

基于 Xilinx Artix7 的实时双目立体视觉系统

第一部分 设计概述

1.1 设计目的

在 FPGA 上实现立体匹配算法时候，资源消耗是其中的一个问题。该问题通常通过优化算法来解决，但也可以通过在不修改算法的情况下实现新的硬件架构来解决此类问题，本作品挑战在资源有限的低端 Artix7 开发板上实现双目立体视觉系统，本作品以最小化资源、高实时性、高准确性、高视差范围为设计目的硬件实现了双目立体视觉 SAD 算法。系统使用两个平行配置的低成本 OV7670 摄像头进行图像采集，最大分辨率为 640*480，最大帧率为 30fps。系统基于 VGA 接口进行图像数据传输和显示，并可利用控制开关切换传输内容，包括左右原始图像和处理后的视差图像。

对比其他项目通过优化算法来解决资源消耗问题，在我们设计中则是采用新的硬件架构，通过采用主动式内存控制器、FIFO 和行缓存解决了内存竞争和带宽限制问题，并利用并行处理，在低时钟频率下实现高处理速度，达到节省逻辑资源的优化目标。

1.2 应用领域

目前，立体视觉技术广泛应用于三维重建、自动驾驶、机器人自动导航等领域。而且，随着 Google Glass、Oculus Rift、HoloLens 等设备的推出，虚拟现实与增强现实这看似未来科技的事物走入了人们的视线，随着这些产品的落地，立体视觉技术必然会在虚拟现实与增强现实中获得更为快速的发展。

1.3 主要技术特点

我的主要挑战是要在有限的资源上运行复杂的图像处理算法。深度图像检测方法上采用了 SAD 算法，其具有运算简单，资源消耗少，易于实时检测的特点。

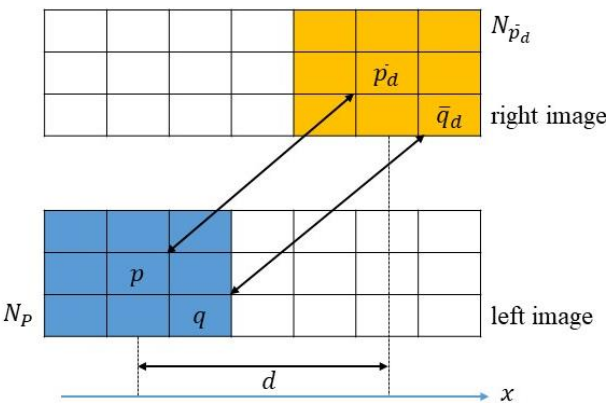


图1 SAD 算法检测过程

SAD 算法基本思想：差的绝对值之和。此算法常用于图像块匹配，将每个像素对应数值之差的绝对值求和，据此评估两个图像块的相似度。

$$D(x, y, d) = |I_l(x, y) - I_r(x - d, y)|^2$$

1.4 关键性能指标

电路连接图（外接 0V7670 双目摄像头和 VGA 输出接口）图 3 所示。

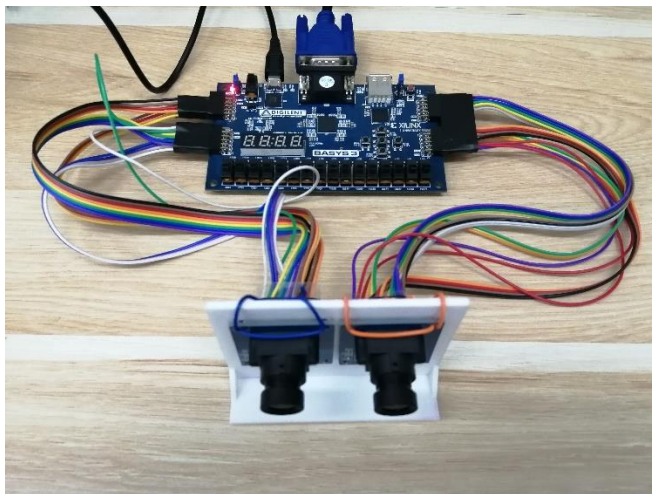
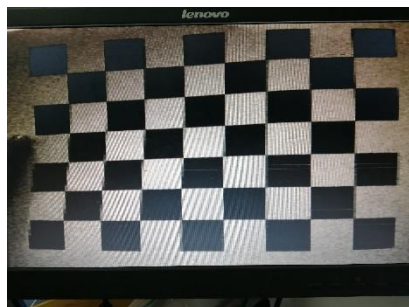
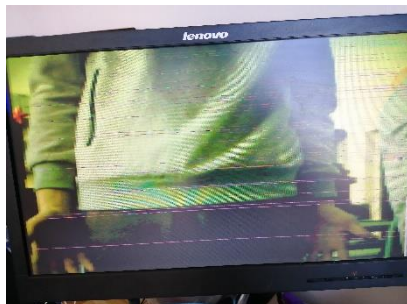
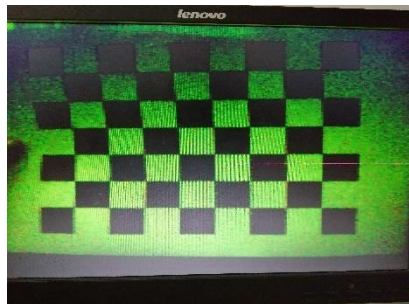


