



(12) 发明专利申请

(10) 申请公布号 CN 112861640 A

(43) 申请公布日 2021.05.28

(21) 申请号 202110050957.7

(22) 申请日 2021.01.15

(71) 申请人 复旦大学

地址 200433 上海市杨浦区邯郸路220号

(72) 发明人 韩军 张永亮 李强 张辉

王威振 曾晓洋

(74) 专利代理机构 上海正旦专利代理有限公司

31200

代理人 陆飞 陆尤

(51) Int. Cl.

G06K 9/00 (2006.01)

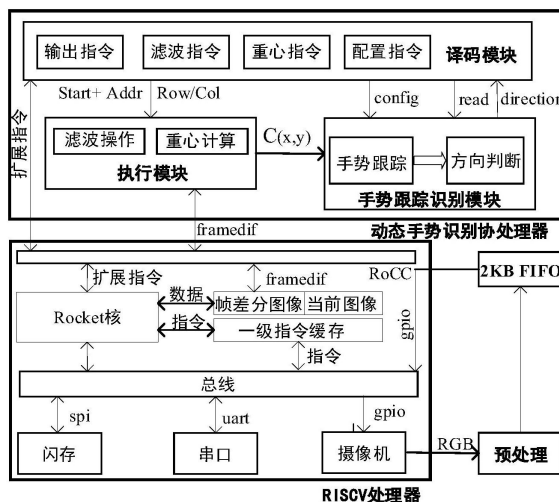
权利要求书3页 说明书6页 附图2页

(54) 发明名称

一种面向智能终端领域的动态手势识别硬件加速器

(57) 摘要

本发明属于集成电路技术领域,具体为一种面向智能终端领域的动态手势识别硬件加速器。本发明电路包括:图像预处理模块,待机模块,RISCV处理器,动态手势识别协处理器;该加速器采用低数学复杂度手势识别算法,将RGB图像转化为YCrCb图像,并根据YCrCb图像提取二值肤色图像,再对二值图像进行帧差分,分离出运动的手势图像。该加速器采用中值滤波电路去除噪声干扰后计算差分图像的手势重心,通过追踪重心的位移得到手势移动方向,具有优越的识别效果,可广泛应用于无接触式智能终端设备;此外该加速器根据图像输入的过程,以及滤波电路的卷积核大小采用逐行代替的帧差分过程,可有效的降低存储,从而降低资源消耗和功耗。



CN 112861640 A

1. 一种面向智能终端领域的动态手势识别硬件加速器,其特征在于,结构包括图像预处理模块、2KB FIFO, RISC V处理器和动态手势识别协处理器;其中:

所述RISC V处理器,由闪存、串口、摄像机、RoCC、总线、rocket核、一级数据缓存、一级指令缓存组成;其中,闪存中保存编译C产生的汇编指令,系统启动后,首先rocket核初始化,然后根据指令不断地访问4KB一级指令缓存,读取下一条指令;然而当一级指令缓存miss后,一级指令缓存会经过spi接口读取闪存保存的汇编代码并反馈给rocket核;rocket核获取指令后,根据指令首先通过gpio端口配置摄像机;其后,rocket核通过RoCC向动态手势识别协处理器发送配置指令,设置参数,参数包括:最小帧数、横向最小位移、纵向最小位移、横向最大位移、纵向最大位移;配置完成后,rocket核会通过总线读取2KB FIFO的empty信号,当empty=0时,此时FIFO中含有数据,rocket核通过gpio和总线读取数据并传输至12KB一级数据缓存,一级数据缓存主要保存5行的帧差分图像和当前图像;每读取一行数据以后,rocket核通过RoCC向动态手势识别协处理器发送执行滤波指令,当完整的一张图像加载完毕后,RISC V处理器通过RoCC向动态手势识别协处理器发送重心计算指令;最后发送输出指令,动态手势识别协处理器根据该指令传输输出方向direction;RISC V处理器接收到direction,将其发送至UART模块进行输出;

所述图像预处理模块,用于接收来自摄像机采集的图像,并将RGB图像转化为YCrCb;然后根据肤色区域在CrCb的分布情况,将YCrCb图像转化为二值肤色图像,二值肤色图像中肤色值为1,非肤色值为0;图像预处理模块发送二值肤色图像至2KB FIFO;

所述2KB FIFO,用于接收并保存图像预处理模块发送的二值肤色图像,并拉低传输至RISC V处理器的empty信号;当2KB FIFO中所有二值肤色图像传输至RISC V处理器,2KB FIFO中的empty信号会拉高;

所述动态手势识别协处理器,包括:译码模块,执行模块和手势跟踪识别模块;其中:

所述译码模块由配置指令寄存器、重心指令寄存器、滤波指令寄存器以及输出指令寄存器构成;

