Fast Fusion of Range and Video Sensor Data

A. Linarth, J. Penne, B. Liu, O. Jesorsky, R. Kompe

Abstract

This paper brings an innovative approach to generate 3D scene views from real world data using a matrix range sensor and a 2D video sensor. Besides the enhanced visualization, combining 2D and 3D data provides also valuable information for object detection and recognition as well as for other image processing methods. The scope of this work includes a discussion on extracting good features from low resolution images, necessary for the system calibration. The registration process is based on a geometrical model which is derived from the calibration parameters of the cameras. Finally, a FPGA-based solution is provided, which generates the 3D scene on-the-fly. The proposed architecture achieves its maximum performance with a fully pipelined implementation from the raw data to the generated fusion coordinates, while limited hardware resources are consumed. In contrast to standard processor approaches, a significant performance boost is achieved with the FPGA architecture.

1 Introduction

Photorealistic 3D models are a desired feature in many computer vision related applications, e.g. virtual and augmented reality. The proposed system takes advantage of the new generation of 3D sensors based on the time of flight principle which provides individual distance information for each element of a sensor matrix besides intensity and amplitude values (see [6] and [3]). The camera is basically composed of a sensor capable to filter the incoming light signal, in order to identify the portion of light absorbed which is emitted by the attached illumination source and reflected by the scene. The measurements are done by calculating the phase shift caused by the reflection time of this modulated light in analogy to standard radar systems. Although the available resolution of these sensors is sufficient for a bunch of image processing methods, current lateral resolutions cannot yet compete with standard 2D video sensors. Relating the low-resolution depth frame to a standard 2D monochrome or color image enhances the results of many image processing methods, such as segmentation, tracking or classification. The complexity level of these operations can also be reduced, which simplifies their use in embedded platforms. The described 2D-3D fusion follows the increasing interest of the automotive industry on imaging systems, with applications that range from pedestrian recognition to parking aid systems, as well as in the manufacturing process like quality inspection.

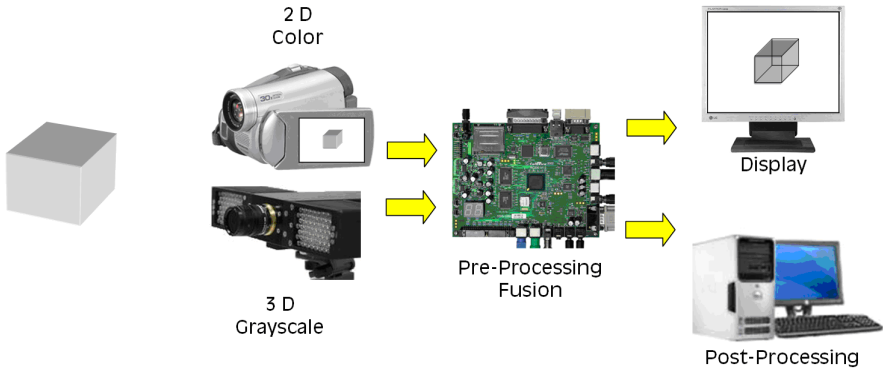

Fig. 1. System Overview.

We discuss in Section 2 about the pattern and the feature extraction used in the calibration procedure. Then a geometrical model in Section 3 is derived for the rigid registration between two camera coordinate systems. An efficient scheme for the fusion system using FPGA is shown in Section 4, followed by an experimental analysis of this work in Section 5.

2 Camera Calibration

The geometrical approach implemented for the fusion process requires a reliable estimation of the cameras positions and their internal parameters. The calibration is performed by using the method presented by Zhang [7] for both cameras simultaneously, which makes it possible to estimate the relative position and orientation between the two cameras at the same time. To choose a good calibration pattern and to determine the most accurate point correspondences are essential conditions for achieving good results. The intensity images with a low resolution of 64x16 supplied by the range sensor and the color images with a high resolution of 720x576 supplied by the 2D video sensor are treated in different manners in the pre-processing phase.

2.1 Calibration Pattern

Choosing a proper calibration pattern was an important issue during the development process due to the low-resolution feature of the time-of-flight camera. A minimum of 4 point correspondences is a prerequisite imposed by the homography calculation used in the Zhang method. A circle based pattern was chosen instead of standard chess boards, since the latter shows higher sensibility to the infra-red lighting condition supplied by the 3D camera. Corners and intersections were by far visibly harder to be identified than the circles' centroids (gravity centers), and increasing the number of squares consequently reduces their sizes making even harder to identify good correspondence points. Figure 2 shows images from these two patterns.

