doi: 10.12068/j.issn.1005-3026.2020.06.004

一种新的基于FPGA的立体图像差异性算法

李贞妮, 王 骄, 李晶皎, 王泽坤
(东北大学 信息科学与工程学院, 辽宁 沈阳 110819)

摘 要: 提出了一种新的基于FPGA的立体图像差异性算法,它以块匹配算法为基础,根据FPGA的特点,对图像相关性的公式进行设计优化,并结合穷尽方式搜索和预测方式搜索,提高算法的执行速度. 设计基于FPGA的立体图像差异性算法IP核,充分利用FPGA独特的并行处理机制和强大的运算能力以提高系统的处理速度和性能. 系统测试结果表明,基于FPGA的立体图像差异性算法,可以达到每秒33帧的处理能力,处理速度能够达到PC机的二百倍以上,具有较好的实时性;且能够连续处理500帧图像数据,具有较好的稳定性.

关 键 词: 立体图像;差异性算法;块匹配;FPGA;IP核
中图分类号: TP 393.0 **文献标志码**: A **文章编号**: 1005-3026(2020)06-0778-06

A New Stereo Image Disparity Algorithm Based on FPGA

LI Zhen-ni, *WANG Jiao*, *LI Jing-jiao*, *WANG Ze-kun*
(School of Information Science & Engineering, Northeastern University, Shenyang 110819, China. Corresponding author: WANG Jiao, E-mail: wangjiao@ise.neu.edu.cn)

Abstract: A new stereo image disparity algorithm was proposed based on FPGA. Based on the block matching algorithm and according to the characteristics of FPGA, the formula of image correlation was optimized. Moreover, by combining exhaustive search and prediction search method, the execution speed of the algorithm was improved. The Stereo image disparity algorithm IP core was designed, and the processing speed and performance of the system could be improved by making full use of the FPGA's unique parallel processing mechanism and powerful computing ability. The test results show that the stereo image disparity algorithm based on FPGA can reach the processing capacity of 33 frames per second and the processing speed can reach more than 200 times of PC, with good real-time performance. It can process 500 frames of image data continuously and has good stability.

Key words: stereo image; disparity algorithm; block matching; FPGA; IP core

随着科学技术和工业生产的快速发展,许多领域都需要对立体物体的形状进行快速准确的测量. 现代大工业生产中,为获取产品的空间信息而进行的三维自动测量是人们普遍关注的技术[1-2]. 而立体图像差异性算法是立体视觉的关键研究内容,因此,研究立体图像差异性算法具有重大的理论和现实意义[3-4].

本文研究的立体图像差异性算法是基于激光散斑的结构光方法[5],该方法的实现需要一个能发射激光散斑的光源和一个灰度摄像头. 每次算法的计算只需要拍摄一张场景图像,然后通过比较场景图像和参考图像之间的差异性得到立体图像. 其计算量非常大,需要进行大量的乘法和除法运算,因此保证该算法处理的实时性是设计立体图像差异性算法处理系统的关键问题.

目前已有的算法在实现的过程中,为了保证系统的实时性,通常采用基于高端显卡的设计. 这种设计架构方式具有很高的实时性,能够满足系

收稿日期: 2019-10-17
基金项目: 国家自然科学基金重点项目(61836011); 中央高校基本科研业务费专项资金资助项目(2020GFYD011, 2020GFZD008).
作者简介: 李贞妮(1982-),女,辽宁沈阳人,东北大学讲师,博士; 王 骄(1978-),男,辽宁沈阳人,东北大学教授,博士生导师; 李晶皎(1964-),女,辽宁沈阳人,东北大学教授,博士生导师.

统处理速度的要求. 但是其电路结构极其复杂,功耗很高,系统可扩展性不高,价格昂贵,设备开发和调试十分困难,后期的升级和维护也十分不易[6-7].