配置指令寄存器接收到RISC V处理器的配置指令,根据其中的序列号将配置参数config发送到手势跟踪识别模块;滤波指令寄存器接收到RISC V处理器的滤波指令,对这些进行译码得到初始地址blur\_addr、blur\_start=1、Row\_addr和Col\_addr并发送至执行模块;重心指令寄存器接收到重心计算指令,对其译码并发送centroid\_start=1至重心计算模块;输出指令寄存器接收到输出指令,对输出指令进行译码并发送读取使能read至手势跟踪识别模块,并在下一周期发送读取的direction至RISC V处理器;

所述执行模块包括滤波操作模块、重心计算模块,用于滤波操作、重心计算;

重心计算模块接收到centroid\_start、sum\_x、sum\_y以及sun\_skin,若centroid\_start为高电平,则重心坐标 $C(x, y)$ 为 $sum\_x/sun\_skin$ 与 $sum\_y/sun\_skin$ ;若centroid\_start为低电平,则重心坐标 $C(x, y)$ 为(0,0);

所述手势跟踪识别模块包括手势跟踪模块和方向判断模块;其中:

所述手势跟踪模块接收到多个轮廓重心值 $C(x, y)$ 和轮廓面积后,取出面积最大的重心坐标作为当前坐标减去上一帧坐标,得到帧间位移 $(\Delta x, \Delta y) = C(x, y) - L(x, y)$ ,并用 $C(x, y)$ 替换 $L(x, y)$ ;

判断 $\Delta x$ 是否处于横向最小位移和横向最大位移以及 $\Delta y$ 是否处于纵向最小位移和纵

向最大位移之间；

(一)若是,则:该重心坐标有效,选出历史位移(dest\_x,dest\_y)中绝对值最大的值进行临界判断:

①若dest\_x绝对值大,判断dest\_x与 $\Delta x$ 是否同号:

同号表示运动方向不变,dest\_x=destx+ $\Delta x$ 与dest\_y=dest\_y+ $\Delta y$ ,累计帧数增加1,并继续下一帧检测;

若不同号,则:

(a)当累计帧数大于等于最小帧数,拉高direct\_en,并将direct\_en与(dest\_x,dest\_y)发送至方向判断模块,下一周期将(dest\_x,dest\_y),累计帧数,direct\_en置0,重新检测;

(b)当累计帧数小于最小帧数,将(dest\_x,dest\_y),累计帧数,direct\_en置0,重新检测;

②若dest\_y绝对值大,判断dest\_y与 $\Delta y$ 是否同号:

同号表示运动方向不变,destx=dest\_x+delt\_x与dest\_y=dest\_y+ $\Delta y$ ,累计检测帧数累计帧数增加1,并继续下一帧检测;

若不同号,则:

(a)当累计帧数大于等于最小帧数,拉高direct\_en并将direct\_en与(dest\_x,dest\_y)发送至方向判断模块,下一周期将(dest\_x,dest\_y),累计帧数,direct\_en均置为0,重新检测;

(b)当累计帧数小于最小帧数,将(dest\_x,dest\_y),累计帧数,direct\_en均置为0,重新检测;

(二)若否,则:

(a)当累计帧数大于等于最小帧数,拉高direct\_en并将direct\_en与(dest\_x,dest\_y)发送至方向判断模块,下一周期将(dest\_x,dest\_y),累计帧数,direct\_en均置为0,重新检测;

(b)当累计帧数小于最小帧数,将(dest\_x,dest\_y),累计帧数,direct\_en均置为0,重新检测;

所述方向判断模块,接收到direct\_en与(dest\_x,dest\_y)信号后,当direct\_en为高电平时,根据(dest\_x,dest\_y)绝对值最大的数的方向进行判断:当dest\_x绝对值大时:若dest\_x为正,则运动方向为向右,direct=1;若为负,则运动方向为左,direct=2;当dest\_y绝对值大时:若dest\_y为正,则运动方向为向下,direct=3;若为负,则运动方向为上,direct=4;当direct\_en为低电平时,direct=0;方向判断模块接收到read使能时,将direct发送至译码模块。

2.根据权利要求1所述的面向智能终端领域的动态手势识别硬件加速器,其特征在于,在图像预处理模块中,所述将RGB图像转化为YCrCb,采用如下颜色域转化公式:

$$Y = (77 * R + 150 * G + 29 * B) \gg 8,$$

$$Cr = (-43 * R - 85 * G + 128 * B) \gg 8 + 128,$$

$$Cb = (128 * R - 107 * G - 21 * B) \gg 8 + 128;$$

所述肤色区域在CrCb的分布情况为:

$skin_{color} = (133 < Cr < 173) \&\& (77 < Cb < 127)$ 。

3. 根据权利要求1所述的面向智能终端领域的动态手势识别硬件加速器,其特征在於,所述执行模块中:

滤波操作模块接收blur\_addr、blur\_start、Row\_addr和Col\_addr后,通过RoCC读取一级数据缓存的帧差分图像,在帧差分图像左右两边添加2列0值;对最左边5\*5的图像块进行求和得到滤波总和,当滤波总和超过12输出为1,否则为0;其后按步长为1不断向右滑动最终得到一行输出;滤波操作对该行输出进行重心累计:统计输出内非0值的横纵坐标总和(sum\_x, sum\_y)以及非0值个数(sum\_skin),其中横纵坐标是由Row\_addr和Col\_addr进行计算;滤波操作模块发送非0值的横纵坐标总和以及非0值个数至重心计算模块;

重心计算模块接收到centroid\_start、sum\_x、sum\_y以及sun\_skin,若centroid\_start为高电平,则重心坐标C(x,y)为sum\_x/sun\_skin与sum\_y/sun\_skin;若centroid\_start为低电平,则重心坐标C(x,y)为(0,0)。

## 一种面向智能终端领域的动态手势识别硬件加速器

### 技术领域

[0001] 本发明属于集成电路技术领域,具体涉及一种面向智能终端领域的动态手势识别硬件加速器。

### 背景技术

[0002] 随着技术的进步,智能终端遍布生活中的方方面面,例如手机、自动售货机、电子手表、IoT产品等。但是公共的智能终端一般使用触控式的方式进行控制,这些智能终端会带来严重的公共卫生安全,很多的传染性疾病是通过间接接触的形式造成的,因而无接触的手势控制是解决智能终端造成的间接接触问题的一种重要的手段。

[0003] 手势识别一般分为静态手势识别和动态手势识别,而对于智能终端实际中最常用的一般为动态手势识别。但是智能终端对于功耗要求极高,在低功耗的基础上识别精度要能达到90%以上的对于设计来说是一种巨大的挑战。现有技术中,动态手势识别一般采用光流法,历史运动图像,GCN,3D-CNN等方法,然而这些方法算法复杂度高,资源消耗高,因而对于功耗的要求也就大幅上升。

### 发明内容

[0004] 为了克服现有技术的不足,本发明提出一种面向智能终端领域的动态手势识别硬件加速器。以期能够实现算法复杂度低,资源消耗少,功耗低,可靠性强的动态手势识别硬件加速器,该加速器可以广泛应用于移动端、IoT、可穿戴设备等智能终端领域。

[0005] 本发明提供的面向智能终端领域的动态手势识别硬件加速器,其结构包括:图像预处理模块,2KB FIFO,RISCV处理器和动态手势识别协处理器;其中:

[0006] 所述RISCV处理器,由闪存、串口、摄像机、RoCC、总线、rocket核、一级数据缓存、一级指令缓存组成;其中,闪存中保存编译C产生的汇编指令,系统启动后,首先rocket核初始化,然后根据指令不断地访问4KB一级指令缓存,读取下一条指令;然而当一级指令缓存miss后,一级指令缓存会经过spi接口读取闪存保存的汇编代码并反馈给rocket核;rocket核获取指令后,根据指令首先通过gpio端口配置摄像机,设置摄像机输出图像大小为320x240,帧频为30帧/秒;其后,rocket核通过RoCC向动态手势识别协处理器发送配置指令,设置参数,包括:最小帧数、横向最小位移、纵向最小位移、横向最大位移、纵向最大位移;配置完成后,rocket核会通过总线读取2KBFIFO的empty信号,当empty=0时,此时FIFO中含有数据,rocket核会通过gpio和总线读取数据并传输至12KB一级数据缓存,一级数据缓存主要保存5行的帧差分图像和当前图像;每读取一行数据以后,rocket核通过RoCC向动态手势识别协处理器发送执行滤波指令,当完整的一张图像加载完毕后,RISCV处理器通过RoCC向动态手势识别协处理器发送重心计算指令;最后发送输出指令,动态手势识别协处理器会根据该指令传输输出方向direction;RISCV处理器接收到direction会将其发送至UART模块进行输出;

[0007] 所述图像预处理模块,用于接收来自摄像机采集的图像,根据颜色域转化公式:

[0008]  $Y = (77 * R + 150 * G + 29 * B) >> 8,$

[0009]  $Cr = (-43 * R - 85 * G + 128 * B) >> 8 + 128,$

[0010]  $Cb = (128 * R - 107 * G - 21 * B) >> 8 + 128,$

[0011] 将RGB图像转化为YCrCb;然后根据肤色区域在CrCb的分布情况:

[0012]  $skin_{color} = (133 < Cr < 173) \&\& (77 < Cb < 127),$

[0013] 将YCrCb图像转化为二值肤色图像,二值肤色图像中肤色值为1,非肤色值为0;图像预处理模块发送二值肤色图像至2KBFI0;