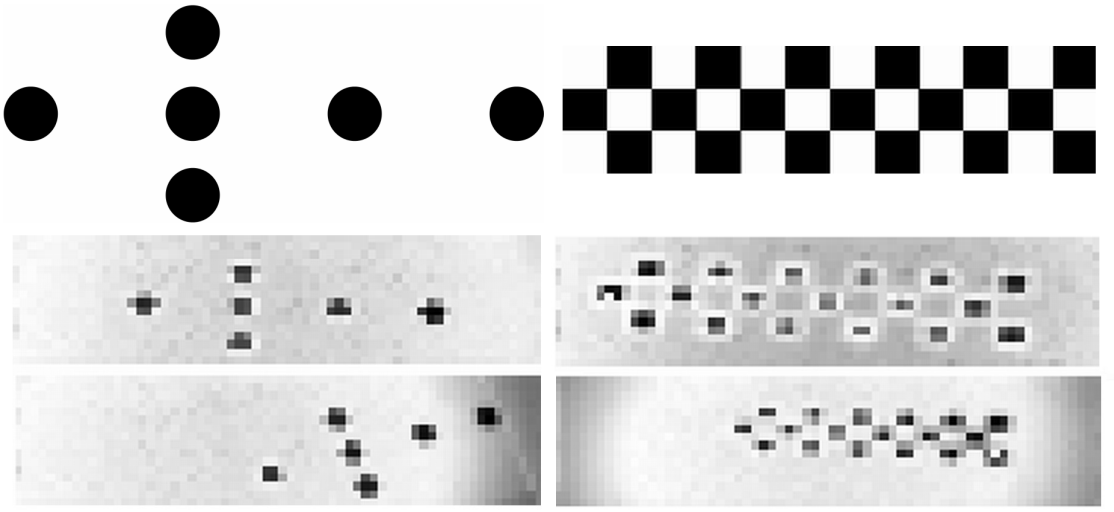

Fig. 2. Video Images and Range Images of the Circle Pattern and Chess-Board Pattern.

Concerning the geometrical relation between both cameras, to improve the precision of the derived model, each point is correlated to its pair in the second image, applying the registration procedure. This method takes the advantage of the depth data grabbed by the 3D range sensor at the given position. Therefore, a final calibration pattern with white circles on black ground was chosen since the low reflectance of the black material yields unreliable distance measurements. The final calibration pattern is shown with two of its 64x16 intensity images from the 3D camera.

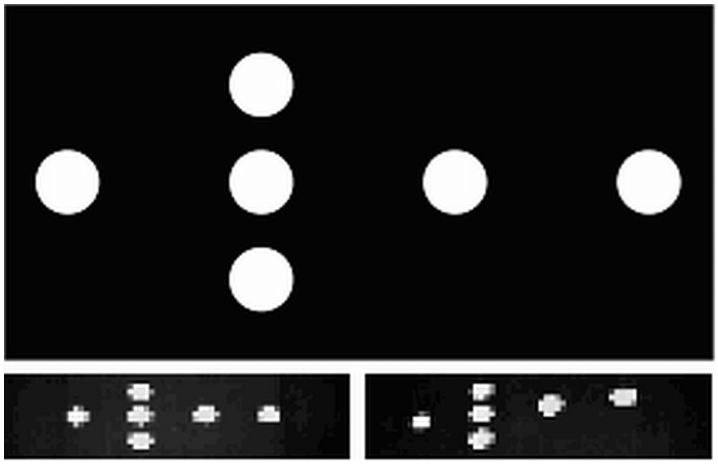

Fig. 3. Final Calibration Pattern

2.2 Point Correspondences

The calibration process is based on the point correspondences between the extracted circular feature points on the images and the corresponding ones on the actual calibration pattern. Applying a simple threshold followed by erosion is enough for separating the interesting regions in the high resolution video image. However, this process does not provide satisfying result for the low resolution grayscale depth image from the 3D camera. Since the circles are very small, eroding them causes a great loss of information. A minimum threshold also cannot filter correctly the information due to lighting effects. The approach proposed in this study is to apply a strong threshold, such that at least one point of each circle remains, followed by a region growing with these points as seeds. A threshold for the maximal gradient change is used as the stop condition for the region growing. The segmented region is then considered as the effective region for the circle. The centroid of the circle is further computed as the weighted average of the points using their intensity values as weights.