图 3 作品概貌

效果图如下所示，其中图 a 是原图场景，图 b 是矫正后效果，图 c 是深度图。



(a)原图场景



(b) 矫正后效果图



(c) 深度图(左图视差图, 右图平均图)

图4 立体视觉图像举例

第二部分 系统组成及功能说明

2.1 基础理论

为了实现立体视觉系统,对摄像机进行了设置,使两个光轴平行。下图是此双目立体视觉系统的垂直投影, A 是被计算对象上的一点。 M 和 N 是两个光学中心, $MN=l$ 是两个光学中心之间的距离。 B 和 C 分别是 A 在左右摄像机焦平面上的投影, 因此,从 A 到相机系统的距离是 AB 。因为两个摄像头 (MN) 之间的距离以几厘米为单位,而物体到摄像头 (AB) 之间的距离则要大得多 ($AB \gg MN$), 也就是说, 三角形 ABC 非常尖锐, 所以 $AB \approx AC$ 。因此, 我们不必计算 AC , 让 $NK \parallel AB$ 。由于两个三角形 ABC 和 NKC 相似, 因此我们具有:

$$\frac{AB}{NK} = \frac{BC}{KC}$$

$$AB = \frac{NK + KC}{KC} \times \sqrt{NH^2 + HK^2}$$

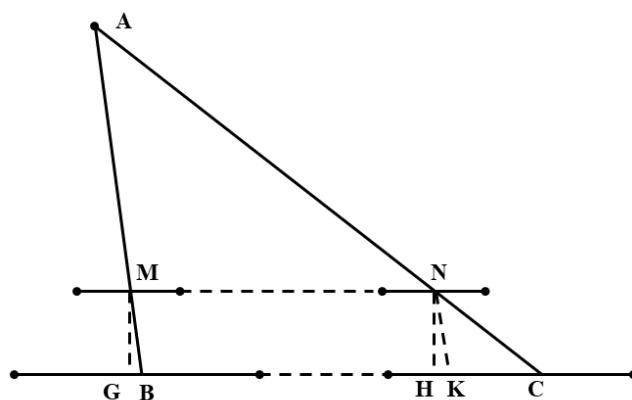


图5 立体视觉系统的垂直投影

令 $KC = d$ 是 A 在右图像上的投影与左图像相比的位移, $MG = NH$ 是焦距 f , $GB = HK$ 。

$$AB = \frac{l + d}{d} \times \sqrt{f^2 + GB^2}$$

GB 是从要计算的像素到图像中心的距离, 两个摄像头之间的距离和焦距 f 是已知的。

通过找到两个摄像头上 A 的两个投影的位移 d 解决了找到 AB 的问题。由于已知相机传感器上的像素大小，因此只需找出 A 的投影，则 $d = \mu \cdot P$ ，其中 μ 是光学传感器上像素的大小，P 是两点 K 和 C 之间的像素数。

则上式可以改写成：

$$d = \frac{l\sqrt{f^2 + GB^2}}{AB - \sqrt{f^2 + GB^2}}$$

分母中的 AB 表示近距离物体产生较大的位移，而远处物体产生较小的位移。位移在结果图像上表示为像素强度。因此，在结果图像上，近处的物体看起来比远处的物体亮。

视差算法通常涉及匹配的成本函数，匹配成本是相似性或差异性的函数，视差值 (d) 作为变量。通常，视差算法会尝试找到相似性函数 $f(d)$ 的最大值或差异函数的最小值。例如，窗口大小为 3x3 像素的 SAD 算法的匹配成本函数为

$$f(d) = \sum_{i=1}^9 |E_i(d)|$$

其中 f 是匹配成本函数， d 是当前视差值； E_i 是两个比较窗口在同一位置的两个像素之间的强度差。当视差值 d 最小化绝对差函数 $f(d)$ 之和时，将这两个窗口视为“匹配”。当差异值最大化相似性函数或最小化差异函数时，就会发现差异值。

2.2 整体介绍

如下图所示系统的框图，该系统具有三个主要模块。第一个模块是相机接口模块，它从两个相机捕获像素数据，并写入两个先进先出缓冲器 (FIFO)，等待写入外部存储器。第二个模块是多端口内存控制器，它管理来自其他模块的所有内存访问请求，并确保没有内存竞争。第三方框代表具有视差匹配模块的处理单元。其他模块是时钟管理模块和 VGA 控制器模块。

