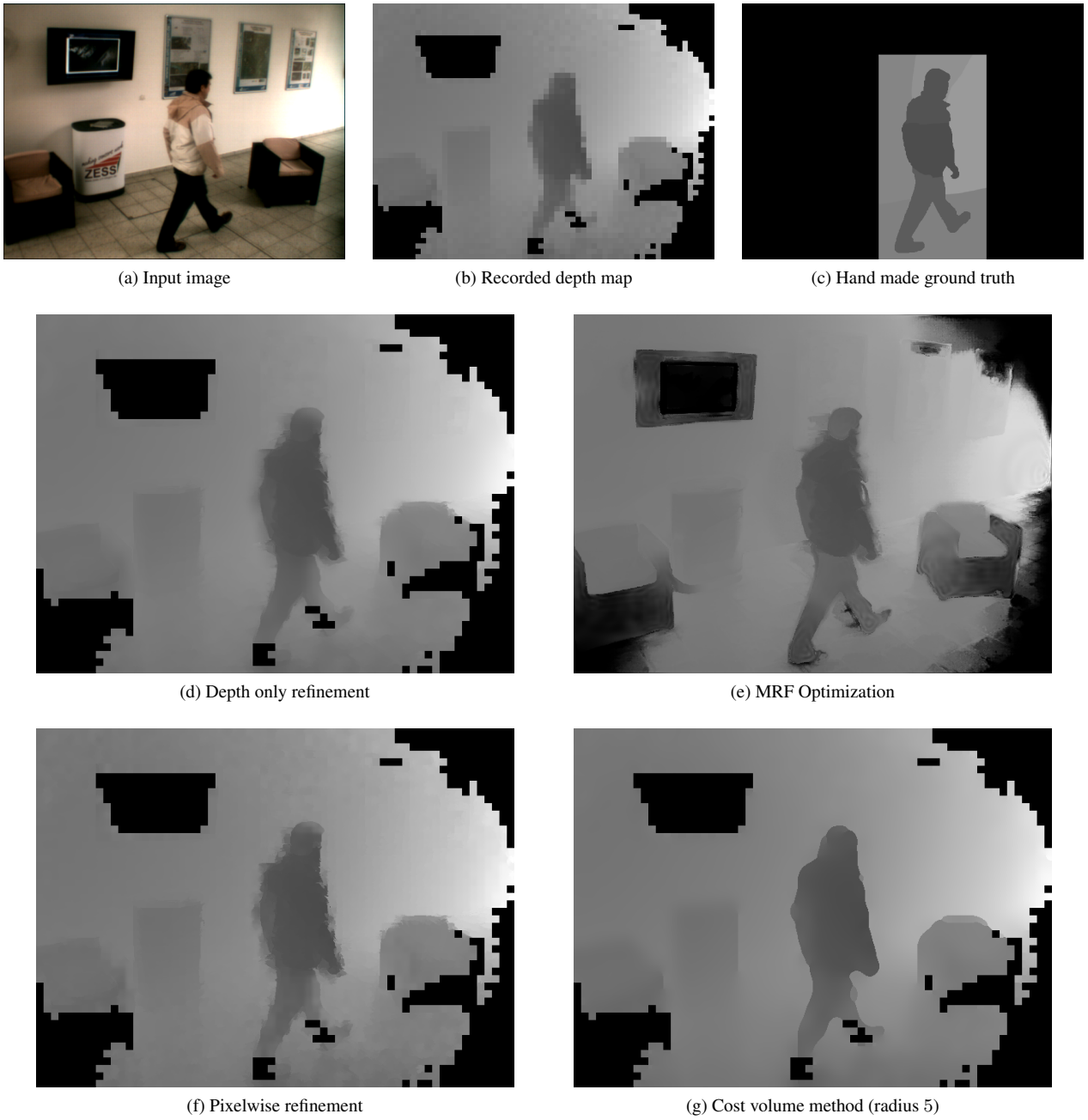


(a) Downsampling factor 4 (b) Downsampling factor 10 (c) Downsampling factor 20

Figure. 6: Comparison of the generated depth images with the original depth for different downsampling factors 4, 10, 20 based on the Middlebury data (Midd1) with $\sigma_s = 10$ and $\sigma_c = 30$.