Identifying a circle in a calibration pattern can be done using the following property:

$$d = \frac{(circunference)^2}{area} = \frac{(2\pi r)^2}{\pi r^2} = \frac{4\pi^2 r^2}{\pi r^2} - 4\pi \approx 12.6 \qquad (1)$$

The ratio between the squared circumference and area of a circle should be equal to the above value, while other figures present a bigger value. In practice, an error level is set, since circles get projected as ellipses. Although this property works so well with the high resolution image, for a 64x16 pixel image a circle can be represented with 10 pixels or even fewer, sometimes with formats that can be easily confused with squares or rectangles. The centroid of such a small circle is also not precise enough for the calibration method. Therefore, adjusting the error level can represent a good method for determining if the circle is good/big enough for the calibration process. In this paper no further studies were done in this direction, but an empirical error level was chosen based on experiments. The next figure shows a reversed image of the calibration pattern with the circle centers marked after ordering them.

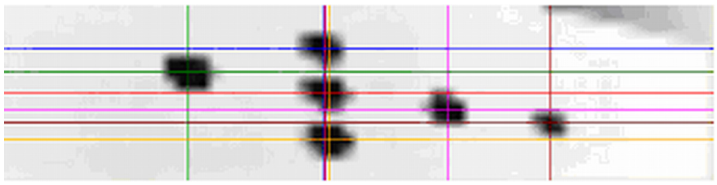

Fig. 4. Estimated Centroids of the Circles on a Low-Resolution Image

2.3 Calibration procedure

- Place the calibration pattern such that it can be clearly sensed by both cameras and then synchronously acquire an image pair.
- Move the pattern such that the pattern plane is not coplanar to those where pictures have already been taken (see Zhang [7]) and take another pair of images. Repeat this step at least 3 times. The more pictures are taken, the higher is the accuracy of the result.
- Calculate point correspondences for each of the images.
- Calculate internal and external parameters using the Zhang method.
- Derive the relative transformation between both cameras and the registration matrix from one of the image pairs. (see the following section)

3 Computation of the Fusion Coordinates

Assuming a rigid configuration of a 3D and a 2D camera, viewing to the same scene, a geometrical model can be found to register the 3D information of time of flight sensor with the 2D information of the video camera. Compared to stereo methods, the approach proposed here does not need feature points to identify correspondences. Instead, with previously calibrated camera parameters and the relative transformation between the two cameras, it is only necessary to calculate the 3D world points from the sensed distance values using the camera parameters achieved from the calibration procedure and to project them into the 2D image.

Taking advantage of the distance d_w from the centre of the time of flight camera to the object, the back projection to world points (q_x, q_y, q_z), can be derived using triangulation, as seen in Fig. 5, where d_i represents the distance from the center of the camera to the i^{th} image point, w the pixel width, h the pixel height, (u_0, v_0) the camera principal point, (p_{ix}, p_{iy}) the i^{th} pixel coordinate and f the focal length.

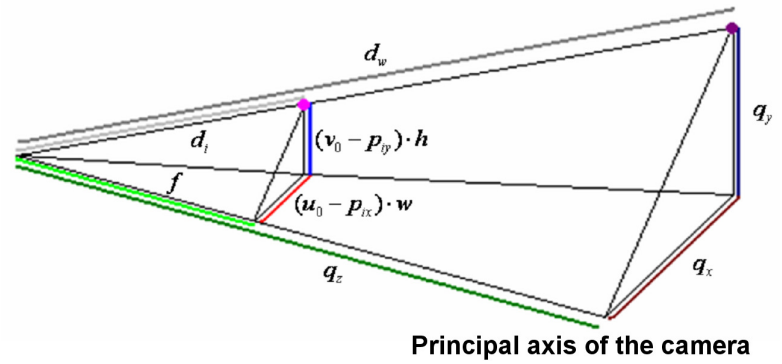

Fig. 5. World Point Similar Triangles

Assuming zero skew, the camera calibration matrix [2] can be written as:

$$K = \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \frac{f}{w} & 0 & u_0 \\ 0 & \frac{f}{h} & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Scaling the image triangle with *1/f*, and denoting as d'_i the scaled distance of the image point to the camera, the world point (q_x, q_y, q_z) can be calculated by:

$$s_x = \frac{u_0 - p_{ix}}{\alpha} \quad (3)$$

$$s_y = \frac{v_0 - p_{iy}}{\beta} \quad (4)$$

$$d'_i = \sqrt{\left(s_x^2 + s_y^2 + 1\right)} \quad (5)$$

$$q_x = s_x \cdot \frac{d_w}{d'_i} \quad (6)$$

$$q_y = s_y \cdot \frac{d_w}{d'_i} \quad (7)$$

$$q_z = \frac{d_w}{d'_i} \quad (8)$$

Once the world points are calculated, the next step is to project it back to the image plane of the video camera. At this point the coordinates of the world points are expressed under the coordinate system of the time-of-flight camera, it is necessary to transform the coordinate to the coordinate system of the video camera (see Fig. 6).

