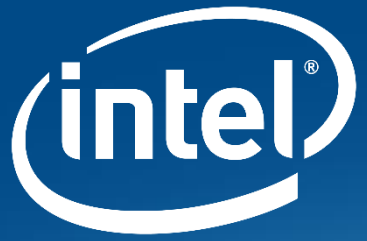


Introduction to Computer Vision

Daniil Osokin

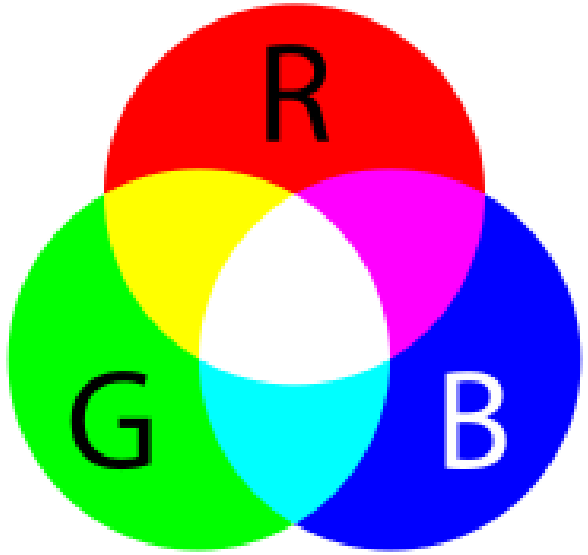
Intel, IOTG/VMC/ICV

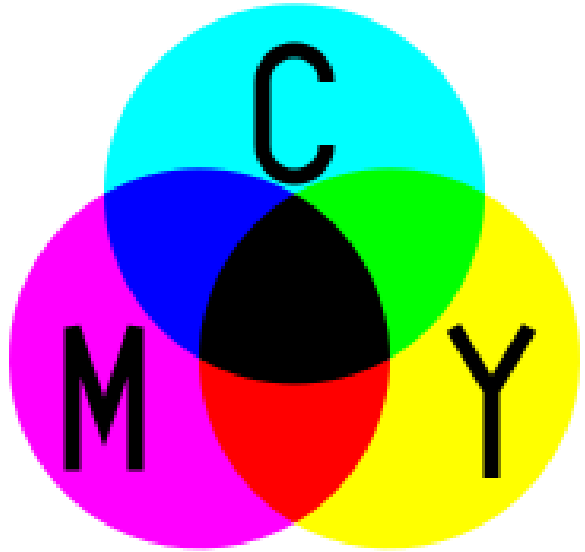
daniil.osokin@intel.com

Internet of Things Group

Color Spaces

- Grayscale, values [0;255]
- RGB, RGBA, BGR, values [0;255] x number_of_channels
- LUV, LAB, HSV
- YUV, YUV 420, YUV 422, YUYV
- CMYK
- etc.



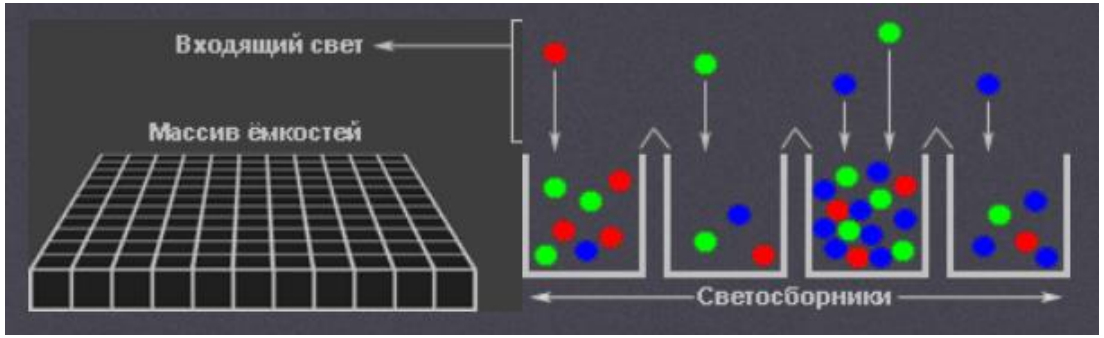


OpenCV reads image in BGR by default.