[0014] 所述2KBFI0,用于接收并保存图像预处理模块发送的二值肤色图像,并拉低传输至RISCV处理器的empty信号;当2KBFI0中所有二值肤色图像传输至RISCV处理器,2KBFI0中的empty信号会拉高;

[0015] 所述动态手势识别协处理器,包括:译码模块,执行模块和手势跟踪识别模块;其中:

[0016] 所述译码模块由配置指令寄存器、重心指令寄存器、滤波指令寄存器以及输出指令寄存器构成;

[0017] 配置指令寄存器接收到RISCV处理器的配置指令,根据其中的序列号将配置参数config(包括最小帧数、最大帧数、横向最小位移、纵向最小位移、横向最大位移、纵向最大位移)发送到手势跟踪识别模块;滤波指令寄存器接收到RISCV处理器的滤波指令,对这些进行译码得到初始地址blur\_addr、blur\_start=1、Row\_addr和Col\_addr并发送至执行模块;重心指令寄存器接收到重心计算指令,对其译码并发送centroid\_start=1至重心计算模块;输出指令寄存器接收到输出指令,对输出指令进行译码并发送读取使能read至手势跟踪识别模块,并在下一周期发送读取的direction至RISCV处理器;

[0018] 所述执行模块包括滤波操作模块、重心计算模块;

[0019] 滤波操作模块接收blur\_addr、blur\_start、Row\_addr和Col\_addr后,通过RoCC读取一级数据缓存的帧差分图像,在帧差分图像左右两边添加2列0值;对最左边5\*5的图像块进行求和得到滤波总和,当滤波总和超过12输出为1,否则为0;其后按步长为1不断向右滑动最终得到一行输出;滤波操作对该行输出进行重心累计:统计输出内非0值的纵横坐标总和(sum\_x, sum\_y)以及非0值个数(sum\_skin),其中纵横坐标是由Row\_addr和Col\_addr进行计算;滤波操作模块发送非0值的纵横坐标总和以及非0值个数至重心计算模块;

[0020] 重心计算模块接收到centroid\_start、sum\_x、sum\_y以及sun\_skin,若centroid\_start为高电平,则重心坐标C(x,y)为sum\_x/sun\_skin与sum\_y/sun\_skin;若centroid\_start为低电平,则重心坐标C(x,y)为(0,0);

[0021] 所述手势跟踪识别模块包括手势跟踪模块和方向判断模块;其中:

[0022] 所述手势跟踪模块接收到多个轮廓重心值C(x,y)和轮廓面积后,取出面积最大的重心坐标作为当前坐标减去上一帧坐标,得到帧间位移 $(\Delta x, \Delta y) = C(x,y) - L(x,y)$ ,并用C(x,y)替换L(x,y);

[0023] 判断 $\Delta x$ 是否处于横向最小位移和横向最大位移以及 $\Delta y$ 是否处于纵向最小位移和纵向最大位移之间;

[0024] (一)若是,则:该重心坐标有效,选出历史位移(dest\_x, dest\_y)中绝对值最大的值进行临界判断;