针对现有技术的不足,本文提出了一种适合于 FPGA 硬件实现的立体图像差异性算法,对立体图像差异性算法进行了改进,并能够充分利用 FPGA 的高速性和并行性,具有实时性、开发周期短、成本低、功耗小、控制方便等优点. 改进后的立体图像差异性算法的实现和处理基于 FPGA 硬件开发平台,充分发挥了硬件加速的优势;利用 FPGA 中的 Nios II 软核处理器对数据进行管理,该过程在 FPGA 片内总线上完成,解决了数据传输的瓶颈,充分发挥了硬件设计的高速性和 Nios II 软核处理器控制的灵活性.

1　立体图像差异性算法描述

1.1　立体图像差异性算法基本描述

立体图像差异性算法的具体方法为:输入为两张图像,一张为需要到投影方向上确定深度的参考图像;另一张为需要测量的未知物体表面的场景图像. 算法的输出为差异性图像,指示了从场景图像到参考图像的每个像素的位置移动,具体如图 1 所示.

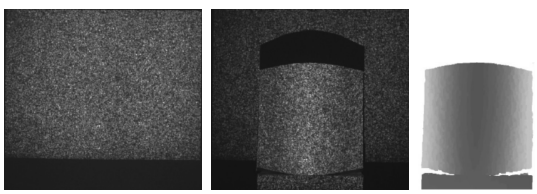

图 1　差异性图像
Fig. 1　Disparity image

1.2　立体图像差异性算法设计

基于散斑匹配,设计立体图像差异性算法步骤如下.

第一步:假设参考图像和场景图像有着相同的大小,均为 $W \times H$. 设有部分重叠的大小为 $m \times H$的来自场景图像的子图像和对应的大小为 $(m+2d) \times H$ 来自参考图像的子图像,具体如图 2 所示.

第二步:从场景子图像中选入大小为 $m \times m$ 部分重叠的块,在对应的参考子图像中找到与其最接近的块,如图 3 所示.

块匹配[8]过程详细描述如下.

1) 初始化大小为 $m \times H$ 的三张图像:分别为水平差异性图、垂直差异性图和相关性图,并将里面所有的元素均设为 0.

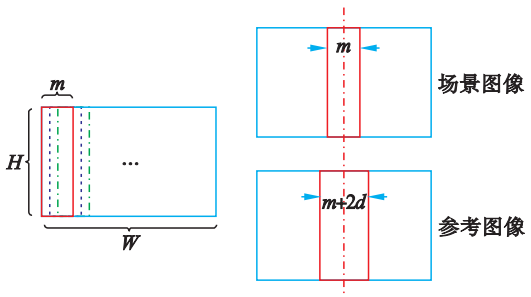

图 2　子图像分割示意图
Fig. 2　Sub-image division

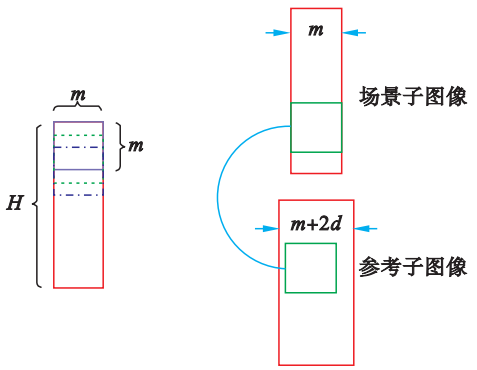

图 3　块匹配示意图
Fig. 3　Block matching

2) 阴影检测. 如果场景块的灰度平均值小于阈值 T_S,则认为这是阴影块并跳过块匹配.

3) 两个图像块之间的归一化互相关[9]的计算:图像相关性 NCC[10]计算公式如式(1)所示:

$$\mathrm{NCC} = \frac{\sum (I_r - \bar{I}_r)(I_s - \bar{I}_s)}{\sqrt{\left(\sum (I_r - \bar{I}_r)^2\right)\left(\sum (I_s - \bar{I}_s)^2\right)}}. \quad (1)$$

其中:I_r 和 I_s 代表参考子图像和场景子图像中各个像素点的灰度值;$\bar{I}_r$ 和 $\bar{I}_s$ 代表了两个像素块的平均灰度. 其中,NCC 的计算在整个算法计算过程中消耗的时间最多,因此本文设计了适合于硬件加速的立体图像差异性算法,从而提高算法的执行效率.

