

改进的尺度不变特征变换算法并行加速双目测距系统及其实现

张志强^{1*}, 施文华²

¹郑州商学院通识教育中心, 河南 巩义 451200;

²Department of Electrical and Computer Engineering, Technical University of Munich, Munich 80809, Germany

摘要 在数字图像处理领域中,尺度不变特征变换(SIFT)算法是特征点识别的代表性算法。以开放运算语言(OpenCL)并行计算为加速手段,建立了基于改进的SIFT算法的双目测距系统,深入研究了如何加快SIFT算法的运算速度。在加快SIFT算法方面,选取了积分均值模糊,并利用OpenCL对其进行并行加速,对算法进行并行优化后,使之能够在NVIDIA GPU硬件平台上进行实现。在获取精确视差方面,对原SIFT匹配方法进行了改进,极大地提高了匹配效率。此外,构建了双目测距异构计算实验平台,并进行实验。实验平台对采集的图像进行了实时处理,验证了基于SIFT算法的并行加速计算的可行性,同时可以直观地看到中间计算过程和测距的结果。实验结果表明,本文方法与已有的加速优化工作相比,计算时间消耗比原方案要少得多。

关键词 机器视觉; 双目测距; 尺度不变特征变换算法; 开放运算语言; 并行加速

中图分类号 TP 391.41

文献标识码 A

doi: 10.3788/LOP56.141502

Research and Implementation of Binocular Distance Measurement System Based on Improved Scale-Invariant Feature Transform Algorithm with Parallel Acceleration

Zhang Zhiqiang^{1*}, Shi Wenhua²

¹General Education Center, Zhengzhou Business University, Gongyi, Henan 451200, China;

²Department of Electrical and Computer Engineering, Technical University of Munich, Munich 80809, Germany

Abstract The scale-invariant feature transform (SIFT) algorithm is the representative approach for key-point detection in the field of digital image processing. A binocular distance measurement system is established herein based on the improved SIFT algorithm by using open computing language (OpenCL) parallel computing as an acceleration method, and a profound study on how to speed up the operation of SIFT algorithm is performed. First, the integral mean blur is selected to speed up the SIFT algorithm operation. OpenCL parallel computing is then used to accelerate it. The parallel optimization of the algorithm is made to be implemented on NVIDIA GPU hardware platforms. The original SIFT matching method is improved to obtain an accurate parallax. Consequently, the matching efficiency has been greatly improved. Finally, a binocular distance measurement system heterogeneous computing experimental platform is constructed. The experimental platform performs a real-time processing on the acquired images. The feasibility of parallel acceleration based on the SIFT algorithm is verified. An intermediate calculation process and the distance measurement results can be directly obtained in the system. The experimental results show that compared with the previous accelerated optimization work to the SIFT, the computational time consumption of the proposed approach is much less than that in the original method.

Key words machine vision; binocular distance measurement; scale-invariant feature transform algorithm; open computing language; parallel acceleration

OCIS codes 150.5670; 330.1400; 330.4060; 100.2000

收稿日期: 2019-01-25; 修回日期: 2019-01-31; 录用日期: 2019-02-21

基金项目: 河南省高等学校重点科研项目(19A140021)

* E-mail: zhangzhiqiang08@gmail.com

1 引言

随着数字图像处理技术和计算机技术的快速发展,单纯的离线处理图像已经无法满足人们的需求,如何让计算机实时地处理图像信息显得尤为重要。双目测距就是利用计算机对图像信息进行实时处理的一个重要应用。对于图像信息实时处理而言,一方面,如何在可以接受的匹配精确度的情况下,大幅缩减计算开销;另一方面,如何将不同种类的计算任务分配到对应承载不同计算任务的硬件平台上。这两点一直是国内外研究的热点。例如,车辆辅助驾驶是当下的热门,从前沿的自动驾驶,到已经实用化的自动泊车,无不与物体识别、特征点检测有关。尺度不变特征变换(SIFT)算法^[1]是特征点识别的代表性算法。相较于其他算法而言,它具有抗尺度变换、抗旋转、抗光照变化等诸多优点,但是其程序复杂,运行所需要的计算开销较大^[2-3]。同时,由于其算法中串并行计算过程互相夹杂,难以剥离,针对该算法的加速并行优化受到了限制^[4-5]。

图形处理器(GPU)备虽然每一个计算单元的速率很低,功能有限,但因其成百上千的计算单元可以独立运算,每个计算单元并不依赖于其他单元的状态,从而能够以相对于中央处理器(CPU)快数倍甚至数百倍的速度完成计算^[6]。开放运算语言(OpenCL)标准是异构系统通用目的并行编程的开放式、免费标准^[7],适用于SIFT算法的并行加速优化^[8-9]。在SIFT算法中,计算过程消耗的时间主要受限于卷积运算,也就是高斯模糊。虽然可以通过使用替代性的分离高斯卷积来降低其运算量,但是其计算复杂度仍然很高。配合以合适的硬件平台,OpenCL能够极快地加速SIFT运算^[10-12]。

在车辆辅助驾驶这一应用场景下的双目测距^[13-15],需要对SIFT计算结果中的关键点进行聚类^[16],形成社区,将不同的物体与背景环境剥离开来。使用OpenCL加速SIFT运算并提供一种切实可行的聚类方法剥离出不同的目标,从而对目标进行实时双目测距,是一项十分有意义的工作。

