高技术通讯 2012 年 第22 卷 第4 期:429 ~ 435

doi:10.3772/j.issn.1002-0470.2012.04.015

基于 FPGA 的 SIFT 特征点检测①

肖 晗② 原 魁③ 何文浩 柴晓杰

(中国科学院自动化研究所 北京 100190)

摘 要 为了在现场可编程门阵列(FPGA)中用硬件电路实现尺度不变特征转换(SIFT)特征点的实时检测,对原算法进行了改进,提出了一种基于面密度插值法和双重极值点约束的新方法。该方法显著提升了 SIFT 特征点的尺度不变性,同时有利于提高定点数的计算精度。在此基础上,设计了一种高效率的硬件计算方案,将所有的计算步骤都实现在了流水线结构中,并在 FPGA 上完成了开发工作。与现有研究成果相比,这种高效的方案非常节约硬件资源,大大降低了硬件成本,同时又大幅度提高了计算精度。系统能够在采集图像的同时完成特征点的检测。对于 360 ×288 大小的图像,其理论最高处理能力为 303 帧/秒;由于受到摄像头的图像采集速度限制,目前的实际处理速度是 25 帧/秒。

关键词 尺度不变特征转换(SIFT),现场可编程门阵列(FPGA),特征点检测,机器视觉,硬件计算

0 引 言

尺度不变特征转换(scale invariant feature transform,SIFT)是一种同时具有平移、旋转、尺度不变性的特征点检测和匹配算法[1,2],对光照和仿射变换也具有一定程度的鲁棒性,因而成为计算机视觉中一种著名的算法。然而,由于计算复杂度很高,在通用计算机上用软件进行 SIFT 特征点的检测与匹配计算,常常不能满足实时性的要求。现场可编程门阵列(field programmable gate array,FPGA)作为一种可定制的逻辑电路,能够通过硬件并行计算实现算法的加速。并且 FPGA 的功耗很低,一般不到 1W,适合应用在嵌入式系统中。因此,不少学者对如何用 FPGA 实现 SIFT 算法的实时计算进行了研究[3-9]。

SIFT 算法可分为特征点位置的检测和特征描述向量的提取及匹配两大部分,本文研究前一部分。特征点的位置检测包括高斯滤波、图像求差、求极值点等计算步骤,计算量很大,而操作相对简单,适合用硬件电路实现。具体的开发工作包括:对原算法进行改造,充分挖掘其中的并行性;设计合适的电路结构,合理地分配和使用硬件资源。现有的工作普遍缺乏对算法的合理改造,在硬件计算方案的设计上也存在一些问题。例如:采用级联的方式进行高斯滤波,会增加硬件成本或者降低计算的并行性;固定使用较小的滤波模板而又不增加滤波级数,会严重影响检测到的特征点的数量和质量;采用的定点数位数太少,会造成较大的计算误差;勉强照搬极值点的"精确定位"计算步骤,会大大增加硬件资源消耗量。因此,虽然这些研究成果能够在 FPGA 上以每秒几十帧的速度检测 SIFT 特征点,但是在计算精度和硬件成本等方面仍有很大的改进余地。为了做出一个具有高性价比的实用系统,本文针对硬件计算的特点,对 SIFT 特征点检测算法进行了改进,并设计了一种高效率的电路计算方案。与现有研究成果相比,本文的方案可以有效节约硬件资源,从而能够在计算中采用更多的定点数位数,因而具有低成本和高精度两个显著特点。本文所开发的 FPGA 模块能够在采集图像的同时完成 SIFT 特征点的检测,具有非常好的实时性能。

1 基本算法步骤及其改进

1.1 SIFT 特征点检测的基本算法

根据文献[1],从灰度图像中检测 SIFT 特征点

① 863 计划(2008AA040204,2008AA040209,2009AA043902-2)和国家自然科学基金(60875051)资助项目。

② 男,1981 年生,博士生;研究方向:嵌入式机器视觉系统;E-mail:han.xiao@ia.ac.cn

③ 通讯作者,E-mail:kui.yuan@ia.ac.cn

(收稿日期:2010-11-29)

