

Data-Fusion of PMD-Based Distance-Information and High-Resolution RGB-Images

Marvin Lindner*, Andreas Kolb* and Klaus Hartmann†

*Computer Graphics Group, Institute for Vision and Graphics, University of Siegen, Germany

Email: {marvin.lindner | andreas.kolb}@uni-siegen.de

†Centrum for Sensor Systems (ZEISS), University of Siegen, Germany

Email: hartmann@zess.uni-siegen.de

Abstract—An important field of research in computer vision is the 3D analysis and reconstruction of objects and scenes for e.g. position determination, online object recognition or collision prevention. Known techniques thereby mainly rely on measuring devices such as laser scanners, stereo camera systems, or comparable algorithms like structure from motion.

The rather new *Photonic Mixing Device (PMD)* technique is based on the time-of-flight principle and measures full-range distance information in real-time. Unfortunately, PMD-based devices have still limited resolution (e.g. 160×120 px) and provide only graylevel information.

This paper describes a fast algorithmic approach to combine high resolution RGB images with low resolution PMD distance data, acquired using a binocular camera setup. The approach relies completely on fast parallel *Graphics Processing Units (GPUs)* resources. Additionally, we introduce a simple but efficient method to detect insufficient color assignments, which are due to the binocular setup.

The resulting combined RGBZ-data not only enhances the visual result, but also represents a basis for advanced data processing in e.g. object recognition.

I. INTRODUCTION

In automatization areas like robotics or automotive, the reconstruction of objects and scenes is a necessary fundamental with respect to computer vision. Information obtained from digitized scenes represent important input data for position determination, online object recognition, or collision prevention.

In this context, typically expensive and complex setups including laser scanners or stereo vision systems are used for distance measurements. However, laser scanning techniques which sample a scene row by row with a single laser device are rather time-consuming and thus impracticable for dynamic scenes. Stereo vision camera systems, on the other hand, completely rely on the fast identification of corresponding points. Mismatching correspondences especially in homogeneous object regions leads to gaps in the reconstruction.

A rather new and promising approach developed in recent years estimates the distance by time-of-flight measurements for modulated, incoherent light based on the new Photo Mixing Device (PMD) technology [1], [2], [3]. A technology which even works for outdoor scenes. Integrating a PMD camera into

a vision system, full-range distance information is available in real-time, i.e. with up to 20 fps. Nevertheless, there are still some problems to be solved. One them is the relatively low sensor resolution. Thus, many algorithms known from image processing cannot be applied to PMD data directly. Another drawback is that, currently, the PMD camera does not provide color information. Only an intensity image, correlating with the incident active light is available.

The contribution of this paper is a combined 2D-3D vision system, consisting of a PMD and a RGB-camera. Our approach improves the capabilities of downstream processing tasks like object recognition. We incorporate a detection of color misassignment, which occurs in areas hidden to the RGB camera but visible to the PMD. Our technique completely runs on commodity Graphics Processing Unit (GPU), thus relieving the CPU from these basic pre-processing steps.

A short overview of the PMD's functionality and known related work is given in Sec. II and Sec. III. The design of our fusion model in general and its difference to existing models is described in Sec. IV. The problem of occlusion and hidden surfaces is treated in Sec. V. Finally, the results are discussed in Sec. VI which leads to a short conclusion of the presented work in Sec. VII.

II. PMD

The *Photonic Mixing Device (PMD)* technology is based on a time-of-flight approach. Here, the scene is illuminated with modulated infrared light. The reflected light is gathered in a smart pixel sensor [1], [2], [3]. By sampling and correlating the incoming optical signal with the reference signal of the modulated, incoherent illumination directly on a pixel, the PMD is able to determine the signal's phase shift and thus the distance to the according object region.

Given a reference signal $g(t)$ and the optical signal $s(t)$ incident to a PMD pixel, the pixel samples the correlation function $c(\tau)$ for a given internal phase delay τ :

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

For a sinusoidal signal $g(t) = \cos(\omega t)$ and the optical response signal $s(t) = k + a \cos(\omega t + \phi)$ basic trigonometric calculus yields (see Lange [2] for more details):

$$c(\tau) = k + \frac{a}{2} \cos(\omega\tau + \phi)$$

where ω is the modulation frequency, a and k are the amplitude and the offset of the correlation function, respectively, and ϕ is the phase shift corresponding to the object distance. The modulation frequency defines the distance unambiguousness of the distance sensing. The demodulation of the correlation function is done using several samples of $c(\tau)$ obtained by four sequential PMD raw images $A_i = c(\tau_i)$ using internal phase delays $\tau_i = i \cdot \frac{\pi}{2}$, $i = 0, \dots, 3$:

$$\begin{aligned} \phi &= \arctan\left(\frac{A_3 - A_1}{A_0 - A_2}\right) \\ a &= \frac{\sqrt{(A_3 - A_1)^2 + (A_0 - A_2)^2}}{2} \\ h &= \frac{A_0 + A_1 + A_2 + A_3}{4}. \end{aligned} \quad (1)$$

The amplitude a is a measure for the quality of the distance measure.