本文结合OpenCL并行计算在图像处理上的应用,对SIFT算法进行了改进,提出了双目测距应用下新的网络探测方法,构建了双目测距OpenCL加速平台,并进行了实验研究。与以往的加速方案相比,算法时间消耗要少得多,速度上要高数倍。

2 双目测距原理

测距系统基于双目视差的原理,用2个相距为 D 的摄像机接收物体(位于 O 点)经过两条光路的光照信息后的成像,物体在2幅画面上存在横向像素位置偏差 x ,利用这个位置偏差和经过建模后得到的 x 与物距 L 之间的关系就可以解算出物体相对于摄像机的位置,如图1所示。

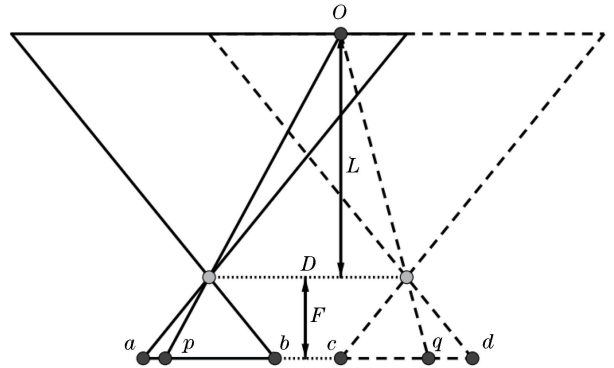


图1 双目测距原理图。实线表示左侧摄像机光路图,虚线表示右侧摄像机光路图

Fig. 1 Principle of binocular distance measurement. Solid lines represent light path of left camera and dotted lines represent light path of right camera

图中的 \overline{ab} 和 \overline{cd} 分别表示左右2个摄像机成像CCD的横向尺寸, p 和 q 表示物体在2个摄像机所成像的画面上的位置。由图可知,物体在2幅画面中的像素位置偏差为

$$x = l_{cq} - l_{ap}, \quad (1)$$

式中: l_{ap} 为物体在左侧摄像机画面中的像素位置到 a 点的距离, l_{cq} 为物体在右侧摄像机画面中的像素位置到 c 点的距离。

根据三角形相似原理,有

$$\frac{D}{l_{pq} - D} = \frac{L}{F}, \quad (2)$$

式中: l_{pq} 为物体在左右两侧摄像机画面中的像素位置之间的距离, F 为摄像机的焦距。由于 $l_{pq} = D + x$,所以有

$$L = \frac{FD}{x}. \quad (3)$$

直接测量 x 的值是比较困难的,但是通过读出摄像机捕获画面上物体位置偏移的百分比再间接测量其偏移是可行的。令 $x' = x/l_{AB}$,其中 l_{AB} 为摄像机成像CCD的横向长度,将(3)式化成一般的形式,得

$$L = \frac{k}{x'}, \quad (4)$$

式中: k 为待定参数。

对于(4)式来说,要确定其参数,有2种途径:第1种途径是测量摄像机的焦距 F 和2个摄像机的水平位置距离 D ;第2种途径是根据已有的系统,标定标准距离 L_0 时 x' 的值,设为 x'_0 ,将 (x'_0, L_0) 代入(4)式,解算出对应的 k 。从实用的角度来说,首先用第1种方法来根据应用场合确定合适的摄像机水平位置距离 D 和摄像机焦距 F ,当整个系统搭建完成后,再用第2种方法对其参数 k 进行校正。

众所周知,摄像机采集的分辨率是有限的,其理论最小单位是1 pixel,但由于其具体的实际尺寸还依赖于等效的 CCD 横向尺寸,不妨设最小值为 Δx_{\min} ,则为了能够识别宽度为 ΔL 的物体,系统的最大工作距离为

$$L_{\max} = \sqrt{\frac{FD}{\Delta x_{\min}}} \Delta L. \quad (5)$$

3 SIFT 算法计算过程

SIFT 算法是数字图像处理领域探测和描述特征点的代表性算法,其主要用途是寻找图像的特征点,进而在此基础上寻求方法来量化描述其特征。此算法由 Lowe 提出并完善^[1],应用范围非常广泛,包括物体辨识、机器人地图感知与导航、影像缝合、三维(3D)模型建立、手势辨识、影像追踪和动作比对等^[3-4, 14, 17]。

SIFT 算法的主要步骤有:1) 构建尺度空间;2) 检测极值点,获取尺度不变性;3) 特征点过滤并进行精确定位;4) 计算每个特征点的方向值;5) 生成特征描述子。之所以选用 SIFT 算法作为特征点识别方式,是因为其对特征点精确定位的步骤适用于双目识别应用场合。只有对目标物体进行精确定位,其左右图像的视差才能够被精确地计算出来。

3.1 构建尺度空间

为了寻找存在于不同尺度空间上的关键点,需要构建一连串的尺度空间。前人工作已经证明,高斯卷积核是唯一实现尺度变换的核,因此需选用高斯模糊来构建尺度空间。

高斯模糊所使用的卷积模板来源于高斯函数(正态分布函数)。将原图像与高斯模糊模板作卷积运算,就可以实现图像的模糊操作:

$$L(X, Y, \sigma) = G(X, Y, \sigma) * I(X, Y), \quad (6)$$

式中: X 为横坐标; Y 为纵坐标; $I(X, Y)$ 表示输入图像的函数表达式; $*$ 表示卷积运算;高斯函数为

$$G(X, Y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{X^2 + Y^2}{2\sigma^2}\right), \quad (7)$$