Images credit: https://en.wikipedia.org/wiki/CMYK_color_model, https://en.wikipedia.org/wiki/RGB_color_model

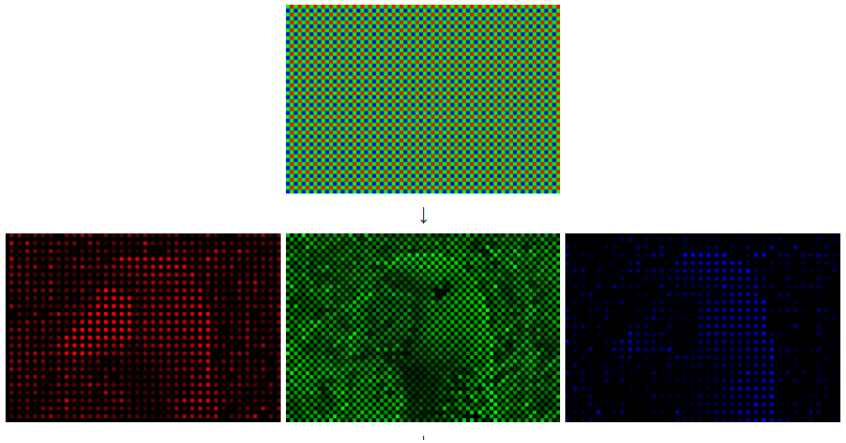
Digital Camera as a Sensor

The Matrix

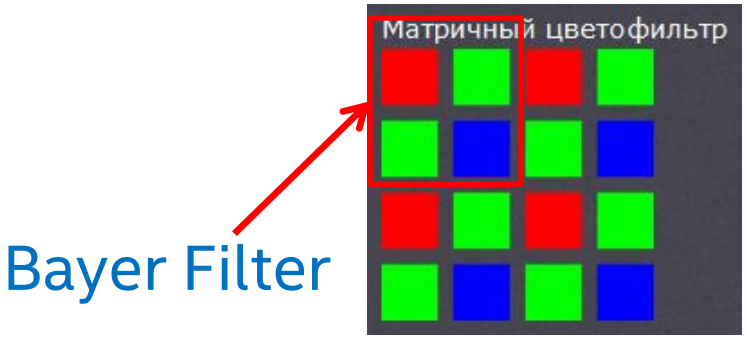


Bryce Bayer

Raw Image



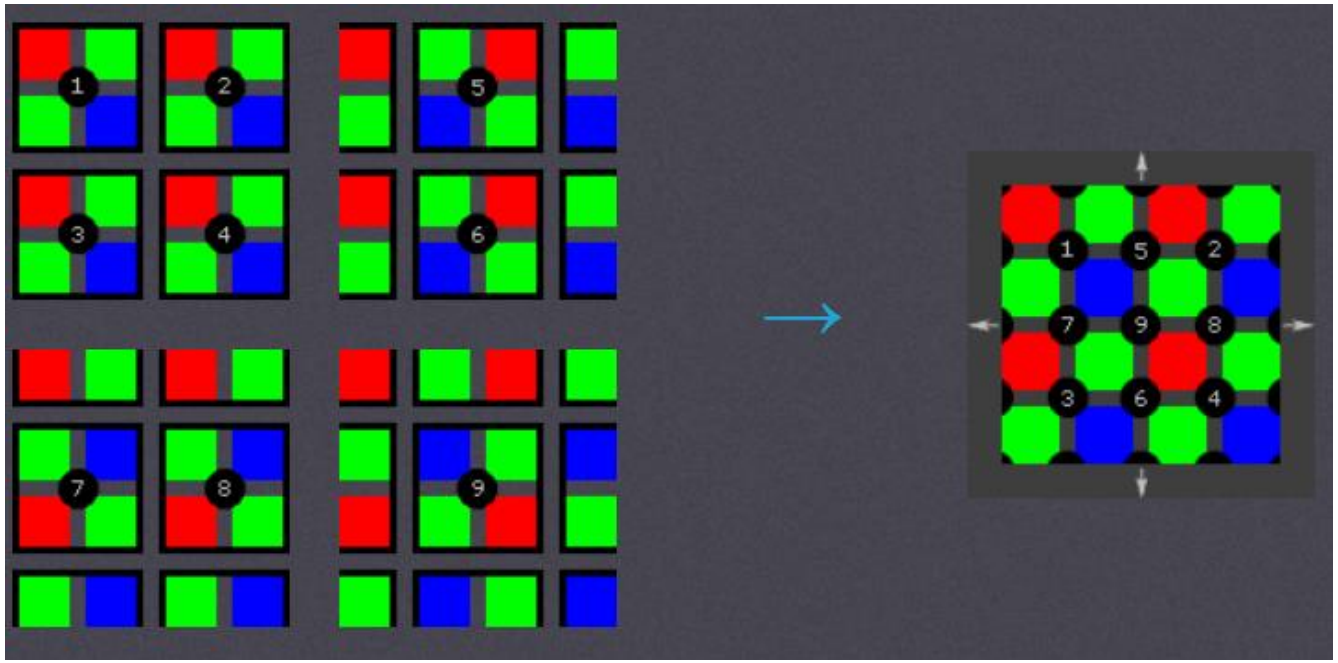
Color Filter Array (Mosaic)

