High Performance Time-of-Flight and Color Sensor Fusion with Image-Guided Depth Super Resolution

Hannes Plank*, Gerald Holweg*, Thomas Herndl*, Norbert Druml*
*Infineon Technologies Austria AG, Graz, Austria
{hannes.plank, gerald.holweg, thomas.herndl, norbert.druml}@infineon.com

***Abstract*—In recent years, depth sensing systems have gained popularity and have begun to appear on the consumer market. Of these systems, PMD-based Time-of-Flight cameras are the smallest available and will soon be integrated into mobile devices such as smart phones and tablets. Like all other available depth sensing systems, PMD-based Time-of-Flight cameras do not produce perfect depth data. Because of the sensor's characteristics, the data is noisy and the resolution is limited. Fast movements cause motion artifacts, which are undefined depth values due to corrupted measurements. Combining the data of a Time-of-Flight and a color camera can compensate these flaws and vastly improve depth image quality.**

This work uses color edge information as a guide so the depth image is upscaled with resolution gain and lossless noise reduction. A novel depth upscaling method is introduced, combining the creation of high quality depth data with fast execution. A high end smart phone development board, a color, and a Time-of-Flight camera are used to create a sensor fusion prototype. The complete processing pipeline is efficiently implemented on the graphics processing unit in order to maximize performance. The prototype proves the feasibility of our proposed fusion method on mobile devices.

The result is a system capable of fusing color and depth data at interactive frame rates. When there is depth information available for every color pixel, new possibilities in computer vision, augmented reality and computational photography arise. The evaluation shows, our sensor fusion solution provides depth images with upscaled resolution, increased sharpness, less noise, less motion artifacts, and achieves high frame rates at the same time; thus significantly outperforms state-of-the-art solutions.

***Index Terms*—Time-of-Flight, 3D sensing, sensor fusion, GPGPU, image processing**

I. Introduction

Indirect depth sensing based on Time-of-Flight works by emitting a modulated light signal. As shown on Fig. 1, the light signal is reflected by the scene and captured by the Time-of-Flight sensor. The Time-of-Flight sensor measures the phase difference of the incoming to the outgoing light for each pixel. A Photonic Mixer Device (PMD) on each pixel enables the conversion of the phase to a voltage, which is proportional to the depth. These devices occupy a lot of space on the sensor and limit the image resolution. In addition, various systematic errors are introduced during the measurement process, causing not all pixels to contain valid depth values. The most common is the low-signal error, when not sufficient light is reflected. This is also the main noise source. Another inherent weakness of the PMD-based Time-of-Flight technology is motion artifacts. These artifacts are caused by a fast moving scenery and due to the fact that several raw images need to be taken consecutively (each with a certain exposure time) in order to compute one depth image. There are numerous countermeasures for these errors, but using additional information is the best way to cope with these weaknesses. While a combination of several different depth sensing systems makes sense and has been thoroughly researched, practical applications are limited due to the increased complexity. Fusing depth and color can enhance depth resolution and reduce noise. High-resolution color cameras are ubiquitous on mobile devices like smart phones and tablets. Research has shown that depth image quality can be improved immensely by using color information, as shown by the authors in [1].

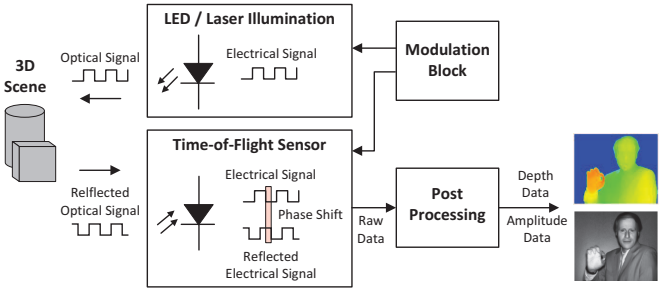

Fig. 1. The working principle of an indirect PMD-based Time-of-Flight depth sensing system. Obtained with changes from [2].