The manufacturing of a PMD chip is based on standard CMOS-processes which allows a very economic production of the device which is, due to an automatic suppression of background light, suitable for indoor as well as for outdoor scenes. Current devices provide a resolution of 48×64 or 160×120 px at 20 Hz. A common modulation frequency is 20 MHz, resulting in an unambiguous distance range of 7.5 m.

III. RELATED WORK

So far, there exist only a few distinguished approaches to combine PMD distance information with high-resolution RGB image data, differing in their particular realization concerning hardware and software implementation.

Prasad et al. [4] describe a hardware-based realization, which combines a traditional CCD and a PMD chip in a monocular device using an optical splitter. Due to the monocular setup, no special mapping transformation between both images has to be done as both images consist of the same view. However, a known disadvantage of this approach consists of the used beam splitter. By deviding the incoming optical signal into two separate parts, the optical power is slightly affected, which influences the underlying measuring process of the distance data.

Another approach using the combination of a 2D- and a 3D camera has been presented by Reulke [5]. Unlike the hardware-based realization, a binocular camera setup is used, as in our case. The software-based data processing consists of a radial undistortion and viewing transformations. The final data fusion is done by an algorithm called *orthophoto generation* [6]. Here, the 2D image is distorted in order to eliminate the perspectivity of the image by taking the 3D information into account. The result is an orthographic image, where the optical rays are parallel. After the image

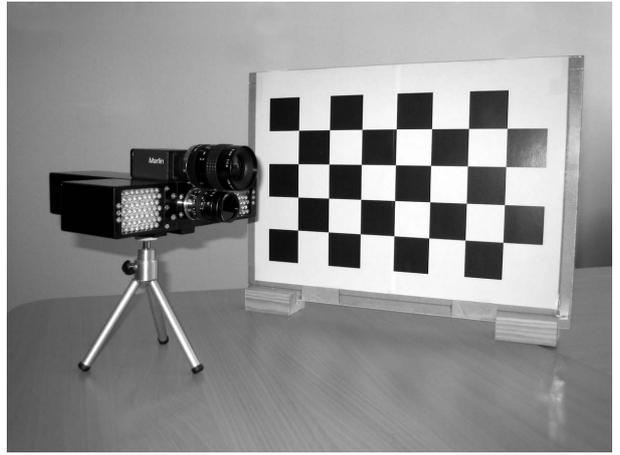


Fig. 1. 2D/3D camera setup consisting of a 19k PMD and a high-resolution CCD-camera. On the right site an exemplary calibration panel can be seen.

rectification, the data fusion is straight forward applying a parallel mapping of the color information onto the appropriate distance data.

IV. DATA FUSION

Unlike the approach taken by Reulke [5], our 2D-3D data fusion approach is based on projective texture maps rather than on orthophoto generation. Projective textures have first been introduced by Segal et al. [7]. Here, the image of the RGB camera is assumed to be projected onto the geometry provided by the PMD camera distance information.

A. Camera Setup and Calibration

As already mentioned, we use a binocular camera setup, by mounting a standard 2D camera on top of the PMD distance sensor. Due to radial lens distortion, additional image undistortion of both images has to be performed in order to prevent a false mapping between both camera views. The required intrinsic parameters, distortion coefficients as well as extrinsic parameters, necessary for the pose estimation of both cameras, have been computed using Intel's computer vision library OpenCV [8].

Unlike camera tracking, where the extrinsic parameters of each camera have to be determined for every frame, the camera setup used for data fusion has been fixed (see Fig. 1). Thus, in order to express the RGB camera position and orientation in the PMD camera coordinate system, the rotation and translation parameters of each camera has to be determined only once during an initial registration step.

B. Projective Texturing on the GPU

Having the acquired 2D image together with the object geometry given from the PMD sensor, the correct RGB image coordinate S^{RGB} , i.e. *texture coordinate*, for a point on the geometry can be determined by projecting the according point onto the image plane of the RGB camera (see Fig. 2).

