

http://bhxb.buaa.edu.cn jbuua@buaa.edu.cn
DOI: 10.13700/j.bh.1001-5965.2018.0438

基于 FPGA 的低复杂度快速 SIFT 特征提取

姜晓明^{1,2,*}, 刘强^{1,2,*}

(1. 天津大学 微电子学院, 天津 300072; 2. 天津市成像与感知微电子技术重点实验室, 天津 300072)

摘 要: 尺度不变特征变换(SIFT)算法具有优良的鲁棒性,在计算机视觉领域得到广泛应用。针对 SIFT 算法高计算复杂度而导致其在 CPU 上运行实时性低的问题,基于现场可编程门阵列(FPGA)设计了一种低复杂度的快速 SIFT 硬件架构,主要对算法的特征描述符提取部分进行优化。通过降低梯度信息(包括梯度幅值和梯度方向)的位宽、优化高斯权重系数的产生、简化三线性插值系数的计算和简化梯度幅值直方图索引的求解等方法,避免了指数、三角函数和乘法等复杂计算,降低了硬件设计复杂度和硬件资源消耗。实验结果显示,提出的低复杂度快速 SIFT 硬件架构,与软件相比,可以获得约 200 倍的加速;与相关研究相比,速度提高了 3 倍,特征描述符稳定性提高了 18% 以上。

关 键 词: 现场可编程门阵列(FPGA); 尺度不变特征变换(SIFT); 硬件设计; 梯度信息; 特征描述符提取

中图分类号: TP391.4

文献标识码: A

文章编号: 1001-5965(2019)04-0804-07

近年来,计算机视觉技术迅猛发展^[1]。在计算机视觉领域,图像特征提取是目标识别的基础,所提取的特征对于尺度和旋转变化必须具有不变性,以保证最终的目标识别效果^[2]。尺度不变特征变换(SIFT)特征提取算法具有优良的鲁棒性,在计算机视觉领域广泛应用^[3-4]。

近几年,深度神经网络在目标识别领域取得了很好的效果^[5-6]。在传统特征描述算法启发下,基于深度神经网络的可学习特征描述子成为新的研究课题^[7]。传统的图像处理方式不但可以为深度神经网络的发展提供思路,而且在一些应用中还表现出优异的性能^[8]。因此,对于传统目标识别算法的研究仍然非常重要。

SIFT 特征提取算法具有较高的计算复杂度。实验发现,对于分辨率为 480 像素 × 320 像素的图像,提取 713 个特征点的特征描述符的时间消

耗约为 2 s(CPU I5 2.6 GHz, Visual Studio 2010, OpenCV2.4.11)。为了达到目标识别实时性的要求,Heymann 等利用 CPU 并行运算加速 SIFT 算法,对于分辨率为 640 像素 × 480 像素的图像处理速度达到 20 帧/s^[9];Zhang 等利用多核系统对 SIFT 算法进行加速,对于大小为 640 像素 × 480 像素的图像,可以获得平均 45 帧/s 的加速效果^[10]。但是,这 2 种方式都存在功耗高的缺点,无法在嵌入式系统中应用。

近年来,现场可编程门阵列(FPGA)由于具有速度高功耗低的特点,得到了广泛应用^[11-12]。因此,为了满足速度和功耗的需求,许多研究者利用 FPGA 对 SIFT 算法进行加速。Qasaimeh 等^[13]基于并行的方式构造包含 6 层高斯图像的高斯金字塔,采用 17 × 17 的特征点邻域窗口进行特征点主方向计算和特征描述符提取。过多的高斯金字

收稿日期: 2018-07-23; 录用日期: 2018-10-19; 网络出版时间: 2018-11-12 11:27
网络出版地址: kns.cnki.net/kcms/detail/11.2625.V.20181109.0852.001.html
基金项目: 国家自然科学基金(61574099)
* 通信作者: E-mail: qiangliu@tju.edu.cn

引用格式: 姜晓明, 刘强. 基于 FPGA 的低复杂度快速 SIFT 特征提取[J]. 北京航空航天大学学报, 2019, 45(4): 804-810.
JIANG X M, LIU Q. Low-complexity fast SIFT feature extraction based on FPGA[J]. Journal of Beijing University of Aeronautics and Astronautics, 2019, 45(4): 804-810 (in Chinese).