的基本步骤如下：

(1)用标准差不同的高斯函数分别对原始图像进行滤波。这一过程也称为“建造高斯金字塔”。

(2)滤波后的图像两两求差，得到高斯差(difference of gaussian，DOG)图像。

(3)从相邻的 DOG 图像中求取极值点。

(4)对极值点进行稳定性检查。其中需要计算 Hessian 矩阵，详见文献[1]。

(5)通过插值将原始图像增大或缩小，然后重复以上步骤，从而在更多的尺度层上检测特征点。

1.2 相关的概念和原理

(1)高斯滤波的叠加原理

对一幅未经滤波的图像进行两次高斯滤波，第一次的标准差为 σ_1，第二次的标准差为 σ_2，则等效于对原图像进行一次标准差为 $\sigma = \sqrt{\sigma_1^2 + \sigma_2^2}$ 的高斯滤波。

(2)尺度层次的概念

用不同大小的高斯核对图像进行滤波，得到的图像位于不同的尺度层次。通过像素插值对滤波后的图像进行大小缩放时，图像的模糊程度发生改变，但其所处的尺度层次不变。

1.3 算法改进

本文对基本算法进行了改进。新方案具有以下几个特点：(1)更适合用并行计算实现；(2)能够提高定点数的计算精度；(3)节约硬件成本，同时又兼顾了特征点的数量和质量；(4)增强了特征点的尺度不变性。

为了提高整个系统的性价比，本文在改进方案中取消了文献[1]推荐的通过插值、迭代对特征点进行“精确定位”的计算步骤。

1.3.1 尺度层的选择

(1)图像增大的等效替代方法

文献[1]中的方法需要通过插值对原始图像进行增大，使其边长增为原来的 2 倍。然而，该操作会使 FPGA 硬件资源的消耗量增加 4 倍以上。根据 1.2 节所述的尺度层次的概念，本文采取了一种改进措施：不进行图像增大，直接用标准差较小的高斯函数对原始图像进行滤波，从而等效地得到较高的尺度层。

(2)尺度层间距的选择

根据文献[1]，最佳的尺度层间距为 $2^{1/3}$，次优方案是 $2^{1/2}$。考虑到开发系统中 FPGA 容量的限制，本文选取了后者。

(3)舍去太低和太高的尺度层

根据实验，太高的尺度层，抗噪声能力较差；太低的尺度层上检测到的特征点又太少。经过权衡，我们最终选择了 3 个特征点尺度层，如图 1 所示。

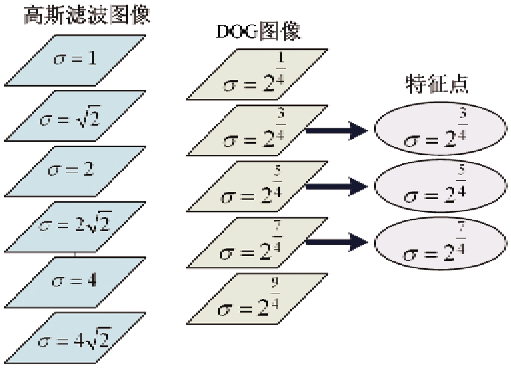


图 1 在 3 个尺度层上检测 SIFT 特征点

最终的方案无需改变图像大小，只需对原始图像进行 6 层高斯滤波即可。此方案便于在统一的流水线结构中实现，大大节省了硬件资源，同时又兼顾了特征点的数量和质量。

1.3.2 采用“面双法”检测特征点

图 1 中，3 个特征点尺度层上图像的模糊程度不同，同样使用 3×3 的窗口检测极值点时，会存在如下差异：(1)模糊程度大的尺度层上，相邻像素值更为接近，在计算 Hessian 矩阵时，差分结果的数值较小，对定点数计算精度不利；(2)模糊程度不同使得 3×3 窗口内的有效信息量不同，从而降低了特征点在不同尺度层间的重复检测率(尺度不变性)。

改进的方案通过像素插值降低局部的模糊程度，从而使得 3 个尺度层上的特征点检测在统一的模糊程度下进行。为了避免检测到相邻特征点(相距 1～2 个像素)，采用了双重极值点约束——如图 2所示，要求两个 3×3 矩阵的中心像素均为极值。其中，由“★”构成的较大的 3×3 像素矩阵通过插值得到。