Color images offer additional information about the depth data, because there is a correlation between depth and color discontinuities. When upscaling a depth image, depth discontinuities can adapt to color discontinuities, improving the resolution and reducing noise. When every color image pixel is associated with a depth value, new applications for depth sensors arise. Fields like augmented reality, computational photography, and computer vision in general can benefit of a unified color and depth sensor.

However, there is major gap in literature concerning efficient sensor fusion solutions targeting mobile devices. This paper tackles this gap and proposes a novel solution which not only improves the resolution of Time-of-Flight depth images, but also performs loss-less noise reduction, reduces motion artifacts, and maintains interactive frame-rates by exploiting massive parallel processing. Summarizing, this paper makes the following contributions:

- It introduces a novel depth upscaling approach, designed to unify depth image refinement with high computation

978-3-9815370-6-2/DATE16/

performance.

- It demonstrates a feasible prototype implementation on a mobile platform using a combined Time-of-Flight and color camera system.
- It presents an evaluation which not only provides a comparison to other upscaling methods but also shows the feasibility of depth and color sensor fusion on mobile platforms.

II. Related Work

Fusing depth information with other camera data is well-explored. A lot of contributions focus on fusing different depth sensing systems, cf. [3]. By fusing Time-of-Flight cameras with stereo systems it is possible to accelerate the stereo matching computation by limiting the search space with Time-of-Flight depth measurements. In contrast to passive stereo systems, Time-of-Flight depth sensing can measure depth on homogenous surfaces and during bad lighting conditions.

Fusing Time-of-Flight depth data with any camera system requires calibration. In [4], Herrera et al. propose a method for calibrating a depth and color camera system. It is however possible to reduce the calibration procedure to a 2D stereo calibration, cf. [5], because recent Time-of-Flight cameras are also able to capture a 2D gray-scale image. This enables the usage of the well-established Bouguet Matlab calibration toolbox.

Since depth cameras usually offer lower image resolutions than RGB cameras, improving the resolution of the depth sensor is a well researched topic. It is possible to increase the depth resolution by regarding color discontinuities. In [1], Langmann et al. review a number of approaches. The joint bilateral filter by Kopf et al. [6] is a popular upsampling method. It is a modification of the original bilateral filter by Tomasi [7] and works by weighting the depth image with the color distance of the color image. Sung et al. [8] extend the joint bilateral filter with color segmentation based boundary refinement. Yang et al. [9] construct a cost volume and apply the bilateral filter. The best cost is then selected and the process is applied iteratively.

A different way to fuse depth and color data is global optimization. These global approaches formulate a cost function, which is numerically minimized, cf. [5] and [10]. This leads to the best results among depth super resolution methods but the optimization is computationally complex. In [11], Dai et al. promise a high performing global approach. The algorithm interprets the pixel-grid as graph with intensity differences as edge weights. The minimum spanning tree (MST) is calculated and the sparsely mapped depth values are interpolated along the tree. The drawback of this method is that the MST computation is hard to implement efficiently on a GPU, as described by the authors in [12].

The demand for an efficient color and depth fusion method, capable of running on limited hardware formed just recently, when depth sensors became available on mobile platforms. In [13], Chuchvara et al. propose a GPU based sensor fusion method, which creates a surface mesh from the mapped depth data. This mesh is rendered to a frame-buffer with OpenGL, and a non-local means based noise reduction method is executed. This method uses the GPU hardware with OpenGL efficiently for interpolation, but does not increase the depth resolution.

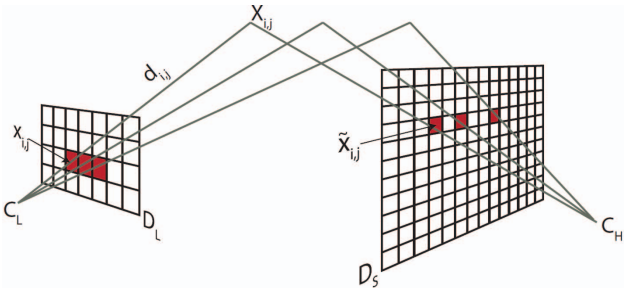

Fig. 2. Depth data mapping from depth D_L to color camera D_S image space. Obtained with changes from [5].

