

文章编号 1004-924X(2019)01-0211-10

光流法运动估计在 FPGA 上的实现与性能分析

王向军^{1,2*}, 张继龙^{1,2}, 阴 雷^{1,2}

- (1. 天津大学 精密测试技术及仪器国家重点实验室, 天津 300072;
2. 天津大学 微光机电系统技术教育部重点实验室, 天津 300072)

摘要: 图像序列的光流估计理论在机器视觉领域已被提出多年, 但算法的高计算复杂度限制了其在工业领域的应用。为了满足应用的实时性要求, 阐述了一种光流实时估计的实现方法。为了提高算法精度及减少 FPGA 片内资源消耗, 对 L&K 光流计算方法进行改进。首先, 通过设计两层光流计算架构来提高精度。针对在此过程中出现的外部存储器读写速率不够的问题, 提出一次读取同时分层缓存、分时计算的方法。考虑到两层光流在计算过程中的迭代关联性, 设计了满足要求的外部存储器数据读出顺序表; 然后, 针对卷积运算资源消耗大的问题, 设计了新的卷积权重函数, 能够将卷积计算量降低 73%, 从而节省了大量逻辑资源; 最后通过实验验证, 所提出的 FPGA 光流计算方法的精度高于运行在 PC 平台的 L&K 方法, 卷积计算资源消耗明显降低。设计的系统可以完成 1280×1024 pixel、60 frame/s 输入视频的计算, 满足光流计算的实时性要求。

关键词: 两层光流; 改进 L&K 算法; 实时计算; 现场可编程阵列

中图分类号: TP391.4; TN79.1 **文献标识码:** A **doi:** 10.3788/OPE.20192701.0211

Implementation and performance analysis of optical flow based motion estimation on FPGA

WANG Xiang-jun^{1,2*}, ZHANG Ji-long^{1,2}, YIN Lei^{1,2}

- (1. State Key Laboratory of Precision Measuring Technology and Instruments, Tianjin University, Tianjin 300072, China;
2. Key Laboratory of MOEMS of the Ministry of Education, Tianjin University, Tianjin 300072, China)

* Corresponding author, E-mail: xdocuxjw@vip.163.com

Abstract: Optical flow estimation theory has been proposed in the field of machine vision for many years, but the high computational complexity of the algorithm limits its application in the industrial field. To meet the real-time requirements, a method was realized by FPGA in this study. To improve the accuracy of the algorithm and reduce the consumption of hardware resources, the Lucas and Kanade optical flow calculation method was improved. First, a two-level optical flow computation framework was designed to improve the accuracy. To address the problem of insufficient read-write rate of the external memory, it was proposed that when the image was read, it was sampled and cached to two spaces at the same time for subsequent computation. Considering the iterative correlation between

收稿日期: 2018-06-28; 修订日期: 2018-08-29.

基金项目: 国家自然科学基金面上项目资助 (No. 51575388)

the two levels of optical flow, we designed the data readout order to be stored in external memory. Then, in this study, a new convolution weight function was designed to reduce the volume of convolution, which was reduced by 73%, thus saving a lot of hardware resources. Experimental results indicate that the accuracy of the hardware implementation is higher than that of the Lucas and Kanade method on a PC, and the convolution computation is significantly reduced. This system fulfills the specified real time constrains of 60 images per second with $1\ 280 \times 1\ 024$ image resolution.

Key words: two levels optical flow; improved L&K algorithm; real time; Field-Programmable Gate Array(FPGA)

1 引言

求取图像稠密速度场的光流法^[1]可以分为基于梯度、基于相关性、基于能量和基于相位的 4 类方法。其常应用于三维重构^[2]、运动目标检测^[3]、视频压缩^[4]及模式识别^[5]等领域。近年来光流算法得到了很大改进,但在 PC 上的运行性能仍然达不到实时应用的要求。一些学者已经对不同的光流方法进行了对比研究^[6]。通过对比结果,基于梯度的 Lucas & Kanade(L&K)光流计算方法^[7]在精度和效率方面表现的较为理想,相对适合应用于实时系统。另一方面,对于许多计算机视觉算法,FPGA 上的并行化和流水线化的实现可以显著提高实时计算性能^[8]。

部分学者对光流计算方法的 FPGA 上实现进行了研究,文献[9]使用 XilinxXC2V1000-4FPGA 平台对 Lucas & Kanade 算法进行设计。该设计在 40 MHz 的时钟下可以完成 340×280 像素,30 frame/s 的图像处理。实现后平均角度误差(AAE)增大 2.39° 。文献[10]在 FPGA Xilinx Virtex II XUPV2P FPGA 实验板上同样的实现了该光流计算方法。所提出的架构能够以每帧 8 ms 的计算速度处理 Yosemite 图像序列(316×256 pixel)。但并未给出实现后的精度分析。文献[11]中提出了一种基于张量的光流计算方法。在 100 MHz 的时钟工作频率下,可以处理 60 frame/s 的 640×480 pixel 图像。该设计实验平台与文献[10]选择的 Xilinx XUPV2P 实验板相同,计算 Yosemite 图像序列光流场的总平均角误差(AEE)为 12.9° 。另一种著名的基于梯度的光流法由 Horn 和 Schunck^[12]提出,文献[13]使用 AlteraFPGA(20K100)开发平台实现 50×50 pixel,19 frame/s 的图像光流计算。文献[14]在 EP20K300EQC240-2 开发平台上实现了 $256 \times$