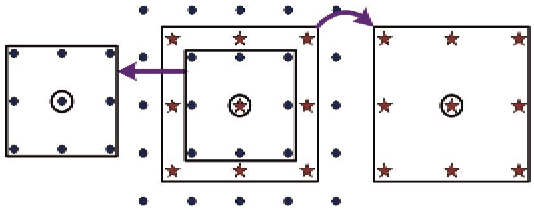


图 2 像素插值与双重极值点约束

在像素插值的方法上，经过实验发现，传统的双线性插值法效果不够理想，即使采用了双重极值点

约束,仍然会检测到一些相邻特征点。为此,我们将地理科学中的面插值法[10]引入到图像缩小的计算中,提出了“面密度插值法”。如图 3 所示,将每个像素的值视为一块正方形区域内的面密度,通过面积加权的方法,计算出插值像素所对应的新正方形内的平均面密度,也就是插值像素值。例如,图 3 中右下角“★”对应的插值像素值为

$$\rho = \frac{\rho_1 A_1 + \rho_2 A_2 + \rho_3 A_3 + \rho_4 A_4}{A_1 + A_2 + A_3 + A_4} \tag{1}$$

其中,$\rho_1 - \rho_4$ 代表面密度,$A_1 - A_4$ 代表对应的矩形区域的面积。实验证明,该方法更适合特征点检测这种精细的局部计算。

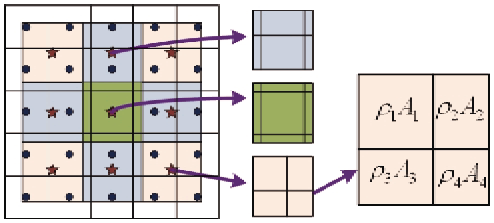


图 3　用面密度插值法计算 3×3 的像素矩阵(第 2 尺度层)

我们将这种基于面密度插值法和双重极值点约束的方法简称为“面双法”。

从相邻的 3 层 DOG 图像中检测极值点,可以用并行计算结构实现。如图 4 所示,在中间一层 DOG 图像上,通过“面双法”检测出中心极值点 B;而对于上下两层 DOG 图像,只采用面密度插值法,从插值后的 9 个像素中求出极值(不一定位于中心)。在 FPGA 内实现时,3 个 DOG 图像层上的计算是同时进行的,A, B, C 是相邻 3 个模块的输出结果,在下一级模块中检测 B 是否为三者的极值。

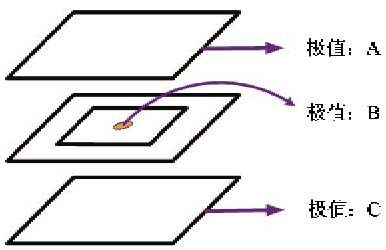


图 4　特征点检测的并行计算

多次实验测试的结果表明,与文献[1]中的原算法相比,采用“面双法”能够使检测到的特征点的数量增加大约 33%,并可以保持并提高特征点的质量。后者主要反映在以下两个方面:(1)特征点的帧间重复检测率,亦即算法的抗噪声能力,基本保持不变;(2)特征点的尺度层间重复检测率获得显著提升,从而提高了特征点的尺度不变性。

2　硬件计算方案的设计

在算法改进的基础上,要实现低成本、高性能的实时计算,还需要设计合理的硬件实现结构。

2.1　高斯滤波

高斯滤波在 SIFT 算法的运算量中占有相当大的比重,而且又处在整个算法的起始阶段。高斯滤波的硬件实现结构,不仅对整个系统的硬件资源使用量和实时性产生重大影响,而且影响到后续计算步骤的硬件实现方式和计算精度。

2.1.1　二维高斯滤波的可拆分性

对图像进行一次标准差为 σ 的二维高斯滤波,等效于分别沿纵向和横向对图像进行一次标准差为 σ 的一维高斯滤波。利用这个性质,可以大大减小高斯滤波的计算量,从而大大降低 FPGA 中硬件资源的使用量。

2.1.2　直接滤波与累进滤波

根据 1.2 节所述的高斯滤波叠加原理可知,各高斯滤波图像既可以在原始图像上通过直接滤波得到,也可以在前一幅标准差较小的高斯滤波图像上通过累进滤波(或称“级联滤波”)得到。