Despite the vast amount of research, there is major gap in literature concerning efficient sensor fusion solutions for mobile devices, which is addressed in our work.

III. Depth Image Mapping

The desired sensor fusion result is a color image with depth associated to each pixel. As a first step, it is necessary to warp the depth data to the color image. Since warping introduces undefined areas, the cameras are mounted as close as possible. Several images of a checkerboard target in various orientations and distances are captured for camera calibration. The checkerboard pattern enables the calibration software to derive edge reference points with sub-pixel accuracy. The Time-of-Flight camera is able to capture infrared-based gray-scale images, enabling to calibrate the cameras like a stereo system. The Bouguet Matlab Calibration toolbox was used to compute the intrinsic and extrinsic camera parameters.

After distortion correction, the depth values with pixel position $x_{i,j}$ and depth $d_{i,j}$ are transformed to 3D points $X_{i,j}$ in color camera space, as defined in (1) and as illustrated in Fig. 2. $\tilde{P}_D$ denotes the pseudoinverse of the depth camera calibration matrix. The extrinsic camera parameters in form of rotation matrix R and translation vector T directly transform the depth measurements to the 3D space of the color camera with center C_H.

$$X_{i,j} = T + Rd_{i,j}\frac{\tilde{P}_D x_{i,j}}{\|\tilde{P}_D x_{i,j}\|} \tag{1}$$

The homogenous 3D depth coordinates $X_{i,j}$ are projected to the 2D image space D_S of the color camera by a multiplication with the color camera projection matrix P_C, cf. (2). This leads to warped depth measurements with the coordinates $\tilde{x}_{i,j}$ in color image space.

$$\tilde{x}_{i,j} = P_C \cdot X_{i,j} \tag{2}$$

After applying the lens distortion of the color camera, the depth coordinates are associated with the nearest color pixel.

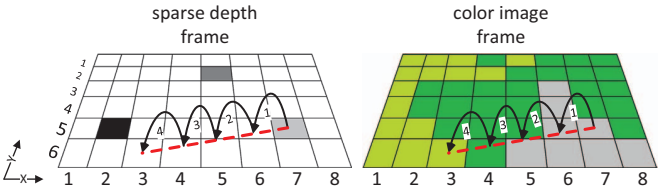

Fig. 3. Pixel parsing along circular paths for each depth value.

Reducing the noise of the depth image with an edge-preserving 3x3 median filter before warping benefits the final depth image greatly while not causing significant computation overhead. Alternatively, other well-established depth image denoising techniques can be applied. The next step after the depth image mapping is to interpolate this sparsely mapped depth frame by using color information.

IV. Image-Guided Depth Upsampling

The depth upscaling is the most computational complex step in the sensor fusion procedure. With a well-designed algorithm, the sparse depth data on the color image is not just interpolated, but also denoised and the resolution refined. This work targets to unify the creation of a high quality 1-1 mapped color and depth image, while maintaining high computation performance. Data parallelism with as less mutual data dependence as possible is desired for an efficient GPU implementation. It is also important to consider memory usage and addressing, as GPU memory operations are slow.

A preparation step before the upscaling is extracting edge information from the color image. While any edge detector can be used, the Sobel operator proved to be simple and effective. If the Sobel operator is not just applied to the intensity of the image but also to the saturation, then the edges between surfaces with equal brightness but different colors are detected as well.

The edge image and the sparsely mapped depth frame form the input data for the upscaling algorithm which is executed in parallel for each mapped depth value. The algorithm can be interpreted as the propagation of depth value influence over a local circular area on the color image. A radial path from the mapped depth value position to each influenced pixel is parsed.