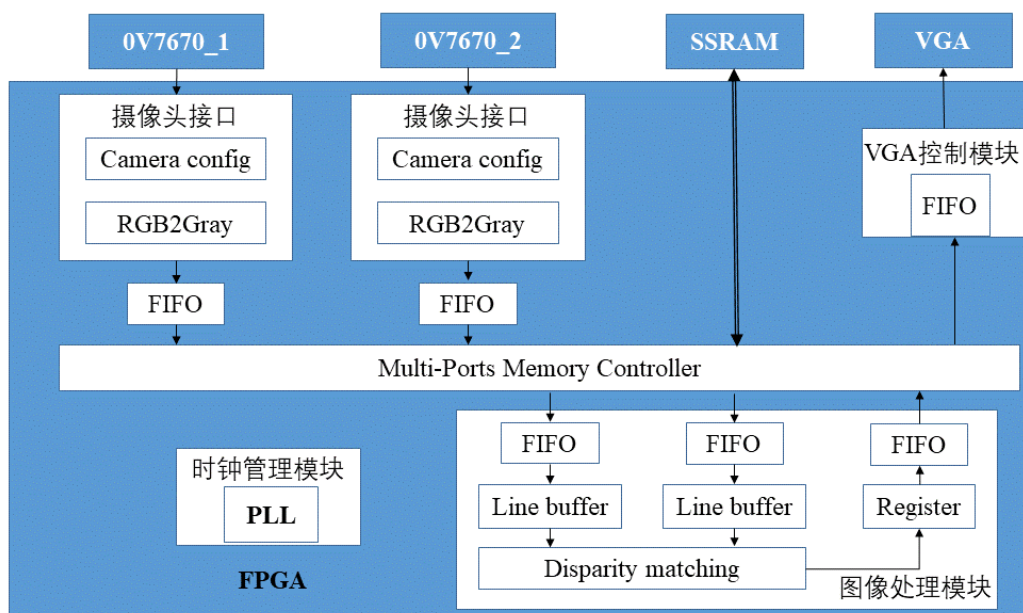


图6 系统框架

2.3 各模块介绍

根据总体系统框图，给出各模块的具体设计说明。

1. Pre-processing

本作品的 OV7670 双摄像头安装架是使用 cad 工具设计的，并通过 3D 打印来安装摄像头。

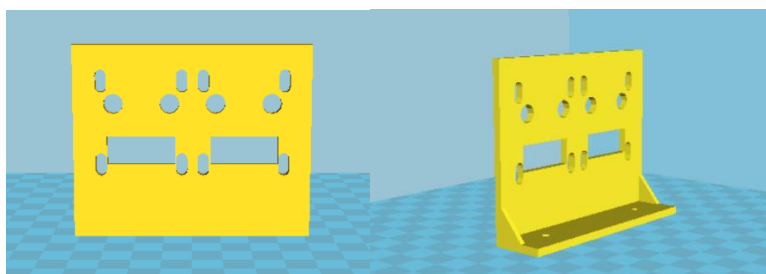


图7 OV7670 双摄像头安装架

本作品的低成本 OV7670 模块且输出可以配置为 8 位并行，采用 IC 通信协议对两个 OV7670 进行功能配置。按照 IIC 协议标准，每次通信分别传送 3 个字节控制字到 OV7670 的内部寄存器，这三个字节分别为 OV7670 器件地址、需要配置的功能寄存器地址、需要写入制。需要配置的功能包括：工作时钟、图像数据格式、输出格式、亮度控制、颜色控制、增益补偿、亮度对比度控制等。

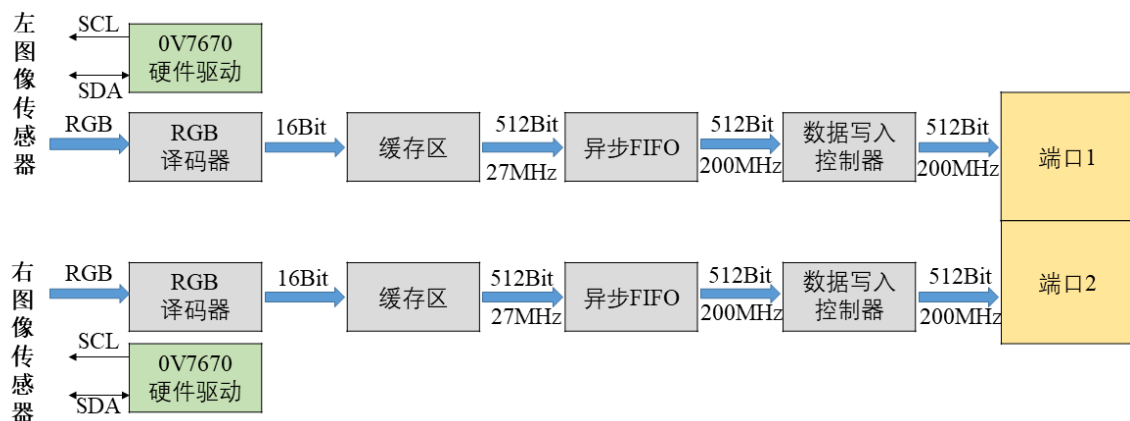


图8 采集控制单元工作流程

像素以 G1, G2, R, B 四种颜色的 bayer 格式从相机输出。每个 bayer 格式彩色像素具有 12 位 ADC 分辨率，其中 G1 和 G2 相同。捕获像素数据后，摄像头接口模块会将其转换为灰度图像。仅灰度像素写入存储器。这样，可以节省大量的内存带宽，或者不存储原始图像数据并再次访问它。