塔层数和大量的用于产生特征点邻域窗口的行缓存增加了硬件资源消耗。Vourvoulakis 等^[14]利用 4 层高斯图像构造高斯金字塔,能够降低硬件存储资源,但是由于没有计算特征点主方向,提取的特征描述符不具备旋转不变性。Jiang 等^[3]为了提高速度,同时进行特征点主方向计算和特征描述符提取,然后根据特征点主方向对特征描述符重新排序。忽略了特征描述符提取与特征点主方向的依赖关系,导致特征描述符稳定性降低。

Vourvoulakis 等^[14]对 SIFT 算法中特征点检测部分进行了优化,使得特征点检测所需时钟周期数越来越接近于图像像素点的数量,特征描述符提取部分逐渐成为计算的瓶颈。本文基于 FPGA 设计了一个 SIFT 特征提取硬件加速架构。在相关研究^[3,13-14]的基础上,通过降低梯度信息(包括梯度幅值和梯度方向)位宽,优化高斯权重系数的产生、简化三线性插值系数的计算和简化梯度幅值直方图索引的求解等技术,重点对特征描述符提取部分进行优化。

1 SIFT 硬件设计

1.1 SIFT 算法原理

SIFT 算法的核心思想是:首先检测图像中稳定的特征点,然后计算特征点的主方向并根据其对特征点邻域旋转,最后提取特征点的特征描述符^[2]。

特征点检测时,首先对输入图像进行包括上采样和初始二维高斯滤波的预处理,在此基础上构建高斯金字塔和高斯差分金字塔,然后在高斯差分金字塔内检测特征点。

特征点主方向的计算在特征点邻域内进行,关键是构建梯度幅值直方图。首先把 360° 平均分成 36 类,每类 10° 且用 1 个直方图表示。对于特征点邻域内每一个像素点,计算它的梯度方向和梯度幅值。然后把高斯权重化的梯度幅值加到与梯度方向相对应的梯度幅值直方图中。最后,计算梯度幅值直方图峰值,该峰值对应的梯度方向为特征点主方向。

在提取特征描述符之前,根据特征点主方向对特征点邻域旋转以使特征描述符具备旋转不变性。特征描述符提取的核心是构建 128 维梯度幅值直方图。在特征点邻域内,计算每一个像素点的梯度幅值和梯度方向,梯度方向和特征点主方向与旋转后的坐标共同决定最终梯度幅值直方图索引,同时对高斯权重化后的梯度幅值进行三线性插值处理并将插值结果加到对应的梯度幅值直

方图中。特征描述符提取伪代码(OpenCV 2.4.11)如下所述。

步骤 1 1) 邻域像素点旋转。
 $c_rot = c \cdot \cos_t - r \cdot \sin_t; r_rot = c \cdot \sin_t + r \cdot \cos_t$
其中:(r, c)和(r_rot, c_rot)分别为区域旋转前、后的像素点坐标; \sin_t 和 \cos_t 分别为特征点主方向的正弦值和余弦值。

2) 梯度信息求解。
 $fastAtan(dy, dx, Ori); magnitude(dx, dy, Mag)$
其中, dy 和 dx 分别为行、列方向的像素值变化量, Ori 和 Mag 分别为所求解的梯度方向和梯度幅值。

3) 高斯权重求解。
 $w = c_rot \cdot c_rot + r_rot \cdot r_rot; \exp(w, w)$
其中: w 为高斯权重系数。

步骤 2 三线性插值系数求解。
 $rbin = r_rot + d/2 - 0.5; cbin = c_rot + d/2 - 0.5$
 $obin = (Ori - Ori) \cdot bpr$
 $r0 = Floor(rbin); c0 = Floor(cbin); o0 = Floor(obin)$
 $rbin = rbin - r0; cbin = cbin - c0; obin = obin - o0$

其中: $rbin, cbin$ 和 $obin$ 分别表示行、列和垂直方向的插值系数; d 为特征点邻域行或列方向的子区域个数; ori 为特征点主方向, $Floor$ 表示取整运算; $bpr = 1/45$,为常数。

步骤 3 1) 梯度幅值直方图索引计算。
 $idx = (r0 \times 4 + c0) \times 8 + o0$
2) 梯度幅值权重化。
 $val = Mag \cdot rbin \cdot cbin \cdot obin$
步骤 4 更新梯度幅值直方图。
 $hist[idx] = val$