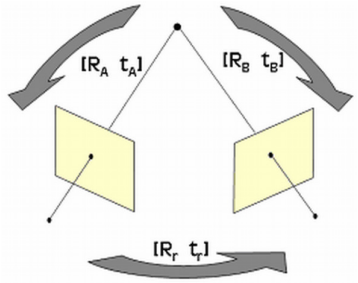

Fig. 6. Relative Transformation

The transformation between the coordinate systems can be calculated by relating the external parameters of them at each scene and it is constant in a rigid configuration. In the following equations footnotes *A* and *B* are added to distinguish the two cameras, and letting m be the 3D world coordinates in homogeneous coordinates. Let the image points p_i [2]:

$$p_{iA} = K_A \begin{bmatrix} R & t \end{bmatrix}_A m \quad (9)$$

$$p_{iB} = K_B \begin{bmatrix} R & t \end{bmatrix}_B m \quad (10)$$

Let:

$$q = K_A^{-1} p_{iA} = \begin{bmatrix} R & t \end{bmatrix}_A m \tag{11}$$

Separating rotation R and translation t components:

$$m = R_A^{-1}(q - t_A) \tag{12}$$

Hence:

$$p_{iB} = K_B \begin{bmatrix} R & t \end{bmatrix}_B R_A^{-1}(q - t_A) \tag{13}$$

$$p_{iB} = K_B \left(R_B R_A^{-1} q - R_B R_A^{-1} t_A + t_B\right) \tag{14}$$

Interpreting q as a world point whose coordinate system's origin is at the centre of the first camera, the relative transformation between both coordinate systems is given by:

$$R_r = R_B R_A^{-1} \tag{15}$$

$$t_r = t_B - R_B R_A^{-1} t_A \tag{16}$$

Therefore the final step on the registration is to apply the projection of the world points with the origin the first camera coordinate system to the second camera, by applying the calculated relative transformation and finally the internal parameters of the second camera:

$$p_{iB} = K_B \begin{bmatrix} R & t \end{bmatrix}_r \begin{bmatrix} q \\ 1 \end{bmatrix} \tag{17}$$

Where $q = (q_x, q_y, q_z,)$.

4 On-the-Fly implementation on FPGA

Developed for the Xilinx Virtex II 3000 FPGA [Xil06] available in the Celoxica RC203-E development kit [Cel06], the realized system targets the optimization of data-flow and the parallelization of processes to achieve a fast fusion method for the 3D data of the time of flight system and the 2D video data. The system developed in this work can be roughly described by the data flow between the three modules, shown in Figure 7.

