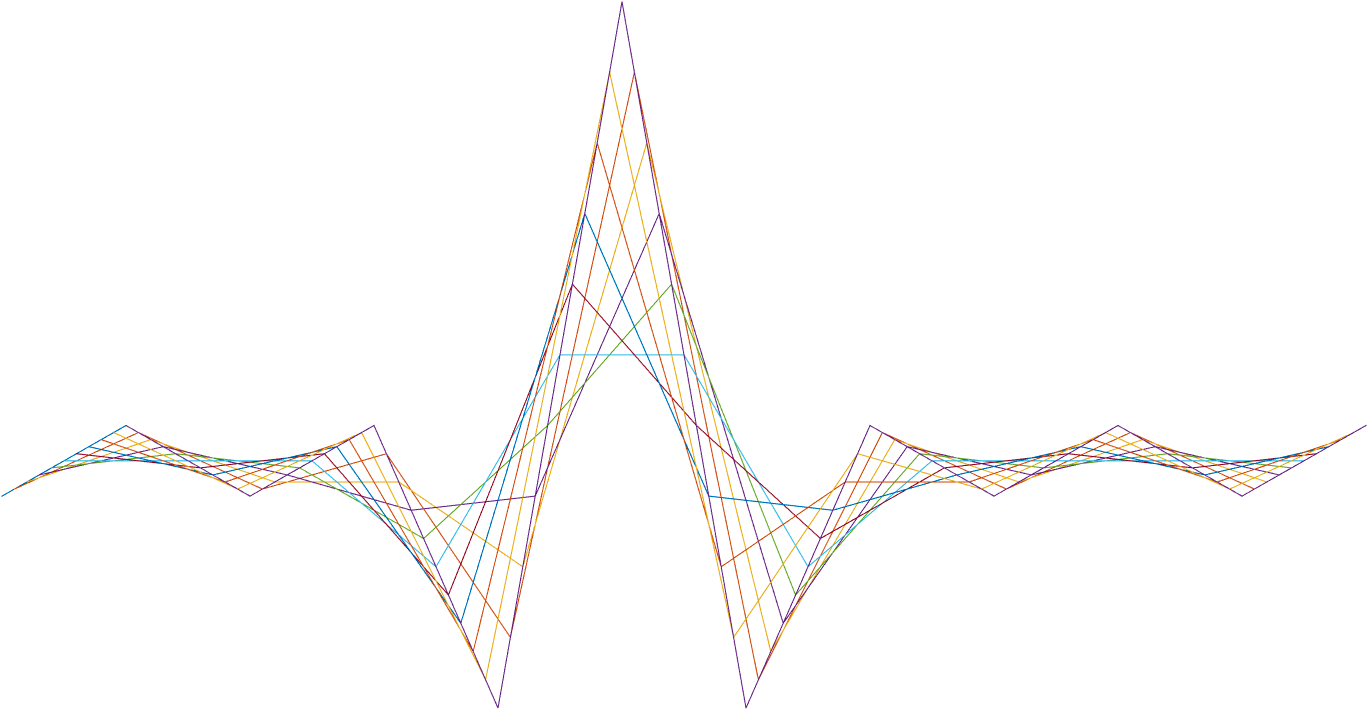
IVCAM2.0 3D Imaging Camera   
IVCAM2.0 3D Imaging Camera



ASIC A0 JFIL Edge filter specification

6 October 2016

Revision 0.6.0

Table 1: Revision history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Matlab Version | Revision Number | Revised by | Description | Revision Date |
| 2016a | 0.5.0 | David Silver | Initial release | July 4, 2016 |
| 2016a | 0.6.0 | Yoni Chechik | Total makeOver | Aug. 31, 2016 |
| 2016a | 0.6.1 | Yoni Chechik | bypass mode renew | Sep. 12, 2016 |
| 0.6.37 |  | Omer Sella | Added a test plan | Sep, 22, 2016 |
|  | 0.6.11 | Yoni Chechik | Line buffers=2 | 10.11.16 |

Contents

[1 Introduction 4](#_Toc460425790)

[1.1 General 4](#_Toc460425791)

[2 Interfaces 5](#_Toc460425792)

[2.1 Memory interfaces 5](#_Toc460425793)

[2.2 Other block interfaces 5](#_Toc460425794)