步骤 5 对邻域内所有像素点重复步骤 1 至步骤 4。

1.2 SIFT 硬件设计优化分析

从特征描述符提取伪代码可见,特征描述符提取涉及指数、三角函数、乘除法复杂数学运算,不易于硬件实现。本节结合子区域划分方式对特征描述符提取部分进行优化,以降低硬件设计复杂度、硬件资源消耗和提高速度。

本文选取 16 × 16 的特征点邻域提取特征描述符,其子区域划分方式如图 1 所示, r, c 分别表示特征点邻域像素点的行、列坐标,从左上角开始每 4 × 4 个像素点作为一个子区域(红色正方形),256 个像素点分成 16 个子区域。图中红色的点表示子区域种子点,蓝色点代表特征点。其中每个小正方形表示一个像素的梯度信息,其梯

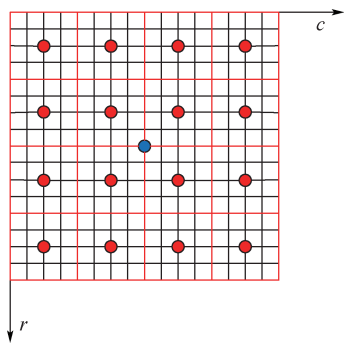


图1 子区域划分
Fig.1 Subregion division

度方向分成 8 类,每类 45°。因此,每个子区域对应一个 8 维梯度幅值直方图,16 个子区域对应 128 维梯度幅值直方图。

特征描述符提取伪代码中获得 128 维梯度幅值直方图需要产生高斯权重系数 w ,它的产生在邻域旋转之后进行,旋转方式如式(1)、式(2)所示。旋转前、后像素点的位置分别为 (r, c) 和 $(r_{\text{rot}}, c_{\text{rot}})$, θ 为特征点主方向,高斯权重的计算如式(3)所示。其中, d 为常数 4,表示图 1 中行、列方向划分的子区域个数。由于式(1)~式(3)包含指数、三角函数以及乘法等数学运算,计算复杂度高,不易于硬件实现。

$$c_{\text{rot}} = c \cos \theta - r \sin \theta \tag{1}$$

$$r_{\text{rot}} = c \sin \theta + r \cos \theta \tag{2}$$

$$w = e^{-\frac{(c_{\text{rot}})^2 + (r_{\text{rot}})^2}{2 \times (0.5d)^2}} \tag{3}$$

但是,将式(1)和式(2)代入式(3)可以看出,高斯权重系数与坐标旋转无关,只与邻域像素点(图 1 中每个小正方形)和特征点之间的相对位置有关。当子区域划分确定后,邻域像素点与特征点之间的相对位置确定,所以 w 也随之确定。因此,本文提前计算图 1 中每个像素点的高斯权重并固化到 ROM(FPGA 片内存储资源)中,硬件执行时直接读取,可以避免指数、三角函数和乘法等数学运算,降低硬件复杂度,提高处理速度。

梯度信息的求解是计算特征点主方向与提取特征描述符的基础,利用硬件实现时,梯度方向一般用 16 bits 表示^[3]。当利用流水线结构计算特征点主方向和提取特征描述符时,需要构造 2 个 16×16 的存储像素梯度信息的寄存器阵列,会消耗大量硬件资源。如 1.1 节所述,梯度方向分成 36 类,主要作用是确定梯度幅值直方图索引。在后续计算中,只需要梯度方向的类别,不需要具体的方向。因此,本文增加了一个梯度方向分类模块,直接将梯度方向转变为 6 bits 梯度方向类别,