256 pixel 60 frame/s 图像光流计算的设计。但没有给出所设计系统的光流计算精度以及对比分析。

以上文献都较为侧重光流法的 FPGA 实现,并没有注重提高实现后精度;并且没有解决权重函数在卷积运算过程中带来的高复杂度计算问题。为此,本文从算法实现后的精度和降低卷积运算资源消耗两方面出发。通过在 FPGA 上实现两层光流架构设计来提高算法实现后的精度。与其他分层设计^[15]不同,本文没有采用缓存图像金字塔的方法,而是在原始图像数据上进行一次读取并同时计算两层光流的设计,以此来降低外部存储器的数据传输压力。针对卷积运算资源消耗大的问题,本文设计了新的权重函数。通过理论推导验证,运用所设计的权重函数,只需要 4 次加法计算就可以实现卷积运算,很大程度降低资源消耗。最后,本文给出了设计的精度、计算速率和资源利用分析。

2 Lucas & Kanade 光流算法

2.1 L&K 光流法基本原理

一个图像序列 $g(x)$ 可以用一个三维的列向量 $\mathbf{x}=(x, y, t)^T$ 来表示,其中 x 和 y 是空间分量, t 是时间分量。根据亮度恒定约束,时空域的物体运动将产生具有一定方向的亮度模型。根据亮度恒定假设有:

$$I(x, y, t) = I'(x + dx, y + dy, t + dt), \quad (1)$$

式中: I 和 I' 表示图像相邻帧, dx 和 dy 表示像素在 dt 时间内沿 x, y 方向的位移增量。根据小运动模型假设,将(1)式 Taylor 展开后舍去高阶项,可得光流计算模型方程为:

$$\mathbf{I}_x \mathbf{V}_x + \mathbf{I}_y \mathbf{V}_y + \mathbf{I}_t = 0, \quad (2)$$

式中: $\mathbf{V}_x, \mathbf{V}_y$ 分别是光流向量在水平和垂直方向的分量, $\mathbf{I}_x, \mathbf{I}_y$ 和 \mathbf{I}_t 分别表示图像的空间和时间维

度的梯度信息。

在三维世界中,如果属于相同物体平面的点具有相同速度,则投影到二维平面对应的点也具有相同的速度。将这一假设表示为:

$$I_x(i)V_x + I_y(i)V_y + I_t(i) = 0, i = (1, 2, \dots, n). \quad (3)$$

将上述约束方程利用最小二乘法求解可得光流的计算公式:

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i \omega I_x^2(i) & \sum_i \omega I_x(i) I_y(i) \\ \sum_i \omega I_x(i) I_y(i) & \sum_i \omega I_y^2(i) \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i \omega I_x(i) I_t(i) \\ -\sum_i \omega I_y(i) I_t(i) \end{bmatrix}, \quad (4)$$

式中 ω 是权重窗函数。

$$AA_{xx} = \sum_i \omega I_x^2(i),$$

$$AA_{yy} = \sum_i \omega I_y^2(i),$$

$$AA_{xy} = \sum_i \omega I_x(i) I_y(i),$$

$$AB_{xt} = \sum_i \omega I_x(i) I_t(i),$$

$$AB_{yt} = \sum_i \omega I_y(i) I_t(i),$$

展开整理可得:

$$\begin{cases} V_x = \frac{AA_{yy} \times AB_{xt} - AA_{xy} \times AB_{yt}}{AA_{xx} \times AA_{yy} - AA_{xy}^2} \\ V_y = \frac{AA_{xx} \times AB_{yt} - AA_{xy} \times AB_{xt}}{AA_{xx} \times AA_{yy} - AA_{xy}^2} \end{cases} \quad (5)$$

2.2 FPGA 光流算法优化

FPGA 的并行处理特点及流水线方法的应用增加了算法的执行效率,使其广泛应用于实时性要求较强的场合。然而,若在 FPGA 上直接对 L&K 算法进行实现则存在两个问题。第一, L&K 算法是一种基于梯度的光流估计方法,通过邻域加权窗内像素对该点进行光流估计,其中窗函数的卷积操作具有较高的计算复杂度,这将导致在实现过程中 FPGA 资源消耗严重。第二, FPGA 数据以二进制形式由独立的寄存器单元存储,在进行数学运算时常通过扩大操作数倍数的方法转换为定点数运算,这也导致算法在 FPGA

上实现后精度的下降。为此,本文通过设计权重函数来降低卷积运算的资源消耗,并设计了两层光流计算架构来提高算法精度,使得算法更加适合于 FPGA 上实现。

2.2.1 光流法权重函数的改进

由式(4)可知,对输入图像 I 求取各方向梯度后,需要对 I_x^2 和 I_y^2 等进行窗函数卷积运算。其中窗函数 ω 起到改变像素权重作用,使得卷积窗中心数据所占比重大,边缘数据占比逐渐减小。窗函数 ω 常选用高斯函数,采用先对列卷积再对行卷积的方式可以将二维图像卷积通过一维卷积实现。由于高斯函数的对称性,在一维行或列图像上,半径为 r 的卷积所需资源 O_1 为:

$$O_1 = (2r - 1) \text{ 次加法} + \frac{2r - 1}{2} \text{ 次乘法}. \quad (6)$$

随着卷积窗半径 r 的增大,所需资源 O_1 将线性增加。考虑到卷积核 ω 是一个权重函数,用来使窗中心数据所占权重增大。为此,本文提出新的权重函数,函数方程为:

$$\omega = \begin{cases} kx, & 0 < x \leq 8, \\ -kx, & 8 < x \leq 15 \end{cases}, \quad (7)$$