was created for a few frames per hand and it was used to measure the filtered depth images. The differences (SSD) using different radii are also given in figure 4 as well as the results using simple upscaling techniques.
The results clearly show that all methods produce depth images that are much more accurate than a simple scaling using standard techniques with the MRF method being an exception. Linear and cubic scaling blur the depth images which results in greater differences. The methods of depth only and depth and color refinement perform very similarly. The pixelwise approach is slightly weaker. The cost volume method produces subjectively the nicest results, but it tends towards to smooth contours especially for higher radii. This behavior is in particular problematic for real recorded images. The MRF optimization method produces visibly good results but it changes to depth values to much and thereby it diverges from the ground truth. It turns out, that $\sigma_d^{-1} = 0$, i.e., neglecting the depth values when determining the weights (Cross Bilateral Filtering) performs best. This assertion holds true for all methods. Even in the MRF optimization ignoring the differences to the available depth measurements performs best. The fact that the results for the synthetic and the real data are very similar proves the validity of the results. Only the MRF optimization performs significantly better for the real scenario when using high radii, although it takes then easily several minutes to process a single image.

In figure 8 processing times per iteration are given for the different methods applied on the Middlebury image (see figure 2) with a resolution of 465×370 pixels and measured on a Intel Pentium IV Core2Duo processor running at 3 GHz. There are also processing times for two GPU implementations of the depth only refinement method given, for which an ATI HD 5770 graphics card was used. In the first implementation all calculations for a pixel in an iteration are done in one thread on the GPU and the second one is a vectorized version. Here the pixelwise method performs best. The runtime of the cost volume method depends on the number of discrete depth steps (slices) of which 100 were used. The processing times using only 10 slices are similar to those of the pixelwise method. As expected the GPU versions perform by more than one order of magnitude better. These numbers should provide an estimate



(a) Input image (b) Recorded depth map (c) Hand made ground truth

(d) Depth only refinement (e) MRF Optimization

(f) Pixelwise refinement (g) Cost volume method (radius 5)

Figure. 7: Comparison of the filtered depth images with the original depth for the or the handmade ground truth respectively using different radii for $\sigma_s = 10$, $\sigma_c = 30$ and a downsampling factor of 10.

of possible applications for these methods. Performance values for the MRF optimization are omitted due to the large dependence on the conjugate gradient method for which many implementations are possible.

VI. Conclusion

In this paper different bilateral filtering techniques are formulated and compared from a theoretic point of view as well as using real 2D/3D recordings. When working with 2D/3D images usually the different types of data have to be fused. In a monocular setup this can simply be done by rescaling using common scaling techniques, otherwise the images have to be registered first. It was found, that all described bilateral filtering methods can be used to scale the depth images and produce more accurate results. Additionally, the cost volume based method results in strong but often also to smooth contours. The depth and the depth and color refinement methods perform very similarly and better than the pixelwise method. On the other hand this is the fastest one as the computational demand of these methods is addressed as well. Two possible solutions to reduce the processing times are discussed: the limitation of the filtering to a region of interest and an implementation on special hardware such as GPUs.

Table 1: Resulting depth images after applying the super-resolution methods on a complete input image and a low resolution depth image using different parameters.