4) 在参考图像中搜索像素块:具体方法分为两种,分别是预测方式搜索和穷尽方式搜索.

穷尽方式搜索:对于场景子图像中的每个像素块,先搜索参考子图像中相同位置的像素块. 然后,一个像素一个像素的移动搜索窗口,具体方式如图 4 所示(先搜索左侧再搜索右侧,然后上移搜索窗口,继续搜索. 不需要搜索下方的像素块,因为在系统中,物体的表面在参考平面的前面). 在整个参考子图像搜索完成之前,记录每个位置 NCC 的值,如果当前 NCC 的值大于阈值 T_B,则跳

出搜索并保存 NCC 的值. 当搜索完成,找出最大的 NCC 的值,并保存更新后的 NCC 值.

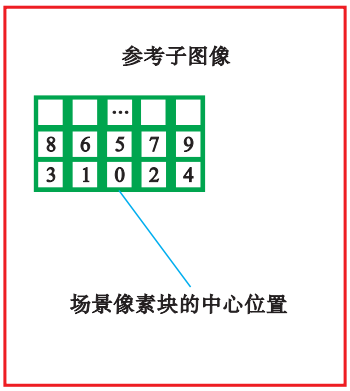

图 4 穷尽方式搜索
Fig. 4 Brute-force mode searching

预测方式搜索:穷尽搜索方式十分耗时,为了加速算法执行,可以采用预测方式搜索. 原因是附近的像素块因为表面光洁会有很大的相似性. 所以,对于每个场景子图像的像素块,可以先检查上一个像素块所指示位置的附近大小为 $2d \times 2d$ 的区域,如图 5 所示. 如果上述区域内最大的 NCC 值大于阈值 T_P,保存这个 NCC 值,否则回到穷尽方式搜索.

5)对重叠像素点的差异性和相关性的更新:在计算得到当前像素块中所有像素的 NCC 值和相关差异性后,如果当前的 NCC 值增加,则与上个块重叠的像素点的 NCC 值和差异性值将被更新.

第三步:当得到所有场景子图像的差异性和相关性图像后,重叠像素点的差异性取决于它们中最大的 NCC 值.

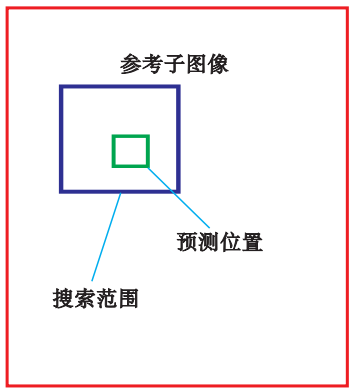

图 5 预测方式搜索
Fig. 5 Prediction mode searching

在本系统中,参数的取值如下:

$m = 16, d = 1.$

$T_B = 0.85, T_P = 0.80, T_S = 30.$

相邻两个子图像和像素块之间重叠的像素为 14,即每次移动 2 个像素点.

由于 NCC 的计算会占用系统大部分时间,所以需要尽可能地优化 NCC 的计算方法. 下面对 NCC 的计算公式进行部分优化,使之计算量减少,并且更适合 FPGA 的实现.

由式(1)可知,NCC 的计算需要先求取参考像素块和场景像素块灰度的平均值 $\bar{I}_r$ 和 $\bar{I}_s$. 但在 FPGA 中,如果先求像素块的灰度平均值,再求取 NCC 的值,这对于并行处理的 FPGA 系统来说影响了其并行性,大大降低了 FPGA 的效率.