式中: x 为权重函数的水平坐标, k 为权重函数斜率。下图 1 是需要进行卷积运算的图像部分 I_c 和本文设计的权重函数 ω , 卷积半径为 7。

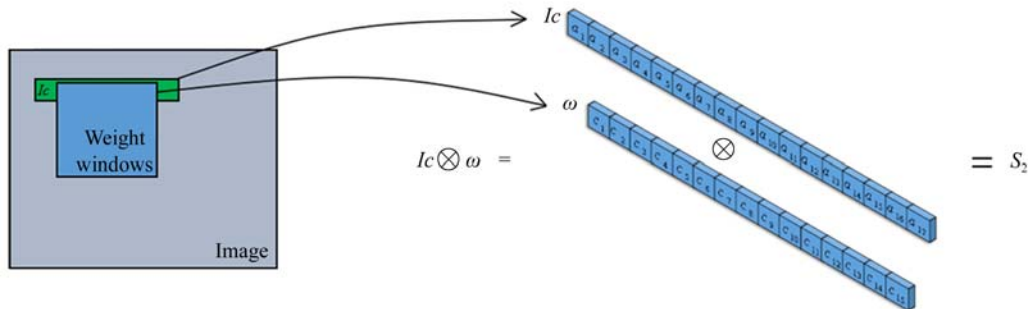


图 1 一维卷积示意图

Fig. 1 Diagram of one-dimensional convolution

令卷积运算结果为 S , 卷积运算结果的差为 Δ , 则有:

$$S_2 = a_2 c_1 + a_3 c_2 + \dots + a_{15} c_{14} + a_{16} c_{15}, \quad (8)$$

$$S_3 = a_3 c_1 + a_4 c_2 + \dots + a_{16} c_{14} + a_{17} c_{15}, \quad (9)$$

$$\Delta_2 = S_3 - S_2 = -a_2 c_1 + a_3 (c_1 - c_2) + \dots + a_{16} (c_{14} - c_{15}) + a_{17} c_{15}, \quad (10)$$

式中: c_i 为窗函数权重系数, a_i 为窗函数对应的卷积区域的像素。

同理:

$$\Delta_1 = -a_1 c_1 + a_2 (c_1 - c_2) + \dots + a_{15} (c_{14} - c_{15}) + a_{16} c_{15}. \quad (11)$$

由于权重函数正梯度为 k , 则:

$$c_{i+1} - c_i = \begin{cases} k, & (i = 8, 9, \dots, 14) \\ -k, & (i = 1, 2, \dots, 7) \end{cases} \quad (12)$$

$$c_1 = c_{15} = k.$$

则 $\Delta_2 - \Delta_1 = ka_1 - 2ka_9 + ka_{17}$, 所以可得权重函数卷积运算表达式:

$$\begin{cases} S_{i+1} = S_i + \Delta_i \\ \Delta_i = \Delta_{i-1} + ka_1 - 2ka_9 + ka_{17} \end{cases} \quad (i = 1, 2, \dots, n), \quad (13)$$

由式(13)可知, 改进后的卷积运算计算量明显减少, 所需资源 O_2 为:

$$O_2 = 4 \text{ 次加法} + 3 \text{ 次乘法}. \quad (14)$$

由式(14)可知, 本文所设计的权重函数进行卷积计算时, 运算量与卷积窗半径无关, 特别地, 当设计的权重函数正梯度 $k=1$ 时, 卷积操作只需要 4 次加法运算。对于 k 值的选取可依照新的权重函数与高斯权重函数尽可能的近似的原则。在光流计算过程中, 高斯权重函数的标准差 σ 的取值是一个参考“ 3σ 原则”的经验值。此外 k 值的选取还要考虑存储资源的占用。设计的权重函数与高斯权重函数越接近, 求取的光流值与用高斯权重函数求取的光流值越接近。

2.2.2 两层光流计算架构设计

与其他光流计算有所不同^[9-11], 本文设计了两层光流计算架构用于提高光流计算的精度, 并且没有将图像降采样为多层金字塔后存储到 SDRAM 中。这种结构方式对存储器的数据传输速率要求很高, 很难达到算法对数据吞吐量的要

求; 此外, 还需要对求解的第一层光流进行暂存, 需要大量的存储资源。为此, 本文提出一种两层光流同时计算的方法。

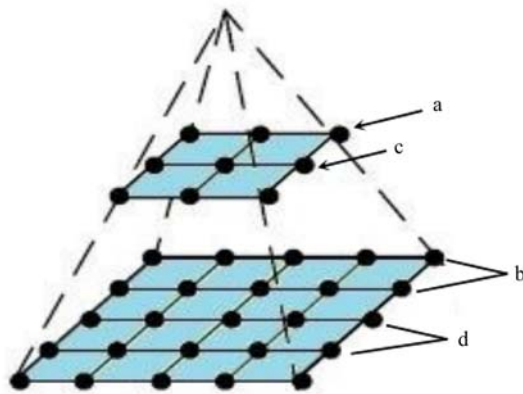


图 2 两层光流计算顺序示意图

Fig. 2 Calculation sequence of two layer optical flow

计算架构设计思想: 通过顶层图像与底层图像对应行像素的交替计算, 取代先计算顶层图像所有像素光流再求取底层图像光流的计算结构。先计算图像金字塔顶层图像第 1 行像素光流值(如图 2 中 a 处), 再计算底层图像第 1, 2 行像素光流(如图 2 中 b 处); 重复此计算过程(如图 2 中 c, d 指示处)。