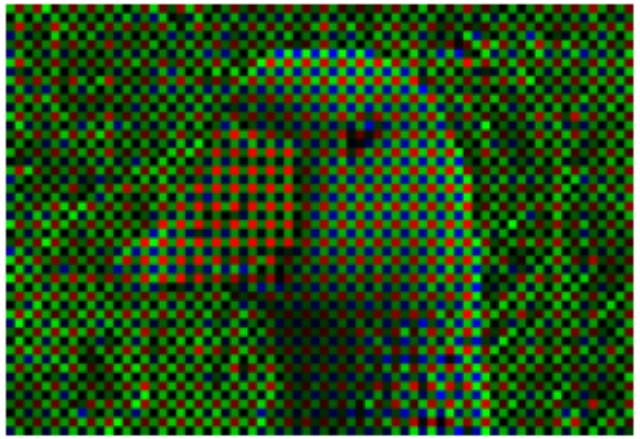


Images credit: https://www.cambridgeincolour.com/ru/tutorials-ru/camera-sensors.htm

Digital Camera as a Sensor

Demosaicing (Debayering)





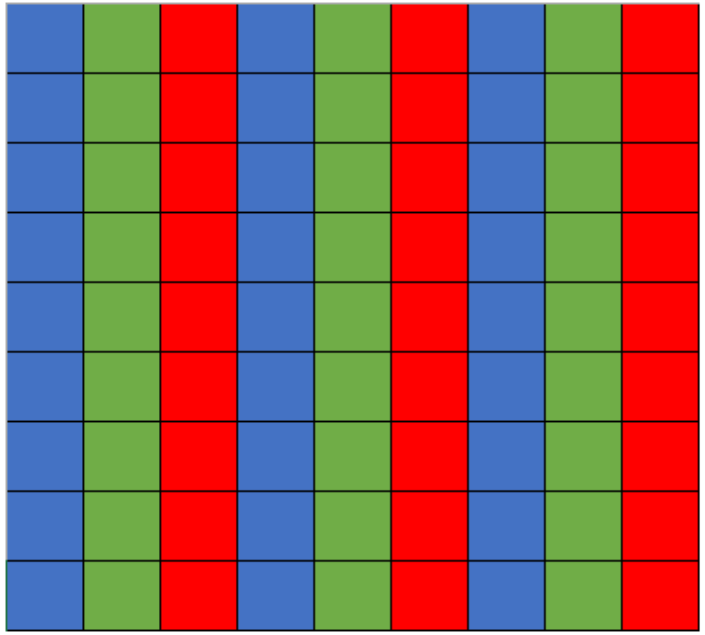
Pixels, which they sell

Pixels, which we obtain

Images credit: https://www.cambridgeincolour.com/ru/tutorials-ru/camera-sensors.htm