其中 σ 是正态分布的标准差,它的值越大,表示图像的模糊程度越大。为了获取连续的尺度空间,使用线性增长的 σ 值来连续地模糊图像。

在确定了尺度空间每个点的坐标之后,开始高斯模糊的具体计算。这里选用积分均值模糊,在计算过程中,中间结果可重复使用。对于一个点来说,它与模板中任意一个值相乘所得的乘积是固定的,也就是说,这个乘积可以参与这个点的所有相邻点的计算,而不必重复乘法。此方法可以将计算复杂度降低两个数量级。

3.2 检测极值点

构建连续的尺度空间后,构建差分高斯空间(difference of Gaussian),从而检测空间中的极值点。选用差分高斯函数来稳定地探测尺度空间的极值点,主要有两个主要原因:一是容易计算;二是差分高斯函数提供了一个尺度归一化的高斯拉普拉斯函数,而尺度归一化的高斯拉普拉斯函数检测出的极大值和极小值,能够得到最为稳定的图像特征。

高斯差分函数定义为

$$\begin{aligned} D(X, Y, \sigma) &= [G(X, Y, \sigma_1) - G(X, Y, \sigma_2)] * I(X, Y) \\ &= L(X, Y, \sigma_1) - L(X, Y, \sigma_2), \end{aligned} \quad (8)$$

式中: σ_1 和 σ_2 分别为2幅相邻图像中使用的高斯卷积核的标准差。只要简单地将2个相邻尺度的图像作减法就可以得到差分高斯空间。

3.3 精确定位

经过以上步骤提取出来的关键点,坐落在不同的尺度内,且图像具有离散性,因此需要通过计算来拟合出图像最佳的关键点,并用得到的关键点来差值计算出其极值。

使用泰勒展开的前3项进行拟合。由于图像是离散的,因此并不知道具体的解析式,但是可以使用有限差分方法来替代微分方程,将连续变化的变量离散化。当计算得到的极值点位置偏差在任意 X 或者 Y 方向上大于0.5时,就代表着极值点已经落在相邻的点上,此时需要以相邻的点为中心,重新进行插值计算。一般而言,当该步骤重复进行5次以上或者到达图像边缘时,极值点并不在这个点附近,也就是说这个关键点可以被舍弃。

3.4 特征点过滤

上节得出的关键点数量巨大,如果仔细观察可以发现,很多不恰当的点也会出现在结果之中。这些不良关键点的来源主要有2大类。

第1类来源于图像采集过程中夹杂着噪声。这类噪声是容易去除的,因为这些噪声的幅值较图像本身而言是很小的。最直接的去噪办法是,选取一个恰当的阈值,将绝对值小于这个阈值的关键点全部剔除。

第2类来源于边缘响应。一个不良的关键点会有一个较大的主曲率,在垂直方向会有一个很小的曲率。主曲率可以通过计算一个关键点所在位置的 2×2 二阶偏导数 Hessian 矩阵得到。由于更值得关心这个 Hessian 矩阵中2个特征值的比值,因而可以通过矩阵的迹和行列式的值来计算,而不必计算出特征值确切的数量,从而避免大量无意义的中间计算。假设这个 Hessian 矩阵 \mathbf{H} 一大一小2个特征值分别是 α 和 β ,那么就有

$$\text{Tr}(\mathbf{H}) = \frac{\partial^2 D}{\partial X^2} + \frac{\partial^2 D}{\partial Y^2} = \alpha + \beta, \quad (9)$$

$$\text{Det}(\mathbf{H}) = \frac{\partial^2 D}{\partial X^2} \frac{\partial^2 D}{\partial Y^2} - \frac{\partial^2 D}{\partial Y \partial X} \frac{\partial^2 D}{\partial X \partial Y} = \alpha\beta, \quad (10)$$

式中: $\text{Tr}(\cdot)$ 为求矩阵的迹; $\text{Det}(\cdot)$ 为求矩阵的行列式的值。

当 Hessian 矩阵的行列式值为负值时,就代表2个曲率有着不同的符号,也就是说对应的点并不是极值点,应当被剔除。假设 $\alpha = r\beta$,又根据

$$\frac{[\text{Tr}(\mathbf{H})]^2}{\text{Det}(\mathbf{H})} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r\beta + \beta)^2}{r\beta^2} = \frac{(r + 1)^2}{r}, \quad (11)$$

$r > 0$ 可知,在 $r = 1$ 时取到最小值,在 $r = 1$ 左侧单调递减,在 $r = 1$ 右侧单调递增。这样就可以发现,如果需要限定 $r < r_0$,则只需要令(11)式满足

$$\frac{[\text{Tr}(\mathbf{H})]^2}{\text{Det}(\mathbf{H})} < \frac{(r_0 + 1)^2}{r_0}. \quad (12)$$

一般而言,取 $r_0 = 10$ 就可以剔除图像在某一尺度下绝大部分的边缘响应点。

3.5 特征点方向

SIFT 算法提供了旋转不变性的特征,其建立在对一个关键点附近邻域内每个点的梯度幅值和方向的统计之上。首先定义一个点的幅值及方向。SIFT 算法达到旋转不变性的途径就是统计关键点附近邻域内每个点的梯度幅值和方向,以总计幅值最大的方向作为这个关键点的主方向,如果这个点有多个方向的幅值达到了主方向幅值的80%,那么就复制多份这个关键点,然后依次分配这些幅值较大的方向。

