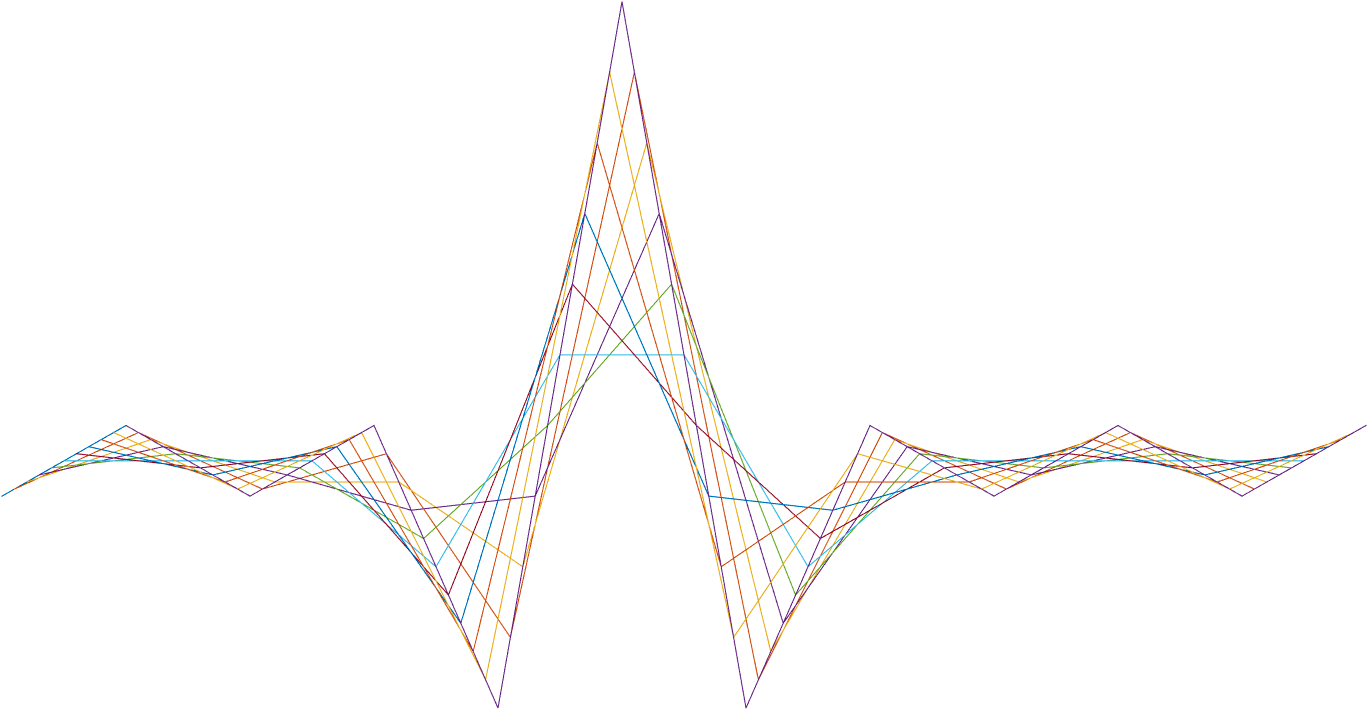
IVCAM2.0 3D Imaging Camera



ASIC A0 JFIL featureExtration specification

10 November 2016

Revision 0.5.0

Intel Top Secret

Table 1: Revision history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Matlab Version | Revision Number | Revised by | Description | Revision Date |
| 2016a | 0.5.0 | Yoni Chechik | Initial release | Sep. 1, 2016 |
| 6.37 | 0.5.1 | Omer Sella | Review   * Added registers. * Added details on output. | Sep. 1 ,2016 |
|  | 0.5.2 | Yoni Chechik | Convolution difference & more regs | Sep. 4 ,2016 |
|  | 0.5.3 | Yoni Chechik | Kernels are now 8 bits | Sep. 12 ,2016 |
| 0.7.38 |  | Omer Sella | Changed FEXT to pass through Edge2 – due to JFIL pipe change. | 22/09/2016 |
| 0.7.44 | 0.7.3 | Ohad Menashe | FP16🡪FP18 | 30-Oct-2016 |
|  | 0.73.1 | Yoni Chechik | JFILdFeaturesIsEnergyPreserve -> RegsJFILdFeaturesNorm | 10.11.16 |

7

Contents

[1. Introduction 3](#_Toc460745401)

[1.1 General 3](#_Toc460745402)

[2. Interfaces 4](#_Toc460745403)

[1.2 Memory interfaces 4](#_Toc460745404)

[1.3 Other block interfaces 4](#_Toc460745405)

[1.4 Input 4](#_Toc460745406)

[1.4.1 4](#_Toc460745407)

[1.4.2 Output 4](#_Toc460745408)

[2 Memory 5](#_Toc460745409)

[2.1 Memory requirement 5](#_Toc460745410)

[3 Detailed Description 6](#_Toc460745411)

[3.1 Outline 6](#_Toc460745412)

[3.2 In -> out (through) 6](#_Toc460745413)

[3.3 Sort 6](#_Toc460745414)

[3.4 Convolution 6](#_Toc460745415)

[3. Registers 7](#_Toc460745416)

Tables

[Table 1: Revision history 1](#_Toc455660006)

[Table 2: Registers 9](#_Toc455660007)

1. Introduction

General

The FeatureExtrationblock is located inside the *JFIL* block*.* It outputs features to the neural network (NN) that comes after it.