对式(1)的求和符号展开后得到式(2):

$$\mathrm{NCC} = \frac{\sum I_r I_s - \sum \bar{I}_s I_r - \sum \bar{I}_r I_s + \sum \bar{I}_r \bar{I}_s}{\sqrt{\left(\sum (I_r^2) - 2\sum \bar{I}_r I_r + \sum (\bar{I}_r^2)\right)\left(\sum (I_s^2) - 2\sum \bar{I}_s I_s + \sum (\bar{I}_s^2)\right)}}. \tag{2}$$

将常数从求和符号提出可得式(3):

$$\mathrm{NCC} = \frac{\sum I_r I_s - \bar{I}_s \sum I_r - \bar{I}_r \sum I_s + \sum \bar{I}_r \bar{I}_s}{\sqrt{\left(\sum (I_r^2) - 2\bar{I}_r \sum I_r + \sum (\bar{I}_r^2)\right)\left(\sum (I_s^2) - 2\bar{I}_s \sum I_s + \sum (\bar{I}_s^2)\right)}}. \tag{3}$$

将 $\bar{I}_s \sum I_r = \frac{\sum I_s \sum I_r}{M^2} = \bar{I}_r \sum I_s = \sum \bar{I}_r \bar{I}_s, \bar{I}_r \sum I_r = \frac{\left(\sum I_r\right)^2}{M^2} = \sum (\bar{I}_r^2)$ 和 $\bar{I}_s \sum I_s = \frac{\left(\sum I_s\right)^2}{M^2} = \sum (\bar{I}_s^2)$ 代入式(3)中得到式(4):

$$\mathrm{NCC} = \frac{\sum I_r I_s - \frac{\sum I_r \sum I_s}{M^2}}{\sqrt{\left(\sum (I_r)^2 - \frac{\left(\sum I_r\right)^2}{M^2}\right)\left(\sum (I_s)^2 - \frac{\left(\sum I_s\right)^2}{M^2}\right)}}. \tag{4}$$

此时，耗时较长的计算为：参考块像素灰度值的累加和 $\sum I_r$，场景块像素灰度值累加和 $\sum I_s$，参考块像素灰度值平方的累加和 $\sum (I_r)^2$，场景块像素灰度值平方的累加和 $\sum (I_s)^2$ 和参考块与场景块对应像素灰度值相乘的累加和 $\sum I_r I_s$. 这些值都可以在 FPGA 内进行并行计算，大大加快了算法的计算速度.

2　基于 FPGA 的硬件加速设计

2.1　硬件系统总体结构

根据系统的功能分析，将系统进行模块化划分，系统的结构示意图如图 6 所示. 该系统分为以下几个模块：SD 卡控制器模块、图像差异性算法 IP 核模块、LTM 控制器模块和 Nios II 嵌入式系统模块.

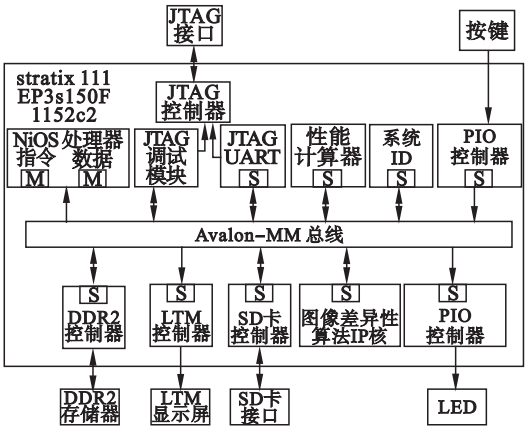

图 6　系统硬件结构示意图

Fig. 6　The schematic diagram of system structure

Nios II 软核处理器模块主要完成总体的核心控制以及各个模块的协调作用. 首先，Nios II 软核处理器控制 SD 卡控制器从 SD 卡中读取图片数据并存入 DDR2 存储器中，之后将图片数据按一定顺序写入图像差异性算法 IP 核中进行差异性处理，等待处理完成后，再将处理结果从算法 IP 核中读取出来，最后将读取的结果数据写入 LTM 控制器中显示，再开始下一张图片的处理. 具体结构如图 7 所示.