Fig. 3 illustrates an example of a depth influence propagation path. The influence of a depth value on the output depends on the sum of edge values S_E between the mapped depth value (at 7/5) and each pixel. The color edges crossed at step 3 and 4 decrease the depth influence. Each pixel of the resulting high-resolution depth image is influenced by several depth values which are then interpolated by their propagated influence. The interpolation weight w quantifies the influence of a depth value. The requirement of a function to convert S_E to w, is that small changes of S_E lead to large changes of w. Depth values need to be restricted from propagating over edges if there are valid depth measurements on the other side of the edge. The natural exponential function e^x suits this purpose well, cf. (3).

$$w = e^{\frac{-S_E}{\sigma}} \tag{3}$$

The parameter σ scales the argument of the exponential function, in order to restrict the range of w to the boundaries of the 32bit floating point data type. The k depth values d_n which influence pixel i are interpolated with the interpolation formula by using the ratios of the weights w_n, resulting in the final depth d_i, cf. (4).

$$d_i = \frac{\sum_{n=1}^{k} d_n \cdot w_n}{\sum_{n=1}^{k} w_n} \tag{4}$$

Depth values propagate their influence to the influence sphere of other depth values to reduce noise. Since color edges greatly reduce the influence, the noise reduction is edge preserving.

A. Efficient Processing

The introduced algorithm can be efficiently implemented with some modifications. The interpolation formula in the last section can be reformulated as update formula. d_{sum} and w_{sum} are stored for each pixel and replace the upper and lower fraction terms in the interpolation formula. They are updated every time a depth value operates on a pixel, cf. (5) and (6).

$$d_{sum} = d_{sum} + d_n \cdot w_{new} \tag{5}$$

$$w_{sum} = w_{sum} + w_{new} \tag{6}$$

The final depth d_i is calculated by dividing d_{sum} and w_{sum} on each pixel i, as given by (7).

$$d_i = \frac{d_{sum}}{w_{sum}} \tag{7}$$

This operation requires at least 3 GPU memory access operations per pixel, which can be reduced. If d_i is calculated and updated during the depth influence propagation, d_{sum} can be extracted by multiplying d_i with w_{sum}. Without storing d_{sum} for each pixel, the depth d_i and w_{sum} are updated at each pixel operation according to (8)-(10).

$$w = e^{\frac{-S_E}{\sigma}} \tag{8}$$

$$w_{sum} = w_{sum} + w_n \tag{9}$$

$$d_i = \frac{d_i \cdot (w_{sum} - w_n) + d_n \cdot w_n}{w_{sum}} \tag{10}$$

To satisfy the requirement that S_E is the sum of all edges along the pixel path between pixel i and the mapped depth value d_n, the surrounding pixels need to be parsed in a specific order. Fig. 4 illustrates the dependence hierarchy, caused by the radial parsing principle. Each newly processed pixel depends on its predecessor along the pixel path. Calculating the pixel paths during computation is avoided by a lookup table. The table stores the relative pixel positions on the paths in the correct order.

A small array is used to buffer the sum of edge weights on the path S_E for each pixel, so it can be looked up during the

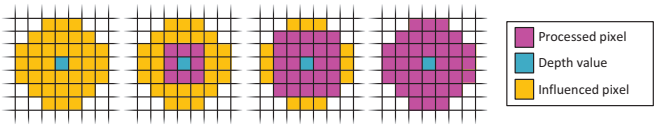

Fig. 4. The pixel parsing hierarchy of each mapped depth value.

next pixel operation on the path. This averts the prerequisite to parse the pixels along the radial pixel paths, as long the hierarchy shown in Fig. 4 is satisfied. As a consequence, the lookup table can be cleaned of all redundant operations, so just n pixel operations for n influenced pixels are required.

A further optimization is to selectively abort calculations on a pixel path. When a pixel path is classified as irrelevant, the latest buffered value of S_E is marked invalid, and all further pixel processing operations are canceled. The abortion criteria is satisfied, when a depth value does not have a significant influence on the final depth. This can happen when a pixel path crosses too many edges and its influence is minuscule compared to the other depth values in the area. Irrelevant paths are detected, by comparing the already existing weights w_n of a pixel to the current weight. If the difference is below a certain value γ, the remaining operations on the current pixel path are canceled. The evaluation showed that stopping the processing on irrelevant influence propagation paths also improves the quality of the upscaled depth image as the irrelevant pixel paths introduce an error to the upscaled depth image.