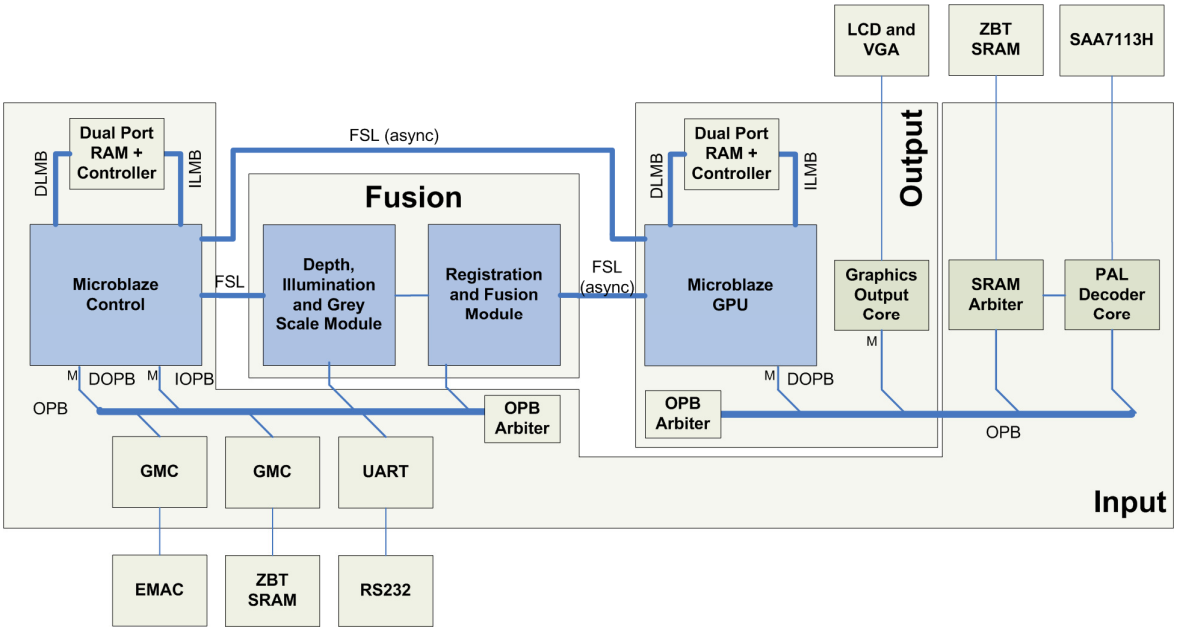

Fig. 7. FPGA Implementation Architecture

While the input and output processes are responsible to interface the external devices, the core of the implementation is represented by the fusion module. The input and output modules were developed to provide an interface to the camera and to a display, respectively. In the final system, the depth calculation procedure can be connected with a very simple interface to the analog to digital converter, and the output module would be connected to further image processing modules.

The fusion module was developed as a fully pipelined architecture. It receives the raw pixel data in a progressive scan fashion and provides at the same frequency depth, grey scale, amplitude of the infrared light as well as the relative coordinates of such points in the second image.

In the equations to calculate q_x, q_y and q_z it is possible to identify that just the depth gets changed at a given time. All other values are constant for each pixel while the configuration of the camera does not change. Therefore, a look-up table can store pre-calculated values, reducing the calculation of the world points to three simple multiplications which can be done in parallel, wasting one single clock cycle. Figure 8 shows the hardware architecture for this module. The look-up tables are stored in dual-port RAMs. This is an efficient approach since one of the ports can be connected to a configuration module (ex. a processor, auto calibration module), while the second one is read in a sequential fashion, to provide the coefficients at each point, avoiding time multiplexing, except the fact that multiple access to the same address should be avoided (implemented with combinational logic, in “read after write” mode). The table should contain one entry for each pixel on the image, which means 64x16 = 1024 entries. By choosing a 32bit representation, each of the three look-up tables can be implemented using 2 Block-RAMs, out of 96 in total of the given FPGA. To connect to the system, a configuration module, representing a core generated OPB bus memory interface was adapted to connect to the described dual-port RAMs.

Several approaches could be used to solve the registration equation in hardware. One of them could be pre-multiply intrinsic an extrinsic parameters and reducing the operation to a simple matrix multiplication. This multiplication, by its time, could be solved, for example, sharing just one adder and one multiplier in a MAC fashion. This could be an interesting method if the data arrives in 32bit words, in a sequential way. However, in the present case, world points are available at a single clock cycle, allowing a more efficient approach to be implemented. The bottom part of Figure 8 shows the proposed architecture for this matrix multiplication.

