Towards Hardware Stereoscopic 3D Reconstruction

A Real-Time FPGA Computation of the Disparity Map

S. Hadjitheophanous, C. Ttofis, A. S. Georgiades, and T. Theocharides
Department of Electrical and Computer Engineering, University of Cyprus
KIOS Research Center for Intelligent Systems and Networks
PO Box 20537, 1678 Nicosia, Cyprus
{hadjitheofanous.stavros, ttofis.christos, athos.georghiades, ttheocharides}@ucy.ac.cy

Abstract—Stereoscopic 3D reconstruction is an important algorithm in the field of Computer Vision, with a variety of applications in embedded and real-time systems. Existing software-based implementations cannot satisfy the performance requirements for such constrained systems; hence an embedded hardware mechanism might be more suitable. In this paper, we present an architecture of a 3D reconstruction system for stereoscopic images, which we implement on Virtex2 Pro FPGA. The architecture uses a Sobel edge detector to achieve real-time (75 fps) performance, and is configurable in terms of various application parameters, making it suitable for a number of application environments. The paper also presents a design exploration on algorithmic parameters such as disparity range, correlation window size, and input image size, illustrating the impact on the performance for each parameter.

Keywords- FPGA Signal Processing; Stereo Vision; Disparity Computation;

I. INTRODUCTION

Stereoscopic 3D reconstruction is an important process in the field of Computer Vision and refers to the ability to infer information on the 3D structure and distance of a scene, from a stereo pair of images [1]. Stereo images are captured in the same way the human eyes capture scenes, and can provide depth information [1], just as human eyes provide perception about depth. While a 2D image provides only height and width, 3D reconstruction of the 2D image extends the information to a third dimension, depth. The major task of 3D reconstruction is to find the location of an object of interest in the left image and the location of that same object in the right image, with the objective of computing the disparity between the two objects, and eventually using this information to reconstruct the 3D scene [1]. 3D image reconstruction algorithms have been widely used in 3D imaging systems and to capture 3D surface information, as well as medical imaging [2]. Emerging 3D reconstruction applications, such as geographic information systems, satellite-based earth and space exploration, autonomous robots and vehicles and security systems, require real-time, high-performance computation. Moreover, computation needs to be done using mobile and embedded equipment, and high resolution stereo images.

Several software implementations of the 3D reconstruction process utilize general purpose processors as well as clusters and multiprocessor systems, GPUs and DSP processors [3, 5, 7, 19]. However, while easy to implement and extremely flexible, software implementations require state-of-the-art equipment in order to reconstruct the 3D image in real time, especially as the resolution of the image increases. Moreover, such systems are not suitable for mobile and embedded applications. The 3D reconstruction process however, potentially can benefit from customization depending on the targeted application; hence a more flexible architecture with flexible memory access patterns is preferred. Custom architectures implemented on ASICs and FPGAs, can exploit the intrinsic parallelism of the 3D reconstruction algorithm, and optimize memory access patterns to provide parallel computation. By taking advantage of the resources integrated into modern FPGAs such as embedded processors and DSP units, dedicated architectures can be designed in order to yield real-time 3D image reconstruction. FPGAs allow the architecture to be reconfigured and adapt to different application demands.

This paper presents an architecture targeting embedded stereo 3D reconstruction applications. The proposed architecture processes calibrated stereo image pairs and computes the disparity and depth maps. One of the novel contributions of this paper is the integration and utilization of the Sobel edge detection algorithm, which aids significantly in reducing the amount of processed data, and subsequently decreases drastically the resulting frame rate. The proposed architecture is parameterizable in terms of correlation window size, image size and disparity levels range. The paper presents a design exploration over these parameters as well. The proposed system was implemented on Virtex2 Pro FPGA platform, yielding 75 frames per second (for a stereo image pair of 320x240 pixels).

The rest of this paper is organized as follows; section II explains the stereo 3D reconstruction process. Section III discusses related work, and section IV presents the proposed architecture. Section V presents the experimental framework and results and section VI concludes the paper, discussing future work.