在 FPGA 上实现时,更适合采用直接滤波的方式,因为这样可以实现缓存的共享(见后文)。

2.1.3　滤波模板的大小

高斯函数的标准差越大,所需要的滤波模板越大。图 5 为一维高斯函数曲线。从图 5 可见,一维高斯核的半径取 3.5σ 是最合适的。由此可以定义高斯滤波模板的必要长度为

$$W = 2 \times Round(3.5\sigma) + 1 \tag{2}$$

其中,$Round$ 代表四舍五入运算。

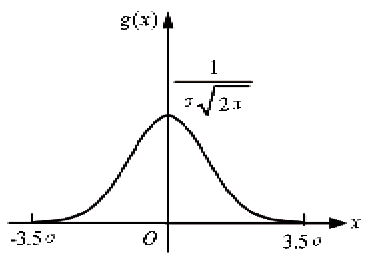


图 5　一维高斯函数曲线

例如 $\sigma = \sqrt{2}$ 时,滤波核的半径 $r = 3.5\sigma = 4.949$,则滤波模板的必要长度为 $W = 2 \times Round$

(4.949) + 1 = 11。

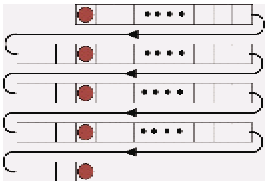
当实际使用的模板长度小于由式(2)计算出的必要长度时,会在高斯滤波的计算结果中引入误差。模板长度比必要长度小得越多,滤波误差越大。

实验证明,如果固定使用较小的模板长度(例如将各级滤波模板的长度都固定为 7 或固定为 5),而又不增加滤波级数,那么会给高斯滤波带来很大误差,严重影响检测到的特征点的数量和质量。

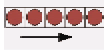
2.1.4 纵向滤波与横向滤波的顺序

在算法上,将二维高斯滤波拆分成两个一维高斯滤波,其顺序对结果没有影响。然而,纵向滤波与横向滤波在硬件资源消耗量上却大不相同,这使得两者的顺序对整个系统的硬件资源使用量具有重大影响。

图 6 示意了纵向滤波与横向滤波所需要的缓存。假设图像宽度为 360 个像素,则对于一个长度为 5 的滤波模板来讲,纵向滤波需要缓存 360 ×4 = 1440 个像素,而横向滤波仅需要缓存 5 个像素。图 6中的大圆点代表与模板系数相乘的像素。



(1) 纵向滤波缓存



(2) 横向滤波缓存

图 6 滤波缓存

可见,高斯滤波对 FPGA 存储资源的消耗主要用在了纵向滤波上。在直接滤波方式下,采用先纵后横的滤波顺序,则只需按照最大的滤波模板长度设置一个纵向滤波缓存即可,其他较小的滤波模板可以共享缓存内的像素数据;而如果采用累进滤波方式,则无法实现缓存的共享。

2.1.5 最节省硬件资源的高斯滤波实现结构

文献[11]在 FPGA 上实现的算法虽然不是 SIFT 算法,但是也涉及到多层高斯滤波。文献[11]对二维高斯滤波进行了拆分,采用直接滤波的方式,先纵后横的滤波顺序,并且对各层高斯滤波共享了纵向滤波缓存。在硬件计算中,乘法所需要的逻辑资源远多于加法。文献[11]注意到一维高斯函数的对称性,提出“将对称位置的像素相加再与模板系数相乘”,从而进一步节省了硬件资源。

本文在高斯滤波的实现结构上借鉴了文献[11]的方案。此方案不仅能够节省大量的硬件资源,而且使得后续的计算步骤——求 DOG 图像时像素的对齐问题——变得极为简单。硬件资源的节省,使得增加定点数位数成为可能,从而能够提高硬件计算的精度。我们的系统在计算精度上能够接近 SIFT 算法的软件实现,而这是现有的大多数研究成果所不能做到的。

2.2 在流水线中进行极值点检测和稳定性检查

由于采用了先纵后横的高斯滤波顺序,求取 DOG 图像的模块接收的是横向滤波模块的输出,各层滤波结果只相差几个到十几个时钟周期,只需很少的延时单元便可实现对齐。在 DOG 图像的基础上,按照面密度插值法的需要,通过开辟 3 ×3 的窗口或 5 ×5 的窗口来实现极值点的检测。极值点的稳定性检查也包括在了其中。整个过程通过几条并行的流水线来实现(见图 7)。