To achieve this, the portion of the scene observed by the PMD sensor is first reconstructed in global world coordinates. Through this, each PMD pixel becomes a 3D point in space consisting of a homogeneous coordinate $P = (x, y, z, w)^T$, $w = 1$.

Next, every vertex point P is transformed into the coordinate system of the RGB camera and perspective projected by

$$P^{\text{RGB}} = T_{\text{proj}}^{\text{RGB}} \cdot T_{\text{PMD} \rightarrow \text{RGB}} \cdot P \quad (2)$$

where $T_{\text{PMD} \rightarrow \text{RGB}}$ is an affine viewing transformation determined by the extrinsic parameter acquired during the registration step and $T_{\text{proj}}^{\text{RGB}}$ is the perspective projection specified by the intrinsic parameter of the RGB camera.

Finally, the RGB image coordinates $S^{\text{RGB}} = (x^{\text{RGB}}, y^{\text{RGB}})$ for the vertex point P are given by the (x, y) -component of the normalized coordinate $P^{\text{RGB}}/w^{\text{RGB}}$.

For many applications, it is not sufficient to compute the color for PMD pixels only. Often a higher resolution is desirable, thus a single PMD pixel is mapped onto several pixels of the final 2D/3D image. This is also true, if we want to generate external views of the sensor setup (see Fig. 4 (top)). Therefore, the high resolution RGB information has to be interpolated properly for each PMD sub-pixel.

Due to the perspective projections involved in the data acquisition process, direct use of the linear interpolation of the rasterization pipeline leads to distorted color mappings. Applying projective texture mapping (see [7]), the distortion of the RGB image on the PMD geometry can be avoided.

Assuming, we have two points S_1^{PMD} and S_2^{PMD} in the PMD image plane resulting from the projective transformation

$$P_i^{\text{PMD}} = T_{\text{proj}}^{\text{PMD}} \cdot P_i, \quad i = 1, 2 \quad (3)$$

with subsequent normalization. Here, $T_{\text{proj}}^{\text{PMD}}$ is the perspective projection specified by the intrinsic parameter of the PMD camera. The rasterization of any intermediate point using linear interpolation

$$S^{\text{PMD}} = (1-t) \cdot S_1^{\text{PMD}} + t \cdot S_2^{\text{PMD}} \quad (4)$$

requires the proper computation of the associate location S^{RGB} in the RGB image plane. According to Segal [7], S^{RGB} is the (x, y) -component of

$$P^{\text{RGB}}/w^{\text{RGB}} = \frac{(1-t) \cdot P_1^{\text{RGB}}/w_1^{\text{PMD}} + t \cdot P_2^{\text{RGB}}/w_2^{\text{PMD}}}{(1-t) \cdot w_1^{\text{RGB}}/w_1^{\text{PMD}} + t \cdot w_2^{\text{RGB}}/w_2^{\text{PMD}}} \quad (5)$$

Since the rasterization stage performs linear interpolation only, (5) needs to be rearranged. This is achieved by linearly interpolating $(x, y)^{\text{RGB}}/w^{\text{PMD}}$ and $w^{\text{RGB}}/w^{\text{PMD}}$. Finally the RGB image coordinate S^{RGB} is calculated in the fragment program by simple normalization with w^{PMD} .

As the described process highly depends on the correct reconstruction geometry and therefore on proper distance information, the PMD has been calibrated first using the calibration model described in [9]. This reduced systematic distance errors due to the discrepancy between the theoretically considered sinusoidal signal and the actual signal form and therefore results in better fusion results.

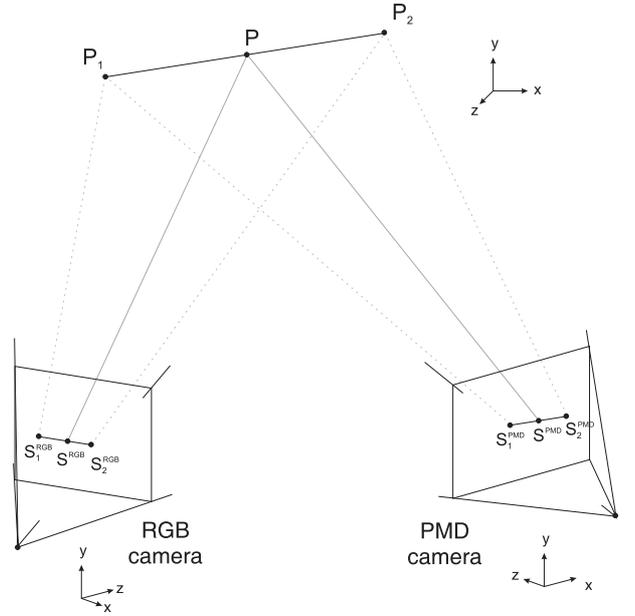


Fig. 2. Overview about the texture projection setup.