B. GPU Implementation

In our prototype, the complete sensor fusion procedure is implemented on the graphics processing unit, using OpenCL. The system features a color and Time-of-Flight camera system as well as a high-end smart phone development kit for computation. The camera system is shown in Fig. 6 and consists of a Logitec B910 color camera and an Infineon REAL3™ sensor based on Time-of-Flight technology of pmdtechnologies. With this prototype, the ability of fusing depth and color while maintaining interactive frame-rates at same time is demonstrated. An Android application implements the framework for handling the cameras and Android NDK is employed to handle the GPU computation.

The processing pipeline consists of 3 GPU programs, for mapping the depth data, calculating the edge image, and upscaling the depth data with the proposed method. The upscaling computation is by far the most computational intensive operation, requiring 95% of the execution time.

Because the parameter space of the local OpenCL workgroup size is restricted to a few dozen configurations, the implementation can optimize itself on a large variety of GPU devices. Local memory buffering was avoided in favor for optimized cache usage. This avoids any synchronization mechanisms and allows to use all computation units simultaneously without data corruption.

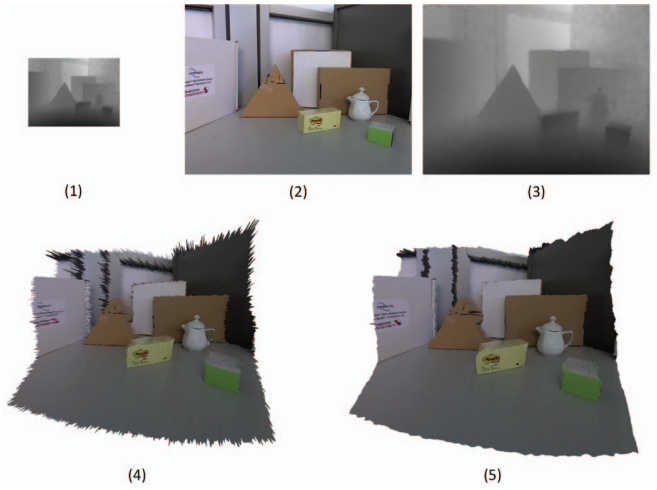

Fig. 5. 1: Original depth image, captured with the Time-of-Flight camera; 2: Color image; 3: The resulting mapped and upscaled depth image; 4 and 5: 3D visualization of raw depth data compared to our upsampling method

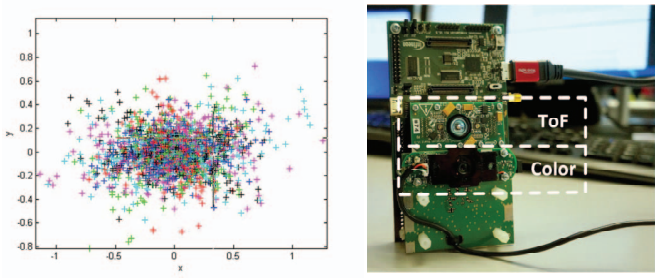

Fig. 6. 1: The reprojection error of the calibration target; 2: The color and depth camera system.

V. Results and Evaluation

A. The Prototype System

One of the goals of this work was to proof that state-of-the art mobile devices are capable of running the proposed sensor fusion method at interactive frame rates. The prototype is capable of fusing 288x256 depth images with 640x480 color images at a rate of 10 frames per second (fps). Fig. 5 shows a scene captured and processed by the prototype. The 3D visualization clearly shows the significant quality impact of our proposed depth interpolation method. The evaluation in Section V-D reveals, the upscaling algorithm can also handle large resolutions, up to several megapixel.