2.2　图像差异性算法模块硬件设计

图像差异性算法 IP 核的设计与实现是整个 FPGA 系统的关键部分，算法 IP 核通过 Avalon 从接口连接到系统总线. 算法 IP 核每次输入为若干个参考子图像和对应的场景子图像，每次输出为对应的差异性子图像.

图像差异性算法 IP 核在子图像级采用并行计算，整个算法 IP 核由多个子图像处理核组成，每个子图像处理核可以处理一对参考和场景子图像，并输出对应的差异性图像. 由于 FPGA 内部资源限制，本系统中算法 IP 核由 18 个子图像处理核组成，每个子图像处理核还配有一个对应的 FIFO，用于缓存结果数据. 图像差异性算法 IP 核内部结构如图 7 所示.

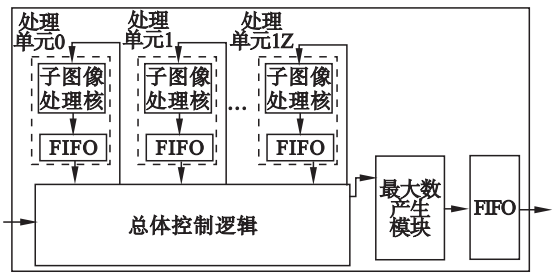

图 7　图像差异性算法 IP 核内部结构图

Fig. 7　The structure diagram of image disparity algorithm IP core

多个参考子图像和场景子图像数据进入算法 IP 核后，由总体控制模块将各个参考子图像和场景子图像按序对应分配给子图像处理核，使每个子图像处理核得到一个参考子图像和对应的场景子图像. 然后，多个子图像处理核并行处理，当处理好每对子图像后，将结果输出到对应的 FIFO 中暂存. 之后，由总体控制模块按序读取各个 FIFO，将数据输出到最大数产生模块. 最后，最大数产生模块生成每个像素块中最大的结果，将结果暂存到结果存储 FIFO 中，当 Nios II 需要读取数据时，读取结果存储 FIFO，将结果读到 Nios II 中.

1）总体控制逻辑模块：算法 IP 核的总体控制模块的主要功能是控制数据的分配和输出，由一个 5 个状态的有限状态机实现. 其状态转移图如图 8 所示.

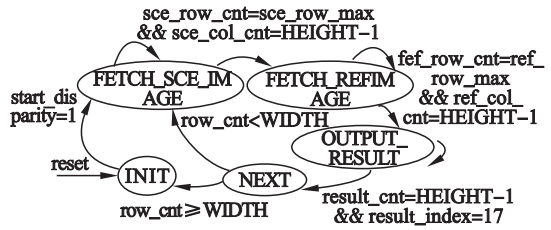

图 8　总体控制模块状态转移图

Fig. 8　The state transition of overall control module

2）子图像处理核：能处理一张场景子图像和对应的参考子图像，并输出各个像素点的 NCC 值和差异性值. 由于像素块每次移动的像素为 2，所以相邻的 4 个像素点会同时处理，本模块将这 4

个像素点作为处理单元,这样可以增大处理带宽,提高系统的并行性. 数据的移动和计算的最小单位是一个处理单元. 子图像处理核进一步可以划分为子图像存储器、控制逻辑和 NCC 计算模块等三个部分. 其中,NCC 计算模块接收来自控制模块的参考像素块数据和场景图像块数据,并计算两个像素块的 NCC 值. NCC 计算模块每次的输入为 8 个参考子图像的像素点和 8 个场景子图像的像素点. 根据算法需求,NCC 计算模块需要并行进行以下操作:将参考数据累加、将场景数据累加、将参考数据求平方再累加、将场景数据求平方再累加、将参考数据乘场景数据再累加. 在一次搜索图像块中,场景像素块并不移动,而相邻的参考像素块中有很大一部分像素点是重叠的,考虑将每 3 个像素块作为一个组合,每次比较一个组合,产生 3 个 NCC 值.