2.异步通信

OV7670 输出的图像数据为 16 位，而片外存储的数据位宽大小为 512 位，因此需要对数据位宽进行匹配。由于每个时钟周期接收到 8 位图像数据，而两个时钟周期才能接收到一个像素点的完整 16 位数据，因此每次可以存储 32 个像素点的数据到片外存储。

OV7670 图像传感器的驱动时钟频率为 24MHz，多端口存储控制器的驱动时钟频率为 200MHz，因此需要采用跨时钟域数据通信设计。由于这里通信的数据位宽大小为 512 位，因而这里使用了异步 FIFO

3. 处理单元

图 3 显示了处理单元的设计。该模块由两个输入 FIFO 组成，它们从存储器读取左右图像的像素数据。像素数据然后被写入到两个行缓冲器中，右上角的一个称为主行缓冲区。它存储两行图像像素，具有一个输入和三个输出。主线缓冲器具有 1280 个寄存器。主线缓冲器输出的数据将传递到左上角的第二个基于寄存器的行缓冲区（辅助行缓冲区）中。第二行缓冲器的目的是在要处理的图像数据行上选择一个片段。左边的辅助缓冲区每行只有三个寄存器（等于窗口大小），而右边的辅助缓冲区有 34 个寄存器，用于 32 个像素的视差范围。当新数据被写入到行缓冲器中并且先前的数据向左移动时，这类似于在输入图像上从左向右移动窗口。

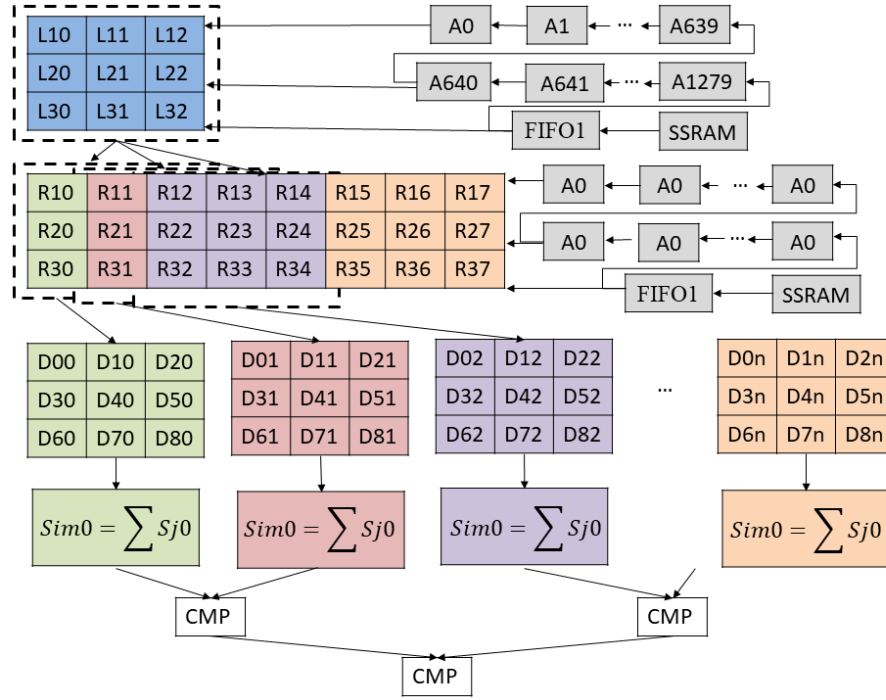


图9 视差匹配模块

将绝对减法 $D[j][k]$ ，绝对差之和 $SAD[k]$ ，总相似度 $Sim[k]$ 和比较器(CMP)建模为组合逻辑电路，其中 $0 \leq j \leq 8$ 和 $0 \leq k \leq 31$ 。在时钟的每个上升沿，一个像素移至寄存器阵列左侧的下一个位置，并且从行缓冲器中写入一个新像素。