两层光流计算架构如图 3 所示, 整个架构主要由外部存储器 DDR2、内部 RAM 及光流计算单元组成。为了节省 FPGA 资源并保证算法具有较高精度, 本文对 1/2 层和 1/4 层两层光流的计算架构进行设计。原始图像从 DDR2 存储器中读出, 将其降采样成 1/4 层和 1/2 层图像并缓存到 RAM1 和 RAM2, 通过对原始图像一次读取并同时采样计算光流的方法以降低 DDR2 的吞吐压力。同样为了平衡资源消耗和算法精度, 本文设计的卷积窗尺度为 15×15 。为此, RAM1 需要缓存 1/4 层图像 17 行; 相邻两帧图像最大移动量不超过 6 个像素时满足光流计算理论的小运动模型假设。由于光流值表示的是相邻两帧图像的像素移动量, 1/4 层光流值与降采样系数的积作为求解 1/2 层光流时的像素初始偏移量。为满足索引偏移不超出范围, 则此时 RAM2 需要缓存 1/2 层图像为 23 行。缓存的图

像通过选择器进入到计算单元进行分时计算。将 1/4 层的计算结果缓存,作为下一层图像的索引偏移量。

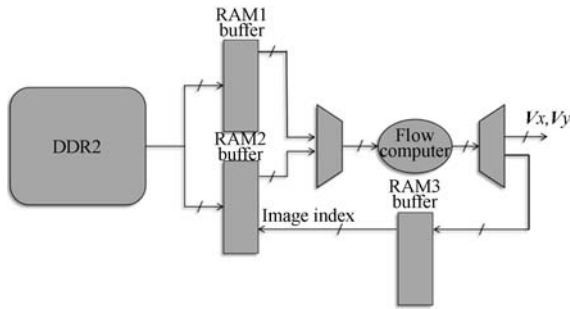


图 3 两层光流计算架构示意图

Fig. 3 Architecture of two layer optical flow

计算架构实现过程为:

- (1) 图像按行格式输入,缓存 17 行 1/4 层图像到 RAM1,缓存 23 行 1/2 层图像到 RAM2。
- (2) 缓存的 1/4 层图像通过选择器进入光流计算单元求取光流矢量值。
- (3) 求解的 1/4 层光流缓存到光流缓存区 RAM3 中。

(4) 1/2 层图像通过选择器进入光流计算单元求取,此时 RAM3 作为 RAM2 的读地址偏移量。需要指出,在计算 1/2 层光流时,1/4 层一行光流索引 1/2 层两行像素。

此外,DDR2 不仅需要将图像缓存到 RAM1 和 RAM2 中,而且要兼顾图像的 VGA 显示。为此,需要在满足 DDR2 吞吐量前提下,对 DDR2 内图像读出顺序进行设计。图像按行读出顺序如表 1,其中 I_1, I_2 表示原始图像相邻两帧, $I^{1/4-1}, I^{1/4-2}$ 表示 1/4 层相邻两帧图像, $I^{1/2-1}, I^{1/2-2}$ 表示 1/2 层相邻两帧图像。

表中 3~6 行表示 1/4 层相邻两帧和 1/2 层相邻两帧图像各行的读出顺序,对应原始图像相邻两帧读出顺序如 1,2 行所示,其中加黑部分对应图像用于 VGA 显示。例如表中第 5 列, I_1 和 I_2 分别对应序列号 1 和 33,表示此时从 DDR2 中读出 I_1 的第 1 行和 I_2 的第 33 行。 $I^{1/2-1}$ 和 $I^{1/4-2}$ 分别对应序列号 1 和 9,表示此时 I_1 第 1 行经 1/2 降采样后得到 1/2 层图像 $I^{1/2-1}$ 的第 1 行, I_2 第 33 行经 1/4 降采样后得到 1/4 层图像 $I^{1/4-2}$ 的第 9 行,同时 VGA 显示 I_1 的第 1 行。

表 1 图像读出行顺序表

Tab. 1 Row order of output images

图像	图像行顺序号																					
I_1	1,5,9~29	33	-	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
I_2	1,5,9~29		1,3,5,7	33	9	37	11	37	13	41	15	41	17	45	19	45	21	49	23	49	25	53
$I^{1/4-1}$	1~8	9	-	-	-	10	-	-	-	11	-	-	-	12	-	-	-	13	-	-	-	14
$I^{1/4-2}$	1~8	-	-	9	-	-	-	10	-	-	-	11	-	-	-	12	-	-	-	13	-	-
$I^{1/2-1}$	-	-	-	1	-	2	-	3	-	4	-	5	-	6	-	7	-	8	-	9	-	10
$I^{1/2-2}$	-	-	1~4	-	5	-	6	-	7	-	8	-	9	-	10	-	11	-	12	-	13	-

3 FPGA 上光流处理的实现方法

3.1 整体架构设计

图 4 为设计的整体架构框图,各模块详细描述如下:

(1) 图像缓存到 DDR2: 首先,连续图像从图像传感器输入到 FPGA,通过 DDR2 控制器将图

像存储到外部存储器 DDR2 中。

(2) 图像从 DDR2 读出:DDR2 控制器按表 1 数据传输格式从外部存储器中读取数据并传输给 VGA 显示模块、1/4 层缓存 RAM1 和 1/2 层缓存 RAM2。

(3) 计算 1/4 层光流:主控制器 Master Controller 控制 RAM1 输出 1/4 层像素,通过选择器后进入光流计算单元 Flow computer。