获得关键点的主方向之后,将该点及其一定范

围内的邻域,按照其主方向角度旋转至一个固定的方向。由于所有的特征点都是在这个方向上建立的描述子,且一个关键点在不同的采样图像中根据其邻域内离散点幅值和方向统计结果确定的其主方向是稳定的,因而通过该途径就能够实现旋转不变性。在双目测距这样的应用场景中,因为2个摄像机的位置是固定的,所以可以假设所有关键点在左右2幅图像中都没有发生旋转。这就代表着:SIFT 算法中为特征点构建主方向并作相应旋转的步骤可以忽略。

3.6 特征描述子

SIFT 算法的最后一个步骤是为筛选出的关键点构建描述子。描述子是一个用来描述关键点以及在其周围邻域一段距离内各个点梯度幅值和方向分布的向量。通常的方法是在关键点附近划分 4×4 个区域,每个区域又含有 4×4 的采样点,这样生成的描述子更加稳定可靠。

统计在这其中的 4×4 个采样点的梯度幅值和方向状况。为了量化这一概念,可将梯度方向固定划分为8个区间,每一个方向上的幅值由该区域内采样点的梯度幅值和方向情况统计而来。

但是,并不能直接把采样点的幅值和方向用作统计的样本,而是要做一定的处理,选择将其乘以高斯函数的结果作为统计的依据。这是因为,距离关键点越远的点,对其描述子的贡献越小。此外,还需要对采样点的幅值进行限制,其绝对值应当被限定在某一定值之内,例如取0.2,以剔除非线性光照的影响。处理之后,将这16个区域中每一个区域的8个方向的幅值按照顺序排列,得到最终的128维的向量 $\mathbf{V} = [v_0, v_1, \dots, v_{127}]$ 。最后,为了消除光照的影响,对其作归一化处理,从而完成一个旋转不变性变换的 SIFT 关键点描述子。

4 OpenCL 并行计算加速方法

OpenCL 是第1个面向异构系统的并行编程的开放式编程环境。OpenCL 提供了基于任务分割和数据分割的并行计算机制,便于开发人员为高性能计算服务器、手持设备编写高效轻便的代码,而且广泛适用于 CPU、GPU、Cell 类型架构,以及数字信号处理器(DSP)等并行处理器。

在 SIFT 算法中,计算过程消耗的时间主要受限于卷积运算,也就是高斯模糊,其计算复杂度很高。卷积计算的特点就是大量的计算过程简单重复且可以并行执行,彼此之间互不影响。基于卷积运

算的特点,将其作并行化处理,提高计算速度。

对一幅宽度、高度分别为 W 和 H 的图像,使用脚本语言来作验证和误差计算,高斯模糊模板的尺寸为 $n \times n$ 。原始高斯模糊、均值模糊、分离均值模

表 1 四种高斯模糊计算方法的消耗时间、复杂度与误差对比

Table 1 Comparison of time consumption, complexity, and error for four Gaussian blur calculation methods

Method	Time consumption/ms		Complexity	Error /%
	$n = 11$	$n = 21$		
Original Gaussian blur	2452	8990	$O(n \times n \times W \times H)$	0
Mean blur	589	2437	$O(n \times n \times W \times H)$	0.04
Separation mean blur	230	435	$O(2 \times n \times W \times H)$	0.04
Integral mean blur	32	34	$O(W \times H)$	0.04

从表 1 可以看出,积分均值模糊是一种在误差微乎其微的情况下替代原始高斯模糊并且将计算复杂度降低了 2 个数量级的方法。

此外,还需要关心的问题是异构计算时如何分配计算任务。在构建差分高斯尺度空间时,大量的计算过程应当发生在 OpenCL 一侧,因为卷积计算的特点就是大量的计算过程简单重复且可以并行执行。当构建完尺度空间之后,对关键点的精确定位、过滤,以及生成关键点描述子的过程也应当发生在 OpenCL 一侧,而不是取回计算的中间结果,让 CPU 接着做后续计算。只有这样,才能够最小化在不同计算空间图像数据传输的时间或者中间计算结果消耗的时间。因此,整个 SIFT 算法的计算过程都应当发生在 OpenCL 侧。

5 特征点的聚类

5.1 关键点之间的相似度

构建特征描述子之后,首先需要的就是定义 2 个关键点之间的相似度,并进行匹配。只有建立了相似度的概念,才能够判断 2 个关键点是否可以成为一对匹配点。同时,也要定义 2 个匹配点之间的相似度,这样才能够构建无标度网络进行社区发现,找到不同的物体。

匹配与匹配之间的相似度有多种假设。第一种常见的相似度定义为闵可夫斯基距离 $M_p(\mathbf{U}, \mathbf{V})$ 的倒数^[18-21],即

$$M_p^{-1}(\mathbf{U}, \mathbf{V}) = \frac{1}{M_p(\mathbf{U}, \mathbf{V})} = \left(\sum_{i=1}^m |u_i - v_i|^p \right)^{-\frac{1}{p}}, \quad (13)$$

式中: \mathbf{U}, \mathbf{V} 为两个描述子; m 为描述子维度; u_i 为描述子 \mathbf{U} 中的元素; v_i 为描述子 \mathbf{V} 中的元素; p 为范数。当 $p = 1$ 的时候,这个度量就是曼哈顿距离 D_M ^[19]。一个关键点描述子的维度是 128 维,如果