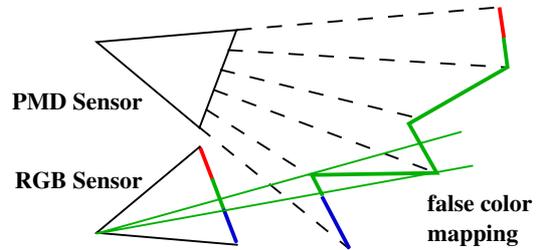


Fig. 3. False color assignment in shadowed region w.r.t. the RGB sensor.

V. HIDDEN SURFACE REMOVAL

Due to the different viewing directions of both cameras, an incorrect mapping mainly in the near range may occur (see Fig. 3). These artifacts are caused by occlusion, e.g. in concave object regions or regions with large distance gradients like object contours. For these surface regions, the RGB camera is unable to provide a proper object color and the projective mapping described in Sec. IV-B selects erroneous color information.

In order to prevent a mismapping, we adopt a render approach comparable to shadow maps [10], [11]. The main idea is to store the closest geometry portion w.r.t. the RGB sensor in a depth-buffer (z-buffer). Therefore, each geometry vertex is transformed into the RGB camera coordinate system using the transformation matrix $T_{\text{PMD} \rightarrow \text{RGB}}$ determined in the registration step. The distance to the RGB camera's origin is automatically interpolated for every fragment and written in an off-screen render target (a so-called *Frame Buffer Object (FBO)*), which stores the minimal per-pixel z-distance $\text{zBuffer}^{\text{RGB}}(x, y)$.

During the color assignment using the projective texture map, again every vertex point is transformed into the RGB

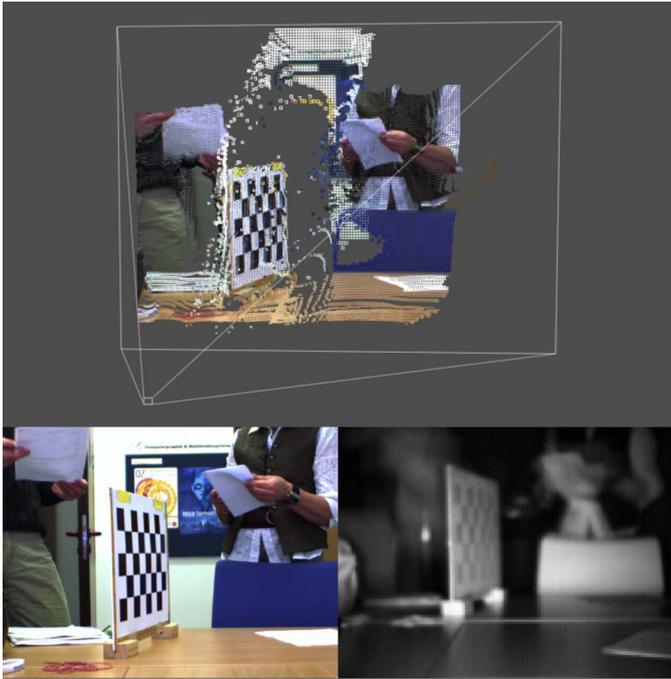


Fig. 4. 2D/3D data fusion for a sample scene (top) consisting of the RGB- and PMD sensor data (bottom left and right, respectively).

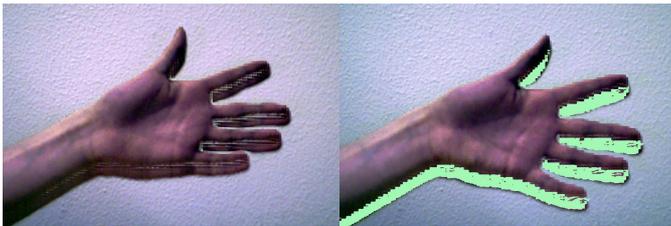


Fig. 5. Image mapping without (left) and with hidden surface removal (right). Areas not seen by the RGB camera are marked green.

camera coordinate system. Now, the current per-pixel distance information $z^{\text{RGB}}(x, y)$ is compared with the z-Buffer entry. In case the current fragment is farther away from the RGB sensor than the distance information stored in the frame buffer, i.e.

$$z_{\text{Buffer}}^{\text{RGB}}(x, y) + \varepsilon < z^{\text{RGB}}(x, y)$$

the fragment is hidden and the color assignment is omitted. The ε -offset is required to account for quantization inaccuracies. Otherwise, the color assignment is performed using the projective approach described in Sec. IV-B.