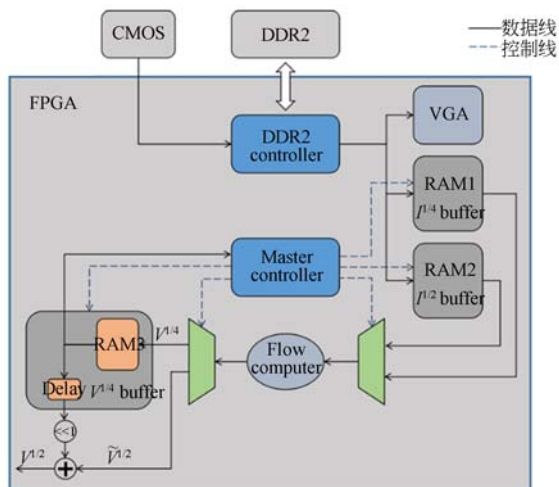


图 4 整体架构框图

Fig. 4 Block diagram of overall framework

(4)缓存 1/4 层光流:光流计算单元对输入的 1/4 层像素求取光流,将计算结果经选择器后存储到 1/4 层光流缓存区 RAM3。

(5)计算 1/2 层光流:主控制器 Master Controller 从 1/4 层光流缓存区 RAM3 读出光流,作为 1/2 层图像像素索引偏移量。控制 1/2 层像素缓存区 RAM2 输出数据,进入光流计算单元求取光流,经选择器后输出。

(6)光流输出:经延迟的 1/4 层光流 $V^{1/4}$ 进行左移 1 位运算,与经索引后求取的 1/2 层光流 $\tilde{V}^{1/2}$ 相加得到 1/2 层光流 $V^{1/2}$ 。

3.2 光流计算单元设计

光流计算单元由梯度单元、乘积单元、卷积单元和最小二乘单元 4 部分组成,设计的架构如图 5 所示。相邻两帧图像 I_1 和 I_2 进入梯度单元求取图像各方向梯度,计算结果经乘积单元运算后进入卷积单元。最后,卷积结果经最小二乘单元计算光流。光流计算单元本质是公式(6)的 FPGA 实现,具体细节实现如下。

3.2.1 梯度求取

像素梯度求取如图 6(a)所示,其中包括像素的水平梯度,竖直梯度和时间方向上的像素梯度。像素 P_i 点水平方向梯度求取如式(15):

$$I_x^i = \frac{P_{i+1}}{2} - \frac{P_{i-1}}{2}, \quad (15)$$

其中像素 P_{i+1} 和 P_{i-1} 为正值,可以用无符号数表

示。运算中的除法操作可以通过移位运算来替换,FPGA 上实现如图 6(b)所示。

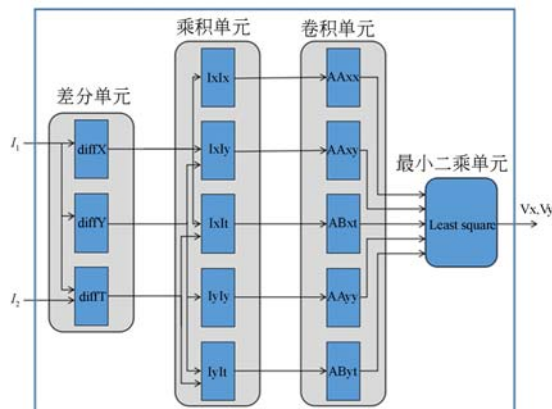
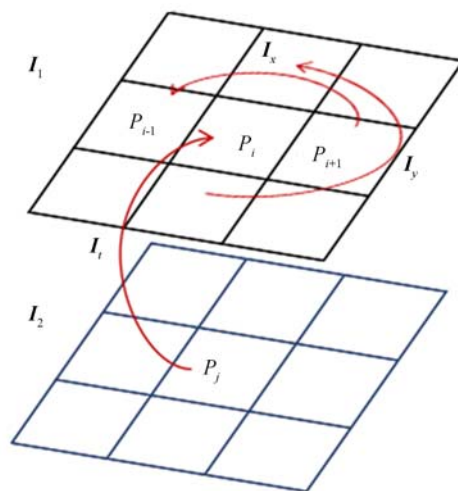