- [0025] ①若 $dest\_x$ 绝对值大,判断 $dest\_x$ 与 $\Delta x$ 是否同号:
- [0026] 同号表示运动方向不变, $dest\_x=dest\_x+\Delta x$ 与 $dest\_y=dest\_y+\Delta y$ ,累计帧数增加1,并继续下一帧检测;
- [0027] 若不同号,则:
- [0028] (a)当累计帧数大于等于最小帧数,拉高 $direct\_en$ ,并将 $direct\_en$ 与 $(dest\_x, dest\_y)$ 发送至方向判断模块,下一周期将 $(dest\_x, dest\_y)$ ,累计帧数, $direct\_en$ 置0,重新检测;
- [0029] (b)当累计帧数小于最小帧数,将 $(dest\_x, dest\_y)$ ,累计帧数, $direct\_en$ 置0,重新检测;
- [0030] ②若 $dest\_y$ 绝对值大,判断 $dest\_y$ 与 $\Delta y$ 是否同号:
- [0031] 同号表示运动方向不变, $dest\_x=dest\_x+\Delta x$ 与 $dest\_y=dest\_y+\Delta y$ ,累计检测帧数累计帧数增加1,并继续下一帧检测;
- [0032] 若不同号,则:
- [0033] (a)当累计帧数大于等于最小帧数,拉高 $direct\_en$ 并将 $direct\_en$ 与 $(dest\_x, dest\_y)$ 发送至方向判断模块,下一周期将 $(dest\_x, dest\_y)$ ,累计帧数, $direct\_en$ 均置为0,重新检测;
- [0034] (b)当累计帧数小于最小帧数,将 $(dest\_x, dest\_y)$ ,累计帧数, $direct\_en$ 均置为0,重新检测;
- [0035] (二)若否,则:
- [0036] (a)当累计帧数大于等于最小帧数,拉高 $direct\_en$ 并将 $direct\_en$ 与 $(dest\_x, dest\_y)$ 发送至方向判断模块,下一周期将 $(dest\_x, dest\_y)$ ,累计帧数, $direct\_en$ 均置为0,重新检测;
- [0037] (b)当累计帧数小于最小帧数,将 $(dest\_x, dest\_y)$ ,累计帧数, $direct\_en$ 均置为0,重新检测;
- [0038] 所述方向判断模块,接收到 $direct\_en$ 与 $(dest\_x, dest\_y)$ 信号后,当 $direct\_en$ 为高电平时,根据 $(dest\_x, dest\_y)$ 绝对值最大的数的方向进行判断:当 $dest\_x$ 绝对值大时:若 $dest\_x$ 为正,则运动方向为向右, $direct=1$ ;若为负,则运动方向为左, $direct=2$ ;当 $dest\_y$ 绝对值大时:若 $dest\_y$ 为正,则运动方向为向下, $direct=3$ ;若为负,则运动方向为上, $direct=4$ ;当 $direct\_en$ 为低电平时, $direct=0$ ;方向判断模块接收到 $read$ 使能时,将 $direct$ 发送至译码模块。
- [0039] 与现有技术相比,本发明的有益技术效果体现在:
- [0040] 1、本发明采用 $5*5$ 的滤波核对帧差分图像进行滤波,结合摄像头的图像发送形式,我们采用5行大小的存储保存帧差分图像,并随着图像行数的增加逐行替换图像,该方法有效的节约了数据的存储大小,从而节约了资源并降低了功耗;
- [0041] 2、本发明采用RISC-V处理器的指令体系,使用两个数组代替差分图像和原始图像,在系统级进行差分计算和替换,有效地提升了计算的灵活性和可靠性,节约了计算资源,提升了芯片的性能。
- [0042] 3、预处理模块将输入的RGB图像转化为YCrCb后再通过肤色提取的方式分离出肤色与背景,将整个图像转化为二值黑白图像表示;该方式能够减少在计算过程中的数据搬

运量,加快计算过程,降低功耗,提升整个芯片的性能。

[0043] 4、协处理器对二值图像进行滤波操作和重心计算将二维图像转化为连续运动点的分析,再通过对历史运动的轨迹进行分析,最后判断出移动方向;识别算法简单快捷,硬件资源消耗小,功耗低,可靠性强,识别精度高。

### 附图说明

[0044] 图1为本发明面向智能终端的动态手势识别硬件加速器系统结构图示。

[0045] 图2为动态手势识别协处理器框图。

[0046] 图3为扩展指令集格式。

[0047] 图4为滤波操作模块。

[0048] 图5为手势跟踪识别电路。

### 具体实施方式

[0049] 本发明设计的动态手势识别硬件加速器,系统结构如图1所示,包括图像预处理模块、2KB FIFO,RISCV处理器和动态手势识别协处理器;其中:

[0050] RISCV处理器,包括:闪存/串口/摄像机/RoCC/总线/rocket核/一级数据缓存/一级指令缓存组成;其中闪存中保存编译C产生的汇编指令,系统启动后首先rocket核初始化以后根据指令不断地访问4KB一级指令缓存,读取下一条指令;然而当一级指令缓存miss后,一级指令缓存会经过spi接口读取闪存保存的汇编代码并反馈给rocket核;Rocket核获取指令后根据指令首先通过gpio端口配置摄像头,设置摄像头输出图像大小为320x240,帧频为30帧/秒;其后,rocket核通过RoCC向动态手势识别协处理器发送配置指令,设置参数:最小帧数/横向最小位移/纵向最小位移/横向最大位移/纵向最大位移;配置完成后,rocket核会通过总线读取2KBFIFO的empty信号,当empty=0时,此时FIFO中含有数据,rocket核会通过gpio和总线读取数据并传输至12KB一级数据缓存,一级数据缓存主要保存5行的帧差分图像和当前图像;每读取一行数据以后,rocket核通过RoCC向动态手势识别协处理器发送执行滤波指令,当完整的一张图像加载完毕后,RISCV处理器通过RoCC向动态手势识别协处理器发送重心计算指令;最后发送输出指令,动态手势识别协处理器会根据该指令传输输出方向direction;RISCV处理器接收到direction会将其发送至UART模块进行输出。

[0051] 图像预处理模块,用于接收来自摄像头采集的图像,根据颜色域转化公式: $Y = (77 * R + 150 * G + 29 * B) >> 8$ ,  $Cr = (-43 * R - 85 * G + 128 * B) >> 8 + 128$ ,  $Cb = (128 * R - 107 * G - 21 * B) >> 8 + 128$ ,将RGB图像转化为YCrCb,其后根据肤色区域在CrCb的分布情况: $skin_{color} = (133 < Cr < 173) \&\& (77 < Cb < 127)$ ,从而将YCrCb图像转化为二值肤色图像,二值肤色图像中肤色值为1,非肤色值为0;图像预处理模块发送二值肤色图像至2KBFIFO。

