



(12)发明专利申请

(10)申请公布号 CN 111028126 A

(43)申请公布日 2020.04.17

(21)申请号 201911125530.8

G06F 12/02(2006.01)

(22)申请日 2019.11.18

(71)申请人 中国航空工业集团公司西安航空计算技术研究所

地址 710065 陕西省西安市雁塔区高新区锦业二路15号

(72)发明人 陈佳 王绮卉 张少锋 任向隆 姜丽云 赵彬

(74)专利代理机构 北京清大紫荆知识产权代理有限公司 11718

代理人 娄华

(51)Int.Cl.

G06T 1/20(2006.01)

G06T 1/60(2006.01)

G06F 17/15(2006.01)

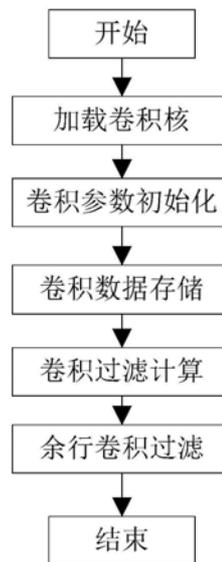
权利要求书3页 说明书7页 附图3页

(54)发明名称

一种GPU图像处理卷积过滤的实现方法

(57)摘要

本发明涉及计算机图形学技术领域,尤其涉及一种GPU图像处理卷积过滤的实现方法,卷积是一种像素过滤器,用像素本身以及邻近像素的加权平均值来替换这个像素。本发明提供了一种GPU图像处理卷积过滤的实现方法,该方法包括如下步骤:1)加载卷积核;2)卷积参数初始化;3)卷积数据存储;4)卷积过滤计算;5)余行卷积过滤。本发明解决了GPU图像处理卷积过滤的问题,包括在有限的DDR存储,和不同的卷积模式情况下,原始数据存储与卷积操作时取数据的计算地址的策略,输出行数的统一,以及最后由于非缩减模式卷积模式导致遗留卷积核高度/2的余行处理,实现了GPU的卷积核加载、像素数据与卷积核算子的卷积过滤功能。



1. 一种GPU图像处理卷积过滤的实现方法,其特征在于:所述方法包括如下步骤:

- 1) 加载卷积核:
- 2) 卷积参数初始化:
- 3) 卷积数据存储:
- 4) 卷积过滤计算:
- 5) 余行卷积过滤。

2. 根据权利要求1所述的GPU图像处理卷积过滤的实现方法,其特征在于:所述步骤1)包括:

1.1) 卷积核参数预处理:

首先从加载卷积核命令获取卷积核的像素格式、数据类型、内部格式以及宽高,然后对像素格式、数据类型以及内部格式进行预处理,根据卷积模式以及卷积核的输入宽高,重置卷积核宽高以及宽高的一半;

1.2) 加载卷积算子:

将按照像素格式和数据类型、内部格式对卷积算子进行转换后写入对应卷积目标的卷积核;

如果是分立卷积核,进行矩阵乘后写入分立卷积核;

加载进卷积核的数据就是卷积算子。

3. 根据权利要求1所述的GPU图像处理卷积过滤的实现方法,其特征在于:所述步骤2)包括:

初始化原始像素写DDR的初始地址和DDR已存储数据的行数都为卷积核高度/2,已完成图像处理行数为0;

如果卷积模式是缩减模式,卷积输出图像宽高等于原始宽高减卷积核宽高加1;

如果是非缩减模式,卷积输出图像宽高等于原始输入宽高。

4. 根据权利要求1所述的GPU图像处理卷积过滤的实现方法,其特征在于:所述步骤3)包括:

3.1) 存储基址和行数统计:

卷积数据是逐像素输入存储,

当输入像素是图像的新一行但非第一行时,存储像素的DDR基址加1;否则基址保持不变;

当DDR基址大于卷积核高度时,DDR基址减卷积核高度后为新的DDR基址;

当像素存储满一行后,DDR已存储数据的行数加1。

3.2) 计算DDR地址:

计算当前像素存储DDR的地址=DDR基址+偏移

其中,DDR基址是由步骤3.1)计算得到;

偏移=(输入像素在当前行图像的序号+卷积核宽度/2)\*4;

根据DDR地址,将原始像素数据写入DDR;

3.3) 判断满足卷积条件:

最后,判断步骤3.1)计算得到的DDR已存储数据的行数,

如果等于卷积核高度,对已存储数据的行数自减,设置存储像素行数满足卷积条件为

true,表示可以DDR中的像素数据可以进行卷积过滤计算;