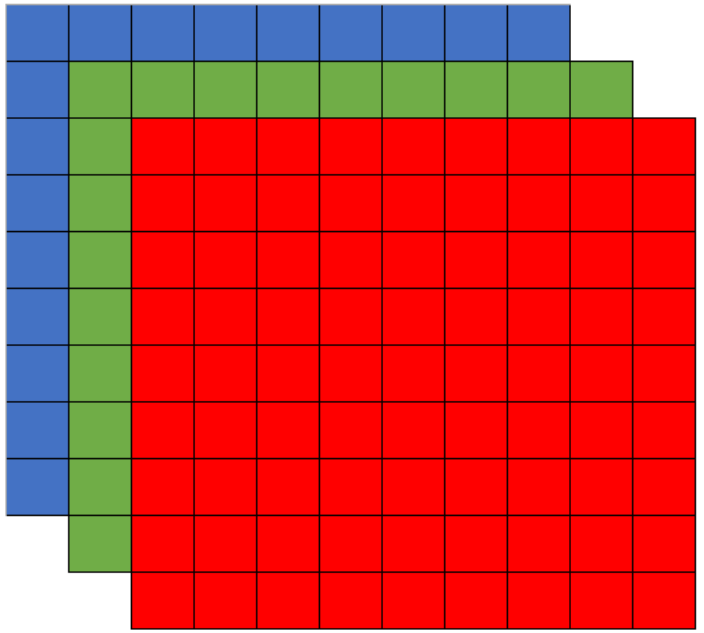
Image Channels

Interleaved (packed) – element of image planes interleaved in memory (BGR, HSV, Luv etc.):



cv::imread default: CV_8UC3.

Non-Interleaved (Planar) – image planes stored in memory one by one (YUV_I420 etc.):