明显降低了后续特征点主方向计算模块和特征描述符提取模块的硬件资源消耗。

对高斯权重化的邻域像素点梯度幅值进行三线性插值是提取特征描述符的关键。假设 r, c 代表水平方向坐标轴, o 代表梯度方向坐标轴。从特征描述符提取伪代码可见,三线性插值主要包括 r, c 水平方向的双线性插值以及 o 方向的插值,本质是求像素点梯度幅值对梯度幅值直方图的贡献,该贡献与距离负相关,距离越小贡献越大^[2]。三线性插值时,首先进行的是 r, c 水平方向的双线性插值。从特征描述符提取伪代码可见,双线性插值系数 rbin, cbin 与式(1)、式(2)相关,涉及三角函数和乘法运算,硬件实现复杂度。为了避免三角函数和乘法计算,本文对双线性插值系数的产生进行简化,特征点邻域旋转利用邻域像素梯度方向类别与特征点主方向类别相减的方式表示。在图 1 中,当子区域划分确定后,邻域像素点 (r, c) 与其种子点之间的相对位置确定,因此可以根据该相对位置确定双线性插值系数并固化到 ROM 中,硬件执行时直接进行读取。简化后的 r, c 水平方向插值系数计算公式如式(4)~式(6)所示。其中, Rbin, Cbin 分别表示行、列方向的插值系数, RCbin 表示 r, c 平面的双线性插值系数, (r, c) 表示邻域像素点与特征点之间的相对位置, seed 代表种子点的位置, k 表示种子点序号, t 表示在 r, c 平面 1 个邻域像素点最多对 4 个种子点有贡献,贡献与距离负相关,最后进行归一化。

$$\text{Rbin}((r-1) \times 16 + c, t) = \frac{(4 - \text{abs}(r - \text{seed}(k, 1)))}{4} \tag{4}$$

$$\text{Cbin}((r-1) \times 16 + c, t) = \frac{(4 - \text{abs}(r - \text{seed}(k, 2)))}{4} \tag{5}$$

$$\text{RCbin} = \text{Rbin} \cdot \text{Cbin} \tag{6}$$

三线性插值还包括 o 方向的插值,在 o 方向上插值前,邻域像素梯度方向类别和特征点主方向类别相减得到相对梯度方向类别。对于在 o 方向上的插值系数,假设相对梯度方向类别为“11”,取其低 2 位为 3,那么它对相邻两个直方图的贡献分别为 0.25 和 0.75。

梯度幅值直方图索引的计算是提取特征描述符的重要一步。从特征描述符提取伪代码可见,梯度幅值直方图索引由 r, c 平面的基准索引和 o 方向的偏移索引两部分组成。基准索引计算与式(1)、式(2)相关,存在三角函数和乘法运算,利用硬件实现复杂度。为了避免指数、三角函数和

乘法运算,本文对梯度幅值直方图基准索引的产生进行简化。与图 1 中红色正方形代表的子区域相对应,直方图基准索引分别为 0,8,⋯,120。在设计时,把图 1 中每个邻域像素点梯度幅值对应的梯度幅值直方图的基准索引(同一子区域中的像素点的基准索引相同,可能不同的是 θ 方向的偏移索引)固化到 ROM 中,硬件执行时直接读取。 θ 方向偏

移索引的计算与梯度方向有关。为了避免利用硬件把 360° 平均分成 8 类,降低复杂度,本文在梯度方向用类别表示的基础上求解 θ 方向的偏移索引,将相对梯度方向类别分成 8 类,分类方式如表 1 所示。对于最终直方图索引,结合表 1,如果 r,c 平面直方图基准索引为 120,相对梯度方向分类后的类别为 7,则对应的最终梯度幅值直方图索引为 127。

表 1 相对梯度方向分类方式
Table 1 Classification method of relative gradient direction

前后类别说明	类别						
分类前类别	0 ~ 7	8 ~ 11	12 ~ 15	16 ~ 19	20 ~ 23	24 ~ 27	28 ~ 31
分类后类别	0	1	2	3	4	5	6

1.3 SIFT 硬件系统设计

本文提出的 SIFT 硬件系统架构如图 2 所示。系统主要包括预处理、尺度空间构建(高斯金字塔构建和高斯差分金字塔构建)、梯度信息计算、特征点检测、主方向计算和特征描述符提取等部分。

在预处理部分,原始图像通过直接存储器访问(DMA)输入至上采样模块^[2],DMA 具备等待功能以保证能够与上采样模块协同工作。上采样模块的输出进入初始二维高斯滤波模块 G0 产生高斯图像,之后再通过 4 个并行的二维高斯滤波器 G1 ~ G4 得到 4 层高斯图像,构成高斯金字塔^[14]。相邻的 2 层高斯图像相减得到的高斯差分图像 D1 ~ D3,构成高斯差分金字塔。特征点的检测在高斯差分金字塔中进行。所检测到的特征点的坐标通过 DMA 传输到双倍速率同步动态随机存储器(DDR,可作为 FPGA 片外存储资源)中进行存储。在进行特征点检测的同时,G2 的输出进入梯度信息计算模块,所求解的梯度信息通过 DMA 传输到 DDR 中进行存储。