![Figure 1. The epipolar geometry.](image-url)
correspondence problem to a search along conjugate epipolar lines, known as the epipolar geometry. This is represented as a common focal length and baseline distance, the first step in 3D reconstruction is the rectification of the stereo images. Rectification is the process of transforming both images on a common plane. An overview of a stereo 3D reconstruction system is shown in Fig. 3. The algorithm receives a pair of stereo images (left and right image) as an input, and outputs a sparse disparity map (or the depth map). The 3D reconstruction algorithm must solve two basic problems: correspondence, which deals with finding an object in the left image that corresponds to an object in the right image, and reconstruction, which deals with finding the depth (i.e. the distance from the cameras which capture the stereo images) and structure of the corresponding point of interest. The correspondence problem is the most demanding in terms of computational complexity, and involves searching and matching techniques (to locate a common object in both images), the robustness of which determines quality and precision of reconstructed 3D data.

In order to locate a common point of reference in the two images, a small window (called correlation window) from both images is being evaluated by comparing the window from the left image to the window from the right image, via the sum of absolute differences (SAD) method. The search is constrained along a horizontal scan line, as the images are rectified. Additionally, an object that appears in both stereo images will be found within a maximum horizontal bound, which depends on the object’s distance from the camera. Henceforth, a search limit can be imposed, known as disparity range, which constrains the search along a bounded horizontal scan line. The size of the search window, and the disparity range, impact the reconstruction algorithm significantly, both in terms of performance, as well as quality of results.

While rectification reduces the search space, further improvement can be obtained, by applying an edge detection process over the input images. Edge detection detects locations in the image where intensities change over a certain threshold through the x and y directions, indicating the presence of an edge. These locations are described by the edge points, which determine the outline (i.e. perimeter) of an object, and distinguish it from the background and other objects. Consequently, the matching procedure can concentrate only on the edge points of the original images, reducing thus the processing time. Moreover, in hardware implementations, edge images that contain only the edge points concentrate only on the edge points of the original images, reducing thus the processing time.

II. 3D RECONSTRUCTION PROCESS OVERVIEW

3D stereo reconstruction is based on the concepts of epipolar geometry, shown in Fig.1. $P(x,y,z)$ is a point in 3D space, and the two points denoted $E_L$ and $E_R$ represent the epipoles. An epipole is the point of intersection of the line across the optical centers, i.e. the baseline, with the image plane. The points $P$, $O_L$ and $O_R$ form a plane called the epipolar plane. The line $P-O_L$ is seen by the left camera as a point $X_L$ because it is directly in line with the camera’s center of projection $O_L$. In the right image plane, that line is represented as $E_R-X_R$ and is called the epipolar line. For each point observed in one image, the same point must be observed in the corresponding epipolar line on the other image. This is known as the epipolar constraint, which reduces the correspondence problem to a search along conjugate epipolar lines. Given pixel coordinates of $P_L(x_L,y_L)$ in the left image, and the corresponding pixel coordinates $P_R(x_R,y_R)$ in the right image, the 3D world coordinates $P(X,Y,Z)$ are computed as:

$$X = \frac{x_L b}{d}, \quad Y = \frac{y_L b}{d}, \quad Z = \frac{f b}{d}$$

where $b$ (baseline) is the distance between the centers of projection $O_L$ and $O_R$, $X_L$ and $X_R$ are the coordinates of $P_L$ and $P_R$ with respect of the principal points $C_L$ and $C_R$, $f$ is the common focal length and $Z$ is the distance between $P$ and the baseline $b$, which represents the depth of the object. The disparity $d$ is defined as the distance of two corresponding points when one of the two images is projected onto the other, and is computed by $disparity = X_R - X_L$ (shown in Fig. 2.) Disparity represents the relative difference in the location between common objects (i.e. the location of the object in the right image, relative to the location of that object in the left image) on the stereo pair. The depth map is simply the reciprocal of the disparity map.