$$L[i][k] \leftarrow L[k][i+1], i = 1, 2, 3; 0 \leq k \leq 3$$

$$R[i][k] \leftarrow R[k][i+1], i = 1, 2, 3; 0 \leq k \leq 3$$

新的像素从 FIFO 送到第一个行缓冲区的开头。当 FIFO 为空时，它将填充图像同一行中的连续像素。

$D[j][k]$ 是两个像素之间的绝对差。

$$D(0+w)[k] = |L1[w] - R1[k+w]|, w = 0, 1, 2$$

$$D(3+w)[k] = |L2[w] - R2[k+w]|, w = 0, 1, 2$$

$$D(6+w)[k] = |L3[w] - R3[k+w]|, w = 0, 1, 2$$

并行执行 $9 \times 32 = 288$ 个绝对减法，将绝对差输入绝对差总和中，所有 $SAD[k]$ 的计算和也并行进行。

$$SAD[k] = \sum D[j][k]$$

如果两个像素之间的强度差小于阈值 T ，则认为两个像素相似。用户可以通过 FPGA 板上的开关设置此阈值。通常，将 T 设置为 10。每个窗口的相似像素数量相加，其值从 0 到 9。

$$S[j][k] = \begin{cases} 1, & \text{if } D[j][k] < T \\ 0, & \text{if } D[j][k] > T \end{cases}$$

$$Sim[k] = \sum_{j=0}^{j=8} S[j][k]$$

比较这些绝对差之和(SAD [k])，以找到最小值 (min SAD [k])。右行缓冲区上产生最小 (SAD [k])和最大相似像素(max(Sim [k]))的窗口被认为与左窗口“匹配”，其对应的位移是视差值，将位移量作为像素强度写入视差图像中，比较器设计为二进制。因此，级别 1 具有 16 个比较器，级别 2 具有 8 个比较器。依此类推，在找到视差值之前，有 5 个比较级别。将数据从行缓冲器传递到最后一个比较器的视差寄存器只需一个时钟周期。

4. 资源约束

我们设计使用了 basys3 FPGA 开发板，Basys3 是围绕着一个 Xilinx Artix®-7 FPGA 芯片 XC7A35T-1CPG236C 搭建的，适合于从基本逻辑器件到复杂控制器件的各种主机电路，但是它不适用于图像处理任务，具体资源如下表所示。

表 1 关键指标特性

LUT	LUTRAM	FF	BRAM	DSP	IO	BUFG	MMCM
20800	9600	41600	50	90	106	32	5

主要困难是 Block 存储器和 LUTRAM，Basys 3 在 50 个单元中有 1,800Kbit 的内存，每个单元中有 36,000 位，需要获得 YCbCr 格式的相机输出。Y 是单个像素大小为 4 位的灰度图像。我们无法处理它们在 BRAM 中的图像，因为 BRAM 是 FIFO（先进先出）存储设备。因此，应该将映像加载到缓存中，以便可以并行访问数据。可以使用 LUTRAM 创建缓存。我们遇到的问题是图像也无法存储以及支持 SAD 算法进行比较。

设计使用了四个行缓冲器，这些行缓冲器使用寄存器建模，从逻辑门创建寄存器。因此，这种设计行缓冲器的方法需要大量的逻辑元件(LE)来创建寄存器，逻辑元素在 FPGA 上非常受限制。

设计行缓冲器的另一种方法是使用片上专用 RAM 模块，行缓冲器用作具有顺序读和写访问端口的 8x1280 存储块，行缓冲区具有一个输入写指针和两个输出读指针，如图 4 所示。

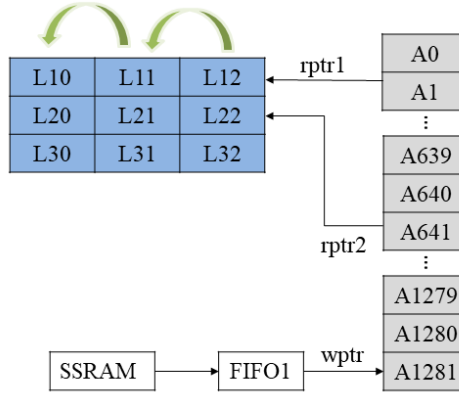


图 11 两种类型的行缓冲器

使用基于存储器的行缓冲器，它只需要 146 个 LE 来作为读写指针，而不是 30272 个 LE。但是，它需要 40,960 个专用存储位来存储像素数据（占可用存储位的 5%）。因此，我们已经从使用关键资源转变为不太重要的资源，存储器位还比逻辑寄存器需要更少的芯片面积。在图 3 中，D20 是 R12 和 L12 的绝对差，因此，使用减法电路来计算 D20，对于 D10 和 D00 类似于：

$$D10 = |L11 - R11|$$

$$D00 = |L00 - R00|$$

但是，一个时钟周期后，L12 的值将移至 L11，同时 R12 的值将移至 R11。因此，L11 - R11 与上一个时钟周期的 L12 - R12 具有相同的值，不必重复相同值的减法三次，代替使用三个减法器来计算 D20、D10 和 D00，可以将两个寄存器与一个减法器一起用于 D20。

$$D20 = |L12 - R12|, D10 \leq D20, D00 \leq D10$$

这样可以减少减法器数量的 2/3，在实际系统上，此步骤显示整个系统的 LE 使用量减少了 24.75%。

通过最小化资源使用，Basys3 FPGA 上有更多资源可以进一步提高系统性能。图 3 中的系统改进到了 64 像素视差范围和 5x5 像素窗口大小。较大的窗口大小和视差范围可提供更好的图像质量，但会使系统变慢。但是，仍然可以满足实时性要求，从 VIVADO 软件的编译报告中，最大时钟频率已从 13.80 MHz 降至 12.17 MHz。而且，在 32 像素视差范围架构中，最大帧速率降低为 39 fps，而不是 41 fps。