特征点检测与梯度信息计算完成后,系统利用 DMA 读取特征点邻域像素点的梯度信息,利用 2 个 16×16 的寄存器阵列组成流水线结构进

行特征点主方向计算和特征描述符提取。

SIFT 特征描述符提取的硬件执行流水线结构和硬件电路分别如图 3、图 4 所示。SIFT 特征描述符的提取主要包括 T0、T1、T2、T3 4 个时刻。在 T0 时刻,系统将特征点邻域像素点的梯度幅值与对应的高斯权重系数相乘产生高斯权重化的梯度幅值 magw 和特征点邻域旋转两个任务并行进行。特征点邻域旋转是指将邻域像素的梯度方向类别 Grc 与特征点主方向类别 Orc 相减得到相对梯度方向类别 $ROrc$ 。在 T1 时刻,确定高斯权重化的邻域像素点梯度幅值在 r,c 平面对周围 4 个种子点的贡献 rcbinmagw;同时,分别取 $ROri$ 的高 4 bits 和低 2 bits 得到 $ROri_4$ 和 $ROri_2$,其中 $ROri_4$ 用于确定最终梯度幅值直方图索引, $ROri_2$ 用于确定在 θ 方向的插值系数。在 T2 时刻,根据 $ROri_4$ 确定最终梯度幅值直方图索引;同时,根据 $ROri_2$ 在 θ 方向上对 rcbinmagw 进行插值得到三线线性插值结果 rcobinmagw。T3 时刻,根据梯度幅值直方图索引,将 rcobinmagw 加到对应的梯度幅值直方图中。当处理完特征点邻域内所有像素点后开始输出提取的特征描述符。输出完毕后,继续提取下一个特征点的特征描述符。