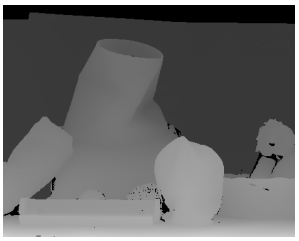
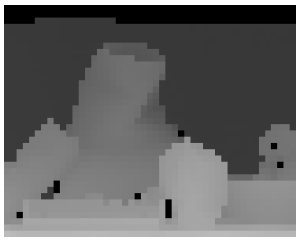
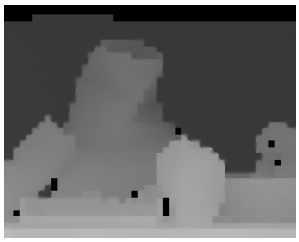
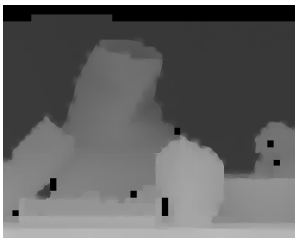
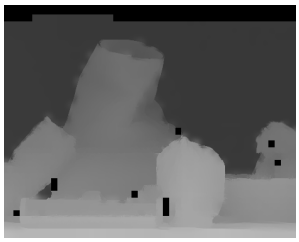
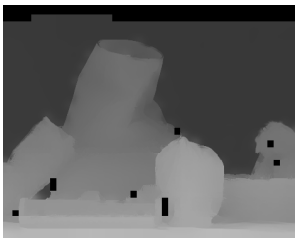
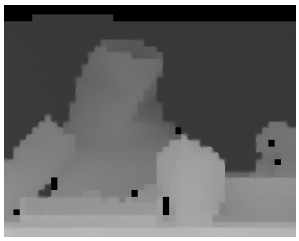
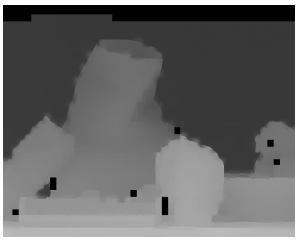
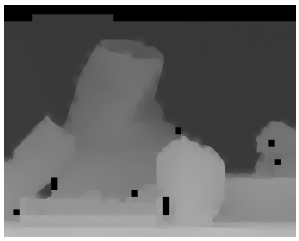
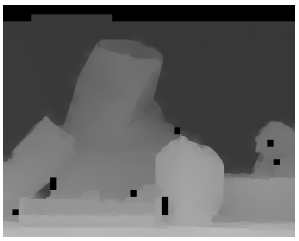
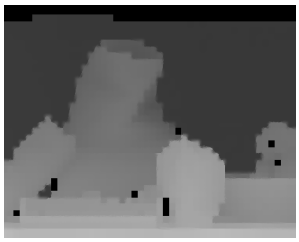
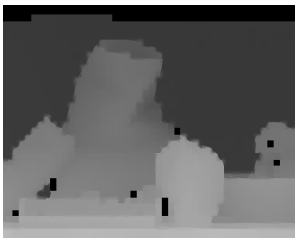
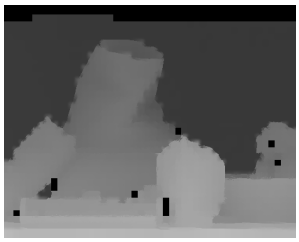
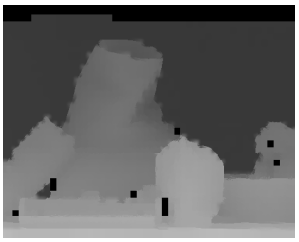
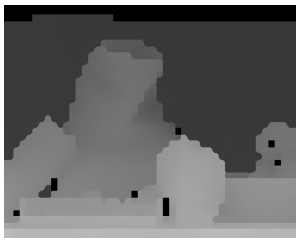
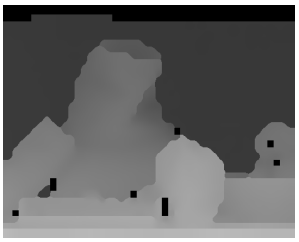
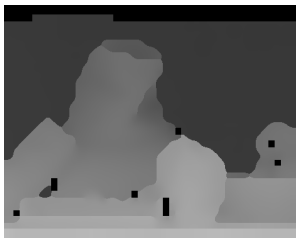
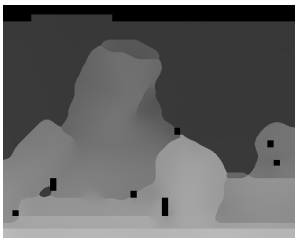
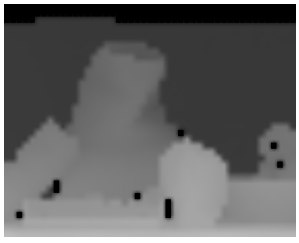
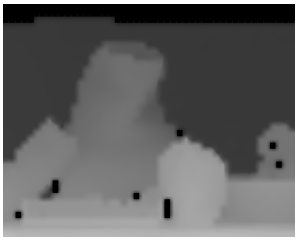
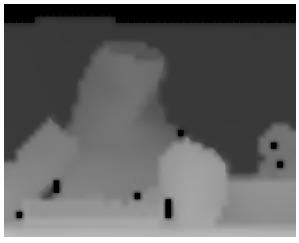
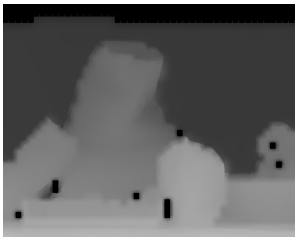
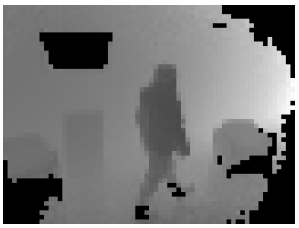
Input image	Depth image	Downsampled factor 10	Linear upscaled
			
Only depth refinement method, Radius 5, $\sigma_s = 10$, $\sigma_c = 30$			
Iteration 1	Iteration 3	Iteration 5	Iteration 10
			
Color and depth refinement method, Radius 5, $\sigma_s = 10$, $\sigma_c = 30$			
Iteration 1	Iteration 3	Iteration 5	Iteration 10
			
Pixelwise method, Radius 5, $\sigma_s = 10$, $\sigma_c = 30$			
Iteration 1	Iteration 3	Iteration 5	Iteration 10
			
Cost volume method, Radius 5, $\sigma_s = 10$, $\sigma_c = 30$			
Iteration 1	Iteration 3	Iteration 5	Iteration 10
			
MRF opt. method, Radius 5, $\sigma_s = 10$, $\sigma_c = 30$			
Iteration 1	Iteration 3	Iteration 5	Iteration 10
			

Table 2: Resulting depth images after applying the super-resolution methods on a real recored input image and a low resolution depth image using different parameters.