As mentioned in Section III, the camera system is calibrated like a stereo camera system, using the 2D amplitude image of the Time-of-Flight camera. Despite the low depth resolution, it is possible to calibrate the system properly. The reprojection error of the interest points on the 46 calibration images is visualized in Fig. 6. It can be observed that the vast majority of reprojected calibration interest points deviate by less than one pixel. A reprojection error below 0.5 pixel is insignificant,

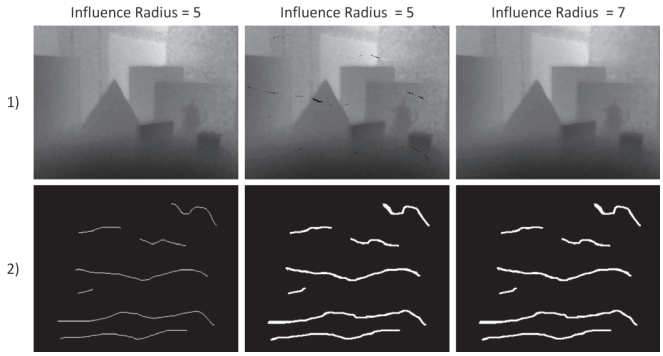

Fig. 7. Interpolation of invalid depth values. 1: The upscaled depth image; 2: The masks which were used to simulate invalid measurements due to, e.g., motion artifacts.

as the mapped depth values are associated with the closest color pixel.

B. Motion Artifacts Compensation

Another important aspect of depth image refinement is the removal of motion artifacts. Advanced Time-of-Flight cameras emit light pulses with different phases and modulation frequencies per depth measurement. Fast movements during this procedure can cause invalid depth measurements. While cameras are able to detect them, it is up to the postprocessing pipeline to interpolate the missing information. The proposed depth upscaling algorithm interpolates these invalid measurements without modifications. As Fig. 7 shows, even larger motion artifacts are compensated if the influence radius of the depth values is increased. Since the cameras are not synchronized, the motion artifacts in Fig. 7 are simulated by creating a mask.

C. Performance

Our prototype system is capable of fusing the sensor data at a rate of 10 fps, when executed on the graphics processor on the mobile platform. In comparison, the sensor fusion procedure runs at a rate 126 fps on an Nvidia Geforce GTX 550 TI desktop GPU.

The depth upscaling stage is the most computational intensive stage, taking 96% of the computation time. The first stage, the depth value mapping takes less than 1%, and the edge detection takes approximately 3% of the computation time. Since the parameter space for the OpenCL local work-group size parameter is limited due to the image dimensions, a brute-force approach quickly determines the optimal configuration. This ensures optimal performance among a wide range of devices.

The complexity class of the upscaling algorithm is $O(k \cdot n_{color})$ where n_{color} is the number of color pixels and k the average number of depth values influencing a color pixel. k depends directly on the influence radius and number of mapped depth pixels. The higher the value of k, the better the noise reduction.

D. Comparison with other depth upscaling methods

With available ground-truth depth data, it is possible to quantify the depth image quality refinement. By calculating the difference between the upscaled depth image and a ground-truth reference image, depth upsampling algorithms can be compared. The popular Middlebury 2006 dataset [14] consists of color and high-quality depth images. By downscaling and adding noise, realistic depth input data is created. Beside the root mean square error (RMSE), the percentage of bad matching pixels is commonly used to evaluate depth upscaling and stereo matching algorithms [15]. The percentage of bad matching pixel (PbmP), is a measure for the number of pixels differing more than a defined tolerance δ from the ground-truth, as given by (11).

$$PbmP = \frac{1}{N} \sum_{(x,y)} (|d(x,y) - g(x,y)| > \delta) \qquad (11)$$

Our evaluation uses the common value 1 for δ, assuming the depth values are mapped to a 0-255 value range.

The proposed solution is compared to simple nearest neighbor and bilinear interpolation, joint bilateral filtering [6] (Kopf et al.), minimax path based interpolation [11] (Dai et al.), and anisotropic total variation upscaling [5] (Ferstl et al.). The joint bilateral filter was selected for comparison because it is a popular and fast upscaling method. It however needs a lot more operations per pixel compared to our proposal and works best if applied iteratively. As our evaluation shows, the joint bilateral filter is significantly outperformed, especially when the upscaling factor is large. The minimax path based upscaling algorithm showed promising results. One of the conditions for its proclaimed fast performance is that every color pixel is just influenced by two depth values. However, due to this fact, it is not capable to reduce depth noise. The anisotropic total variation upscaling method is based on energy minimization. It leads to better results, but is not suitable for real-time applications due to its global approach. We compare with this method to demonstrate what is possible with global optimization, and how the other faster methods perform.