[0052] 2KBFIFO接收到图像预处理模块发送的二值肤色图像,2KBFIFO会保存二值肤色图像并拉低传输至RISCV处理器的empty信号;当2KBFIFO中所有二值肤色图像传输至RISCV处理器,2KBFIFO中的empty信号会拉高。

[0053] 动态手势识别协处理器,包括:译码模块,执行模块和手势跟踪识别模块,如图2所示。



[0054] 译码模块是由配置指令寄存器、重心指令寄存器、滤波指令寄存器以及输出指令寄存器构成;其中指令为RISCV的扩展指令集,格式如图3所示。

[0055] 配置指令模块接收到RISCV处理器的配置指令,会根据其中的序列号将配置参数config(最小帧数/最大帧数/横向最小位移/纵向最小位移/横向最大位移/纵向最大位移)发送到手势跟踪识别模块;滤波指令寄存器接收到RISCV处理器的滤波指令,对这些进行译码得到初始地址blur\_addr、blur\_start=1、Row\_addr和Col\_addr并发送至执行模块;重心指令寄存器接收到重心计算指令,对其译码并发送centroid\_start=1至重心计算模块;输出指令寄存器接收到输出指令,对输出指令进行译码并发送读取使能read至手势跟踪识别模块,并在下一周期发送读取的direction至RISCV处理器。

[0056] 执行模块包括膨胀操作、重心计算模块,如图4所示。

[0057] 滤波操作接收blur\_addr、blur\_start、Row\_addr和Col\_addr后,通过RoCC读取一级数据缓存的帧差分图像,在帧差分图像左右两边添加2列0值;对最左边5\*5的图像块进行求和得到滤波总和,当滤波总和超过12输出为1,否则为0;其后按步长为1不断向右滑动最终得到一行输出;滤波操作对该行输出进行重心累计:统计输出内非0值的横纵坐标总和(sum\_x,sum\_y)以及非0值个数(sum\_skin),其中横纵坐标是由Row\_addr和Col\_addr进行计算;滤波操作发送非0值的横纵坐标总和以及非0值个数至重心计算模块。

[0058] 重心计算模块接收到centroid\_start、sum\_x、sum\_y以及sun\_skin,若centroid\_start为高电平,则重心坐标C(x,y)为sum\_x/sun\_skin与sum\_y/sun\_skin;若centroid\_start为低电平,则重心坐标C(x,y)为(0,0)。

[0059] 手势跟踪识别模块包括手势跟踪模块和方向判断模块,如图5所示。

[0060] 手势跟踪模块接收到多个轮廓重心值C(x,y)和轮廓面积后,取出面积最大的重心坐标作为当前坐标减去上一帧坐标,得到帧间位移 $(\Delta x, \Delta y) = C(x, y) - L(x, y)$ ,并用C(x,y)替换L(x,y);判断 $\Delta x$ 是否处于横向最小位移和横向最大位移以及 $\Delta y$ 是否处于纵向最小位移和纵向最大位移之间,若是:则该重心坐标有效,选出历史位移(dest\_x,dest\_y)中绝对值最大的值进行临界判断:若dest\_x绝对值大,判断dest\_x与 $\Delta x$ 是否同号,同号表示运动方向不变,dest\_x=dest\_x+ $\Delta x$ 与dest\_y=dest\_y+ $\Delta y$ ,累计帧数增加1,并继续下一帧检测;若否:当累计帧数大于等于最小帧数,拉高direct\_en,并将direct\_en与(dest\_x,dest\_y)发送至方向判断模块,下一周期将(dest\_x,dest\_y),累计帧数,direct\_en置0,重新检测;当累计帧数小于最小帧数,将(dest\_x,dest\_y),累计帧数,direct\_en置0,重新检测;若dest\_y绝对值大,判断dest\_y与 $\Delta y$ 是否同号,同号表示运动方向不变,dest\_x=dest\_x+delt\_x与dest\_y=dest\_y+ $\Delta y$ ,累计检测帧数累计帧数增加1,并继续下一帧检测;若否:当累计帧数大于等于最小帧数,拉高direct\_en并将direct\_en与(dest\_x,dest\_y)发送至方向判断模块,下一周期将(dest\_x,dest\_y),累计帧数,direct\_en均置为0,重新检测;当累计帧数小于最小帧数,将(dest\_x,dest\_y),累计帧数,direct\_en均置为0,重新检测;若否:当累计帧数大于等于最小帧数,拉高direct\_en并将direct\_en与(dest\_x,dest\_y)发送至方向判断模块,下一周期将(dest\_x,dest\_y),累计帧数,direct\_en均置为0,重新检测;当累计帧数小于最小帧数,将(dest\_x,dest\_y),累计帧数,direct\_en均置为0,重新检测。

