

# 1 TODO

- (03.02.) Investigate data movement in ArrayFire
- (02.02.) Checkout OpenCV CUDA/OpenCL performance
- (29.01.) Compare CUDA vs. VIGRA vs. ArrayFire
- (29.01.) Implement CUDA c++ gaussian (ignore padding)
- DONE: (29.01.) Implement ArrayFire c++ gaussian (ignore padding)
- DONE: (29.01.) Implement VIGRA c++ gaussian (ignore padding)

## 2 Current setup

### 2.1 `vigra.filters.gaussianSmoothing(img, sigma)`

- separated x-,y- convolution
- kernel is initialized from scratch for every call
- kernel element:  $k[i] = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{i^2}{2\sigma^2}}$
- kernel radius:  $r = \text{round}(3 * \sigma)$

### 2.2 `vigra.filters.laplacianOfGaussian(img, sigma)`

### 2.3 `vigra.filters.gaussianGradientMagnitude(img, sigma)`

### 2.4 `vigra.filters.gaussianSmoothing(img, sigma)` - `vigra.filters.gaussianSmoothing(img, 0.66*sigma)`

### 2.5 `vigra.filters.structureTensorEigenvalues(img, sigma, sigma/2.0)`

### 2.6 `vigra.filters.hessianOfGaussianEigenvalues(img, sigma)`

## 3 HiWi HCI Notes

### 3.1 02.02.2016 (2h)

Worked on arrayfire c++ implementation. Set up vigra and such on personal notebook.

## 3.2 02.02.2016 (1.5h)

### 3.2.1 OpenCV

- CUDA support
- [http://docs.opencv.org/3.0.0/d4/d25/classcv\\_1\\_1cuda\\_1\\_1Convolution.html](http://docs.opencv.org/3.0.0/d4/d25/classcv_1_1cuda_1_1Convolution.html)
- Comparison OpenCV vs. ArrayFire (LibJacket) <http://mcclanahoochie.com/blog/2011/09/opencv-vs-libjacket-gpu-sobel-filtering/>

## 3.3 29.01.2016 (9:30 - 12:00, 12:30 - 14:30, 4.5h)

### 3.3.1 ArrayFire

- handles 1D,2D,3D convolutions
- allows batch processing (1 Kernel, n images or n Kernels, 1 image and so on)
- Does it separate Kernels?
- Example: [http://www.arrayfire.com/docs/getting\\_started\\_2convolve\\_8cpp-example.htm](http://www.arrayfire.com/docs/getting_started_2convolve_8cpp-example.htm)

### 3.3.2 Talked to Sven

- ArrayFire
- nexts steps (see todo)
- received Michaela's code

### 3.3.3 Commands

Show vigra compiler options

```
# vigra-config --cpp-flags --libs
```

## 3.4 26.01.2016 (3h)

Base on CUDA samples. separableConvolution and textureConvolution are already implemented.

### 3.4.1 snippets

---

Listing 1: "print out gaussian 1D kernel"

---

```
import vigra

sigma = 3
```

```
kernel = vigra.filters.Kernel1D()
kernel.initGaussian(sigma)

for i in range(kernel.left(), kernel.right()+1):
    print kernel[i]
```

---

### 3.4.2 Addresses

- bigheron - GPU system
- hciwiki.iwr.uni-heidelberg.de

## 3.5 22.01.2016 (3h)

### 3.5.1 Ideas

- Vigna efficiency improvements
- CUDA/OpenCL (CUDA samples!)
- OpenGL
- FFT approach

### 3.5.2 Commands

```
# source miniconda2/bin/activate ilastik-devel \\
# source miniconda2/bin/activate root \\
# conda create -n ilastik-devel -c ilastik ilastik-everything-but-tracking
```