All-hardware acceleration for binocular vision 3D positioning system

Chengbin Yin$^1$ and Xiaosheng Wu$^2$

$^1$Shanghai Jiaotong University
$^2$Shanghai Jiao Tong University

October 16, 2022

Abstract

Traditionally, stereo vision was mainly based on FPGA+CPU method, which was limited by lower computing capability of CPU. An innovative all-hardware acceleration solution of stereo vision is proposed for positioning application, in which image acquisition, caching, preprocessing and 3D positioning modules are all implemented in FPGA. A novel median filter for image preprocessing is proposed to match higher clock frequency. A complex 3D positioning algorithm is given to simplify the process without reasonable reduction in operation accuracy, which makes the system more conducive to implement in FPGA. The experiments show that the all-hardware acceleration solution has outstanding advantages in computing speed, power consumption and response time compared with the traditional FPGA+CPU one.

Introduction:

Stereo vision technology is an important application field of modern computer vision, such as obstacle avoidance of drones, automatic driving of cars, etc. Stereo vision is the process of recovering three-dimensional spatial information by collecting two-dimensional images through cameras. Since the two cameras are in different spatial positions, the images of the same target collected by the two cameras are different. According to the parallax data combined with the camera’s own parameters, the spatial position deviation of the target object relative to the binocular camera can be calculated.[1]

In the early days, Stereo vision technology was mainly implemented based on CPU/GPU, which can achieve positioning accuracy and algorithm flexibility to a large extent.[2-4] However, due to the characteristics of CPU serial computing, its data processing throughput is limited. In the application of image processing, it is difficult to achieve a satisfactory balance between speed, real-time performance, cost and power consumption.

In recent years, some FPGA-based hardware acceleration implementation solutions have been released. [5-9] They all use FPGA+CPU to accelerate the binocular vision system. Compared with the traditional CPU or GPU implementation, the speed and power consumption are greatly improved. These solutions only
accelerate the simple and easy-to-hardware parts of binocular vision for FPGA acceleration, while the more complex part of positioning is still calculated by CPU. Compared with the traditional CPU implementation, such a solution can improve the performance greatly, but it is still limited by the interaction bandwidth between the CPU and the FPGA. Because of the bottleneck of the CPU computing speed, it is difficult to further improve the speed and the real-time performance of binocular vision.

On this basis, this paper proposes a novel FPGA based all-hardware acceleration for binocular vision implementation as shown in the Fig.1. The system, including image acquisition, buffering, preprocessing and 3D positioning modules, are all implemented in FPGA. A novel median filter and more complex positioning algorithm are optimized to suit the characteristics of the FPGA hardware. The CPU is still implemented in the scheme proposed, but it is only used for flexible interaction with the host computer, and the real-time system configuration or station feedback are under control of the host computer to facilitate the debugging. The comparison between the traditional hardware acceleration scheme and the full hardware acceleration scheme proposed is given in this paper.

The paper is organized as follows: in section II, the optimized all-hardware acceleration algorithm of stereo vision are given. The proposed hardware system is presented in section III. The results are provided in section IV. The conclusions are drawn in section V.

![Fig. 1 The comparison between the traditional hardware acceleration scheme and the full hardware acceleration scheme proposed in this paper](image)

**Design specifications:** Impulse noise is common noise in images, generally caused by transmission errors or bit loss. Impulse noise is obviously different from other pixels in terms of gray characteristics, and is generally the grays extreme point in its neighborhood.

The median filter algorithm is simply to sort the pixels in the window neighborhood according to the gray level, and then select the middle value of the group as the output pixel value. Because impulse noise is an isolated extreme point in the image, median filtering is very effective for impulse noise, but the disadvantage is that it is not conducive to hardware implementation. Based on the characteristics of hardware circuits, this paper proposes a median filter circuit scheme that is easy to implement in hardware. The main idea is to divide the complex sorting problem into several simple comparison problems, and implement the sorting circuit by pipeline. The data throughput of the filtering circuit is guaranteed, and the speed of the median filtering of the image is further guaranteed.
Fig. 2 The proposed median filter steps

The median filter is divided into 4 steps. The specific steps are shown in the Fig.2. The first step is to sort the three pixels in each row. The second step is to sort the pixels of each column. In this process, the size relationship shown by the blue arrow can be guaranteed. The two black pixels in the upper right corner must be smaller than the middle value, and the two pixels in the lower left corner must be smaller than the middle value. must be greater than the median value, which can be excluded from the sorting queue. The third step is to sort the four pixels in the lower right corner. After sorting, it can be ensured that the two pixels in the upper right and lower left are larger than the median and smaller than the median, and they are also excluded from the sorting queue. The last step is to sort the three pixels from the upper left to the lower right, and the middle value of the three pixels is the middle value of the nine pixel matrix.