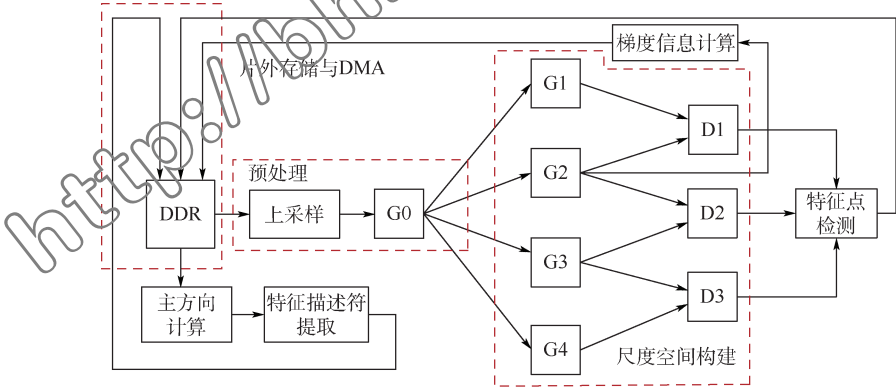


图 2 SIFT 硬件系统架构
Fig. 2 SIFT hardware system architecture

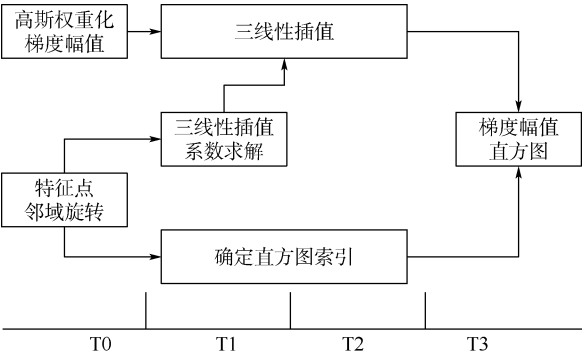


图3 特征描述符提取流水线结构

Fig.3 Feature descriptor extraction pipeline structure

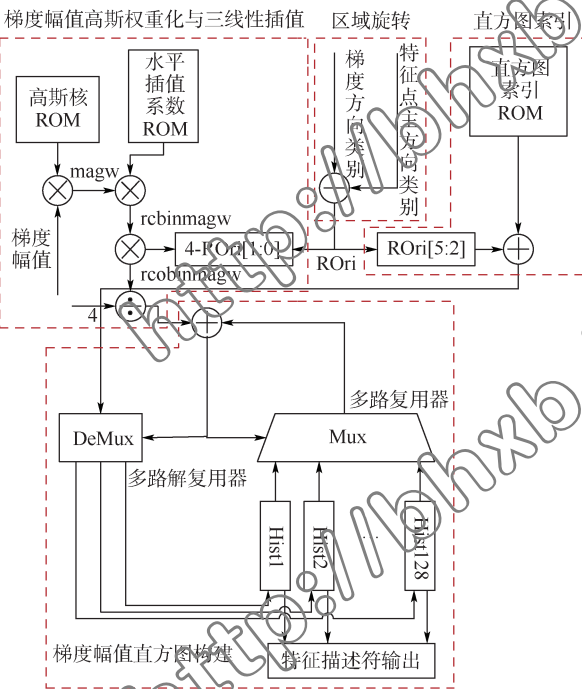


图4 特征描述符提取硬件电路

Fig.4 Hardware circuit of feature descriptor extraction

2 实验与分析

为了评估本文低复杂度快速 SIFT 硬件架构,从特征描述符稳定性、硬件资源消耗、硬件执行速度3个方面与其他研究进行对比。对特征描述

符进行稳定性检测时,选取图 5(a)作为原始图像,对其进行尺度变化、旋转、模糊、光照等处理后得到图像图 5(b),分别提取图 5(a)、(b)的特征描述符进行匹配。在匹配结果中选取20个最佳匹配点对,匹配率等于正确匹配点对的数量与20之比,匹配率越高表示特征描述符稳定性越高。

2.1 特征描述符稳定性检测

图5为硬件提取的特征描述符的匹配结果,从图中可以看出,尺度变化后在20个匹配点对中存在2个明显的错误匹配,正确匹配点对的数量为18,匹配率为0.9,特征描述符稳定性高。

表2为图像逆时针旋转变化的特征描述符稳定性检测结果,随着旋转程度的增加,匹配率呈下降趋势,旋转变化平均匹配率用于与表3中的仿射变化和相机变化进行比较。表3为特征描述符稳定性对比,与文献[3]相比,本文提出的低复杂度快速 SIFT 硬件架构所提取的特征描述符在 JPEG 压缩和光照变化方面稳定性最佳;在仿射变化和相机变化方面,稳定性分别提高了22%、25%;在模糊处理方面,稳定性提高了18%。

2.2 硬件资源消耗对比

表4给出了本文提出的 SIFT 硬件架构与相关文献[3,5,15]的硬件资源消耗对比情况。值得注意的是,本文所需片上存储资源 RAM 与文献[3]相比降低了85%,与文献[13]相比降低了50%;与文献[3]、文献[13]相比本文所需 LUT (查找表)资源和 DSP(乘法器)资源较低,但是本文消耗了较多的 Register(寄存器)资源;而文献[15]消耗了大量的 LUT 与 Register 资源。

表5表示用于特征点主方向计算和特征描述符提取的2个16×16寄存器阵列的资源使用情况,m表示梯度幅值位宽,g表示梯度方向位宽。本文提出的梯度方向表示方法减少了5120个 Register,与传统方法相比,系统 Register 资源降低了21%,而梯度方向分类模块仅消耗102个 LUT 与6个 Register。

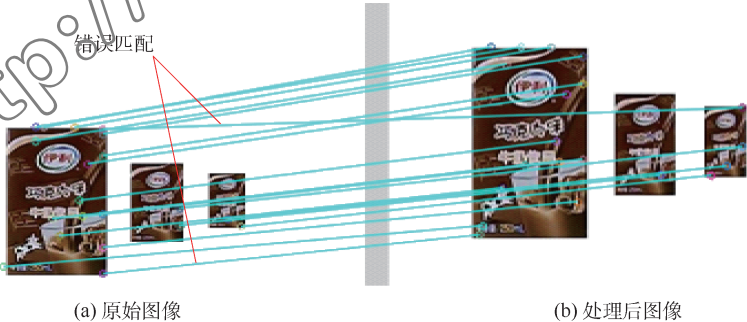


图5 尺度变化特征匹配结果

Fig.5 Feature matching result of scale change

表 2 旋转变化特征描述符稳定性检测

Table 2 Stability detection of feature descriptors for rotation change

旋转程度/(°)	匹配率
5	0.80
10	0.85
15	0.75
20	0.5
25	0.45
30	0.35

注:旋转变化平均匹配率为 0.62。

表 3 特征描述符稳定性对比

Table 3 Stability comparison of feature descriptors

模糊程度(σ)及对比模式	文献[3]匹配率	本文匹配率
0.2	无	0.5
0.4	无	0.5
0.6	无	0.8
0.8	无	0.55
1	无	0.2
模糊处理 ^①	0.42	0.6
JPEG 压缩	0.41	1
光照变化	0.69	1
仿射变化	0.4	0.62 ^②
相机变化	0.37	0.62 ^②