If we assume that a set of stereo images come from calibrated cameras (with known focal length, and fixed baseline distance), the first step in 3D reconstruction is the rectification of the stereo images. Rectification is the process that transforms each image into a common image plane, aligning the pairs of conjugate epipolar lines to a common image axis (usually the horizontal). Rectification allows the computation of stereo correspondence, which involves a 2D search in unrectified images, to be reduced to 1-D search along the epipolar line. Rectification produces a transformation function used to project one of two images in the other’s common plane; the transformation function can remain the same if the stereo camera system remains calibrated, otherwise, a new transformation function is necessary. In this work, we assume a steady and calibrated camera system, so we do not focus on the rectification; instead we take advantage of modern FPGA technology that includes embedded processors, utilizing an embedded processor to transform both images on a common plane.
points, can be represented in binary form (i.e. black or white) and encoded using only one bit per pixel, rather than 8 bits (as used in grayscale images), reducing drastically the computation space and complexity of the search and match operations.

When an object is located in both images, its pixel coordinates are then used to derive the disparity map, and by using triangulation, the 3D structure of the scene. Triangulation is dependent on the expected view of the object under different angles, and includes a projection of the depth map on a 3D space; this is however dependent on the host application and specific lighting conditions, and is left as part of future work [1].

III. RELATED WORK

Most solutions of the 3D reconstruction problem have been implemented in software running on general purpose processors or cluster computers, with relatively adequate frame rates. In [3], a method to implement correlation based disparity calculation is presented. It uses the SAD correlation measure for RGB color images, and can achieve 20 fps. In [4], a dynamic programming (DP) based algorithm for calculating the disparity map is presented. It achieves qualitatively good disparity maps for real scenes, while providing a performance of 30 fps on a 2GHz PC. In [5], a fast area-based stereo matching algorithm is presented. The algorithm rejects previous matches as soon as more reliable ones are detected, and achieves almost 40fps. Such techniques require high-end hardware to run, and performance deteriorates rapidly as the image size increases. Other implementations utilize specialized hardware such as Intel MMX processors [6] and graphics processing units (GPUs) [7].

Recent work suggests an emergence in dedicated hardware architectures however, as means to address software constraints. A few algorithms have been implemented on FPGA platforms, such as [8], where a stereo depth measurement system on FPGA, is presented. The system generates disparities on 512x480 images at a rate of 30 frames per second. The algorithm implements a window-based scan-line correlation search technique, without an edge detector however. A promising implementation of a stereo system in FPGA, is presented. The system consists of a Sobel edge detection unit, a memory controller to optimize memory accesses based on the algorithm requirements, on-chip buffers for image storage, the stereo correlation unit, and memory storage for the disparity and depth maps for debugging and verification purposes. The memory controller also acts as the control unit for the entire operation, coordinating memory accesses as well as data transfers and handshakes between system components. The memory controller acts as an I/O unit and also as a control unit. The controller fetches data from the input images which can be stored on an external memory. The system uses two on-chip buffers, which hold the part of the image which is being searched (one buffer for the left image and one for the right image). The size of the buffers depends on the correlation window size and the disparity range. The memory controller starts filling the first buffer with pixel data, and alternates by sending the data to the edge detector and by filling up the second buffer, maintaining the computation continuity. The detector output is then fed to a second set of on-chip buffers (left and right again), used in similar fashion as the previous ones, that contain the edge points.

The Sobel edge detector is implemented in order to reduce the amount of data and speed up the operation as described in section II. The black and white output image pair of the detector only characterizes the feature boundaries, (edges in black while the remaining image is white). The correlation window now consists of binary values, (black and white), simplifying the search operation. The Sobel detector involves convolution of the input image over two convolution masks; the masks hold data values between -2 and 2; thus the overall convolution does not need to involve a multiplier. A set of additions and subtractions, depending on the mask value (i.e. one addition for a value of +1, two additions for a value of +2, and so on) was used. By avoiding the costly multiplication operation, a higher frequency as well as fewer clock cycles could be obtained, at the cost of quality however. A binary edge representation could potentially return two edges which belong to different components as matching edges, resulting in erroneous computation of the disparity. This however can be addressed by using more intensity levels to represent the edge points [1][14], at the expense of having to compute more data of course. The Sobel architecture is shown in Fig. 5.