Three-dimensional positioning requires the construction of a three-dimensional world coordinate system and a two-dimensional code image coordinate system. The coordinate system is shown in the left figure. A point in the space will be imaged in the imaging plane of the left and right cameras.

Due to the different spatial positions of the left and right binocular cameras, this results in a parallax between the two images, as shown on the right. Due to the principle of similar triangles, it can be concluded that:

\[
\frac{Z_{W_f}}{Z_w} - \frac{f}{Z_w} = \frac{b - d}{b} \quad (1)
\]

\[
\frac{u_l - u_o}{X_w} = \frac{f}{Z_w} \quad (2)
\]

\[
\frac{v_l - v_o}{Y_w} = \frac{f}{Z_w} \quad (3)
\]

Simplify the above formula to get:

\[
X_w = \frac{b(u_l - u_o)}{d} \quad (4)
\]

\[
Y_w = \frac{b(v_l - v_o)}{d} \quad (5)
\]
\[ Z_w = \frac{bf}{d} \quad (6) \]

\((u_l, u_r)\) is the imaging point of the target in the left eye image, \((v_l, v_r)\) is the imaging point of the target in the right eye image. \((X_w, Y_w, Z_w)\) are the three-dimensional coordinates of the target, \(b\) is the distance between the left and right cameras, and \(f\) is the focal length of the camera. The above formula (1-6) is based on the principle of 3D positioning of binocular vision.

**Fig. 3** The top-level architecture of the proposed stereo vision hardware system

**System architecture and hardware implementation:** The top-level architecture of the proposed stereo vision hardware system is depicted in Fig.3. System mainly divided into: image acquisition module, image cache module, image 3D positioning module, HDMI drive module, DMA transmission module and ARM CPU module.

The image acquisition module is responsible for configuring the resolution and frame rate of the binocular camera, stitching and aligning the 8bits/cycle image data collected by the binocular camera, and then aligning the complete 24bits/pixel data with it. The coordinates in the image are sent to the image cache module aligned.

The image cache module is responsible for caching images through off-chip DDR. Among them, MIG is the DDR controller IP of AXI4 interface provided by vivado. The controller module built in this paper is mainly responsible for arbitrating and splicing the binocular images input by the image acquisition module, and converting the image data into AXI format and inputting it to MIG, and then stored in off-chip DDR3. In addition, according to the configuration of the CPU, the data stored in the DDR is continuously output to the HDMI driver module, filter module or DMA module.

The HDMI module is responsible for outputting the collected images to the display for real-time display; the DMA transmission module is responsible for transmitting the image data at various stages to the host computer at high speed.

The filtering module is responsible for median filtering of image data, and its pipelined implementation is one of the innovations of this paper. Throughout the data path, image data is transmitted and processed as a stream of pixel data. As shown in the Fig.4 below, in the median filter module, first use two end-to-end shift registers to cache the data of the previous two lines, and then transmit the output ports of the two shift registers and the image data input synchronously to 3*3 matrix, so that a 3*3 pixel matrix can be formed, and the subsequent median filtering is to operate the 3*3 pixel matrix. The pixel matrix is processed by a
4-stage pipeline and outputs the median value of 9 pixel points, which represents the median filtering result of the current pixel point.

Fig. 4 The proposed median filter pipelined implementation

In this paper, the edge detection algorithm is used to locate the target. Since the data collected by the binocular camera is in the RGB888 format, the image data needs to be converted from the RGB format to the GRAY format first, and then the SOBEL algorithm is used to detect the image edge of the GRAY image. The conversion of RGB format to GRAY format and SOBEL edge detection are similar to median filtering. The pixel data of two lines is buffered by two lines of shift registers, and is input into the 3*3 pixel matrix synchronously with the input pixel data. In this way, a 3*3 pixel matrix is formed in the continuous pixel data for subsequent calculation.

The target center point calibration adopts the algorithm of traversing the entire image, that is, traversing the pixel points of the entire image, and recording the coordinates at the edge. The three-dimensional positioning of the target is to obtain the three-dimensional coordinates through the calculation of the above formula on the basis of determining the center points of the left and right image targets. Among them, multiplication and division are implemented using the corresponding IP core of XILINX.