注: σ 为二维高斯滤波器标准差;①文献[3]中模糊处理特征描述符稳定性检测只进行了 1 组实验,本文进行 5 组实验取平均值与其进行对比;②本文利用表 2 中旋转变化平均匹配率与文献[3]的仿射变化和相机变化匹配率进行对比。

表 4 硬件资源消耗对比

Table 4 Comparison of hardware resource consumption

FPGA 与资源消耗对比	文献[3]	文献[13]	文献[15]	本文
FPGA	Virtex-5	Virtex-5	Virtex-6	Virtex-7
LUT/个	26 398	38 179	57 598	23 726
Register/个	10 310	9 646	24 988	18 788
DSP/个	89	52	8	43
RAM/Mbit	7.8	2.4	1.2	1.2

表 5 梯度方向优化前后硬件资源消耗对比

Table 5 Comparison of hardware resource consumption before and after gradient direction optimization

模块	LUT/个	Register/个
梯度阵列($m=14, g=16$)	0	7 680 × 2
梯度阵列($m=14, g=6$)	0	5 120 × 2
梯度方向分类模块	102	6

2.3 硬件执行速度对比

表 6 给出了本文提出的 SIFT 硬件架构与相关文献[3,5,15]的硬件执行速度对比。由于图像分辨率不同,为了便于比较,本文使用参数 PFS(每秒处理的像素个数)进行讨论。 N 为特征点的数量,为了便于分析,本文取 $N=1\ 000$ 时的硬件执行时间。本文提出的 SIFT 硬件设计,处理速

表 6 硬件执行速度对比

Table 6 Comparison of hardware execution speed

类别	文献[3]	文献[13]	文献[15]	本文
图像大小/(像素×像素)	512×512	640×480	640×480	480×320
硬件平台	Virtex-5	Virtex-5	Virtex-6	Virtex-7
时钟频率/MHz	50(特征点检测);100(特征描述符提取)	50	100	100
执行时间($N=1\ 000$)/ms	8.14	27.28	30	10.37
PFS(归一化)/个	1.46	0.5	0.66	1

度可达 100 帧/s,与文献[13,15]相比,获得3 倍的提升;与文献[3]相比速度降低 21.5%,因为其忽略了特征描述符提取对特征点主方向计算的依赖关系,以降低特征描述符稳定性为代价提高速度。

3 结 论

针对 SIFT 算法高计算复杂度的问题,本文基于 FPGA 设计了一种低复杂度的快速 SIFT 特征提取的硬件实现架构,主要对特征描述符提取部分进行了优化,具有如下特点:

- 1) 通过优化高斯权重系数的产生、简化三线性和插值系数的计算和简化直方图索引的求解等技术,避免了指数的三角函数、乘法等复杂数学运算,降低了硬件设计复杂度。
- 2) 通过降低梯度信息位宽,降低了求解特征点主方向和提取特征描述符的硬件资源消耗。
- 3) 在速度方面,与软件相比,可以获得约200 倍的加速;与文献[13,15]相比,速度提升了 3 倍。
- 4) 在特征描述符稳定性方面,与文献[3]相比,提高了 18% 以上。

参考文献 (References)

[1] 王亭亭,蔡志浩,王英勋. 无人机室内视觉/惯导组合导航方法[J]. 北京航空航天大学学报,2018,44(1):176-186.
WANG T T, CAI Z H, WANG Y X. Integrated vision/inertial navigation method of UAVs in indoor environment[J]. Journal of Beijing University of Aeronautics and Astronautics,2018,44(1):176-186 (in Chinese).

[2] LOWE D G. Distinctive image features from scale-invariant keypoints[J]. International Journal of Computer Vision,2004,60(2):91-110.