Input image	Depth image	Linear upscaled	Ground truth
			
Only depth refinement method, Radius 5, $\sigma_s = 10$, $\sigma_c = 30$			
Iteration 1	Iteration 3	Iteration 5	Iteration 10
			
Color and depth refinement method, Radius 5, $\sigma_s = 10$, $\sigma_c = 30$			
Iteration 1	Iteration 3	Iteration 5	Iteration 10
			
Pixelwise method, Radius 5, $\sigma_s = 10$, $\sigma_c = 30$			
Iteration 1	Iteration 3	Iteration 5	Iteration 10
			
Cost volume method, Radius 5, $\sigma_s = 10$, $\sigma_c = 30$			
Iteration 1	Iteration 3	Iteration 5	Iteration 10
			
MRF opt. method,, Radius 5, $\sigma_s = 10$, $\sigma_c = 30$			
Iteration 1	Iteration 3	Iteration 5	Iteration 10
			

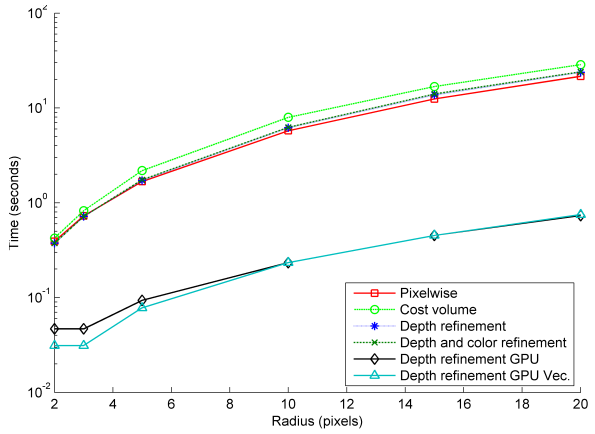


Figure. 8: Processing time per iteration for the different methods.

Acknowledgments

This work was funded by the German Research Foundation (DFG) as part of the research training group GRK 1564 'Imaging New Modalities' and the authors would like to thank the members of the ZESS for the help with working with the 2D/3D camera and especially Seyed E. Ghobadi for the valuable discussions.

References

[1] T. Prasad, K. Hartmann, W. Weihs, S. Ghobadi, and A. Sluiter, "First steps in enhancing 3D vision technique using 2D/3D sensors," in *Computer Vision Winter Workshop*, Telc, Czech Republic, 2006, pp. 82–86.

[2] D. Scharstein and C. Pal, "Learning conditional random fields for stereo," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Minneapolis, Minnesota, 2007, pp. 1–8.

[3] R. Crabb, C. Tracey, A. Puranik, J. Davis, C. Santa Cruz, I. Canesta, and C. Sunnyvale, "Real-time foreground segmentation via range and color imaging," in *Proceedings of the CVPR Workshop on Time-of-flight Computer Vision*, Anchorage, Alaska, 2008, pp. 1–5.

[4] O. Wang, J. Finger, Q. Yang, J. Davis, and R. Yang, "Automatic natural video matting with depth," in *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, Maui, Hawaii, 2007, pp. 469–472.

[5] S. Schuon, C. Theobalt, J. Davis, and S. Thrun, "High-quality scanning using time-of-flight depth superresolution," *Computer Vision and Pattern Recognition Workshop*, vol. 0, pp. 1–7, 2008.

[6] D. Chan, H. Buisman, C. Theobalt, and S. Thrun, "A Noise-Aware Filter for Real-Time Depth Upsampling," in *Proceedings of the ECCV Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, Marseille, France, 2008, pp. 1–12.

[7] Q. Yang, R. Yang, J. Davis, and D. Nister, "Spatial-depth super resolution for range images," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 0, pp. 1–8, 2007.

[8] B. Bartczak and R. Koch, "Dense Depth Maps from Low Resolution Time-of-Flight Depth and High Resolution Color Views," in *Proceedings of the 5th Int. Symposium on Advances in Visual Computing*. Las Vegas, Nevada: Springer, 2009, pp. 228–239.