如果存储行数还不够卷积核高度,那么继续存储像素数据,重复步骤3)。

5.根据权利要求1所述的GPU图像处理卷积过滤的实现方法,其特征在于:所述步骤4)包括:

4.1)获取边框常量颜色:

首先获取卷积模式为GL\_CONSTANT\_BORDER时设置的边框常量颜色;

4.2)计算取卷积数据起始地址:

计算取卷积数据的起始地址=存储数据时最新一行DDR基址+1,

如果起始地址大于卷积核高度,起始地址减卷积核高度为新的起始地址,即DDR中现存数据中最早的原始行;

4.3)计算选取原始数据的地址:

每个像素卷积计算以当前像素为中心,选取卷积核宽度\*高度的原始像素个数,与卷积核相对应,对于当前像素卷积需要的每个卷积算子对应的每个原始像素,

先初始化替换标志为false,然后计算存储在DDR的坐标(x,y);

其中,x=当前像素位置+卷积算子的列数,

y=取卷积数据的起始地址+卷积算子的行数,

如果y结果大于卷积核高度,那么y=y-卷积核高度;

4.4)坐标的边界处理:

对于在边界上的像素,需要根据卷积模式进行特殊处理:

左边界判断:

如果x坐标小于卷积核宽度/2,x=卷积核宽度/2,边界替换标志为true;

右边界判断:

如果x坐标大于原始图像宽度+卷积核宽度/2-1,x=原始图像宽度+卷积核宽度/2-1,边界替换标志为true;

下边界判断:

如果卷积完成行数小于卷积核高度/2,并且当前卷积算子行数小于卷积核高度/2,y=卷积核高度/2,边界替换标志为true;

上边界判断:

如果卷积完成行数大于原始图像高度-卷积核高度/2-1,并且当前卷积算子行数大于卷积核高度/2,y=起始地址+卷积核高度/2,如果y大于卷积核高度,那么y=y-卷积核高度,边界替换标志为true;

4.5)原始数据取值:

如果卷积模式为GL\_CONSTANT\_BORDER并且边界标志为true,原始像素采用边框常量颜色,否则根据原始像素坐标(x,y)在DDR中读取原始像素;

4.6)卷积计算:

从步骤1)加载的卷积核中取出当前位置的卷积算子,将卷积算子与原始像素相乘,并且相乘结果与上次结果进行累加;

当一整行像素都卷积完成后,进入步骤3)开始新一行数据的存储。

6.根据权利要求1所述的GPU图像处理卷积过滤的实现方法,其特征在于:所述步骤5)

包括：

通过判断卷积完成行数是否达到卷积输出图像高度,如果已达到卷积完成行数,则图像处理卷积过滤结束;

否则重新进入步骤4)开始剩余行的卷积过滤计算,直到完成所有的卷积行数。

## 一种GPU图像处理卷积过滤的实现方法

### 技术领域

[0001] 本发明属于计算机图形学技术领域,涉及一种GPU图像处理卷积过滤的实现方法。

### 背景技术

[0002] 在图形处理器芯片(下简称GPU)设计中,卷积过滤是图像处理中一个重要功能,OpenGL API定义了加载卷积核的函数,包括`glConvolutionFilter*D()`、`glCopyConvolutionFilter*D()`、`glSeparableFilter2D()`,卷积核是卷积计算的算子。OpenGL API定义的`glTexImage*D`、`glTexSubImage*D`、`glCopyTexImage*D`、`glCopyTexSubImage*D`、`glDrawPixels`、`glCopyPixels`、`glReadPixels`是在卷积使能的情况下会进行卷积计算的函数。OpenGL API还定义了三种卷积模式,这些都是必须要支持的功能。卷积过滤计算的因子包括了卷积核算子、纹理或者像素矩形的原始像素数据。如何在有限的DDR存储,和不同的卷积模式情况下,将原始数据存储与卷积操作时取数据的地址配合正确,并且输出行数的统一,以及最后由于非缩减模式卷积模式导致遗留的余行数据未处理,这都是需要解决的技术问题。

### 发明内容

[0003] 本发明目的是提供一种GPU图像处理卷积过滤的实现方法。

[0004] 本发明的技术解决方案是:

[0005] 提供了一种GPU图像处理卷积过滤的实现方法,该方法包括如下步骤:

[0006] 1) 加载卷积核:

[0007] 2) 卷积参数初始化:

[0008] 3) 卷积数据存储:

[0009] 4) 卷积过滤计算:

[0010] 5) 余行卷积过滤。

[0011] 进一步的,所述步骤1)包括:

[0012] 1.1) 卷积核参数预处理:

[0013] 首先从加载卷积核命令获取卷积核的像素格式、数据类型、内部格式以及宽高,

[0014] 然后对像素格式、数据类型以及内部格式进行预处理,根据卷积模式以及卷积核的输入宽高,重置卷积核宽高以及宽高的一半;

[0015] 1.2) 加载卷积算子:

[0016] 将按照像素格式和数据类型、内部格式对卷积算子进行转换后写入对应卷积目标的卷积核;

[0017] 如果是分立卷积核,进行矩阵乘后写入分立卷积核;

[0018] 加载进卷积核的数据就是卷积算子。

[0019] 进一步的,所述步骤2)包括:

[0020] 初始化原始像素写DDR的初始地址和DDR已存储数据的行数都为卷积核高度/2,已

完成图像处理行数为0;

[0021] 如果卷积模式是缩减模式,卷积输出图像宽高等于原始宽高减卷积核宽高加1;

[0022] 如果是非缩减模式,卷积输出图像宽高等于原始输入宽高。

[0023] 进一步的,所述步骤3)包括:

[0024] 3.1) 存储基址和行数统计:

[0025] 卷积数据是逐像素输入存储,

[0026] 当输入像素是图像的新一行但非第一行时,存储像素的DDR基址加1;否则基址保持不变;

[0027] 当DDR基址大于卷积核高度时,DDR基址减卷积核高度后为新的DDR基址;

[0028] 当像素存储满一行后,DDR已存储数据的行数加1。

[0029] 3.2) 计算DDR地址:

[0030] 计算当前像素存储DDR的地址=DDR基址+偏移

[0031] 其中,DDR基址是由步骤3.1)计算得到;

[0032] 偏移=(输入像素在当前行图像的序号+卷积核宽度/2)\*4;

[0033] 根据DDR地址,将原始像素数据写入DDR;

[0034] 3.3) 判断满足卷积条件:

[0035] 最后,判断步骤3.1)计算得到的DDR已存储数据的行数,

[0036] 如果等于卷积核高度,对已存储数据的行数自减,设置存储像素行数满足卷积条件为true,表示可以DDR中的像素数据可以进行卷积过滤计算;

[0037] 如果存储行数还不够卷积核高度,那么继续存储像素数据,重复步骤3)。

[0038] 进一步的,所述步骤4)包括:

[0039] 4.1) 获取边框常量颜色:

[0040] 首先获取卷积模式为GL\_CONSTANT\_BORDER时设置的边框常量颜色;

[0041] 4.2) 计算取卷积数据起始地址:

[0042] 计算取卷积数据的起始地址=存储数据时最新一行DDR基址+1,

[0043] 如果起始地址大于卷积核高度,起始地址减卷积核高度为新的起始地址,即DDR中现存数据中最早的原始行;

[0044] 4.3) 计算选取原始数据的地址:

[0045] 每个像素卷积计算以当前像素为中心,选取卷积核宽度\*高度的原始像素个数,与卷积核相对应,对于当前像素卷积需要的每个卷积算子对应的每个原始像素,

[0046] 先初始化替换标志为false,然后计算存储在DDR的坐标(x,y);

[0047] 其中,x=当前像素位置+卷积算子的列数,

[0048] y=取卷积数据的起始地址+卷积算子的行数,

[0049] 如果y结果大于卷积核高度,那么y=y-卷积核高度;

[0050] 4.4) 坐标的边界处理:

[0051] 对于在边界上的像素,需要根据卷积模式进行特殊处理:

[0052] 左边界判断:

[0053] 如果x坐标小于卷积核宽度/2,x=卷积核宽度/2,边界替换标志为true;

[0054] 右边界判断:

- [0055] 如果x坐标大于原始图像宽度+卷积核宽度/2-1,  $x = \text{原始图像宽度} + \text{卷积核宽度} / 2 - 1$ , 边界替换标志为true;
- [0056] 下边界判断:
- [0057] 如果卷积完成行数小于卷积核高度/2, 并且当前卷积算子行数小于卷积核高度/2,  $y = \text{卷积核高度} / 2$ , 边界替换标志为true;
- [0058] 上边界判断:
- [0059] 如果卷积完成行数大于原始图像高度-卷积核高度/2-1, 并且当前卷积算子行数大于卷积核高度/2,  $y = \text{起始地址} + \text{卷积核高度} / 2$ , 如果y大于卷积核高度, 那么  $y = y - \text{卷积核高度}$ , 边界替换标志为true;
- [0060] 4.5) 原始数据取值:
- [0061] 如果卷积模式为GL\_CONSTANT\_BORDER并且边界标志为true, 原始像素采用边框常量颜色, 否则根据原始像素坐标(x,y)在DDR中读取原始像素;
- [0062] 因为即使在卷积模式为GL\_REPLICATE\_BORDER且为边界情况时, 上述对边界处理已将地址指向边界地址;
- [0063] 4.6) 卷积计算:
- [0064] 从步骤1) 加载的卷积核中取出当前位置的卷积算子, 将卷积算子与原始像素相乘, 并且相乘结果与上次结果进行累加;
- [0065] 当一整行像素都卷积完成后, 进入步骤3) 开始新一行数据的存储。
- [0066] 进一步的, 所述步骤5) 包括:
- [0067] 通过判断卷积完成行数是否达到卷积输出图像高度, 如果已达到卷积完成行数, 则图像处理卷积过滤结束;
- [0068] 未达到卷积完成行数, 是因为在为非缩减模式时, 会有卷积核高度/2的余行还在DDR中, 需要利用边界像素或者边框常量颜色再进行卷积过滤计算,
- [0069] 则重新进入步骤4) 开始剩余行的卷积过滤计算, 直到完成所有的卷积行数。
- [0070] 本发明的有益效果:
- [0071] 本发明解决了GPU图像处理卷积过滤的问题, 包括在有限的DDR存储, 和不同的卷积模式情况下, 原始数据存储与卷积操作时取数据的计算地址的策略, 输出行数的统一, 以及最后由于非缩减模式卷积模式导致遗留卷积核高度/2的余行处理。正确的实现了GPU的卷积核加载、像素数据与卷积核算子的卷积过滤功能。

## 附图说明

- [0072] 图1为本发明提供的GPU图像处理卷积过滤的实现方法的流程简图;
- [0073] 图2为本发明的加载卷积核算法流程图;
- [0074] 图3为本发明的卷积数据存储与卷积过滤计算算法流程图;

## 具体实施方式

- [0075] 下面结合附图和具体实施例, 对本发明的技术方案进行清楚、完整地表述。显然, 所表述的实施例仅是本发明一部分实施例, 而不是全部的实施例, 基于本发明中的实施例, 本领域技术人员在没有做出创造性劳动前提所获得的所有其他实施例, 都属于本发明的保

护范围。

[0076] 本发明提供了一种GPU图像处理卷积过滤的实现方法,提供了一种GPU图像处理卷积过滤的实现方法,该方法包括如下步骤:

[0077] 1) 加载卷积核:

[0078] 1.1) 卷积核参数预处理:

[0079] 首先从加载卷积核命令获取卷积核的像素格式、数据类型、内部格式以及宽高,

[0080] 然后对像素格式、数据类型以及内部格式进行预处理,根据卷积模式以及卷积核的输入宽高,重置卷积核宽高以及宽高的一半;

[0081] 1.2) 加载卷积算子:

[0082] 将按照像素格式和数据类型、内部格式对卷积算子进行转换后写入对应卷积目标的卷积核;

[0083] 如果是分立卷积核,进行矩阵乘后写入分立卷积核;

[0084] 加载进卷积核的数据就是卷积算子;

[0085] 2) 卷积参数初始化:

[0086] 初始化原始像素写DDR的初始地址和DDR已存储数据的行数都为卷积核高度/2,已完成图像处理行数为0;

[0087] 如果卷积模式是缩减模式,卷积输出图像宽高等于原始宽高减卷积核宽高加1;

[0088] 如果是非缩减模式,卷积输出图像宽高等于原始输入宽高;

[0089] 3) 卷积数据存储:

[0090] 3.1) 存储基址和行数统计:

[0091] 卷积数据是逐像素输入存储,

[0092] 当输入像素是图像的新一行但非第一行时,存储像素的DDR基址加1;否则基址保持不变;

[0093] 当DDR基址大于卷积核高度时,DDR基址减卷积核高度后为新的DDR基址;

[0094] 当像素存储满一行后,DDR已存储数据的行数加1。

[0095] 3.2) 计算DDR地址:

[0096] 计算当前像素存储DDR的地址=DDR基址+偏移

[0097] 其中,DDR基址是由步骤3.1) 计算得到;

[0098] 偏移=(输入像素在当前行图像的序号+卷积核宽度/2)\*4;

[0099] 根据DDR地址,将原始像素数据写入DDR;

[0100] 3.3) 判断满足卷积条件:

[0101] 最后,判断步骤3.1) 计算得到的DDR已存储数据的行数,

[0102] 如果等于卷积核高度,对已存储数据的行数自减,设置存储像素行数满足卷积条件为true,表示可以DDR中的像素数据可以进行卷积过滤计算;