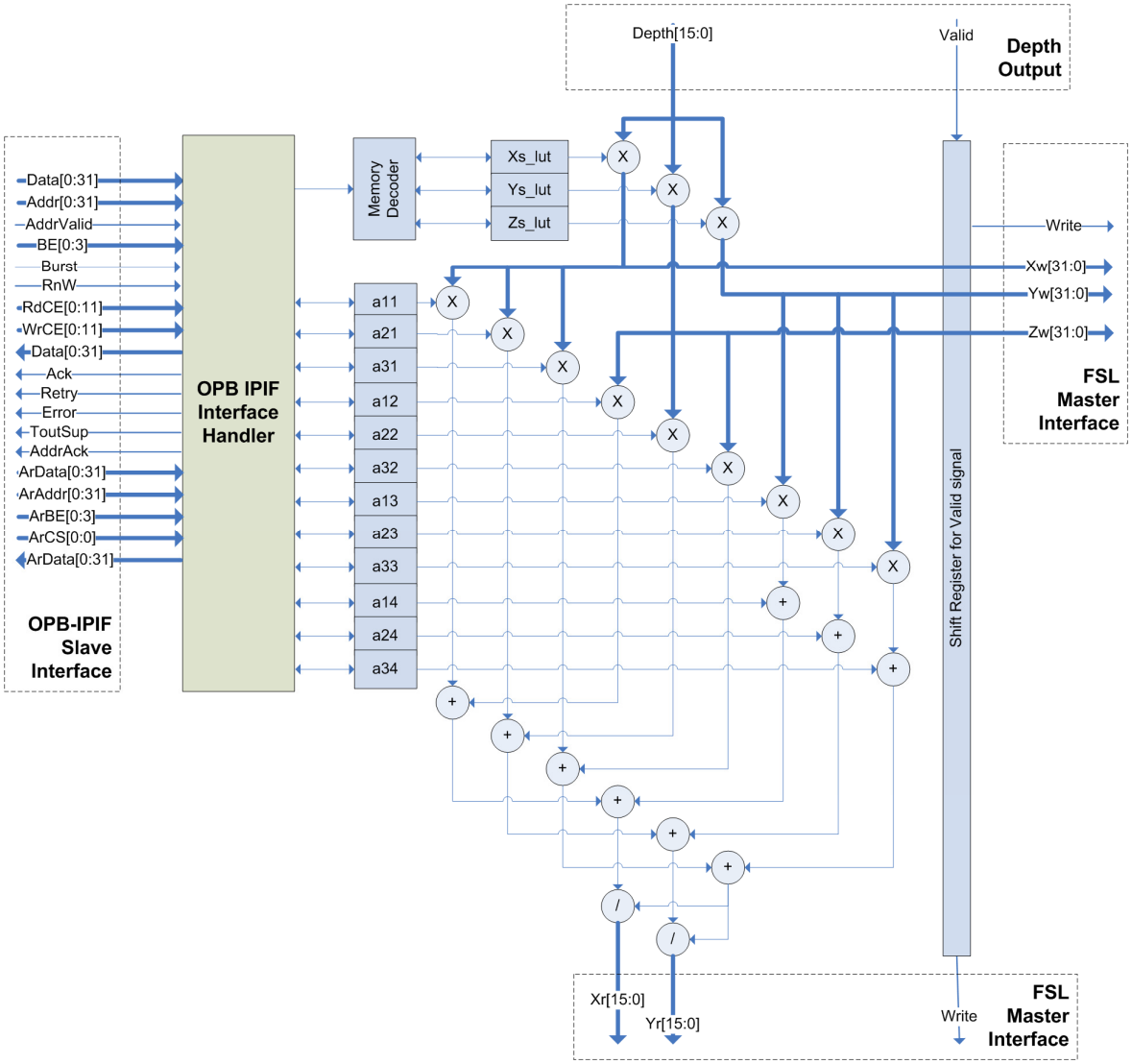

Fig. 8. Registration Hardware Architecture

Applying this method, an on-the-fly computation is achieved. Multiplication of X, Y and Z coordinates are done in parallel and stored in intermediate registers (in the diagram, all the operations present an internal register at the output). At the two next levels, these registers are added, providing the result for the matrix multiplication in just 3 clock cycles. 2 or 1 clock cycles would also be possible but to reduce the overall latency, this is avoided. Since the coordinates were until now represented homogeneously, this is a good time to dehomogenize them. Therefore two core generated pipelined dividers were added, whose results give the coordinate on the second image of the respective point of the time of flight camera.

5 Experiments

5.1 Error analysis

Experimental error can be evaluated by applying the registration process over the circular marks extracted from calibration pattern images. The absolute error is given by the difference between the registered points to the correspondent circle centre in the second image. Once these points lie between the pixels, the used depth is calculated through bilinear interpolation of its neighbors. The error for 96 points at x and y coordinates is shown in

Figure 9. The outliers are caused by bad depth data from the camera, when not enough light is absorbed by the sensor. Figure 10 shows the depth of each point. Each pattern image provides 6 points, which should remain at close depth values, making easy to observe invalid values.