糊和积分均值模糊的结果如表 1 所示。算法时间消耗部分,采用的是图像尺寸均为 $800 \text{ pixel} \times 200 \text{ pixel}$ 的同一幅图像,表中 $O(\cdot)$ 表示算法复杂度。

使用 $p = 0.5$ 或者傅氏距离、谷本距离^[20],都将牵涉到成百上千次的乘法运算或者开方运算,这都是不能够被接受的;而曼哈顿距离^[21],也就是 $p = 1$ 的闵可夫斯基距离则不一样,它只涉及 128 次减法运算、求绝对值的操作和加法运算,这样的运算规模是可以接受的,更何况其本身就是一个良好的相似度定义。

5.2 聚类方法

为了将得到的网络进行划分,并聚类成不同的社区,需要运用社区发现的相关方法。标签传播算法(LPA)^[22]的划分虽然存在一定的误差,但是也有其无法被替代的优点。1) 对于存在多个社区的网络而言,LPA 并不依赖于已知的社区数量,不仅如此,它还能够自主发现网络中存在的社区;2) 它的执行时间几乎是线性复杂度的,因而对于一个较大的网络或者一个较大的数据库而言,使用 LPA 是较好的选择。

由复杂网络的理论可知,对于 N 个顶点、 M 条边的网络来说,快速贪婪算法的复杂度为 $O(N \lg N)$,极值优化算法的复杂度为 $O(N^2 \lg N)$ 。LPA 标签传播算法虽不像前两者能够提供比较精确的划分,但是因其时间复杂度 $O(N + M)$ 几乎是线性的,故也缩短了计算消耗的时间。

5.3 关键点网络的构建

由于本文方法需要作两次匹配,所以在第一步中对关键点的匹配可以容纳更多的噪声匹配。对于次匹配点相似度达主匹配点相似度 80% 的情况而言,可将这个匹配复制一份,将主匹配点替换成次匹配点,以备后用。

为了加速匹配的过程,需要考虑双目测距场景下关键点的特征。对于一个良好的双目测距系统而言,由镜头产生的畸变应该是较小的。这就意味着,同一个物体在左右 2 个摄像机捕获的画面中,其纵

坐标的偏差应该是不大的,即在一个较小的偏差范围内。同时,其在尺度范围内的坐标也应当是接近的。依照这 2 个特征,就能够通过限定每次匹配的范围来减小需要计算的量,从而将 2 个关键点之间匹配计算过程的复杂度由 $O(N_k^2)$ 降低到稍大于 $O(N_k)$,其中 N_k 为一侧画面关键点的数量。

此时需要做的,是通过寻求某种算法,划分出不同物体之间的界限。显而易见,通过拟合出其 Δx 的质心,能够较好地测算出物体与摄像机之间的距离。这种算法对正确划分的要求不太高,但不能容忍错误划分,即可以通过牺牲正确划分来减少错误划分。这是因为,拟合 Δx 质心的工作不需要太多的正确匹配,但一旦有错误匹配,其结果就会出现比较大的偏差。同时,这种比较算法相较其他算法而言,在计算时更改简便,而且不会占用太多的计算时间。

6 实验结果与分析

6.1 实验环境

为了验证基于 OpenCL 加速 SIFT 算法的效果,搭建双目测距平台,并通过此平台对现实场景进行图像采集和实时处理工作,观察算法的实用性。

该平台基于 ThinkPad T450 搭建。中央处理器(CPU)为 Intel i7-5500U 2.4G 双核四线程;图像处理器(GPU)为 NVIDIA 940M,核心频率 1100 MHz,单精度浮点性能 1 TFlops;内存为 12 GB DDR3 内存,工作频率 1600 MHz;摄像头型号是 UVC NT99141, 640 pixel \times 480 pixel, 30 frame/s。操作系统为 Ubuntu 16.04 LTS x64, GPU/OpenCL 驱动版本为 NVIDIA Driver 361.42, OpenCL 为 1.2 版本, Linux 内核为 4.4.0-22 版本, Node.js 版本为 V6.2.0。

摄像机组为标准的 VGA (video graphics array) 分辨率的 UVC Webcam, 采样制式为 YUYV, 帧率为 30 frame/s。程序通过 Linux 系统的 V4L2 接口标准控制并获取采集图像内容。

实验平台选用的 NVIDIA 940M 显卡理论支持的 PCI-E 总线接口带宽为 PCI-E 3.0 x8, 但受限于 CPU 的性能, 只能工作在 PCI-E 2.0 x4, 其单向带宽为 2 GB/s, 限制了系统的性能。

程序使用的 OpenCL API 版本为 1.2, 程序主体使用 Node.js 作为胶水语言, 其主要作用为获取 V4L2 缓冲区, 并递交 OpenCL 任务序列, 处理最终匹配结果, 并打包传送给前端 GUI 呈现出来。

6.2 程序计算流程

程序由 3 个运行主体构成,如图 2 所示:第 1 列为用户界面侧,由 HTML 实现;第 2 列为 Node.js 胶水层,负责图像的获取、OpenCL 执行队列的维护,以及最终结果的筛选;第 3 列为 OpenCL 执行层,每一个执行步骤都为 OpenCL 核心。程序中,关键步骤为积分均值模糊、关键点检测、精确定位与社区发现。