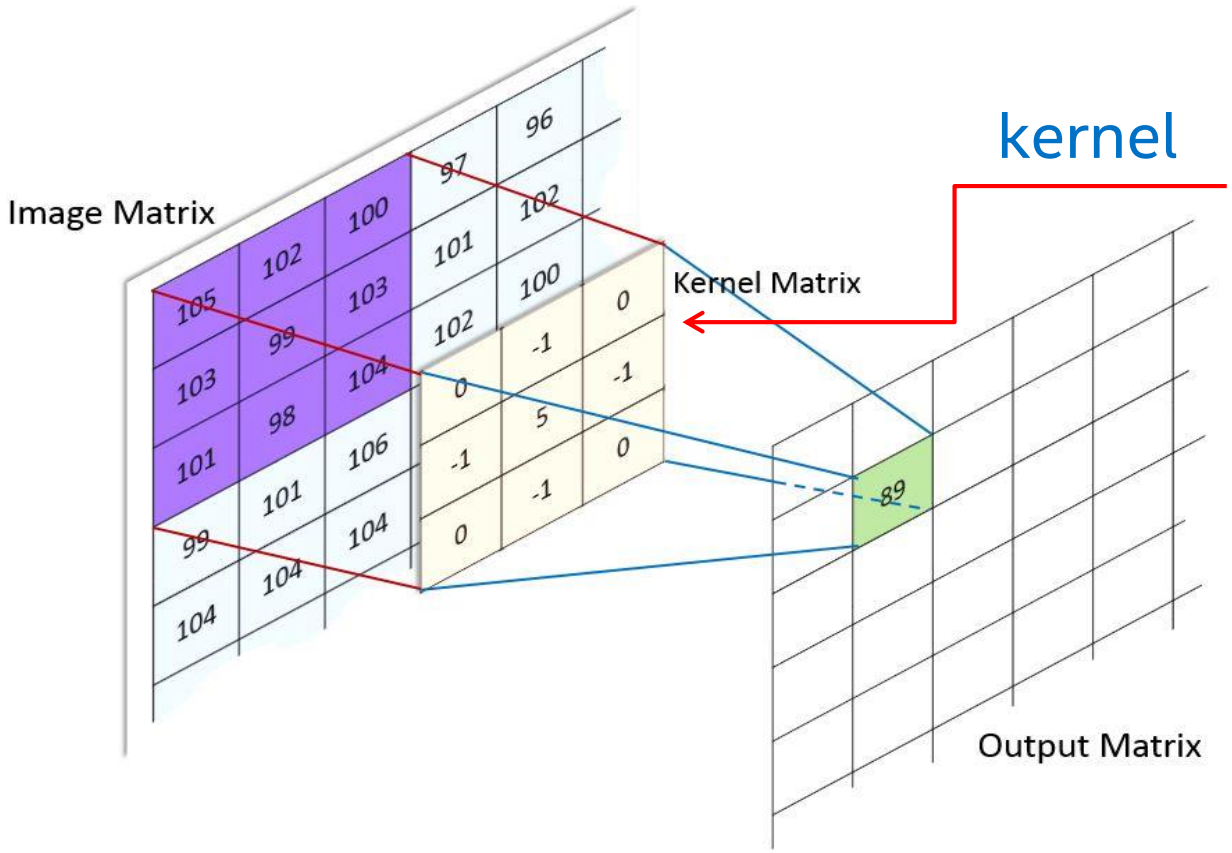


Used as input for Neural Nets (usually).

Computer Vision Topics

- Image Processing
- Keypoint Detection
- Segmentation
- Motion Analysis
- Structure from Motion
- Computational Photography
- Stereo, 3D Reconstruction
- Recognition

Image Processing: Convolution



$$y(t) = x(t) ** h(t) = \int_{-\infty}^{+\infty} x(\tau)h(t-\tau)d\tau$$

$$y[n] = x[n] ** h[n] = \sum_{k=-\infty}^{+\infty} x[k]h[n-k]$$

Images credit: http://machinelearninguru.com/computer_vision/basics/convolution/image_convolution_1.html

Image Processing: Convolution

- Low-pass (Blur, Gaussian Blur, Box Blur…)

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} * \frac{1}{9} \qquad \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} * \frac{1}{25} \qquad \frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Image Processing: Convolution

- Low-pass (Blur, Gaussian Blur, Box Blur…)

Q: Which kernel doesn't change the image?
Q: Which kernel shifts image one pixel down?

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} * \frac{1}{9}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} * \frac{1}{25}$$

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Image Processing: Convolution (Bonus: Zacepila by Artur Pirozhkov)

- Low-pass (Blur, Gaussian Blur, Box Blur…)

Images credit: https://www.youtube.com/watch?v=O0EYAQhLzK0, https://www.youtube.com/watch?v=XQYNUwYHV1c

Image Processing: Convolution

- High-pass (Sharpening, Sobel)