2.4 Vivado Block Design

下图是 vivado 的 block design，详细模块链接在 rar 重点代码压缩包中 blockdesign.pdf 查看。

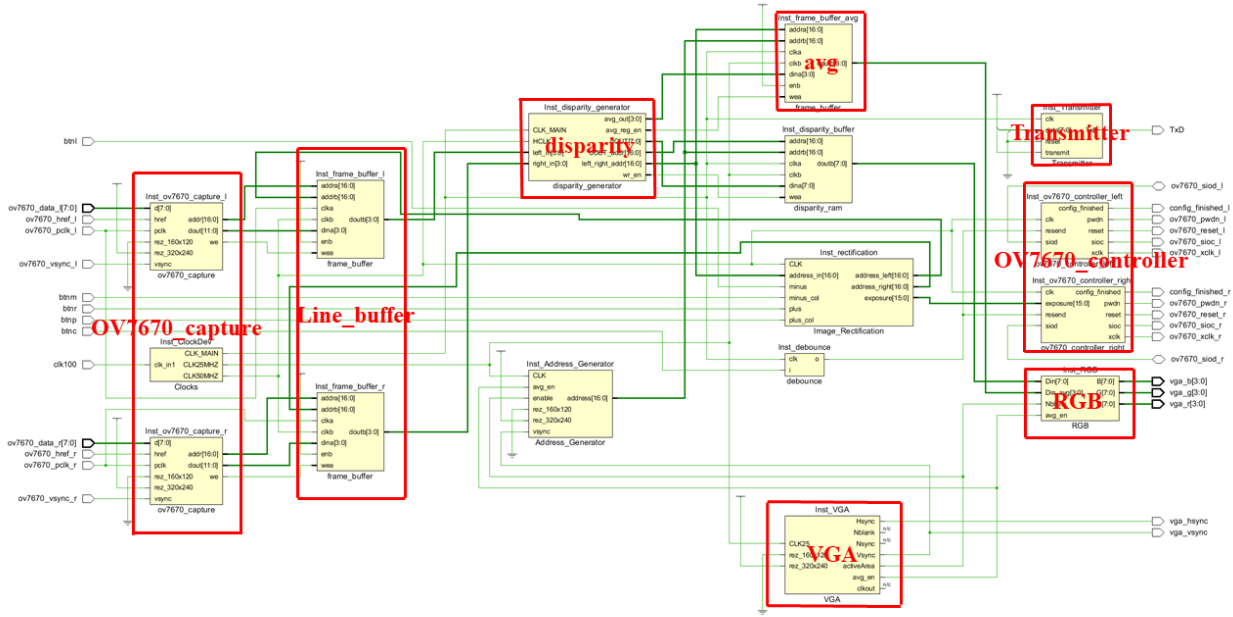


图 12 block design

表 2 Block Design 中主要模块

OV7670_capture OV7670_controller	采用 IC 通信协议对两个 OV7670 进行功能配置。按照 IIC 协议标准，每次通信分别传送 3 个字节控制字到 OV7670 的内部寄存器，这三个字节分别为 OV7670 器件地址、需要配置的功能寄存器地址、需要写入制。需要配置的功能包括：工作时钟、图像数据格式、输出格式、亮度控制、颜色控制、增益补偿、亮度对比度控制等。
RGB	经过配置后的 OV7670 输出的图像格式为 RGB565640*480 分辨率、30fps。其输出图像的同时会伴随行 H、场 V 同步信号来表征正在传输的位置。RGB565 即采用 RGB 数据格式表示图像信息，其中 R.B 通道分别有 5 位数据，G 通道有 6 位数据，这样 R、G、B 通道共 16 位数据可以用来表示一个像素点的完整彩色信息。由于 OV7670 每个时钟周期只能传输 8 位数据，因此两个时钟周期可以传输一个像素点的完整 RGB 数据。而 OV7670 使用的时钟为 24MHz，因此 RGB565 译码器同样采用 27MHz 的时钟驱动。
Line_buffer	该模块由两个输入 FIFO 组成，它们从存储器读取左右图像的像素数据。像素数据然后被写入到两个行缓冲器中，它存储两行图像像素，具有一个输入和三个输出，主线缓冲器具有 1280 个寄存器。主线缓冲器输出的数据将传递到的第二个基于寄存器的行缓冲区(辅助行缓冲区)中，第二行缓冲器的目的是在要处理的图像数据行上选择一个片段。左辅助缓冲区每行只有三个寄存器（等于窗口大小），而的辅助缓冲区有 34 个寄存器，用于 32 个像素的视差范围。当新数据被写入到行缓冲器中并且先前的数据向左移动时，这类似于在输入图像上从左向右移动窗口。