[0103] 如果存储行数还不够卷积核高度,那么继续存储像素数据,重复步骤3);

[0104] 4) 卷积过滤计算:

[0105] 4.1) 获取边框常量颜色:

[0106] 首先获取卷积模式为GL\_CONSTANT\_BORDER时设置的边框常量颜色;

[0107] 4.2) 计算取卷积数据起始地址:

- [0108] 计算取卷积数据的起始地址=存储数据时最新一行DDR基址+1,
- [0109] 如果起始地址大于卷积核高度,起始地址减卷积核高度为新的起始地址,即DDR中现存数据中最早的原始行;
- [0110] 4.3) 计算选取原始数据的地址:
- [0111] 每个像素卷积计算以当前像素为中心,选取卷积核宽度\*高度的原始像素个数,与卷积核相对应,对于当前像素卷积需要的每个卷积算子对应的每个原始像素,
- [0112] 先初始化替换标志为false,然后计算存储在DDR的坐标(x,y);
- [0113] 其中,x=当前像素位置+卷积算子的列数,
- [0114] y=取卷积数据的起始地址+卷积算子的行数,
- [0115] 如果y结果大于卷积核高度,那么y=y-卷积核高度;
- [0116] 4.4) 坐标的边界处理:
- [0117] 对于在边界上的像素,需要根据卷积模式进行特殊处理:
- [0118] 左边界判断:
- [0119] 如果x坐标小于卷积核宽度/2,x=卷积核宽度/2,边界替换标志为true;
- [0120] 右边界判断:
- [0121] 如果x坐标大于原始图像宽度+卷积核宽度/2-1,x=原始图像宽度+卷积核宽度/2-1,边界替换标志为true;
- [0122] 下边界判断:
- [0123] 如果卷积完成行数小于卷积核高度/2,并且当前卷积算子行数小于卷积核高度/2,y=卷积核高度/2,边界替换标志为true;
- [0124] 上边界判断:
- [0125] 如果卷积完成行数大于原始图像高度-卷积核高度/2-1,并且当前卷积算子行数大于卷积核高度/2,y=起始地址+卷积核高度/2,如果y大于卷积核高度,那么y=y-卷积核高度,边界替换标志为true;
- [0126] 4.5) 原始数据取值:
- [0127] 如果卷积模式为GL\_CONSTANT\_BORDER并且边界标志为true,原始像素采用边框常量颜色,否则根据原始像素坐标(x,y)在DDR中读取原始像素;
- [0128] 4.6) 卷积计算:
- [0129] 从步骤1) 加载的卷积核中取出当前位置的卷积算子,将卷积算子与原始像素相乘,并且相乘结果与上次结果进行累加;
- [0130] 当一整行像素都卷积完成后,进入步骤3) 开始新一行数据的存储;
- [0131] 5) 余行卷积过滤。
- [0132] 通过判断卷积完成行数是否达到卷积输出图像高度,如果已达到卷积完成行数,则图像处理卷积过滤结束;
- [0133] 则重新进入步骤4) 开始剩余行的卷积过滤计算,直到完成所有的卷积行数。
- [0134] 实施例:
- [0135] 下面结合附图1-3对本发明做进一步详细描述;
- [0136] 一种GPU图像处理卷积过滤的实现方法,提供了一种GPU图像处理卷积过滤的实现方法,该方法包括如下步骤:

- [0137] 1) 加载卷积核:
- [0138] 1.1) 卷积核参数预处理:
- [0139] 首先从加载卷积核命令获取卷积核的像素格式、数据类型、内部格式以及宽高,
- [0140] 然后对像素格式、数据类型以及内部格式进行预处理,如果卷积模式为GL\_REDUCE,原始卷积核宽高的一半为0;否则,根据输入的宽高计算原始卷积核宽高的一半,
- [0141] 如果卷积核宽高有一维为0,设置卷积核宽高都为1,卷积核宽高的一半都为0,否则,设置卷积核宽高为输入的宽高,卷积核宽高的一半为原始卷积核宽高的一半。
- [0142] 1.2) 加载卷积算子:
- [0143] 将按照像素格式和数据类型、内部格式对卷积算子进行转换后写入对应卷积目标的卷积核:
- [0144] 如果是分立卷积核,进行矩阵乘后写入分立卷积核;
- [0145] 加载进卷积核的数据就是卷积算子;
- [0146] 2) 卷积参数初始化:
- [0147] 初始化原始像素写DDR的初始地址和DDR已存储数据的行数都为卷积核高度/2;
- [0148] 如果卷积模式是缩减模式,卷积输出图像宽高等于原始宽高减卷积核宽高加1;
- [0149] 如果是非缩减模式,卷积输出图像宽高等于原始输入宽高;
- [0150] 初始化已完成图像处理行数为0;
- [0151] 3) 卷积数据存储:
- [0152] 3.1) 存储基址和行数统计:
- [0153] 卷积数据是逐像素输入存储,
- [0154] 当输入像素是图像的新一行但非第一行时,存储像素的DDR基址加1;否则基址保持不变;
- [0155] 当DDR基址大于卷积核高度时,DDR基址减卷积核高度后为新的DDR基址;
- [0156] 当像素存储满一行后,DDR已存储数据的行数加1。
- [0157] 3.2) 计算DDR地址:
- [0158] 计算当前像素存储DDR的地址=DDR基址+偏移
- [0159] 其中,DDR基址是由步骤3.1) 计算得到;
- [0160] 偏移=(输入像素在当前行图像的序号+卷积核宽度/2)\*4;
- [0161] 根据DDR地址,将原始像素数据写入DDR;
- [0162] 3.3) 判断满足卷积条件:
- [0163] 最后,判断步骤3.1) 计算得到的DDR已存储数据的行数,
- [0164] 如果等于卷积核高度,对已存储数据的行数自减,设置存储像素行数满足卷积条件为true,表示可以DDR中的像素数据可以进行卷积过滤计算;
- [0165] 如果存储行数还不够卷积核高度,那么继续存储像素数据,重复步骤3);
- [0166] 4) 卷积过滤计算:
- [0167] 4.1) 获取边框常量颜色:
- [0168] 首先获取卷积模式为GL\_CONSTANT\_BORDER时设置的边框常量颜色;
- [0169] 4.2) 计算取卷积数据起始地址:
- [0170] 计算取卷积数据的起始地址=存储数据时最新一行DDR基址+1,

- [0171] 如果起始地址大于卷积核高度,起始地址减卷积核高度为新的起始地址,即DDR中现存数据中最早的原始行;
- [0172] 4.3) 计算选取原始数据的地址:
- [0173] 每个像素卷积计算以当前像素为中心,选取卷积核宽度\*高度的原始像素个数,与卷积核相对应,对于当前像素卷积需要的每个卷积算子对应的每个原始像素,
- [0174] 先初始化替换标志为false,然后计算存储在DDR的坐标(x,y);
- [0175] 其中, $x = \text{当前像素位置} + \text{卷积算子的列数}$ ,
- [0176]  $y = \text{取卷积数据的起始地址} + \text{卷积算子的行数}$ ,
- [0177] 如果y结果大于卷积核高度,那么 $y = y - \text{卷积核高度}$ ;
- [0178] 4.4) 坐标的边界处理:
- [0179] 对于在边界上的像素,需要根据卷积模式进行特殊处理:
- [0180] 左边界判断:
- [0181] 如果x坐标小于卷积核宽度/2, $x = \text{卷积核宽度} / 2$ ,边界替换标志为true;
- [0182] 右边界判断:
- [0183] 如果x坐标大于原始图像宽度+卷积核宽度/2-1, $x = \text{原始图像宽度} + \text{卷积核宽度} / 2 - 1$ ,边界替换标志为true;
- [0184] 下边界判断:
- [0185] 如果卷积完成行数小于卷积核高度/2,并且当前卷积算子行数小于卷积核高度/2, $y = \text{卷积核高度} / 2$ ,边界替换标志为true;
- [0186] 上边界判断:
- [0187] 如果卷积完成行数大于原始图像高度-卷积核高度/2-1,并且当前卷积算子行数大于卷积核高度/2, $y = \text{起始地址} + \text{卷积核高度} / 2$ ,如果y大于卷积核高度,那么 $y = y - \text{卷积核高度}$ ,边界替换标志为true;
- [0188] 4.5) 原始数据取值:
- [0189] 如果卷积模式为GL\_CONSTANT\_BORDER并且边界标志为true,原始像素采用边框常量颜色,否则根据原始像素坐标(x,y)在DDR中读取原始像素;
- [0190] 4.6) 卷积计算:
- [0191] 从步骤1) 加载的卷积核中取出当前位置的卷积算子,将卷积算子与原始像素相乘,并且相乘结果与上次结果进行累加;
- [0192] 当一整行像素都卷积完成后,进入步骤3) 开始新一行数据的存储;
- [0193] 5) 余行卷积过滤。
- [0194] 通过判断卷积完成行数是否达到卷积输出图像高度,如果已达到卷积完成行数,则图像处理卷积过滤结束;
- [0195] 则重新进入步骤4) 开始剩余行的卷积过滤计算,直到完成所有的卷积行数。