3 系统测试及结果分析

基于 FPGA 的立体图像差异性算法采用 System Verilog 语言,在 Quartus II 平台上实现并仿真测试. PC 机平台为 Intel Core 2 T6500 CPU,主频为 2.1GHz,双核,内存为 8GB. 系统性能测试使用的硬件平台为 Intel 公司的 Stratix III 系列的 FPGA 开发板 DE3 - 115,FPGA 的芯片型号为 EP3S150F1152C2N. 测试文件为 1 张参考图片和 500 张场景图片,大小为 600 × 450. 系统实物图如图 9 所示.

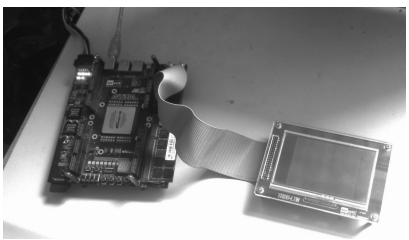

图 9 图像差异性处理系统实物图

Fig. 9 The photo of the image disparity processing system

3.1 1 张参考图像和 1 张场景图像的差异性比较

对 1 张参考图像和 1 张场景图像的差异性进行测试,测试选用的场景图像和参考图像如图 10 所示.

分别在 PC 机上和 FPGA 上运行图像差异性算法程序,得到的差异性图像如图 11 所示.

通过单张场景图像和参考图像差异性算法在 PC 和 FPGA 上的测试,得出立体图像差异性算法在 PC 和 FPGA 上运行的结果基本相同,运行所需的时间及其比较如表 1 所示.

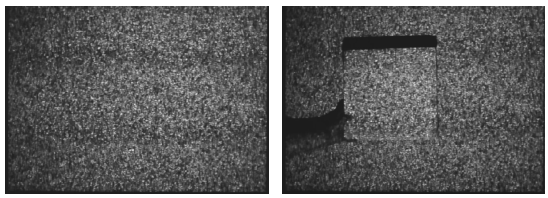

图 10 参考图像(左)和场景图像(右)

Fig. 10 Reference image (left) and scene image(right)

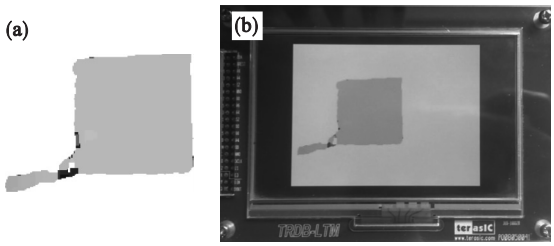

图 11 图像差异性算法运行结果

Fig. 11 The result of image disparity algorithm

(a)—PC 上运行结果; (b)—FPGA 上运行结果.

表 1 图像差异性算法运行时间对比

Table 1 The comparison of calculating time of image disparity algorithm s

测试	PC 机	FPGA	运行时间比
第 1 次测试	9.680 0	0.036 8	262.7
第 2 次测试	10.380 0	0.039 3	264.3

由表 1 可知,由于 FPGA 的并行性和高速性,立体图像差异性算法在 FPGA 上运行的时间远小于 PC 上软件运行的时间,其运行速度大概是 PC 机的两百多倍.

3.2 1 张参考图像和 500 张场景图像的差异性比较

本测试采用 1 张参考图像和 500 张场景图像进行差异性比较,将差异性算法比较的结果连续输出形成视频,测试系统的稳定性和实时性. 参考图像如图 10 中左图所示,分别测试第 100,200,300,400,500 帧的原始图像在 PC 机上和 FPGA 上的处理结果,这里只给出第 400 和第 500 帧图像在 FPGA 上的处理结果如图 12 所示. 在 FPGA 上使用立体图像差异性算法处理 500 张场景图像需要的总时间为 14.897 s,处理每张图像所需的平均时间为 0.029 s,每秒可处理 33 张以上的图像,达到了实时处理和显示的要求.