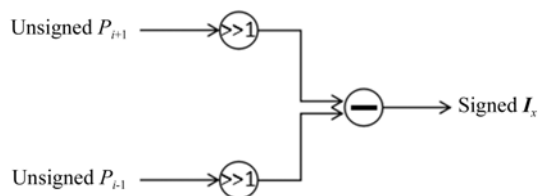
图 5 光流计算单元示意图

Fig. 5 Diagram of optical flow calculation unit



(a) 梯度求取示意图

(a) Diagram of calculating gradient



(b) 梯度计算的逻辑实现

(b) Diagram of calculating gradient on FPGA

图 6 梯度求取

Fig. 6 Calculation of gradient

3.2.2 卷积运算流水线实现

为了降低卷积运算的资源消耗,本文设计了新的权重函数如式(7),取系数 $k=1$ 。图 7 为此运算 FPGA 实现示意图,通过流水线操作可以实时完成卷积计算。

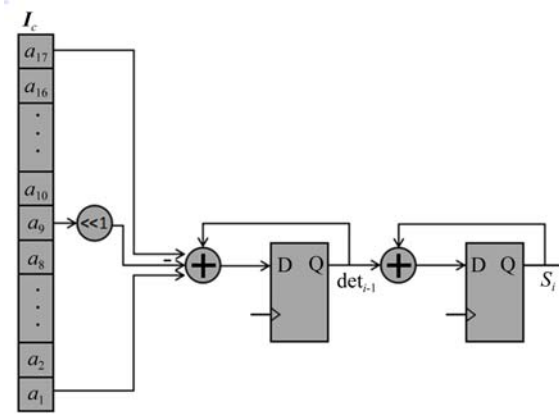


图 7 卷积运算示意图

Fig. 7 Diagram of convolution operation

3.2.3 最小二乘运算并行化实现

本文通过将整数扩大来实现定点数运算。为了满足十进制下小数点后两位精度,移位宽度选择 8,两个维度光流数据 V_x, V_y 分别由高 8 位整数部分和低 8 位小数部分组成。此运算对应图 5 中最小二乘单元,其 FPGA 实现如图 8。

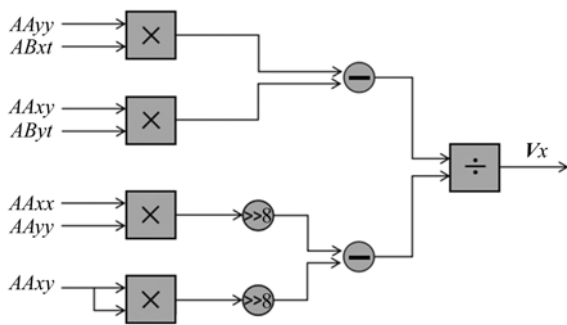


图 8 最小二乘逻辑图

Fig. 8 Diagram of least squares operation

4 实验结果与分析

4.1 实验环境和测试数据

本文算法是在一块载有 Alter Cyclone IV E

系列 EP4CE40F23C7-FPGA 芯片的实验板上实现的。所选 FPGA 包含 39 600 个逻辑单元, 1 134 KB RAM 存储资源和 232 个 9-bit 嵌入式乘法器。详细的资源利用见表 3。此外,设计工具选择的是 Quartus II 13.0,仿真工具是 Modelsim SE 10.0,以及设计语言使用 Verilog HDL。为了与实验结果对比,PC 上使用 Matlab-R2014b 以及 Piotr's Computer Vision Matlab Toolbox 工具。光流真值已知的测试图像已在很多文献中用于测试算法的精度^[6]。本文选取四组光流真值已知测试图像^[16],通过文献[6]的评价方法对算法精度进行评估。

4.2 实验结果和精度分析

为了评估本文提出的光流计算方法在 FPGA 上的实现性能,本文进行了对比试验。将测试图像传输到 FPGA 架构计算光流值,并在 PC 上通过 Matlab-R2014b 以及 Piotr's Computer Vision Matlab Toolbox 对相同测试图像进行了 L&K 光流计算实验,图 9 展示了所有实验结果。其中第 1 列为测试图像,第 2 列为真实光流图像,第 3 列为本文提出的 FPGA 架构计算光流结果,第 4 列为传统 L&K 方法 PC 计算实验结果。

对比图 9 中第 2 列与第 3 列、第 2 列与第 4 列,可初步判断本文提出的光流计算方法在精度上优于传统 LK 方法。为进一步对精度进行分析及对比,本文采用文献[6]评估方法对平均角度误差(AAE)和平均终点误差(AEE)进行计算分析,计算结果如表 2。从表 2 可以看出,本文提出的 FPGA 光流计算架构在测试数据集上的 AAE 和 AEE 都低于 PC 端的 L&K 计算方法。特别是对于 Grove2 测试集,其计算精度得到了很大提高。对该测试集进一步分析,Grove2 图像序列在 FPGA 上进行单独 1/2 层光流计算精度为 $AAE=30.07^\circ$, $AEE=1.84$ pixel。计算单层光流得到的精度略低于 PC 端的 L&K 方法。这主要是 FPGA 内资源有限,数据位宽不足引起的精度下降。但该测试集运行在两层光流计算架构的 FPGA 上时,AAE 和 AEE 分别为 9.88° 和 0.72 pixel,精度得到了很大提高。这得益于本文的两层光流计算架构设计。