6.3 实验结果

本平台在室内测距的中间计算过程及最终计算结果如图 3~6 所示。

图 3 中 SIFT 特征点探测的结果相当令人满意,SIFT 的稳健性也由此可见一斑。左右 2 幅图像在生产约 6000 个关键点,并经过特征点匹配的情况下,平均耗时仅 300 ms,也就是说一侧图像平均需要约 150 ms 的计算时间。经过测算,当只进行关键点探测和描述子生成时,平均耗时约为 100 ms。

图 4 展示了关键点匹配的结果,可见生成的匹配非常之多。该步骤中影响匹配结果的关键因素是主匹配未与次匹配拉开距离。

图 5 和图 6 显示了最终的社区发现及对应的双目测距结果。可以发现,对于社区成员较多的情况,解算出的距离比较稳定,如图 5 中橙色和绿色的社区;但对于处于图像边缘的社区,以及社区成员数量较小的社区,如图 5 中下方蓝色的社区,测距解算结果不够稳定。本实验使用的摄像机组间距为 4 cm。由此可见,在较短间距的情况下,SIFT 算法也能提供比较稳定的关键点探测结果。在实际运用时,通过拉大摄像机间距,能够有效地提高双目测距结果的精度。在进行多次测量取平均值之后,将实验测得的 SIFT 关键点探测及描述子生成结果与已有的工作^[23]进行对比,结果如表 2 所示。

表 2 的结果表明,本文方法相比已有的工作的计算速度要高数倍。此外,在生成尺度空间时,通过进一步的优化,如避免起初的插值放大图像步骤等,还能够进一步减少约 1/3 的计算开销。

整个系统中测距的误差来源主要是 3 部分: 1) 图像采集环节,主要是图像的畸变,可以通过使用更好的摄像机硬件来减小测距误差; 2) SIFT 运算环节,在匹配过程中,并不是每个点都会匹配到对应图像中的另一个点,这也会带来一定的误差,通过合理的匹配筛选过程,可以有效减小其带来的误差; 3) 聚类环节,聚类质量直接影响了最终的测距