文献[1]中推荐的“极值点精确定位”,是一种适合软件的计算步骤,用硬件实现时成本很高,因此我们在实现方案中取消了该步骤。

2.3 定点数位数选择

FPGA 内的电路使用定点数进行运算。对每一个参与计算的变量或常量,在保证计算精度的前提下,选用最少的二进制位数,是节约硬件资源的关键。原始图像的像素灰度值是 8 位无符号数。对于其他变量和常量,通过软件模拟有限精度计算,确定其最佳位数如表 1 所示。

表 1 最合适的定点数位数

高斯滤波模板系数	14 位无符号数
滤波后的图像像素	13 位无符号数
DOG 图像像素	12 位有符号数
面密度插值法的系数(第 2 尺度层)	10 位无符号数

实验证明,当定点数位数少于表 1 中的推荐值时,特征点检测的性能会有显著下降;而多于表 1 中的推荐值时,特征点检测的性能只有少许提高。

2.4 整体实现结构

本文提出的硬件计算方案如图 7 所示。图中所有的模块并行工作,能够在采集图像的同时完成处理,其处理速度只取决于图像的传输速度。

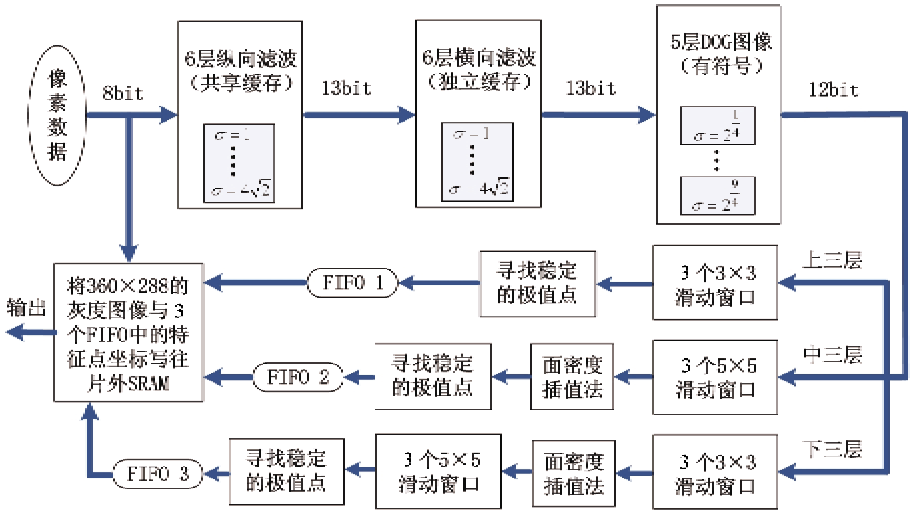


图 7　SIFT 特征点检测在 FPGA 内的实现结构

2.5　系统的处理能力

图 7 中的所有模块都在同一个时钟驱动下工作。目前的时钟频率为 27MHz,系统的处理速度为 25 帧/秒。该速度实际上受到了模拟摄像头的视频采集速度和图像格式的限制,并不是 FPGA 真正的处理能力。视频信号为逐行倒相(phase alternating line,PAL)制式彩色图像,大小为 720 ×576,分为奇偶两场,每采集一帧图像需要 40ms。由于只对奇场图像进行处理,并且进行了间隔采样,所以实际处理的图像大小为 360 ×288。再考虑到 SIFT 算法只处理像素的亮度分量,不处理色差分量,所以在每一帧图像的传送时间里,FPGA 实际上只有 1/8 的时间在工作。因此,在 27MHz 的工作频率下,对于 360 ×288 大小的图像,FPGA 的实际处理能力是200 帧/秒。而根据开发工具汇报的时序分析结果,SIFT 特征点检测模块在 FPGA 内的最高工作频率可达 41MHz,因此,对于 360 ×288 大小的图像,FPGA 的理论最高处理能力是 303 帧/秒。