### Table I. Stereo Vision FPGA Implementations

<table>
<thead>
<tr>
<th>FPGA</th>
<th>Xilinx Virtex</th>
<th>Xilinx Virtex-II</th>
<th>Altera APX20KE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image Size</td>
<td>320x240</td>
<td>640x480</td>
<td>64x64</td>
</tr>
<tr>
<td>Disparity Range</td>
<td>16</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>Window Size</td>
<td>7x7</td>
<td>32x32</td>
<td>8x8(max)</td>
</tr>
<tr>
<td>Speed(fps)</td>
<td>71</td>
<td>30</td>
<td>5063</td>
</tr>
</tbody>
</table>

![Figure 5. Sobel edge detection architecture.](image-url)
Correlation works by comparing the edge points from the two buffers, using the SAD matching algorithm. The edge points are compared by a correlation window from the one image (stored in the left buffer), that is correlated with a second window from the other image (stored in the right buffer). The second window is shifted through all possible positions in the second image as shown in Fig. 6. The possible positions that the search window moves, are bounded by the maximum disparity range. The position where correlation search yields the highest value, determines the pixel in the second image that corresponds to the pixel of interest in the first image, and subsequently to a potentially matching object in the two images. The correlation window size impacts the performance, as larger correlation windows can reduce the effect of noise and improve the quality of results. A large correlation window also results in an increase in the frame rate, since more search operations can be performed in parallel, and therefore the total time for the search operations is reduced. However, a large search window faces the possibility of treating edges which represent two different objects as a single object, thus potentially failing to capture the disparity between the two objects. A smaller correlation window reduces the problem, as it looks over smaller regions. Obviously, a smaller window will result in a large number of correlation searches, which tend to suffer more as the original input image size increases. Generally, the choice of the correlation window size is a tradeoff between the quality of results and the speed of the system. The architecture proposed in this work supports windows of different sizes, and experimental results were carried to determine the effect on the output depth map.

Fig 7. shows the SAD matching technique between correlation windows, which is implemented in the correlation unit (Fig. 4). Each window in the right image is shifted on a predetermined offset (one pixel in our implementation), and compared to the correlation window in the left image. The process is repeated for the maximum disparity range (from 0 to 31 in our case).

The architecture can be easily modified to adapt to several applications, depending on the targeted frame rate or the quality of results. By varying the choice of the edge detector where non-binary edge representation can be used, and by varying the correlation window size and maximum disparity range, the architecture can provide comparable quality-performance tradeoffs for a variety of applications.

V. EXPERIMENTAL PLATFORM AND RESULTS

The proposed stereo system was implemented on a Virtex2 Pro XC2VP30 FPGA platform. We used ten calibrated stereo image pairs [20] that were first loaded into the compact flash memory card, and used the built-in PowerPC processor to generate rectified images that were stored in Virtex2 Pro board.
DRAM, and were fed into the system shown in Fig.4. We displayed the sparse disparity maps on a VGA monitor in order to verify the system. Fig. 8 illustrates the experimental setup of the system and an example display. The system is parametrizable in terms of correlation window size, disparity range and input image sizes, as the performance depends also on a combination of these parameters in addition to the operating frequency. Fig. 9 shows sample 100x100 input images, and Fig. 10 presents the sparse disparity maps for these images when using a 3x3 correlation window size and a disparity range of 25. Fig. 11 shows the output of the Sobel edge detection for all three sample images.

Table II illustrates the synthesis results when using different correlation window sizes up to 11x11. The size of the on-chip memory depends on the correlation window size and disparity range (i.e. buffer size).