Input image

Horizontal Sobel

Vertical sobel

Combined Sobel

−1	0	1
−2	0	2
−1	0	1

−1	−2	−1
0	0	0
1	2	1

direction of gradients

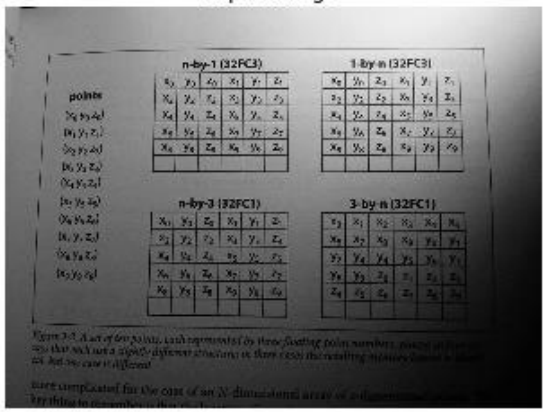
-1	-1	-1
-1	8	-1
-1	-1	-1

0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0

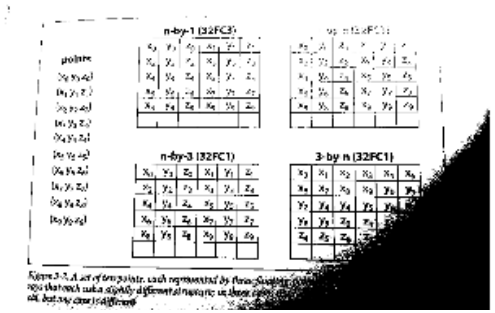
Image Processing: Threshold

- Threshold – the simplest segmentation method
- Adaptive Threshold, Otsu

Input Image

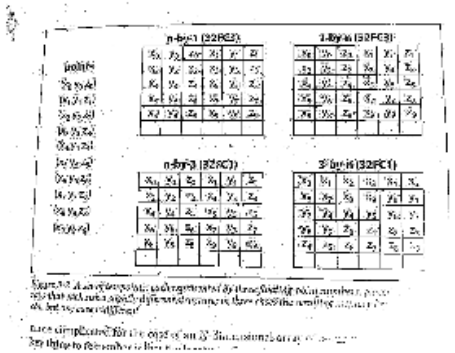


Fixed Thresholding



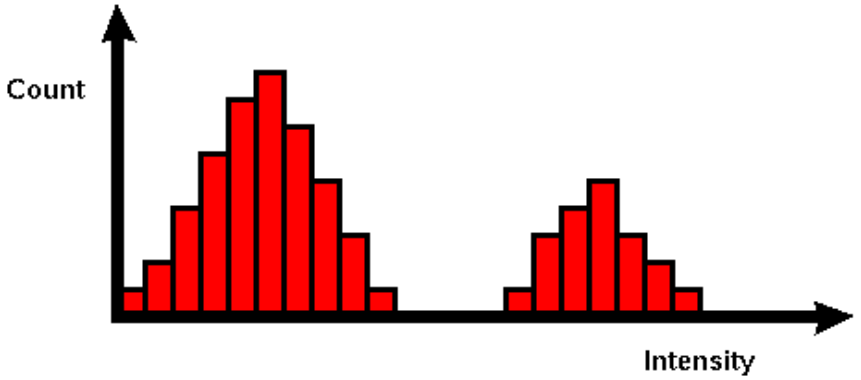
$$\texttt{dst}(x,y) = \begin{cases} \texttt{maxVal} & \text{if } \texttt{src}(x,y) > \texttt{thresh} \\ 0 & \text{otherwise} \end{cases}$$

Adaptive Thresholding with mean weighted average



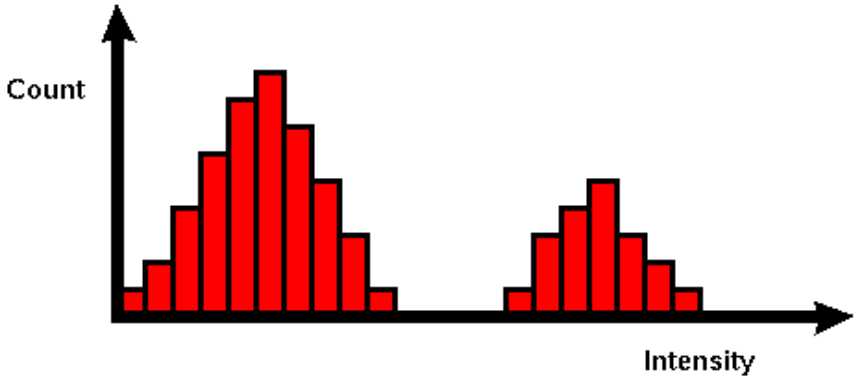
$$\texttt{dst}(x,y) = \begin{cases} \texttt{maxValue} & \text{if } \texttt{src}(x,y) > T(x,y) \\ 0 & \text{otherwise} \end{cases}$$