Three scenes of the Middlebury 2006 dataset were selected, and the ground-truth depth images were downscaled by the factor of 8 and 16. Gaussian noise was added and the depth images were upscaled with the selected algorithms. The Aloe dataset was specifically selected, because it features a salient texture on flat surfaces, which misguides a lot of depth upscaling algorithms. Table I shows the root mean square error and the percentage of bad matching pixel for three scenes of the Middlebury 2006 dataset.

Besides the quantitative comparison, it is also interesting compare the visual results. Fig. 8 shows the resulting depth image of the previously compared algorithms.

VI. Conclusion

Sensor fusion and Time-of-Flight 3D imaging in general is a fast-paced field of research. Fusion of depth and color data on mobile platforms will enable whole new fields of applications.

	Aloe				Art				Moebius			
	8x		16x		8x		16x		8x		16x	
	RMSE	PbmP	RMSE	PbmP	RMSE	PbmP	RMSE	PbmP	RMSE	PbmP	RMSE	PbmP
Nearest Neighbor	4.89	77.76	5.96	77.14	5.73	78.49	7.50	80.47	4.91	77.83	5.62	78.42
Bilinear	4.21	72.17	5.32	73.27	4.98	74.50	6.79	77.24	4.18	73.05	4.92	74.26
Kopf et al. [6]	3.45	57.13	5.47	59.35	4.20	59.41	3.00	54.75	2.89	54.37	4.05	56.65
Dai et al. [11]	3.81	59.72	5.03	69.30	4.06	61.87	4.11	58.58	4.25	64.79	4.03	68.44
Ferstl et al. [5]	2.28	31.69	3.50	49.94	2.83	38.98	1.92	27.89	2.66	52.45	2.33	41.53
Ours	3.36	54.31	4.99	65.44	3.59	53.28	2.59	47.82	2.59	47.82	3.94	63.70

TABLE I
COMPARISON OF UPSCALING ALGORITHMS ON THE MIDDLEBURY 2006 DATASET, USING THE ROOT MEAN SQUARE ERROR (RMSE) AND THE PERCENTAGE OF BAD MATCHING PIXEL (PBMP).

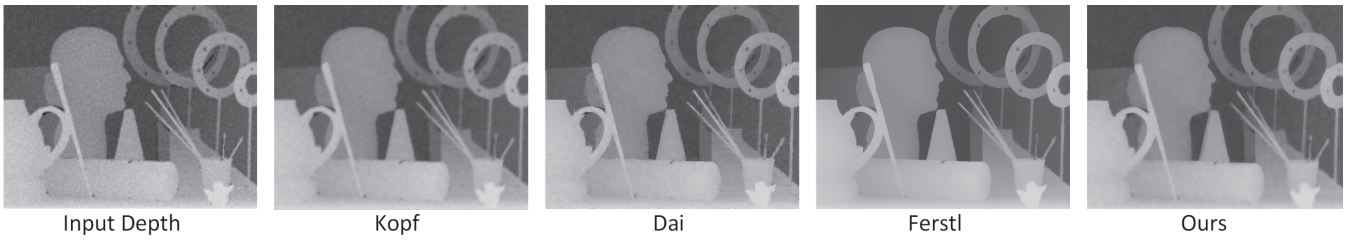

Fig. 8. Visual comparison of depth upscaling methods on the Middlebury 2006 dataset Moebius.

Currently, there are several sensor fusion approaches which can increase depth resolution and reduce noise, but the design goal of reaching high performance on mobile devices was neglected so far.

This work demonstrates the feasibility of high-performance and high-quality depth and color sensor fusion targeting mobile devices by employing a state-of-the-art smart phone development platform. It uses an Infineon REAL3™ image sensor based on Time-of-Flight technology of pmdtechnologies, which is on the brink of being integrated into new generations of smart phones and tablets (cf. Google Tango project). The output of this system is a color and depth image, where each color pixel is directly associated with a depth pixel.