[0061] 方向判断模块接收到direct\_en与(dest\_x,dest\_y)信号后,当direct\_en为高电

平时,根据  $(dest\_x, dest\_y)$  绝对值最大的数的方向进行判断:当  $dest\_x$  绝对值大时:若  $dest\_x$  为正,则运动方向为向右,  $direct=1$ ;若为负,则运动方向为左,  $direct=2$ ;当  $dest\_y$  绝对值大时:若  $dest\_y$  为正,则运动方向为向下,  $direct=3$ ;若为负,则运动方向为上,  $direct=4$ ;当  $direct\_en$  为低电平时,  $direct=0$ ;方向判断模块接收到  $read$  使能时,将  $direct$  发送至译码模块。

[0062] 本发明方法有效地降低片上存储容量,降低数据搬运功耗,进而提升整个芯片的能效比;该加速器采用中值滤波电路去除噪声干扰后计算差分图像的手势重心,通过追踪重心的位移得到手势移动方向,具有优越的识别效果,可以广泛应用于无接触式智能终端设备;此外该加速器根据图像输入的过程,以及滤波电路的卷积核大小采用逐行代替的帧差分过程,该方式有效的降低存储,从而降低资源消耗和功耗;综上所述,本发明具有低功耗,高精度,采用低数学复杂度算法有效地降低芯片硬件资源,提升芯片的能效比,可以广泛应用于IoT、可穿戴设备等智能终端。