3　硬件系统与实验结果

3.1　实验系统简介

本文工作的硬件基础是课题组研制开发的一种基于 FPGA 和数字信号处理器(DSP)的图像采集处理卡,如图 8 所示。FPGA 为 Altera 公司 Cyclone III 系列的 EP3C40F484 型号产品,DSP 为 TI 公司的 TMS320DM642 型号产品。本文所阐述的特征点检测工作完全在 FPGA 上完成。DSP 主要负责 SIFT 算法后一阶段的计算,即特征描述向量的提取和匹配以及目标识别等应用,本文对此不做介绍。系统在工作时,通过 A/D 转换芯片将 PAL 制式的模拟视频信号转换为符合 BT.656 标准的数字信号,输入 FPGA 进行处理。FPGA 的引脚与两块静态随机存取存储器(static random access memory,SRAM)相连。在 FPGA 将当前帧的图像数据和特征点坐标写入其中一块 SRAM 时,DSP 可以从另一块 SRAM 中读取上一帧的结果并进行后续处理,从而实现并行工作。

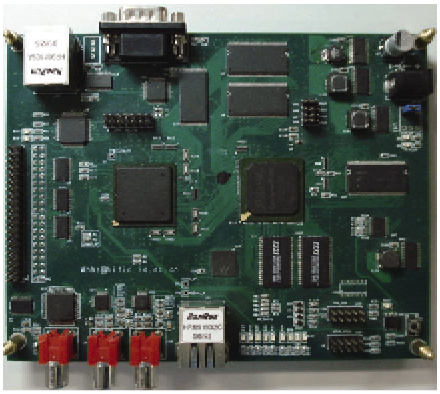
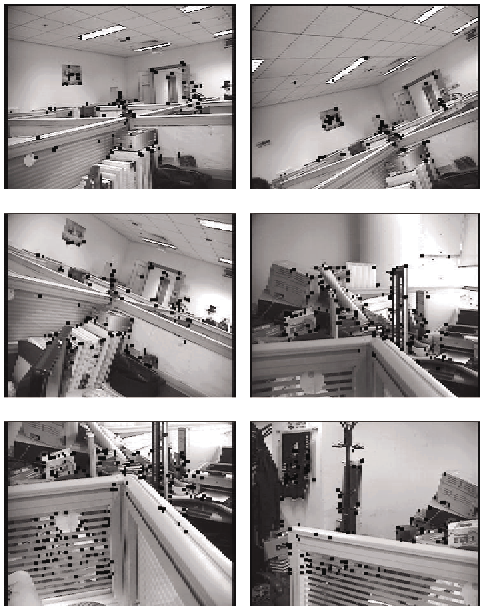


图 8　基于 FPGA 与 DSP 的图像采集处理卡

3.2　实验结果

如图 9 所示,摄像头从 6 个角度拍摄室内场景,对每个角度取两帧,共取 6 对(即 12 幅)图像进行实验。表 2 对 FPGA 与 MATLAB 的特征点检测结果进行了对比,列出了两者检测到的特征点个数、重合个数、以及重合率。从表中可见,两者的检测结果相当接近。



（来自图像采集处理卡，其中的黑点为 FPGA 检测到的特征点位置）

图 9 六幅室内图像

表 2 FPGA 与 MATLAB 的特征点检测结果对比（12 幅图像）

尺度层	重合	FPGA	MATLAB
1	576	761	604
	—	75.7%	95.4%
2	411	418	436
	—	98.3%	94.3%
3	284	329	346
	—	86.3%	82.1%

当景物和摄像头都保持不动时，SIFT 特征点在不同帧之间的重复检测率反映了算法的抗噪声性能。从表 3 给出的 FPGA 和 MATLAB 的帧间重复检测率的对比可以看出，两者的结果也非常接近。

表 3 SIFT 特征点帧间重复检测率

尺度层	MATLAB	FPGA
1	80.8%	78.9%
2	91.3%	88.5%
3	89.6%	89.4%

3.3 FPGA 的资源使用量

特征点检测模块在 EP3C40F484 型 FPGA 上的资源使用量如表 4 所示。

因为高斯滤波模板系数是固定的，所以高斯滤波在硬件实现中是通过移位和加法操作来实现的，没有用到专用乘法器资源。表中的专用乘法器资源用在了极值点稳定性检查的计算上。

表 4 SIFT 特征点检测模块的 FPGA 资源使用量

	使用量	片内资源总量	百分比
逻辑资源（LEs）	22067	39600	64%
存储资源（bit）	230744	1161216	21%
专用乘法器（9-bit）	18	252	7%