Transmitter	OV7670 图像传感器的驱动时钟频率为 24MHz，多端口存储控制器的驱动时钟频率为 200MHz，因此需要采用跨时钟域数据通信设计。由于这里通信的数据位宽大小为 512 位，因而这里使用了异步 FIFO。
Dsparity	将绝对减法 $D[j][k]$ ，绝对差之和 $SAD[k]$ ，总相似度 $Sim[k]$ 和比较器 (CMP)建模为组合逻辑电路，其中 $0 \leq j \leq 8$ 和 $0 \leq k \leq 31$ 。在时钟的每个上升沿，一个像素移至寄存器阵列左侧的下一个位置，并且从行缓冲器中写入一个新像素。比较这些绝对差之和($SAD[k]$)，以找到最小值 ($\min SAD[k]$)。缓冲区上产生最小($SAD[k]$)和最大相似像素 ($\max(Sim[k])$)的窗口被认为与左窗口匹配
avg	OV7670 模块输出可以配置为 8 位并行，两个摄像头使用 I2C 接口与主机通信，用于从两个头读取图像并显示来自 VGA 输出的两个图像的平均值。
VGA	VGA 接口传输的是模拟视频数据信号，需要将数字图像数据分别解析成 R、G、B 三原色和 H 行、V 场同步信号后利用 VGA 接口进行传输，由于传输的是模拟信号其传输损耗较小。

第三部分 完成情况及性能参数

3.1 最终完成效果

电路连接图，在文档的第 2 页，也就是第一部分中图 3 所示。效果图，在文档的第 3 页，也就是在第一部分中图 4 所示。

3.2 功能验证

为了对基于 Xilinx Artix7 的实时双目立体视觉系统进行功能验证，本文设计并依次完成了三个验证实验：(实验一)原图视频的 VGA 实时显示、(实验二)视差图实时显示、(实验三)优化后视差图实时显示。表 3 给出了实验的目的，具体细节将在下文展开。

表 3 验证实验的概况

序号	实验目的
实验一	验证采集模块、存储模块、VGA 模块和矫正效果
实验二	验证采集模块、存储模块和双目立体视觉 SAD 算法
实验三	验证采集模块、存储模块和优化后图像处理模块

实验一：

经过配置后获得的图像数据经过 VGA 接口连接到显示器后就可显示出来，由于图像采集模块采用的双 CMOS 图像传感器进行图像采集并输出视频流，采用控制开关来控制输出左视频流或右视频流。本实验中利用采集控制单元配置 OV7670 使之输出 RGB565 格式图像数据，图像分辨率为 640*480，帧率为 30fps，分别为左右两个 OV7670 采集到的图像。



(a)左图像



(b)右图像

图1 灰度化实时显示

校正后的图像对将被移动到相同的图像平面上，它们的坐标将是彼此平行的，这使得线搜索的立体匹配可以进行。



(a)左图像



(b)右图像

图1 图像矫正效果

实验二：

采用 SAD 算法进行处理，得到视差图后通过存储控制单元存储到外部存储。经过上述配置后同样采用数据线连接系统与显示器后，可以实时显示处理后的视差图，这里同样可以采用控制开关的形式控制输出的图像内容，左图是视差图显示、右图是两个 OV7670 采集图像的平均值显示。



图1 视差图的实时显示（左视差图，右平均图）

实验三：

优化后的视差图实时显示，其基本配置与实验二视差图的实时显示相同，唯一的不同的是把分别校准照相机中的曝光并改善图像校正模块。此时输出的视差图的输出处于良好状态，由于摄像机无法正确识别功能，因此存在计算点混乱问题，但是输出是平滑清晰的。