The presented innovative method works non-iteratively and has an $O(n \cdot k)$ complexity. Our sensor fusion solution provides depth images with upscaled resolution, increased sharpness, less noise, less motion artifacts, and achieves high frame rates at the same time. Given these facts, this work significantly outperforms state-of-the-art and thus sets new benchmarks in the field of mobile computing and will enable a set of new types of mobile applications.

Our future work concerns the development of an interface abstracting the color and Time-of-Flight camera into a unified RGB-D sensor. This involves further optimizations, resolution scalability, camera synchronization, and auto-calibration.

ACKNOWLEDGMENTS

The authors would like to thank the Austrian Federal Ministry for Transport, Innovation and Technology as well as the ARTEMIS Joint Undertaking, which funded the EMC^2 project under the grant agreement n° 621429.

REFERENCES

[1] B. Langmann, K. Hartmann, and O. Loffeld, "Comparison of Depth Super-Resolution Methods for 2D/3D Images," *International Journal of Computer Information Systems and Industrial Management Applications*, vol. 3, pp. 635 –645, 2011.

[2] N. Druml, G. Fleischmann, C. Heidenreich, A. Leitner, H. Martin, T. Herndl, and G. Holweg, "Time-of-Flight 3D Imaging for Mixed-Critical Systems," in *13th International Conference on Industrial Informatics (INDIN)*, July 2015, pp. 1432–1437.

[3] R. Nair, K. Ruhl, F. Lenzen, S. Meister, H. Schfer, C. Garbe, M. Eisemann, M. Magnor, and D. Kondermann, "A Survey on Time-of-Flight Stereo Fusion," *Lecture Notes in Computer Science*, vol. 8200, pp. 105–127, 2013.

[4] D. Herrera C., J. Kannala, and J. Heikkil, "Accurate and Practical Calibration of a Depth and Color Camera Pair," in *Computer Analysis of Images and Patterns*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, vol. 6855, pp. 437–445.

[5] D. Ferstl, C. Reinbacher, R. Ranftl, M. Ruether, and H. Bischof, "Image Guided Depth Upsampling Using Anisotropic Total Generalized Variation," in *IEEE International Conference on Computer Vision (ICCV)*, December 2013, pp. 993–1000.

[6] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint Bilateral Upsampling," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 3, July 2007.

[7] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Sixth International Conference on Computer Vision*, January 1998, pp. 839–846.

[8] S.-Y. Kim, J.-H. Cho, A. Koschan, and M. Abidi, "Spatial and Temporal Enhancement of Depth Images Captured by a Time-of-Flight Depth Sensor," in *20th International Conference on Pattern Recognition (ICPR)*, August 2010, pp. 2358–2361.

[9] Q. Yang, R. Yang, J. Davis, and D. Nister, "Spatial-Depth Super Resolution for Range Images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007, pp. 1–8.

[10] J. Park, H. Kim, Y.-W. Tai, M. Brown, and I. Kweon, "High quality depth map upsampling for 3D-TOF cameras," in *IEEE International Conference on Computer Vision (ICCV)*, November 2011, pp. 1623–1630.

[11] L. Dai, F. Zhang, X. Mei, and X. Zhang, "Fast Minimax Path-Based Joint Depth Interpolation," *IEEE Signal Processing Letters*, vol. 22, no. 5, pp. 623–627, May 2015.

[12] V. Vineet, P. Harish, and P. J. Narayanan, "Large Graph Algorithms for Massively Multithreaded Architectures," 2009.

[13] A. Chuchvara, M. Georgiev, and A. Gotchev, "A speed-optimized RGB-Z capture system with improved denoising capabilities," *SPIE Proceedings*, vol. 9019, March 2014.

[14] H. Hirschmuller and D. Scharstein, "Evaluation of Cost Functions for Stereo Matching," in *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, June 2007, pp. 1–8.

[15] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *International Journal on Computer Vision*, vol. 47, no. 1-3, pp. 7–42, April 2002.