从表 4 中可以看到，FPGA 还有足够的剩余资源，可以用来开发一些供 DSP 调用的辅助计算模块，对 SIFT 算法的后半部分（描述向量的提取）进行加速。

4 结 论

本文研究了 SIFT 算法的前半部分——特征点的检测——在硬件中的实时计算。一方面，针对硬件计算的特点，对原算法进行了有效改进；另一方面，设计了一种高效率的硬件实现结构。与现有研究成果相比，本文的研究成果具有以下优点：（1）算法具有高性能，特征点的尺度不变性更高；（2）采用了足够的定点数位数，获得了与软件相媲美的计算精度；（3）节省硬件资源，成本更低。

本文的方案实现了完全并行的计算结构，能够在采集图像的同时完成特征点检测。目前的处理速度受到模拟摄像头图像采集速度的限制，为25 帧/秒。当采用速度更快的数字摄像头后，系统的处理速度还会获得进一步提升，而整体结构无需大的变动。

参考文献

[1] Lowe D G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004, 60(2): 591-110

[2] Lowe D G. Object recognition from local scale-invariant features. In: Proceedings of the International Conference on Computer Vision, Corfu, Greece, 1999. 1150-1157

[3] Chati H D, Mühlbauer F, Braun T, et al. Hardware/software co-design of a key point detector on FPGA. In: Proceedings of the 15th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, Napa Valley, USA, 2007. 355-356

[4] Bonato V, Marques E, Constantinides G A. A parallel hardware architecture for scale and rotation invariant feature detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 2008, 18(12): 1703-1712

[5] Qiu J, Huang T, Ikenaga T. A 7-round parallel hardware-saving accelerator for gaussian and DoG pyramid construction part of SIFT. In: Proceedings of the 9th Asian Conference on Computer Vision, Xi'an, China, 2009. 75-84

[6] Qiu J, Huang T, Ikenaga T. A FPGA-based dual-pixel processing pipelined hardware accelerator for feature point detection Part in SIFT. In: Proceedings of the 5th International Joint Conference on INC, IMS and IDC, Seoul, South Korea, 2009. 1668-1674

[7] Chang L, Hernández-Palancar J. A hardware architecture for SIFT candidate keypoints detection. In: Proceedings of the Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications——14th Iberoamerican Conference on Pattern Recognition, Havana, Cuba, 2009. 95-102

[8] Yao L, Feng H, Zhu Y, et al. An architecture of optimised SIFT feature detection for an FPGA implementation of an image matcher. In: Proceedings of the International Conference on Field-Programmable Technology, Sydney, Australia, 2009. 30-37

[9] Pettersson N, Petersson L. Online stereo calibration using FPGAs. In: Proceedings of the 2005 IEEE Intelligent Vehicles Symposium Proceedings, Las Vegas, USA, 2005. 55-60

[10] 潘志强, 刘高焕. 面插值的研究进展. 地理科学进展, 2002, 21(2):146-152

[11] Cabani C. Implementation of an Affine-invariant Feature Detector in Field-Programmable Gate Arrays: [Master dissertation]. Toronto: University of Toronto, 2006. 32-34

SIFT keypoints detection based on FPGA

Xiao Han, Yuan Kui, He Wenhao, Chai Xiaojie

(Institute of Automation, Chinese Academy of Sciences, Beijing 100190)

Abstract

In order to achieve real-time detection of scale invariant feature transform (SIFT) keypoints via hardware circuits in field programmable gate array (FPGA), the original algorithm was ameliorated and a new method based on area density interpolation and dual extremum restriction was proposed, which can greatly enhance the scale invariance of SIFT keypoints while benefiting the computation accuracy of fixed-point numbers. After that, a highly efficient hardware computation scheme was designed and implemented in a FPGA chip with all the computational procedures arranged in a pipelined structure. Compared with existing research achievements, the above-mentioned scheme needs much less hardware resources, resulting in the great reduction of hardware costs and the great improvement of computation accuracy. The system can perform image acquisition and keypoint detection at the same time. For an image size of 360 × 288, its theoretical maximum throughput can reach 303 fps (frames per second), while its current actual processing rate is 25 fps because it is limited by the speed of the camera.

Key words: scale invariant feature transform (SIFT), field programmable gate array (FPGA), keypoint detection, machine vision, hardware computation