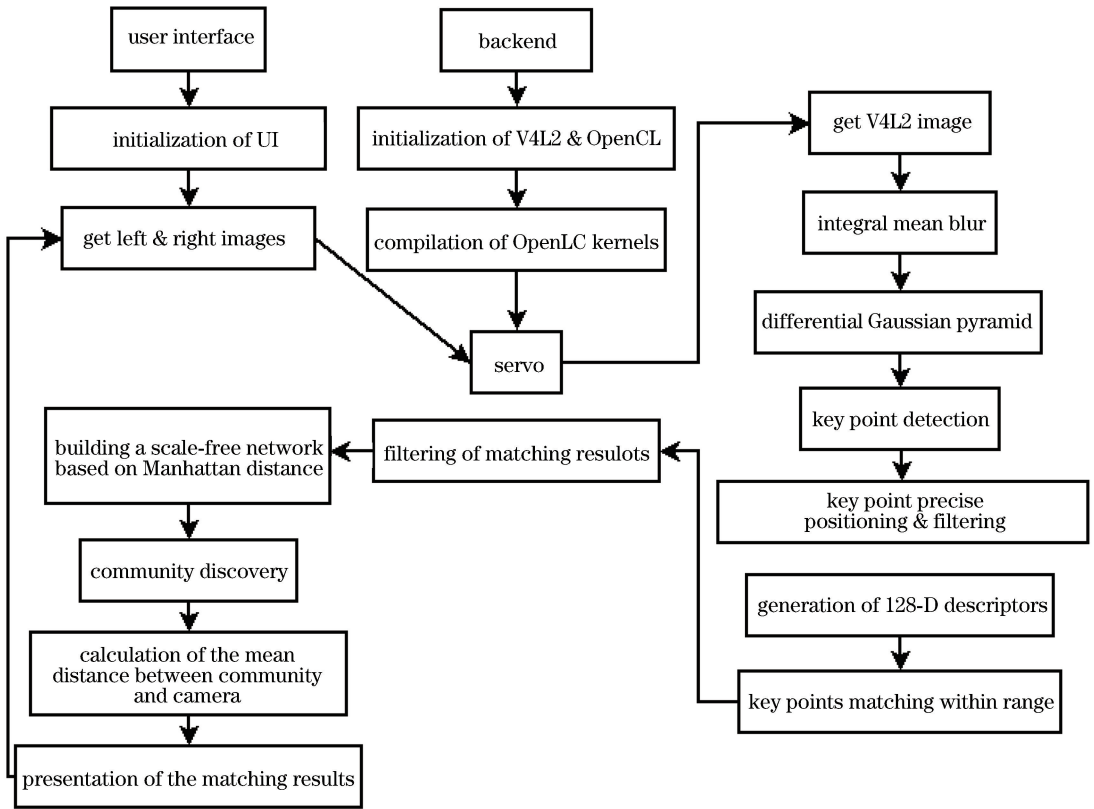


图 2 程序流程图

Fig. 2 Flow chart of program

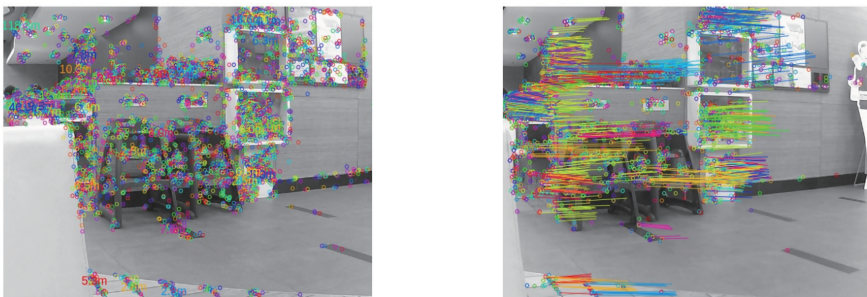


图 3 关键点探测结果

Fig. 3 Results of key-point detection

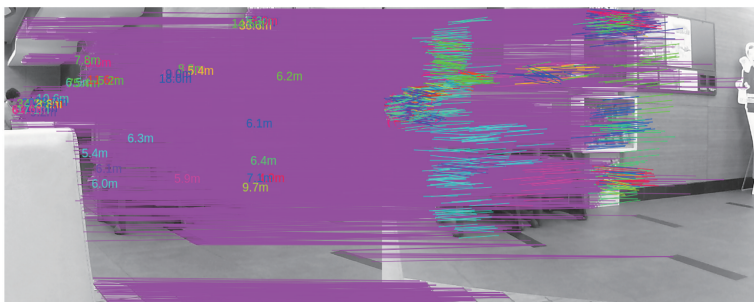


图 4 关键点匹配结果

Fig. 4 Results of key-point matching

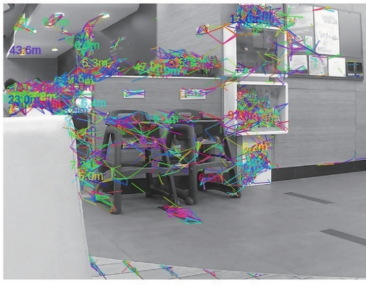


图 5 社区发现结果

Fig. 5 Results of community discovery



图 6 双目测距结果

Fig. 6 Results of binocular distance measurement

表 2 本文 SIFT 运算开销与已有工作的对比

Table 2 Comparison of time consumption between proposed SIFT algorithm and previous work

Item	Proposed algorithm	Previous work in Ref. [23]
Convolution core width	31×31	19×19
Image size / (pixel \times pixel)	640×480	640×480
GPU hardware	GT940M	GTX560
Theoretical floating-point computing power /TFlops	1	1
Time consumption /ms	100	634

结果,在实际应用中应当选取合适的聚类方法来平衡计算开销和测距精度。

7 结 论

选取积分均值模糊,利用 OpenCL 技术对其进行并行加速,应用由摄像机组捕获的左右 2 幅图像,以曼哈顿距离配对结果中的关键点,以关键点描述子之间的坐标差作为点的位置。通过设定阈值,根据曼哈顿距离作为相似度度量连接或断开两点,构建无标度网络。使用标签传播算法对无标度网络进行聚类,得到各个社区的平均视差,套用双目测距理论公式,计算出物体与摄像机的距离。搭建实验平台进行测试,在相同的理论浮点运算能力下,本文方法与已有的加速优化工作相比,时间消耗要少得多,平均耗时仅 100 ms。

参 考 文 献

- [1] Lowe D G. Distinctive image features from scale-invariant keypoints [J]. International Journal of Computer Vision, 2004, 60(2): 91-110.
- [2] Deng B S. An improved algorithm of the screen space ambient occlusion[J]. Modern Computer, 2017(9): 54-57.
邓伯胜. SIFT 算法 GPU 并行化研究[J]. 现代计算机, 2017(9): 54-57.
- [3] Fu W P, Qin C, Liu J, *et al.* Matching and location of image object based on SIFT algorithm[J]. Chinese Journal of Scientific Instrument, 2011, 32(1): 163-169.
傅卫平, 秦川, 刘佳, 等. 基于 SIFT 算法的图像目标匹配与定位[J]. 仪器仪表学报, 2011, 32(1): 163-169.
- [4] Pang R, Xu C R. Parallel implementation and optimization of remote sensing image matching based on SIFT feature[J]. Journal of Jiangxi University of Science and Technology, 2016, 37(5): 28-33.
庞瑞, 徐昌荣. 遥感影像 SIFT 特征匹配的并行实现及优化[J]. 江西理工大学学报, 2016, 37(5): 28-33.
- [5] Wang B L, Zhu Z L, Meng L. CUDA-based acceleration algorithm of SIFT feature extraction[J]. Journal of Northeastern University(Natural Science),