Implementation and results: Benefiting from the hardware programmable features of FPGA, the accuracy and speed of the 3D positioning system are no longer limited by the computing power bottleneck of the embedded CPU and the bandwidth of data exchange between the CPU and FPGA. In the hardware system implemented in this article, the data throughput bandwidth of core computing modules such as image preprocessing and 3D positioning reaches more than 4GB/s, which is much higher than the bandwidth of the data exchange port (HP port) between CPU and FPGA. In addition, the hardware system proposed in this paper is easy to expand, and the data transmission bandwidth is no longer the bottleneck of performance. More complex positioning algorithms can be implemented in FPGA, thereby further improving the accuracy of 3D positioning.

The hardware acceleration solution proposed in this paper also has outstanding advantages in power consumption and resource consumption. As can be seen in the figure below, the overall power consumption of the hardware acceleration system proposed in this paper is nearly 2W, which is much lower than the traditional CPU+FPGA solution. The system proposed in this paper has outstanding advantages in terms of hardware resource consumption. Among them, LUT occupies 14908 and FF occupies 23230, which is only about 5% of the ZUNY development board selected by the system, and has reserved a lot of room for
expansion. The system can be transplanted into a lower cost FPGA system to adapt to the application scenarios that are more sensitive to area and cost.

The hardware 3D positioning system designed in this paper is applied to the actual object positioning: the test is divided into horizontal, front and rear and vertical position changes. The test results are shown in Fig.5: The positioning error is the arithmetic square root of the X, Y and Z direction errors. It can be seen that the positioning error is around 4.5cm, of which the Z-direction error is large due to the long distance in the Z-direction (~70cm).

![Fig. 5](image)

**Fig. 5 The error of the implementation in this paper**

As can be seen in the Table 1 below, the error of the implementation in this paper is basically at the same level of accuracy compared with other similar articles, but it has advantages in power consumption, speed, and system data throughput.

<table>
<thead>
<tr>
<th>Design Proposed</th>
<th>[10]</th>
<th>[11]</th>
<th>[12]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>6.76%</td>
<td>6.22%</td>
<td>7.45%</td>
</tr>
<tr>
<td>Device</td>
<td>XC7Z100T</td>
<td>XC7VX9880T</td>
<td>XC6VLX760</td>
</tr>
<tr>
<td>Image size</td>
<td>640×640</td>
<td>640×480</td>
<td>640×480</td>
</tr>
<tr>
<td>Resources</td>
<td>LUT:14908</td>
<td>Slices: 134153</td>
<td>Slices: 80270</td>
</tr>
<tr>
<td>Speed</td>
<td>150MHz (366fps)</td>
<td>54fps</td>
<td>68fps</td>
</tr>
<tr>
<td>Design</td>
<td>Proposed</td>
<td>[10]</td>
<td>[11]</td>
</tr>
</tbody>
</table>

**Table 1: the error of the implementation in this paper and other similar articles**

**Conclusions:** This article proposes an innovative binocular vision 3D positioning scheme, including architecture design, core filtering algorithm, hardware implementation of positioning algorithm, etc. Under the premise that the positioning accuracy is equal to several state-of-the-art competitors, it has outstanding advantages in terms of speed, system data throughput, power consumption and resource consumption.

Chengbin Yin, Xiaosheng Wu

E-mail: xswu@sjtu.edu.cn

**References**
1. O. Faugeras, Three Dimensional Computer Vision: A Geometric Viewpoint, Cambridge, MA, USA, MIT


aggregation and dynamic programming. In Proceedings of the 3rd International Symposium on 3D Data
Processing, Visualization, and Transmission. IEEE, 798–805


7. M. Perez-Patricio, A. Aguilar-Gonzalez, "FPGA implementation of an efficient similarity-based adaptive
window algorithm for real-time stereo matching", in press on Journal of Real-Time Image Processing DOI

8. G. Cocorullo, P. Corsonello, F. Frustaci, S. Perri, "An efficient hardware oriented stereo matching algo-

rithm and its VLSI Architecture Design", IEEE Transactions on Circuits and Systems for Video Technology,
Vol.25, n°6, 2015, pp.1038-1050.

10. G. Cocorullo, P. Corsonello, F. Frustaci, S.Perri, "An efficient hardware oriented stereo matching algo-


rithm and its VLSI Architecture Design", IEEE Transactions on Circuits and Systems for Video Technology,
Vol.25, n°6, 2015, pp.1038-1050.