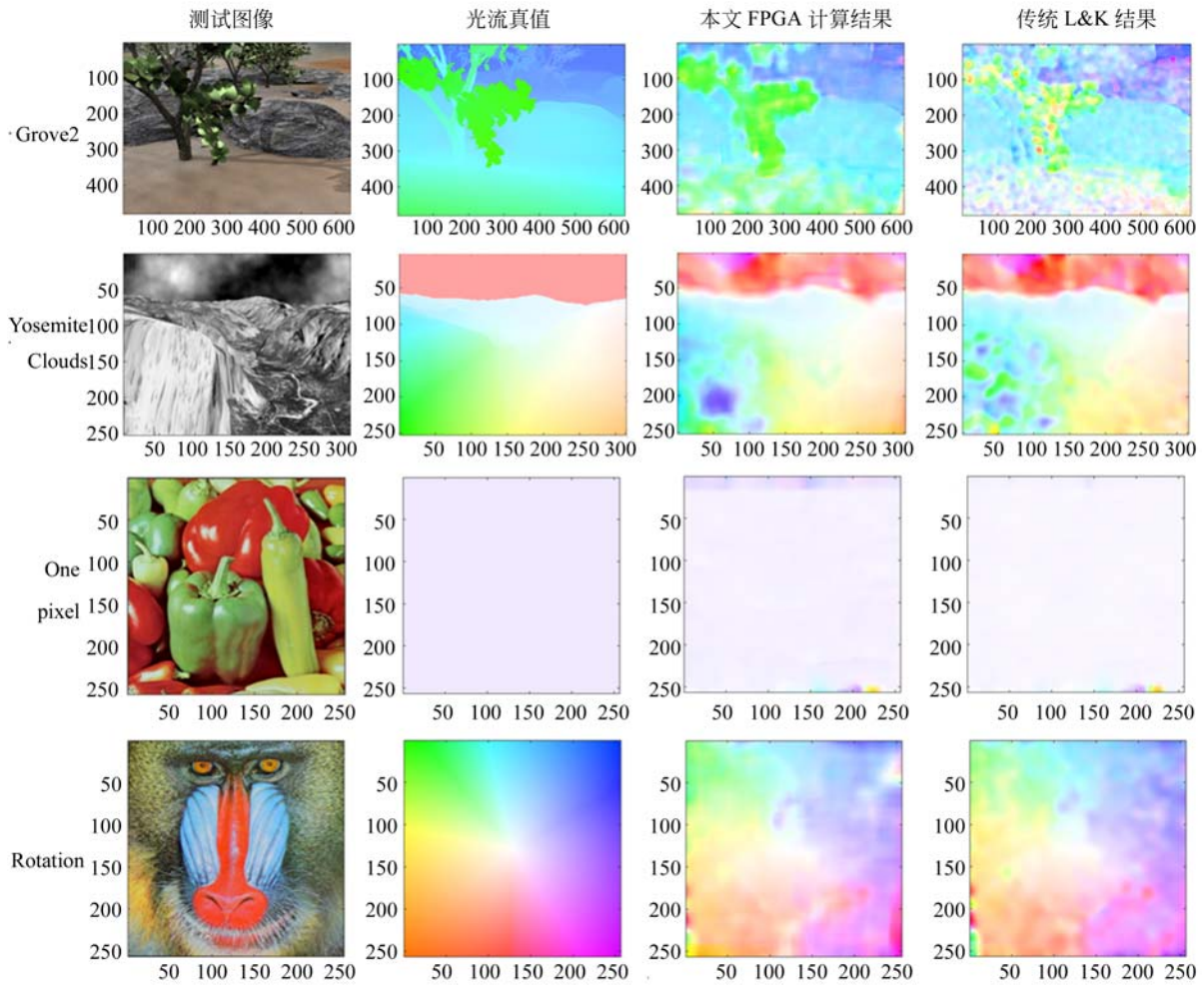


图 9 实验结果对比

Fig. 9 Comparison of experimental results

表 2 计算精度对比

Tab. 2 Comparison of calculation precision

测试图像 序列	AAE/(°)		AAE/pixel	
	FPGA	PC	FPGA	PC
Grove2	9.878 9	29.37	0.7215	1.83
Yosemite clouds	15.442 5	16.409 6	0.810 7	0.862 7
One pixel	6.908 6	7.283 2	0.276 4	0.305 3
Rotation	9.364 6	9.916 0	0.307 7	0.310 1

4.3 计算速度与资源利用分析

在吞吐量计算方面,本文提出的 FPGA 光流计算架构可以实现至少 60 frame/s 的处理速度,满足指定的实时处理显示要求。表 3 列出了本文方法的处理性能并与其他类似系统进行了比较。Diaz. 等人采用 L&K 算法^[10-11,17], Cobos. 等人以

及 Aitzol. 等人采用 Horn Schunck 算法 (H&S)^[13-14]。表中所列的文献多偏向于光流计算方法在 FPGA 平台上的实现,没有体现算法在

表 3 计算精度对比

Tab. 3 Comparison of computing speed

实现方法	吞吐量 /(Kpixel · s ⁻¹)	图像尺寸 /pixel	图像帧率 /(frame · s ⁻¹)
L&K(here described)	76 800	1 280×1 024	60
L&K(Pyda et al. [10])	9 875	316×256	125
L&K(Wei et al. [11])	18 000	640×480	60
L&K(Diaz et al. [17])	1 776	320×240	24
H&S(Aitzol et al. [14])	3 840	256×256	60
H&S(Cobos et al. [13])	46	50×50	19

FPGA 特有的运算环境下的改进。与之相比,本文方法对图像从外部存储器读入顺序进行优化设计(见表 1),提高了时钟利用率,使得计算速率增加。此外高吞吐量还得益于本文计算架构的深流水线设计。

在资源利用方面,本文方法是以模块化的方式设计。表 4 总结了使用 EP4CE40f23c7 FPGA 实现各模块设计的资源利用情况。

表 4 各模块资源利用

Tab. 4 Summary of FPGA resource utilization

	LE 使用 数量	M9K 使用 数量	9-bit 乘法器 使用数量
图像缓存模块	1 559/(3.9%)	74/(58.7%)	0
梯度模块	812/(2.1%)	0	0
乘积模块	2/(~)	0	75/(32.3%)
卷积模块	4 318/(10.9%)	0	0
最小二乘模块	11 371/(28.7%)	0	48/(20.7%)
光流缓存模块	32/(0.1%)	10/(7.9%)	0
DDR2 控制模块	5 031/(15.2%)	9/(7.1%)	0
主控制模块	137/(0.3%)	0	0