[9] A. Rajagopalan, A. Bhavsar, F. Wallhoff, and G. Rigoll, "Resolution Enhancement of PMD Range Maps," in *Proceedings of the 30th DAGM symposium on Pattern Recognition*, vol. 5096. Munich, Germany: Springer, 2008, pp. 304–313.

[10] J. Diebel and S. Thrun, "An application of markov random fields to range sensing," in *Proceedings of Conference on Neural Information Processing Systems*, vol. 18, 2005, pp. 291–298.

[11] M. Lindner, M. Lambers, and A. Kolb, "Sub-pixel data fusion and edge-enhanced distance refinement for 2D/3D images," *International Journal of Intelligent Systems Technologies and Applications*, vol. 5, no. 3-4, pp. 344–354, 2008.

[12] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the Sixth International Conference on Computer Vision*, vol. 846, Bombay, India, 1998, pp. 839–846.

[13] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[14] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, Sep. 1975.

Benjamin Langmann received the Diploma degree in computer science (2009) and in mathematics (2010) from the Technical University of Braunschweig, Germany. Since 2009 he is working as a research assistant at the Center for Sensor Systems (ZESS) at the University of Siegen as a member of the research training group GRK 1564 'Imaging New Modalities' funded by the German Research Foundation (DFG).
His research includes 2D/3D imaging starting with image acquisition to super-resolution methods. Additionally, he works in the field of computer vision, where he worked on background subtraction and tracking utilizing multiple 2D/3D cameras.

Klaus Hartmann received the Diploma degree in electrical engineering and the Dr. Eng. degree from the University of Siegen, Siegen, Germany, in 1983 and 1989, respectively. After receiving the Diploma degree he was with an Industrial Company, where he was engaged in the development of distributed real-time measurement systems until 1985. Since 1985 he has been a Researcher with the Institute of Signal Processing and Communication Theory, Department of Electrical Engineering and Computer Science. In 1989, he became a member of the Center for Sensor Systems (ZESS), which is a central scientific research establishment at the University of Siegen. He also became the general management position of the ZESS in 1989. He founded a research group on Embedded System Design and Image Processing. He cofounded also some spinoffs. His current research interests comprise multisensorysystems, real-time processing for data fusion, sensor networking and image processing.

Otmar Loffeld received the Diploma degree in electrical engineering from the Technical University of Aachen, Aachen, Germany, in 1982 and the Eng. Dr. degree and the Habilitation degree in the field of digital signal processing and estimation theory from the University of Siegen, Siegen, Germany, in 1986 and 1989, respectively. In 1991, he was appointed Professor of digital signal processing and estimation theory with the University of Siegen. Since then, he has been giving lectures on general communication theory, digital signal processing, stochastic models and estimation theory, and synthetic aperture radar. He is the author of two textbooks on estimation theory. In 1995, he became a member of the Center for Sensor Systems (ZESS), which is a central scientific research establishment at the University of Siegen. Since 2005, he has been the Chairman of ZESS. In 1999, he became a Principal Investigator (PI) on baseline estimation for the X-band part of the Shuttle Radar Topography Mission (SRTM), where ZESS contributed to the German Aerospace Center (DLR)s baseline calibration algorithms. He is a PI for interferometric techniques in the German TerraSAR-X mission, and, together with Prof. Ender from FGAN, he is one the PIs for a bistatic spaceborne airborne experiment, where TerraSAR-X serves as the bistatic illuminator, whereas the Fraunhofer Institute for High Frequency Physics and Radar Techniques (FHR)s PAMIR system mounted on a Transall airplane is used as a bistatic receiver. In 2002, he founded the International Postgraduate Programme (IPP) Multi Sensorics. In 2008, based on the aforementioned program, he established the NRW Research School on Multi Modal Sensor Systems for Environmental Exploration and Safety (MOSES) at the University of Siegen as an upgrade of excellence. He is the Speaker and Coordinator of both doctoral degree programs, which are hosted by ZESS. Furthermore, he is the universitys Scientific Coordinator for Multidimensional and Imaging Systems. His current research interests comprise multisensor data fusion, Kalman filtering techniques for data fusion, optimal filtering and process identification, SAR processing and simulation, SAR interferometry, phase unwrapping, and baseline estimation. His recent field of interest is bistatic SAR processing. Dr. Loffeld is a member of the Information Technology Society (ITG) of the German Association for Electrical, Electronic and Information Technologies (VDE) and a Senior Member of the IEEE Geoscience and Remote Sensing Society. He was the recipient of the Scientific Research Award of North Rhine- Westphalia (Bennigsen-Foerder Preis) for his works on applying Kalman filters to phase estimation problems such as Doppler centroid estimation in SAR, phase, and frequency demodulation.