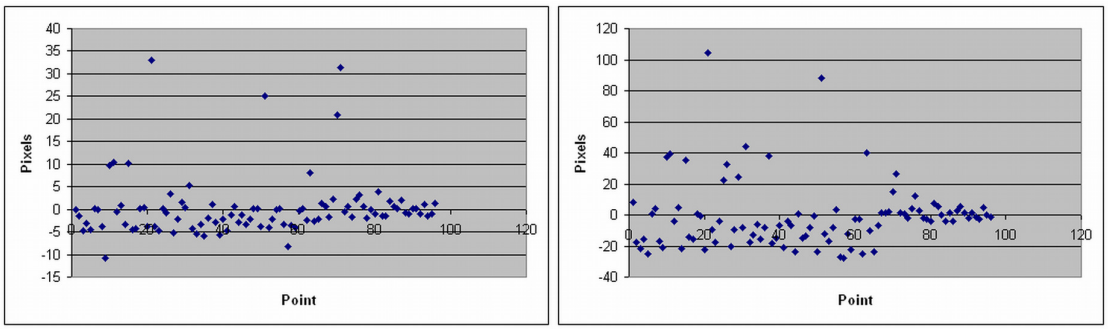

Fig. 9. Error on X and Y Coordinates

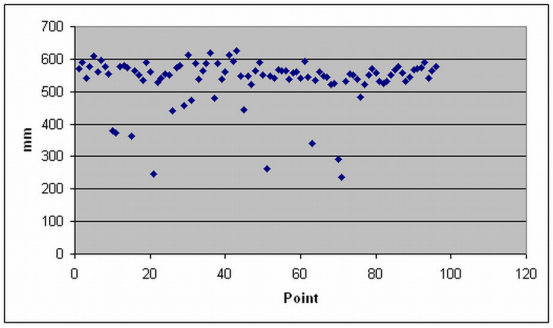

Fig. 10. Depth of the Analyzed Points

It is also important to remark that the given image resolution is 64x16 for the 3D camera and 720x576 for the 2D camera. This means that one pixel of the low resolution image is represented by 11.25 pixel width and 36 pixel height in the high resolution image.

5.2 Performance

Developed in VHDL and synthesized by the Xilinx ISE 7.1 [Xil06], the fusion core presents the following characteristics:

Tab. 1. Hardware configuration in the system

	Depth		World		Projection	
Slices	2336	16%	132	<1%	4848	33%
BlockRAMs	3	3%	6	6%	-	-
Mult 18x18	6	6%	6	6%	36	37%
Max. Frequency (MHz)	88		280		127	
Latency (clock cycles)	63		1		37	

Assuming the system running at maximum synthesizable (in the given FPGA) frequency (88MHz), a comparison to an optimized algorithm implemented in a standard PC (Pentium 4 3GHz - 1MB cache - 1GB memory) is presented in next table for processing one complete frame.

Tab. 2. System performance for dealing with one frame

Module	PC (3GHz)	FPGA (88MHz)	Speed-up factor
Depth	406µs	47µs	8.6
Registration	37µs	12µs	3.1
Complete Fusion	443µs	59µs	7.5

5.3 Example

This section presents an example of the images generated by the system. First, the following equations show respectively calibration parameters matrix of the 3D camera (K_{3D}), calibration parameters of the 2D camera (K_{2D}) and the relative transformation matrix (T).

$$K_{3D} = \begin{bmatrix} 71.090520 & 0 & 39.849392 \\ 0 & 52.923175 & 8.663061 \\ 0 & 0 & 1 \end{bmatrix} \tag{18}$$

$$K_{2D} = \begin{bmatrix} 925.760849 & 0 & 373.528280 \\ 0 & 1013.488068 & 273.059471 \\ 0 & 0 & 1 \end{bmatrix} \tag{19}$$

$$T = \begin{bmatrix} 0.991048 & -0.046965 & 0.124974 & 3.997788 \\ 0.039669 & 0.997394 & 0.060243 & 110.319310 \\ -0.127477 & -0.054746 & 0.990329 & 147.667955 \end{bmatrix} \tag{20}$$

Figures 11 and 12 show respectively the original and registered images of a person.

Fig. 11. Original Images

Fig. 12. Registered Images

The 3D representation uses squares to show each points of the low resolution camera in 3D, while the registered 2D image is mapped as textures on the points. Observe that, on the object borders where the light gets reflected from multiple objects, the depth values of the points get smoothed. As shown in the results, the alignment error is lower than one pixel width, which can be also seen clearly at the horizontal and vertical borders..