从表 4 分析可知,LE 资源使用比重最大的是最小二乘模块。从公式(5)可知求解最小二乘过程需要一次除法运算,没有嵌入式除法器的 FP-

GA 会占用大量 LE 资源来完成这一过程。改进后的卷积运算模块占用 LE 资源 10.9%,且这一模块是由 5 个相同的卷积运算单元组成(见图 5),则每个卷积单元占用 LE 资源不超过 2.2%,由此表明本文改进的卷积运算方法在节省 FPGA 内部资源上效果明显。由于 L&K 光流是一种基于最小二乘的估计算法,需要对估计过程中所用到的像素缓存,因此 M9K 主要用于图像及计算结果的缓存过程。乘法器用于差分数据相乘运算和最小二乘计算。各部分资源都有侧重利用,片上资源利用均衡。

5 结 论

本文描述了基于 FPGA 架构的改进 L&K 光流计算方法实现过程。该方法与传统的 L&K 光流计算方法不同^[17]。通过对图像各行的输入顺序和光流索引下层像素的控制,实现两层光流同时计算的架构,提高算法计算精度。算法在精度上表现高于 L&K 在 PC 端表现。通过设计权重函数,行数据一次卷积操作只需 4 次加法运算,与卷积窗半径无关,很大程度减少资源消耗。FPGA 的并行性以及灵活的流水线设计使得算法在运算速度上表现优秀。本文方法可以实现图像尺寸 1 280×1 024、帧率 60 frame/s 光流计算,满足大尺寸图像光流计算的实时性要求。

参考文献:

- [1] 孙辉,赵红颖,熊经武,等. 基于光流模型的图像运动估计方法[J]. 光学精密工程,2002,10(5): 443-447. SUN H, ZHAO H Y, XIONG J W, et al.. Method of estimating image motion based on the optical flow model[J]. *Opt. Precision Eng.*, 2002, 10(5): 443-447. (in Chinese)
- [2] BAGHAIE A, TAFTI A P, OWEN H A, et al.. Three-dimensional reconstruction of highly complex microscopic samples using scanning electron microscopy and optical flow estimation[J]. *Plos One*, 2017, 12(4): e0175078.
- [3] 刘洪彬,常发亮. 权重系数自适应光流法运动目标检测[J]. 光学精密工程,2016,24(2): 460-468. LIU H B, CHANG F L. Moving object detection by optical flow method based on adaptive weight co-efficient[J]. *Opt. Precision Eng.*, 2016, 24(2): 460-468. (in Chinese)
- [4] CARVALHO B M, SANTOS T S, OLIVEIRA L M. Fuzzy segmentation of video shots using hybrid color spaces and motion information[J]. *Pattern Analysis & Applications*, 2014, 17(2): 249-264.
- [5] 王世刚,鲁奉军,赵文婷,等. 应用在线随机森林投票的动作识别[J]. 光学精密工程,2016,24(8): 2010-2017. WANG SH G, LU F J, ZHAO W T, et al.. Action recognition based on on-line random forest voting[J]. *Opt. Precision Eng.*, 2016, 24(8): 2010-2017. (in Chinese)
- [6] BARRON J L, FLEET D J, BEAUCHEMIN S S. Performance of optical flow techniques[J]. *International Journal of Computer Vision*, 1994, 12(1): 43-77.

- [7] LUCAS B D, KANADE T. An iterative image registration technique with an application to stereo vision[C]. *International Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers Inc. 1981:674-679.
- [8] MACLEAN W J. An evaluation of the suitability of FPGAs for embedded vision systems[J]. *Research Gate*, 2005, 408: 131-131.
- [9] DIAZ J, ROS E, PELAYO F, *et al.*. FPGA-based real-time optical-flow system[J]. *IEEE Transactions on Circuits & Systems for Video Technology*, 2006, 16(2):274-279.
- [10] PYDA B, BRINDHA R. A novel high speed L-K based optical flow computation[C]. *International Conference on Communication and Computational Intelligence*. IEEE, 2011:104-108.
- [11] WEI Z, LEE D J, NELSON B E. FPGA-based real-time optical flow algorithm design and implementation[J]. *Journal of Multimedia*, 2007, 2(5): 38-45.
- [12] BERTHOLD KP. Horn, Brian G. Schunck. Determining optical flow [J]. *Artificial Intelligence*, 1981, 17(1-3):185-203.
- [13] ARRIBAS P C, MACIA F M H. FPGA implementation of Santos-Victor optical flow algorithm for real-time image processing: an useful attempt [J]. *Proceedings of SPIE - The International Society for Optical Engineering*, 2003, 5117(5117):23-32.
- [14] ZULOAGA A, CUADRADO C, BIDARTE U. Hardware implementation of optical flow constraint equation using FPGAs[J]. *Computer Vision & Image Understanding*, 2005, 98(3): 462-490.
- [15] ZHANG K, JIN X, WU A. Accelerating eulerian video magnification using FPGA[C]. *International Conference on Advanced Communication Technology*. IEEE, 2017:554-559.
- [16] Harn-Schunck Optical Flow with a Mutti-Scale Strategy. [OL](2005). <http://demo.ipol.im/demo/20/>.
- [17] DÍAZ J, ROS E, MOTA S, *et al.* Real time optical flow processing system[J]. *Field Programmable Logic & Application*, 2004, 3203:617-626.

作者简介:



王向军(1955—),男,黑龙江哈尔滨人,博士,教授,博士生导师,主要从事精密测试技术与仪器、光电探测与传感技术、影像与视觉测量方面的研究。E-mail: xdocuxjw@vip.163.com



张继龙(1993—),男,黑龙江双鸭山人,硕士研究生,主要研究方向为嵌入式图像处理。E-mail: zjl_tju@163.com