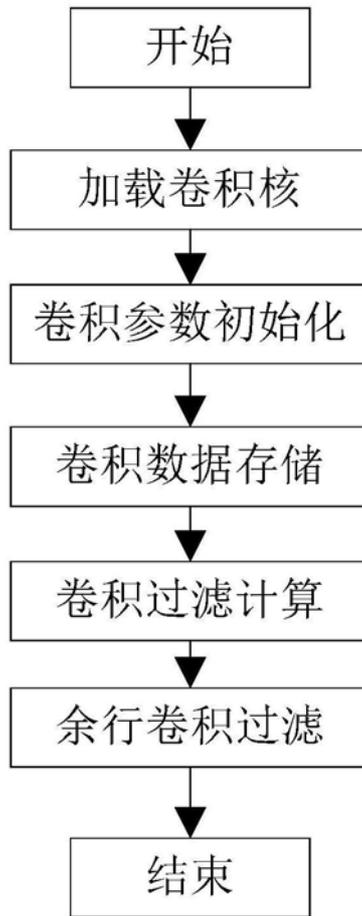


图1

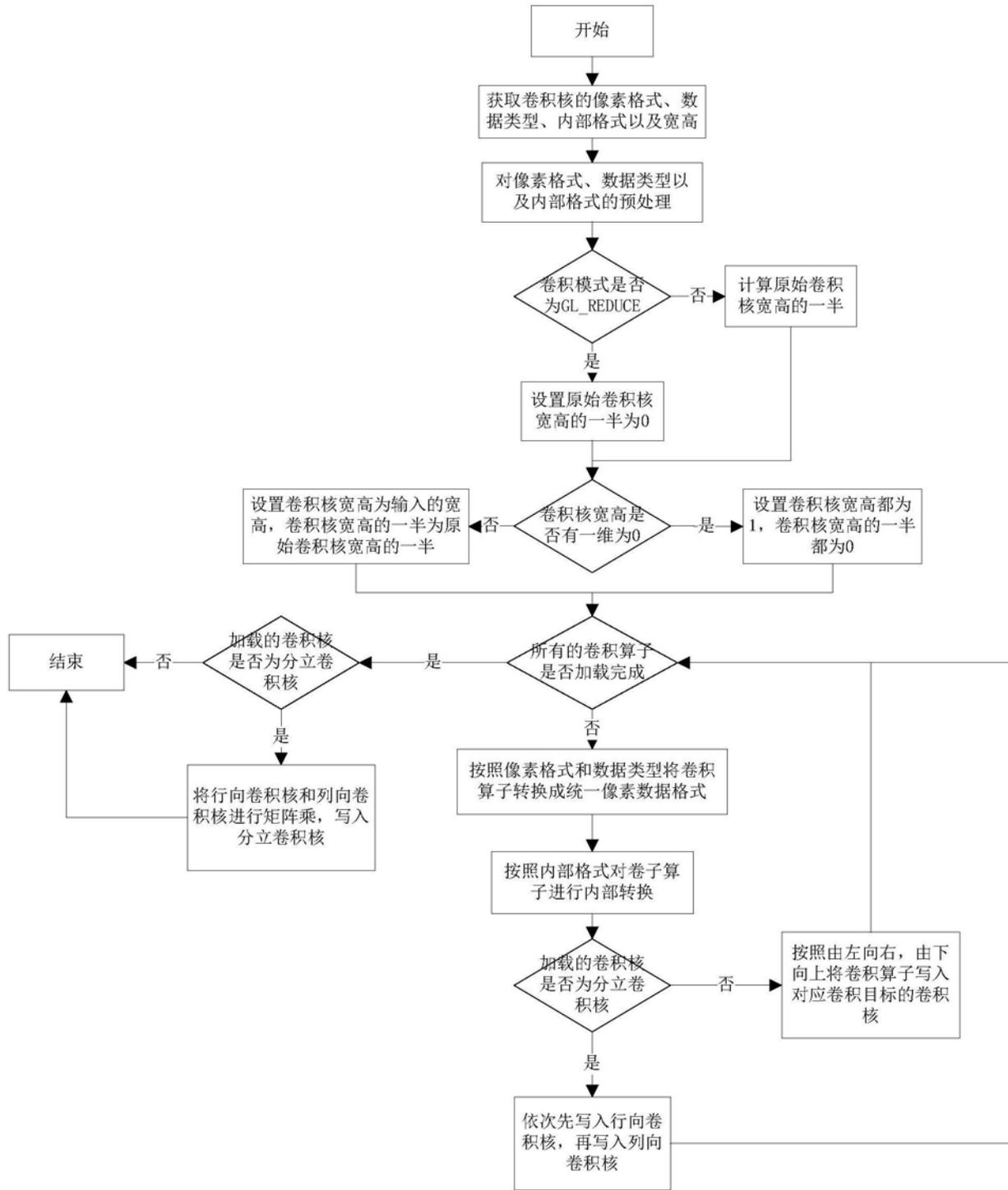