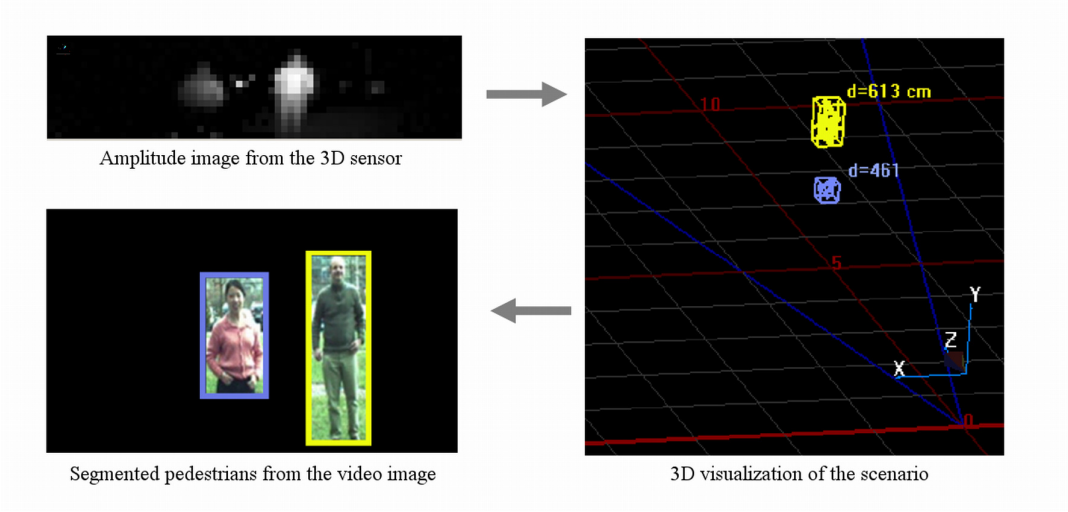

Fig. 13. Scenario for Pedestrian Tracking and Classification.

5.4 Further Related Experiment Scenario

Moreover, we have also built another experiment scenario closely related to this work. See figure 13. The experiment aims to track and classify pedestrians, cyclists and vehicles in traffic scenes. Currently the tracking and classification algorithms are based on the analysis of the range images grabbed from the 3D sensor, and the video images from the 2D sensor are fused for favorable 2D and 3D visualization. An interesting future research is to combine the 2D and 3D image-processing methods to classify the segmented objects.

6 Conclusions and Outlook

An efficient method for the fusion of 3D and 2D sensor was presented in this paper. The results show that this approach is already precise enough for applications that do not require critical information on object borders. Accuracy on object borders could be increased by local analysis using the given fusion result as a first guess. Moreover, auto-calibration methods might be further applied to constantly improve the alignment. The actual solution is compact, which is convenient to be used in embedded systems. Applications like tracking can be improved with the increasing information given by the fusion. The described method is not limited to time of flight and standard video sensors. Other sensors like thermal sensors (e.g. used in night vision systems) and laser scanning systems could be used in equivalent manner.

References

[1] Celoxica: RC203 Website, 2006, http://www.celoxica.com/products/rc203.

[2] R. Hartley, A. Zisserman: Multiple View Geometry In Computer Vision, Cambridge, 2000.

[3] X. Luan: Experimental Investigation of Photonic Mixer Device and Development of TOF 3D Ranging Systems Based on PMD Technology, PhD thesis, University Siegen, 2001.

[4] L. Tao, H. Ngo, M. Zhang, A. Livingston, V. Asari: A Multi-sensor Image Fusion and Enhancement System for Assisting Drivers in Poor Lighting Conditions, in AIPR'05: Proceedings of the 34th Applied Imagery and Pattern Recognition Workshop (AIPR'05), IEEE Computer Society, Washington, DC, USA, 2005, S. 106–113.

[5] Xilinx: Website, 2006, http://www.xilinx.com.

[6] Z. Xu, R. Schwarte, H. Heinol, B. Buxbaum, T. Ringbeck: Smart Pixel – Photonic Mixer Device (PMD): New System Concept of a 3D-imaging Camera-on-a-Chip, in International Conference on Mechatronics and Machine Vision in Practice, Nanjing, China, 1998, S. 259–264.

[7] Z. Zhang: A Flexible New Technique for Camera Calibration, IEEE Transactions on Pattern Analysis and Machine Intelligence, Bd. 22, Nr. 11, 2000, S. 1330–1334.

A. Linarth, B. Liu, O. Jesorsky, R. Kompe

Driving Assistance and Sensor Information
Elektrobit Automotive Software
Frauenweiherstr. 14, 91058 Erlangen, Germany
E-mail: andre.linarth@elektrobit.com, bing.liu@elektrobit.com, oliver.jesorsky@elektrobit.com, ralf.kompe@elektrobit.com

J. Penne

Department of Pattern Recognition
Friedrich-Alexander University, Erlangen-Nuremberg
Martensstr. 3, 91058 Erlangen, Germany
E-mail: penne@informatik.uni-erlangen.de

Keywords: 2D-3D Fusion, Rigid Registration, FPGA, Calibration