[3] JIANG J, LI X, ZHANG G. SIFT hardware implementation for real-time image feature extraction[J]. IEEE Transactions on Circuits and Systems for Video Technology,2014,24(7):1209-1220.

[4] MA W, WEN Z, WU Y, et al. Remote sensing image registration

with modified SIFT and enhanced feature matching[J]. IEEE Geoscience and Remote Sensing Letters, 2017, 14(1): 3-7.

[5] LONG J, SHELHAMER E, DARRELL T. Fully convolutional networks for semantic segmentation[C] // IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE Press, 2015:3431-3440.

[6] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[C] // IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE Press, 2016: 770-778.

[7] VIJAY K B G, CARNEIRO G, REID I. Learning local image descriptors with deep siamese and triplet convolutional networks by minimizing global loss functions[C] // IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE Press, 2016:5385-5394.

[8] 王琳,刘强. 基于局部特征的多目标图像分割算法[J]. 激光与光电子学进展, 2018, 55(6): 061002.

WANG L, LIU Q. A multi-object image segmentation algorithm based on local features[J]. Laser & Optoelectronics Progress, 2018, 55(6): 061002 (in Chinese).

[9] HEYMAN S, MÜLLER K, SMOLIC A, et al. SIFT implementation and optimization for general-purpose GPU [J]. Media Culture & Society, 2007, 67(1): 7-13.

[10] ZHANG Q, CHEN Y, ZHANG Y, et al. SIFT implementation and optimization for multi-core systems[C] // IEEE International Symposium on Parallel and Distributed Processing. Piscataway, NJ: IEEE Press, 2008: 1-8.

[11] LIU Q, LIU J, SANG R, et al. Fast neural network training on fpga using quasi-Newton optimization method[J]. IEEE Transactions on Very Large Scale Integration Systems, 2018, 26(8): 1575-1579.

[12] VOURVOULAKIS J, KALOMIROU J, LYGOURAS J. FPGA accelerator for real-time SIFT matching with RANSAC support [J]. Microprocessors and Microsystems, 2017, 49: 105-116.

[13] QASAIMI M, SAGAHYROON A, SHANABLEH T. FPGA-based parallel hardware architecture for real-time image classification[J]. IEEE Transactions on Computational Imaging, 2015, 1(1): 56-70.

[14] VOURVOULAKIS J, KALOMIROU J, LYGOURAS J. Fully pipelined FPGA-based architecture for real-time SIFT extraction [J]. Microprocessors and Microsystems, 2016, 40: 53-73.

[15] CHIU L C, CHANG T S, CHEN J Y, et al. Fast SIFT design for real-time visual feature extraction[J]. IEEE Transactions on Image Processing, 2013, 22(8): 3158-3167.

作者简介:
姜晓明 男, 硕士研究生。主要研究方向: 图像处理及其 FPGA 硬件加速系统设计。
刘强 男, 博士, 副教授, 博士生导师。主要研究方向: 数字集成电路设计、高速低功耗电路系统设计等。

Low-complexity fast SIFT feature extraction based on FPGA

JIANG Xiaoming^{1,2}, LIU Qiang^{1,2,*}

1. School of Microelectronics, Tianjin University, Tianjin 300072, China;
2. Tianjin Key Laboratory of Imaging and Sensing Microelectronic Technology, Tianjin 300072, China)

Abstract: Scale invariant feature transform (SIFT) algorithm is widely used in the field of computer vision because of its excellent robustness. In order to solve the problem of low real-time performance of computation-intensive SIFT algorithm on CPU, a fast SIFT hardware architecture is proposed based on field programmable gate array (FPGA), with reduced complexity by optimizing the feature descriptor extraction part of the algorithm. By reducing the bit width of gradient information (including gradient amplitude and gradient direction), optimizing the generation of the Gauss weight coefficients, simplifying the calculation of the three linear interpolation coefficients and simplifying the computation process of the histogram index of the gradient amplitude, the proposed design avoids complex computations such as exponent, trigonometric function and multiplication, and reduces the complexity of hardware architecture and hardware resource consumption. The experimental results show that the proposed low-complexity fast SIFT hardware architecture can speed up by about 200 times compared to the software implementation. Compared with the related research, the speed is improved by 3 times and the stability of the feature descriptor is increased by more than 18%.

Keywords: field programmable gate array (FPGA); scale invariant feature transform (SIFT); hardware design; gradient information; feature descriptor extraction