图2

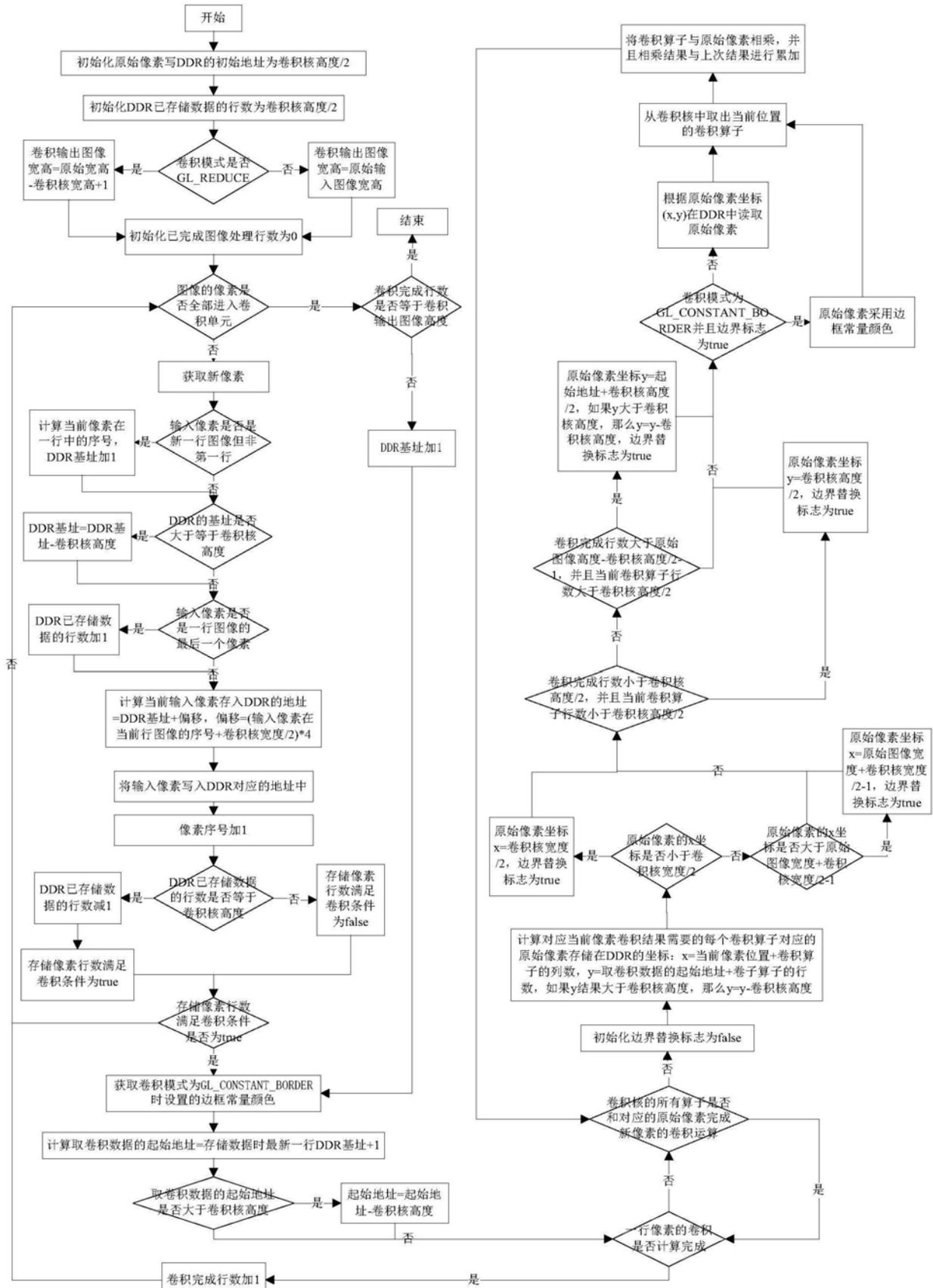


图3