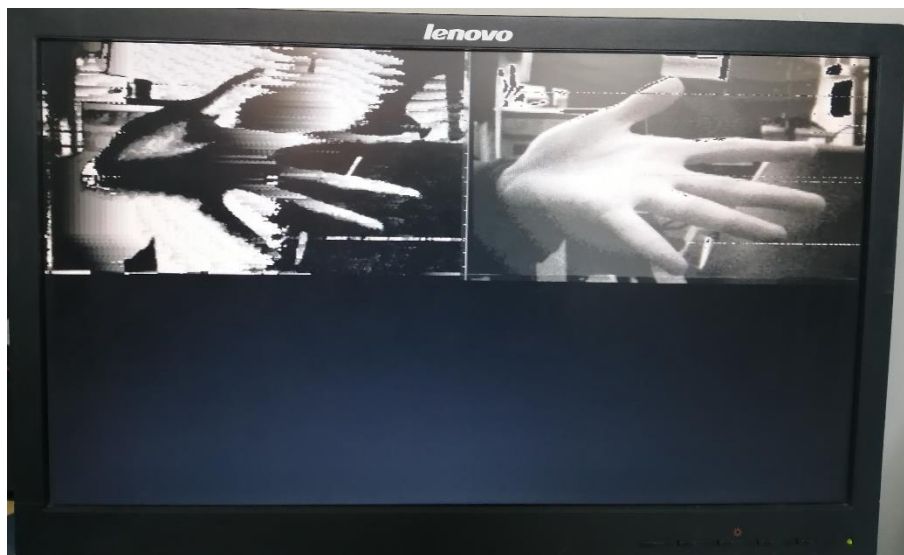


图1 优化视差图的实时显示（左视差图，右平均图）

3.3 性能参数

型号：xc7a35tcp236-1

时序报告图所示。

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 6.587 ns	Worst Hold Slack (WHS): 1.103 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 1	Total Number of Endpoints: 1	Total Number of Endpoints: 2

图 30 Timing Summary

功耗图所示。

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power: 0.08 W
Design Power Budget: Not Specified
Power Budget Margin: N/A
Junction Temperature: 25.4°C
 Thermal Margin: 59.6°C (11.8 W)
 Effective θ_{JA} : 5.0°C/W
 Power supplied to off-chip devices: 0 W
 Confidence level: Low

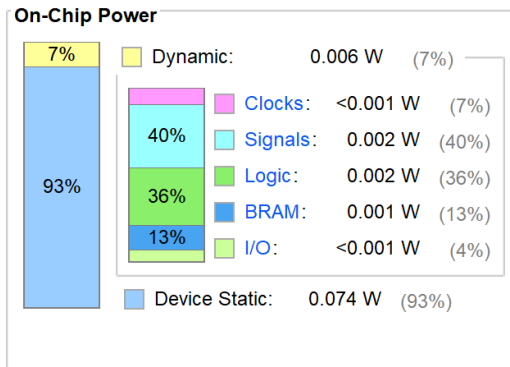


图 31 功耗情况

第四部分 总结

4.1 可扩展之处

(1) 基于 Xilinx Artix7 的实时双目立体视觉系统能够扩展其他功能，例如可以根据需求增加无线或有线以太网传输功能。还可以考虑引入 FPGA+ARM 芯片，采用硬件与软件相结合的方式来提升处理能力。

(2) 可以考虑将现有系统与室内导航系统进行结合。可以利用系统实时输出的深度信息结合 SLAM 算法进行室内导航和重建，可以考虑将实时双目立体视觉系统与实现 SLAM 算法的导航系统及自动机器人结合实现完整的室内自动导航与重建系统。

4.2 心得体会

本作品能够基于双 CMOS 图像传感器进行图像的获取、存储、传输和双目立体视觉算法处理。采集上使用平行排布的两个 CMOS 图像传感器实时采集分辨率为 640*480、帧率为 30fps 的 RGB 彩色视频图像；存储上使用多端口存储控制器与片外存储单元 DDR3 SDRAM 进行数据交互；处理上使用基于自适应权重算法的图像处理模块从采集到的左右视频图像中进行深度信息提取；传输上使用 VGA 接口及协议进行原始视频图像以及处理后的视差图像的显示。

刚开始入门是很痛苦的，嵌入式设计需要从硬件到驱动到软件全部熟悉，硬件系统问题还不是很大(以前做过单片机，DSP 等 MCU),处理器的架构心里还有点数，对于驱动和软件工程，刚开始学习真的很头痛。我们自己对 FPGA 的学习有一段时间了，练习了很多实

例，我们也编写了不少程序，也有了一些项目经验，算是对 FPGA 有些了解了。在不断的学习中发现 FPGA 不仅可以做逻辑设计，算法设计等，还能做嵌入式开发，学习真的是无止境，又一个全新的领域摆在我们面前。

我们作品可扩展之处还有很多，以后的路还很长，不过也有很多惊喜在等着我们.....

第五部分 参考文献

- [1] 零基础 5 分钟上手 FPGA 硬件 Basys3 <http://www.digilent.com.cn/studyinfo/66.html>
- [2] Basys 3 数电基础课程实验官方指导手册及工程文档
- [3] 《搭建你的数字积木—数字电路与逻辑设计》汤勇明编著/2017-06-01/清华大学出版社
- [4] 《基于 Xilinx Vivado 的数字逻辑实验教程》廉玉欣编著/2016-08/电子工业出版社
- [5] 于勇，张晖，林茂松，基于双目立体视觉三维重建系统的研究与设计[J].计算机技术与发展.2009,6,19（6）.127-131.
- [6] Tan X, Sun C, Sirault X, et al.Stereo matching using cost volume watershed and region merging[J].Signal Processing Image Communication, 2014, 29（10）: 1232-1244.
- [7] 刘然，基于计算机立体视觉的双目立体成像研究[D].重庆大学,2007.
- [8] Qian Z B, Qiu Z G, Zhang Y Q.Reviewing the Difficult Problems Facing the Researchers of Stereoscopic VisionfJ].Journal of the Pla Institute of Surveying&Mapping,2001.