The system performance and quality vary depending on system parameters as mentioned earlier. In this context, system quality refers to the percentage of correctly computed disparity values, while system performance is measured in frames per second. The quality represents the ability to correctly distinguish the depth of each object relative to the other objects in the image, and depends on the quality of the edge detection and the disparity range. The performance is affected by the correlation window size, the disparity range and the initial stereo pair size. As the correlation window size increases, the performance increases, since more search operations can be performed in parallel; as the disparity range increases, the performance decreases as there are more searches required; and as the image size increases, the performance decreases since there is more data to be reconstructed.

The system was also evaluated with and without the edge detection, in order to compare the performance between the two different implementations. By using the edge detection before stereo correspondence, the search data is reduced on an average of 35-55%. Experimental results without the edge detector, over the ten sample images, indicate that the quality of the disparity map computation, drops approximately by 7% when compared to the quality of the system without the edge detector; while this might be a considerable drop (and in large images potentially this drop might rise), a better edge detector, such as the Canny edge detector [1], could improve the overall quality. We leave this as future work however.

Performance simulations were done using an array of parameter configurations, and Tables III-V show simulation results for a range of 10 stereo image pairs, with and without the edge detection unit. The average performance of the system against the disparity range is given in Table III. The performance decreases as the disparity range increases. The edge detection almost doubles the performance for all disparity levels, due to its ability to reduce the search data. Table IV illustrates the impact of the input image size on the system performance, while the disparity level and the window size are kept constant. It is obvious that the performance is inversely proportional with the input image size. The impact of the edge detector is even more emphasized as the input image size increases, as the overall data reduction rate increases over. Lastly, performance results with respect to the correlation window size are given in Table V, while the input image size and disparity range are kept constant. As expected, system performance increases when larger window sizes are used, taking advantage of the intrinsic parallelism of the algorithm, something which software implementations achieve only by using expensive and high-end computers.

Table II: Resource Utilization for Virtex2 XC2VP30 Pro Mapping

<table>
<thead>
<tr>
<th>Correlation Window Size</th>
<th>3x3</th>
<th>5x5</th>
<th>7x7</th>
<th>9x9</th>
<th>11x11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Slices (over 13696)</td>
<td>6447 (47%)</td>
<td>8012 (58.5%)</td>
<td>9568 (69.8%)</td>
<td>10987 (80.2%)</td>
<td>12560 (91.7%)</td>
</tr>
<tr>
<td>Number of slice Flip-Flops (over 27392)</td>
<td>12048 (44%)</td>
<td>13531 (49.4%)</td>
<td>17597 (64.2%)</td>
<td>21912 (79.9%)</td>
<td>26014 (94.9%)</td>
</tr>
<tr>
<td>Number of 4-input LUTs (over 27392)</td>
<td>73830 (27%)</td>
<td>91072 (33%)</td>
<td>12561 (45.8%)</td>
<td>16759 (61.1%)</td>
<td>21426 (78.2%)</td>
</tr>
</tbody>
</table>

Figure 8. Experimental testbed with results shown in enlarged window.

Figure 9. Image stereo pairs: a) Tsukuba b) Baby c) Monopoly.
Simulation results suggest a performance advantage over other comparable hardware implementations, primarily due to the integration of the edge detection algorithm. Even with the 100MHz limitation of the Virtex II Pro FPGA, it yields real-time results, while maintaining a very high quality on the disparity map. We anticipate that future optimizations on the architecture can potentially yield even higher frame rates.

VI. CONCLUSION AND FUTURE WORK

This paper presented an architecture for real-time 3D reconstruction of stereo images. The architecture features a Sobel edge detection that speeds up the system performance, and was designed with emphasis on a number of configuration parameters. A prototype implementation indicated real-time performance under a variety of configurations.

Future work includes experimentation with various edge detection algorithms to improve the quality of the results, and further hardware optimizations to improve the operational frequency and memory accesses. Additionally, we plan on implementing the algorithm on a faster and larger FPGA, to study potential performance benefits in terms of frequency, and other design parameters such as search window size and disparity range, for which existing FPGA resources were limited. Lastly, we plan on explore the rectification process either through a custom rectification unit, or through the use of a faster embedded processor.

REFERENCES