- 2013, 34(2): 200-204.
王蓓蕾, 朱志良, 孟球. 基于 CUDA 加速的 SIFT 特征提取[J]. 东北大学学报(自然科学版), 2013, 34(2): 200-204.
- [6] Zhao P, Yan M, Li S K. Performance optimization of application algorithms for heterogeneous multi-processor system-on-chips[J]. Journal of Software, 2011, 22(7): 1475-1487.
赵鹏, 严明, 李思昆. 异构多处理器 SoC 的应用算法性能优化方法[J]. 软件学报, 2011, 22(7): 1475-1487.
- [7] Feng Y, Yuan Q H, Shen J W. Study on programming methods based on CPU+GPU hybrid computing[J]. Communications Technology, 2011, 44(2): 141-143.
冯颖, 袁庆华, 沈健炜. 基于 CPU+GPU 异构计算的编程方法研究[J]. 通信技术, 2011, 44(2): 141-143.
- [8] Jiang C, Geng Z X, Lou B, *et al.* Parallel processing research on SIFT feature matching algorithm based on GPU[J]. Computer Science, 2013, 40(12): 295-297, 307.
姜超, 耿则勋, 娄博, 等. 基于 GPU 的 SIFT 特征匹配算法并行处理研究[J]. 计算机科学, 2013, 40(12): 295-297, 307.
- [9] Jiang L Y. Research on image matching functions performance optimization using OpenCL [D]. Beijing: University of Chinese Academy of Sciences, 2013: 23-35.
蒋丽媛. 基于 OpenCL 的图像匹配算法并行与性能优化研究 [D]. 北京: 中国科学院大学, 2013: 23-35.
- [10] Peng X X. Research on parallel accelerating algorithm based on OpenCL and realization on FPGA [D]. Wuhan: Wuhan Institute of Technology, 2014: 17-27.
彭新显. 基于 OpenCL 并行加速算法研究及其 FPGA 实现[D]. 武汉: 武汉工程大学, 2014: 17-27.
- [11] Xiao H, Guo Y H, Zhou Q L. Parallel algorithm of CPU and GPU-oriented heterogeneous computation in SIFT feature matching [J]. Journal of Tongji University(Natural Science), 2013, 41(11): 1732-1737.
肖汉, 郭运宏, 周清雷. 面向 CPU+GPU 异构计算的 SIFT 特征匹配并行算法[J]. 同济大学学报(自然科学版), 2013, 41(11): 1732-1737.
- [12] Xu C P, Wang G. Parallel design and implementation of scale invariant feature transform algorithm based on OpenCL[J]. Journal of Computer Applications, 2016, 36(7): 1801-1806.
许川佩, 王光, . 基于 OpenCL 的尺度不变特征变换算法的并行设计与实现[J]. 计算机应用, 2016, 36(7): 1801-1806.
- [13] Li Q, Yin Y A. Research on ranging system and stereo matching algorithm based on binocular camera [J]. Telecom Power Technology, 2016, 33(2): 66-67.
李千, 尹业安. 基于双目摄像的测距系统及立体匹配算法研究[J]. 通信电源技术, 2016, 33(2): 66-67.
- [14] Li S X. A method and experiment for measuring distance of object parallax from image[D]. Beijing: Beijing Institute of Technology, 2015: 21-26.
李士玺. 一种利用图像上目标物视差进行测距的方法与实验[D]. 北京: 北京理工大学, 2015: 21-26.
- [15] Wang D D. Research and application of the method to distance measurement based on binocular vision[D]. Chongqing: Chongqing University of Technology, 2016: 37-41.
王代东. 基于双目视觉的测距方法研究与应用[D]. 重庆: 重庆理工大学, 2016: 37-41.
- [16] Sun J G, Liu J, Zhao L Y. Clustering algorithms research[J]. Journal of Software, 2008, 19(1): 48-61.
孙吉贵, 刘杰, 赵连宇. 聚类算法研究[J]. 软件学报, 2008, 19(1): 48-61.
- [17] Shao N, Li H G, Liu L. Binocular stereo distance-measurement method based on feature matching algorithm[J]. Journal of Yanshan University, 2012, 36(1): 57-61.
邵暖, 李惠光, 刘乐. 基于特征匹配算法的双目视觉测距[J]. 燕山大学学报, 2012, 36(1): 57-61.
- [18] Zhang X, Jin Y X, Xue D. Image matching algorithm based on SICA-SIFT and particle swarm optimization [J]. Laser & Optoelectronics Progress, 2017, 54(9): 091002.
张鑫, 靳雁霞, 薛丹. SICA-SIFT 和粒子群优化的图像匹配算法[J]. 激光与光电子学进展, 2017, 54(9): 091002.
- [19] Jiang X D, Yu J Y, Zhu L K, *et al.* Self-calibrated binocular ranging system based on hardware SURF algorithm[J]. Acta Optica Sinica, 2018, 38(10): 1036001.
蒋晓东, 于纪言, 朱立坤, 等. 基于硬件 SURF 算法的自校准双目测距系统[J]. 光学学报, 2018, 38(10): 1036001.
- [20] Yuan R F, Liu M, Hui M, *et al.* Depth map

- stitching based on binocular vision [J]. *Laser & Optoelectronics Progress*, 2018, 55(12): 121013.
- 袁瑞峰, 刘明, 惠梅, 等. 基于双目视觉的深度图拼接[J]. *激光与光电子学进展*, 2018, 55(12): 121013.
- [21] Liu Y Y, Feng P, Long Z R, *et al.* Research and implementation of binocular location system based on region of interest segmentation [J]. *Laser & Optoelectronics Progress*, 2018, 55(5): 051102.
- 刘远远, 冯鹏, 龙邹荣, 等. 基于靶标区域分割的双目定位系统研究与实现[J]. *激光与光电子学进展*, 2018, 55(5): 051102.
- [22] de Arruda G F, da Fontoura Costa L, Rodrigues F A. A complex networks approach for data clustering [J]. *Physica A: Statistical Mechanics and its Applications*, 2012, 391(23): 6174-6183.
- [23] Zhan Z F. Research of SIFT stereo matching algorithm based on GPU [D]. Harbin: Harbin Institute of Technology, 2014: 53-61.
- 占正锋. 基于 GPU 的 SIFT 立体匹配算法研究[D]. 哈尔滨: 哈尔滨工业大学, 2014: 53-61.