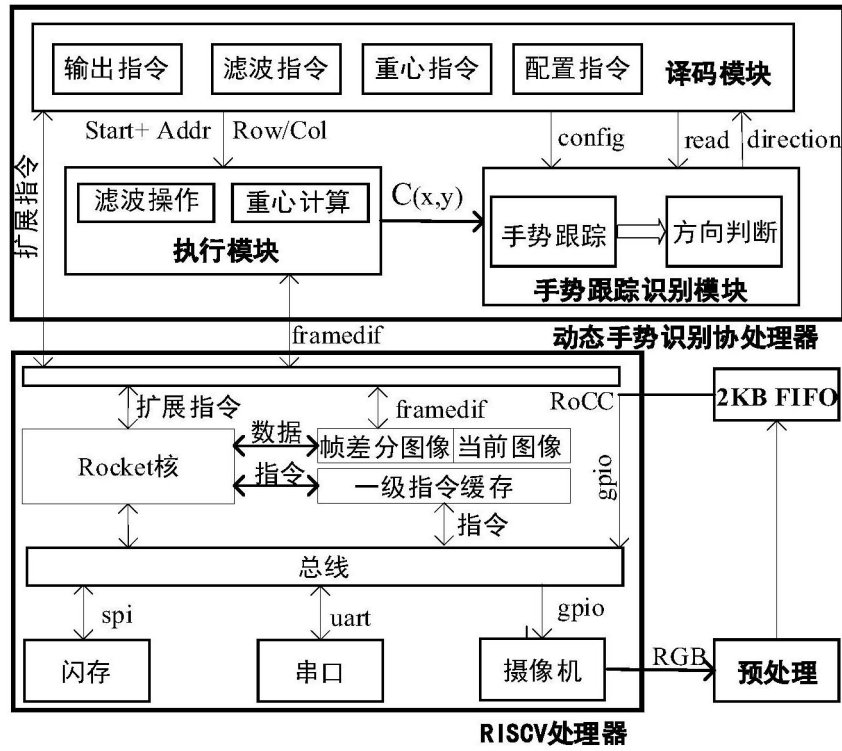


图1

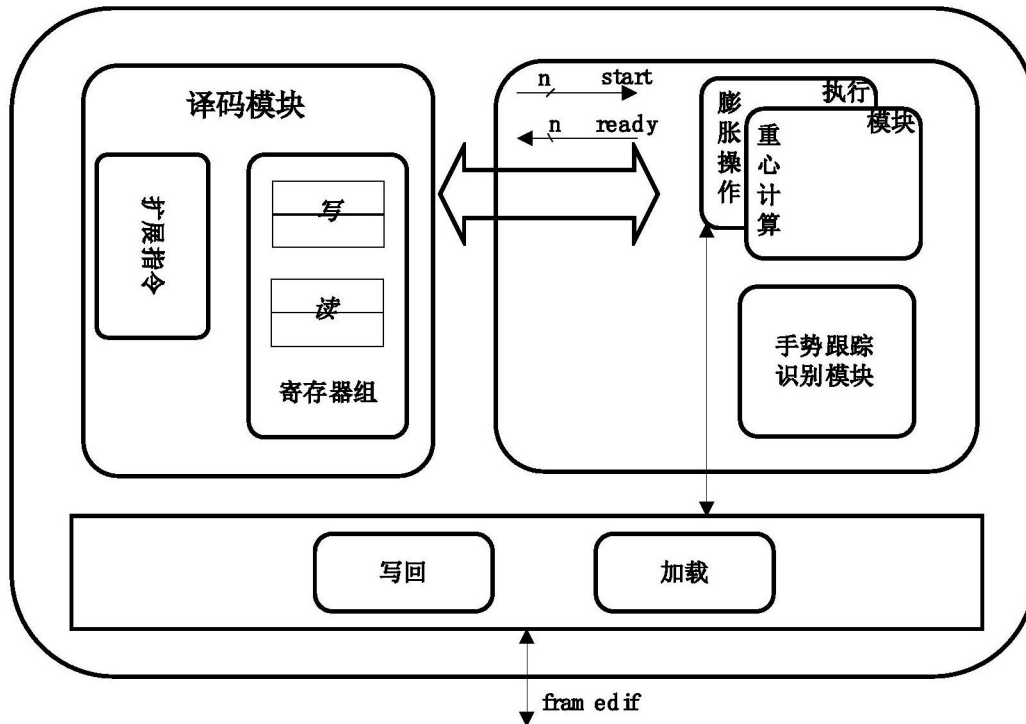


图2

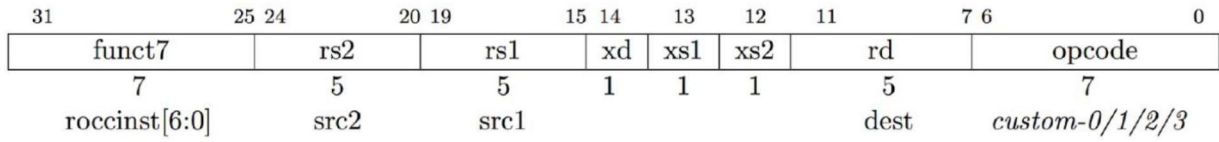


图3

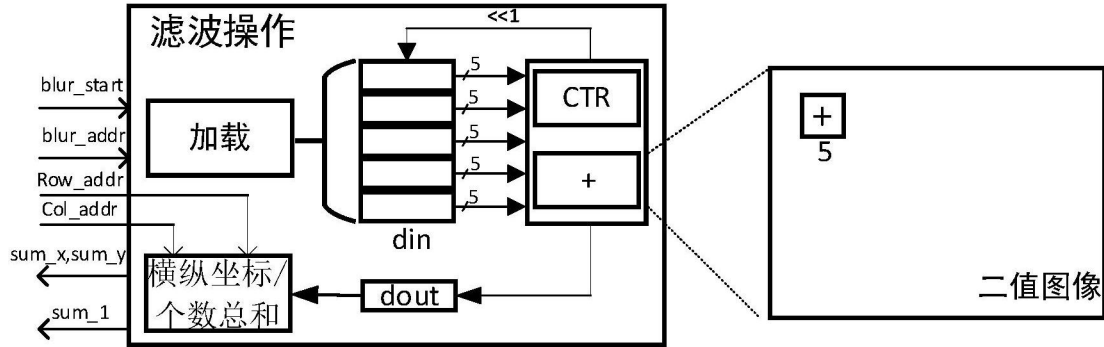


图4

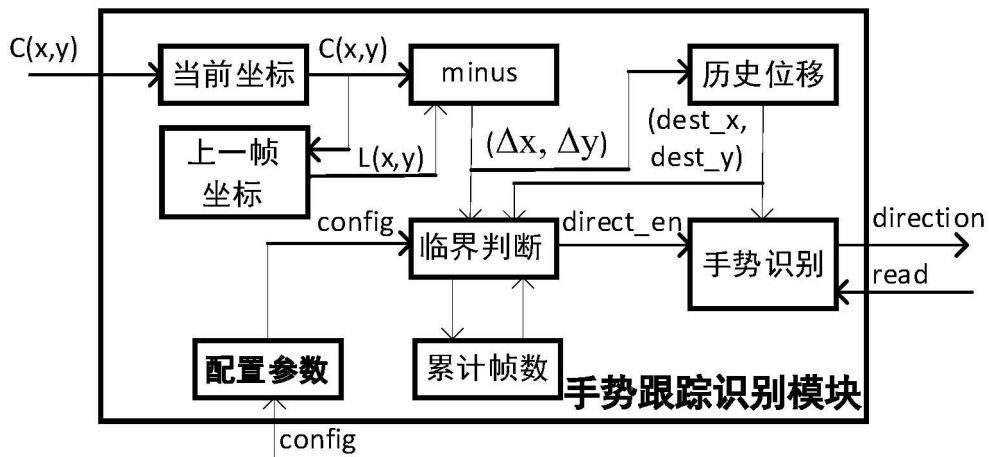


图5