[3 Detailed Description 6](#_Toc460425795)

[3.1 Outline 6](#_Toc460425796)

[3.2 Validness generation 6](#_Toc460425797)

[3.3 Filter structure 6](#_Toc460425798)

[4 Registers 8](#_Toc460425799)

[4.1 New registers definition 8](#_Toc460425800)

[5 Memory and computation 9](#_Toc460425801)

[5.1 Memory requirement 9](#_Toc460425802)

[5.2 Computational load 9](#_Toc460425803)

[6 Test plan 10](#_Toc460425804)

# Introduction

## General

The Edge Filter is located inside the *JFIL* block*.* It is a small median filter that preserves edges.

# Interfaces

## Memory interfaces

The block require memory interface for 2 line buffers.

## Other block interfaces

* 1. Input

The block receive the following as input:

1. Depth 16-bit & IR 12-bit.
2. Confidence 4-bit.
3. Flags 4-bit(not used).
   1. Output

The block send output to the output depth stream of the camera.

1. Depth 16-bit & IR 12-bit.
2. Confidence 4-bit.
3. Flags 4-bit.

# Detailed Description

## Outline

The edge filter operates on a depth+ IR+ confidence frame + a binary mask of depth validness (generated from the confidence). The filter is using a 3x3 sliding window. For each 3x3 neighborhood we estimate if there’s an edge going through the central pixel. If there is, we average the depth value of the two neighbors in that edge direction (and this is how we preserve the edge). Else, we do a simple median of the valid pixels.

## Validness generation

Invalid pixels are pixels with 0 in the confidence & 0 in the depth pixel (will always come together!).

## Filter structure

The filter works as 3x3 sliding window, with the following scheme:

Let x be the center pixel, and  its 8 neighbors. Like this:



We look for the pair such that the values are the closets **only in the depth frame**. That is, we look for which direction of edge going through the pixel.

First, we check if the edge pair is under a given threshold (RegsJFILEdge#MaxTh) – which means that this pair is on the same plane.

Second, we check if we have at list one other pair that is over a given threshold (RegsJFILEdge#DetectTh) – which means that it’s indeed an edge.

In math we can write this algorithm as follows:

In the computation of differences between values, we **ignore** pixels that are marked invalid by assigning the difference |Valid-Invalid|=216-1.

Likewise, when finding the max diff, we assign to pairs with at least one invalid as |Valid-Invalid|=0.

When taking median, we replace invalid in 0 and 216-1 alternately (starting from 0), hence if all of the pixels are invalid we will get 0 in depth (stays invalid) - this can happen in depth or IR.

If it happens in a depth window - we also keep the confidence 0.

Bypass modes

% | config (2b) | bypass(1b) | -> total 3b

% XX1 - bypass

% 000 - apply on all pixels

% 010 - do not apply on valid IR

% 100 - do not apply on valid depth

% 110 - do not apply on both valid IR and valid depth

# Registers

## New registers definition

Table 3: Register definitions

| **Name** | **Size** | **Default** | **Range** | **Special values/ description** |  |
| --- | --- | --- | --- | --- | --- |
| RegsJFILEdge#Bypass | 1 | 110 | Valid values:  001  000  110  100  010 | 001 Bypass  000 Apply on all.  110 do not apply on valid IR/depth  100 – do not apply on valid depth  010 – do not apply on valid IR  Other modes are invalid. |
| RegsJFILEdge#MaxTh | 16 | 100 | 0 – 216-1 | The maximal threshold for the difference between two pixels to take their average. |
| RegsJFILEdge#DetectTh | 16 | 0 | 0 – 216-1 | A minimal threshold for a pixel to be considered to be on edge |

# Memory and computation

## Memory requirement

## Computational load

# Test plan

Table 3: Randomization table

| **Name** | **Range** | **Distribution** |  |
| --- | --- | --- | --- |
| RegsJFILEdge#Bypass | 001 | 4% |
| 000 | 24% |
| 010 | 24% |
| 100 | 24% |
| 110 | 24% |
| RegsJFILEdge#MaxTh | 0-300 | 30% |
| 301-1000 | 30% |
| 1001-2^16-1 | 40% |
| RegsJFILEdge#DetectTh | 0-300 | 30% |
| 301-1000 | 30% |
| 1001-2^16-1 | 40% |