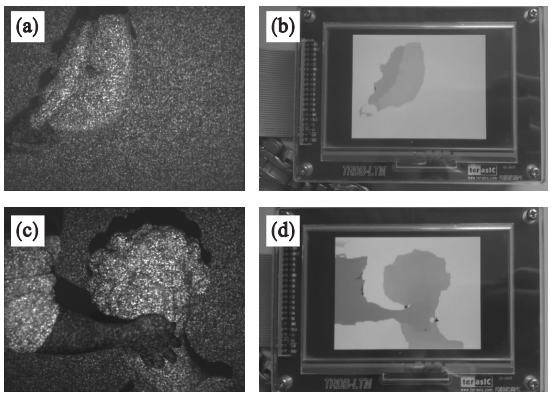

图 12　500 张场景图像的运行结果

Fig. 12　The result of 500 scene images

(a)—第 400 帧原始图像；(b)—第 400 帧 FPGA 上运行结果；(c)—第 500 帧原始图像；(d)—第 500 帧 FPGA 上运行结果.

4　结　　论

本文提出了一种基于 FPGA 的立体图像差异性算法，并通过设计硬件算法的 IP 核，充分利用了 FPGA 独特的并行处理机制和强大的运算能力，极大提高了系统的处理速度和性能. 利用 FPGA 中的 Nios II 软核处理器对数据进行控制，该过程在 FPGA 片内总线上完成，解决了数据传输的瓶颈，充分发挥了硬件设计的高速性和 Nios II 软核处理器控制的灵活性. 系统测试结果表明，立体图像差异性算法在 PC 机和在 FPGA 上的运行结果基本相同，但 FPGA 的处理速度是 PC 机上处理速度的两百多倍. FPGA 可以流畅处理 500 张场景图像且 FPGA 每秒可处理 33 张以上的图像，说明该系统具有稳定性和实时性. 立体图像差异性算法是计算机视觉和图像理解领域的基础研究问题，因此本文提出的基于 FPGA 的立体图像差异性算法处理系统具有较好的应用前景.

参考文献：

[1] 汤锐彬，陈芬，彭宗举. 基于三维感知的立体虚拟视点图像质量评价方法[J]. 光电子·激光，2018，29(8)：893－902.
(Tang Rui-bin，Chen Fen，Peng Zong-ju. Stereo virtual view quality assessment based on three-dimensional perception [J]. *Journal of Optoelectronics·Laser*，2018，29(8)：893－902.)

[2] Jawed K，Morris J，Khan T，et al. Real time rectification for stereo correspondence [C]//International Conference on Computational Science and Engineering. Vancouver，2009：277－284.

[3] Du Q，Shi X，Dai B，et al. Binocular stereo vision system for humanoid robot [J]. *International Journal of Computer Applications in Technology*，2013，46(4)：316－322.

[4] Affendi H R，Haidi I. Literature survey on stereo vision disparity map algorithm[J]. *Journal of Sensors*，2016，2016：1－23.

[5] 张翰，包国琦，刘凯. 一种结构光三维成像系统的简易标定方法[J]. 激光与光电子学进展，2019(14)：148－156.
(Zhang Han，Bao Guo-qi，Liu Kai. A simple calibration method for 3d imaging system of structured light[J]. *Laser & Optoelectronics Progress*，2019 (14)：148－156.)

[6] Yang Z，Xiong Z，Zhang Y，et al. Depth acquisition from density modulated binary patterns[C]//Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Washington DC，2013：25－32.

[7] Zhang S，Royer D，Yau S T. GPU-assisted high-resolution，real-time 3-D shape measurement[J]. *Optics Express*，2006，14(20)：9120－9129.

[8] Nahhas I. Fast computational four-neighborhood search algorithm for block matching motion estimation [J]. *International Journal of Engineering Research and Applications*，2015，5(1)：20－24.

[9] Sun Y，He X，Song H，et al. A block-matching image registration algorithm for video super-resolution reconstruction[J]. *Acta Automatica Sinica*，2011，37(1)：37－43.

[10] Li Q，Li F，Shi G M，et al. One-shot depth acquisition with a random binary pattern[J]. *Applied Optics*，2014，53(30)：7095－7102.