VI. RESULTS

Several geometry representations based on different geometric primitives for the PMD-based distance information are used for the 2D-3D data fusion and in order to visualize the fusion results from a virtual camera position (see Fig. 4, top). Used geometric representations are quads, points and wireframe. The quad- or point-based approach better resembles the pixel-based 3D data acquisition of the PMD sensor. The quads are scaled w.r.t. the distance value yielding a completely

covered solid angle for each pixel. The wireframe represents linearly interpolated neighboring distance values, modelling a continuous geometric shape.

From the fusion perspective, pixel-based render modes provide an implicit depth segmentation and thus reduce color-bleeding artefacts caused by outliers. At the same time, they offer only piecewise constant distance information, which results in imprecise mapping of color values. Here, the wireframe representation offers a higher order geometric approximation and leads to better color mapping results, if the underlying geometry is continuous.

Despite the drawbacks of the different geometric primitives, the pure visual impression compared to the original PMD data is much better. Definitely, the fused 2D-3D data open up new possibilities for scene analysis, e.g. for object recognition or distance refinement. Generally, a proper distance calibration is absolutely necessary due to the algorithm's dependence on an accurate geometric reconstruction.

VII. CONCLUSION

This paper presents an enhanced 2D-3D data fusion approach. The fusion is based on projective texture maps, allowing a fast and accurate fusion performed on the GPU. Additionally, we presented an efficient way to false mappings mapping due to hidden surfaces problems.

ACKNOWLEDGMENT

This work is partly supported by the German Research Foundation (DFG), KO-2960/5-1. Furthermore, we would like to thank PMDTechnologies Germany.

REFERENCES

- [1] H. Kraft, J. Frey, T. Moeller, M. Albrecht, M. Grothof, B. Schink, H. Hess, and B. Buxbaum, "3D-camera of high 3D-frame rate, depth-resolution and background light elimination based on improved PMD (photonic mixer device)-technologies," in *OPTO*, 2004.
- [2] R. Lange, "3D time-of-flight distance measurement with custom solid-state image sensors in CMOS/CCD-technology," Ph.D. dissertation, University of Siegen, 2000.
- [3] Z. Xu, R. Schwarte, H. Heinol, B. Buxbaum, and T. Ringbeck, "Smart pixel - photonic mixer device (PMD)," in *Proc. Int. Conf. on Mechatron. & Machine Vision*, 1998, pp. 259-264.
- [4] T. Prasad, K. Hartmann, W. Weihs, S. Ghobadi, and A. Sluiter, "First steps in enhancing 3D vision technique using 2D/3D sensors," in *11th Computer Vision Winter Workshop 2006*, V. Chum, O. Franc, Ed., 2006, pp. 82-86.
- [5] R. Reulke, "Combination of distance data with high resolution images," in *Image Engineering and Vision Metrology (IEVM)*, 2006.
- [6] T. Luhmann, *Nahbereichsphotogrammetrie*, 2nd ed. Wichmann Verlag, 2003.
- [7] M. Segal, C. Korobkin, R. van Widenfelt, J. Foran, and P. Haeberli, "Fast shadows and lighting effects using texture mapping," in *Proc. SIGGRAPH*, 1992, pp. 249-252.
- [8] Intel Inc., "Open source computer vision library (OpenCV)," www.intel.com/technology/computing/opencv/.
- [9] M. Lindner and A. Kolb, "Lateral and depth calibration of PMD-distance sensors," in *Int. Symp. on Visual Computing (ISVC)*, vol. 2. Springer, 2006, pp. 524-533.
- [10] W. T. Reeves, D. H. Salesin, and R. L. Cook, "Rendering antialiased shadows with depth maps," in *SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press, 1987, pp. 283-291.
- [11] M. Stamminger and G. Drettakis, "Perspective shadow maps," in *Proc. SIGGRAPH*, 2002, pp. 557-562.