We have two quite similar blocks:

1. FeatureExtrationD for depth NN
2. FeatureExtrationI for IR NN. The difference between them is how they deal with low confidence pixels.

The following diagrams show the I/O and position of the feature extractor blocks.





1. Interfaces

Input

The block receive the following as input:

* 16b depth
* 12b ir
* 4b confidence
* In featureExtrationD:
  + 2\*16b depth – 2 pixels of different filtered depth inputs via bilateral filters (pass through).

### 

Output

The block sends output to the bilateral filter in the Depth Filter block:

* 16b depth
* 12b ir
* 4b Confidence
* For featureExtrationD:
  + 7\*16b depth – 7 pixels of different filtered (convolution) depth.
  + 1\*16b depth – 1 pixels of filtered (convolution) confidence.
  + 2\*16b depth – 2 pixels of different filtered depth inputs via bilateral filters (pass through).
  + 1\*16b depth – 1 pixel output from Edge2 filter (pass through).
  + 9\*16b depth – sorted pixels in sliding window of 3\*3.
* In featureExtrationI:
  + 2\*12b IR – 2 pixels of different filtered IR.
  + 9\*12b IR – sorted pixels in sliding window of 3\*3.

Memory

The block requires memory interface for 4 line buffers – for inputs depth (16b), ir (12b), conf(4b), 2\*biletaral(16b).

# Detailed Description

Outline

The different NN (IR NN & depth NN) both require features to work on. The FeatureExtration blocks are located before the NN blocks and right after the image processing filter blocks in the JFIL main block.

All the outgoing outputs are in fp18.

In -> out (through)

For both FeatureExtration (depth & IR) blocks we assign to outputs for the untouched depth, IR & confidence pixels.

In FeatureExtrationD we also assign outputs for 2 pixels from 2 bilateral filters that were performed before this block (so through as well). In addition we pass through 1 pixel from Edge2.

Sort

For both FeatureExtration blocks we output a stream of sorted pixels in a 3\*3 sliding window in the pixel (In FeatureExtrationD we output sorted depth and in FeatureExtrationI we output sorted IR).

In the edges of the pixel we replicate the nearest neighbors to obtain 3\*3 window.

The main difference between the two sort functions is that FeatureExtrationI does standard sort, while FeatureExtrationD has several sort types:

% 0 - low conf are 0 and place at the beginning

% 1 - low conf are fp18(nan)= and place at the beginning

% 2 - low conf are fp18(nan) and place at both ends equally (start from left==beginning)

% 3 - valid min and max are stretch to fill the ends. if all low conf put all fp18(nan)

The confidence threshold is determined by the register dFeaturesConfThr. Pixels that are below that threshold will be set to 0.

The sort type is determined by the register dFeaturesSortType.

Convolution

For both FeatureExtration blocks we output a stream of convolved input pixels:

In the FeatureExtrationD we output 7 different streams of depth after 7 different convolution filters, as well as an 8th stream of confidence, which goes through a convolution filter as well.

In the FeatureExtrationI we output 2 different streams of IR after 2 different filters.

The convolution calculation is done with 8b 2’s complement fixed point precision weights that are saved in LUTs (one for each block of FeatureExtration), and in the edges we obtain the nearest neighbor to make a 5\*5 convolution filter. The LUTS are saved is hex int32, 4 weights in a cell fixed point with 7 bits shift, like so:

%this is how the weights are in the LUT (in HEX)

%[w00 w06 w20;

% ... …

% w05 w07 w24;

%and the next filter concatenated after w27

Again, we have a difference between D & I:

I is a regular convolution.

D has a confidence threshold (RegsJFILdFeaturesConfThr) which under it the weight of the depth in the corresponding place is 0.

Furthermore, in D we divide the kernels to energy preserving and not energy preserving (saved in a binary array of 8 places corresponding to the 8 kernels: RegsJFILdFeaturesNorm). If for the given kernel, the corresponding energy preserve bit is ‘1’, we normalize the outcome: it’s a division of fp18 of the convolution outcome with the sum of the kernels’ high confidence weights of that pixel.

1. Registers

| **Name** | **Size** | **Default** | **Range** | **Special values/ description** |
| --- | --- | --- | --- | --- |
| **JFIL** |  |  |  |  |
| RegsJFILdFeaturesConfThr | 4 | 0 | 0-15 | Conf below this number, will be considered low. |
| RegsJFILdFeaturesNorm | 1 | 11011000 | 0-2^8-1 | Binary- for each kernel is energy preserving |
| RegsJFILdFeaturesSortType | 1 | 3 | 0-3 | Select D sort type |

Table 2: Registers

1. Test Plan

| **Name** | **Range** | **Distribution** |
| --- | --- | --- |
| **JFIL** |  |  |
| RegsJFILdFeaturesConfThr | 0-4 | 60% |
| 5-12 | 35% |
| 13-15 | 5% |
| RegsJFILdFeaturesNorm | 0-2^8-1 | 100% |
| RegsJFILdFeaturesSortType | 0 | 25% |
| 1 | 25% |
| 2 | 25% |
| 3 | 25% |

Table : Registers