Image Processing: Histogram

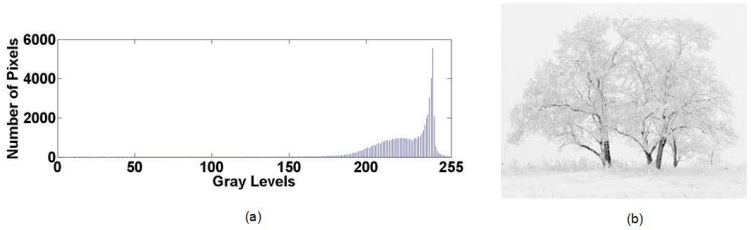


$$Hist[i] = \sum_{y,x} (Image[y,x] == i)$$

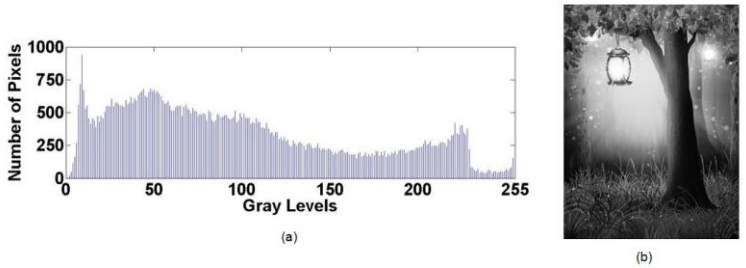
Image Processing: Histogram



$$Hist[i] = \sum_{y,x} (Image[y,x] == i)$$

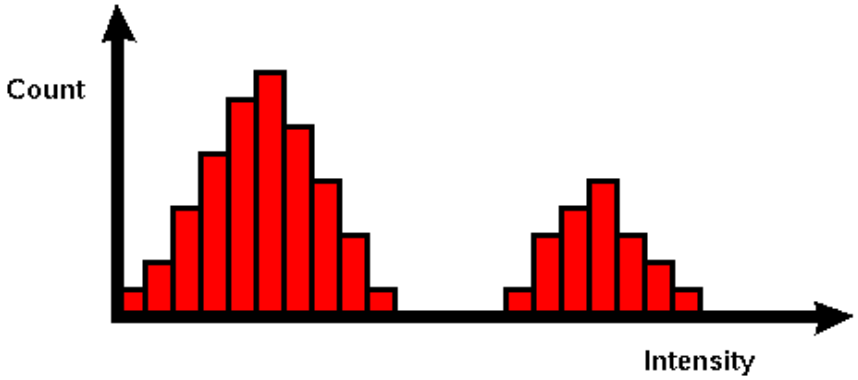


Histogram of a bright image



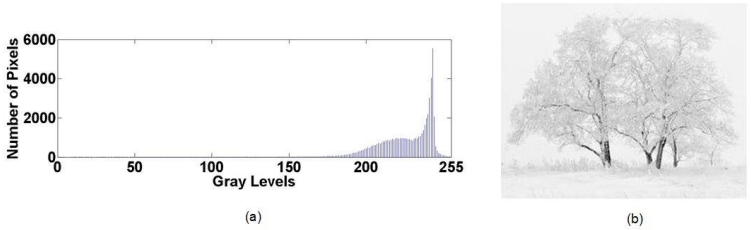
Histogram of a high-contrast image

Image Processing: Histogram

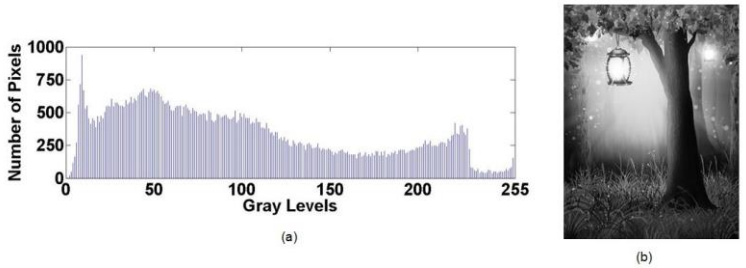


$$Hist[i] = \sum_{y,x} (Image[y,x] == i)$$

Q: How looks histogram of this image?

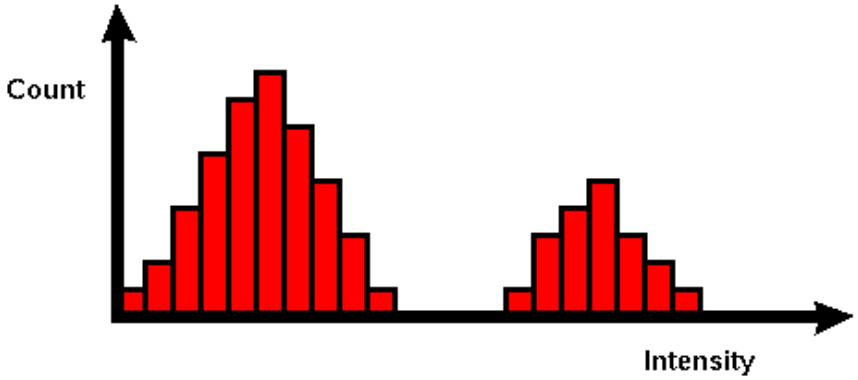


Histogram of a bright image

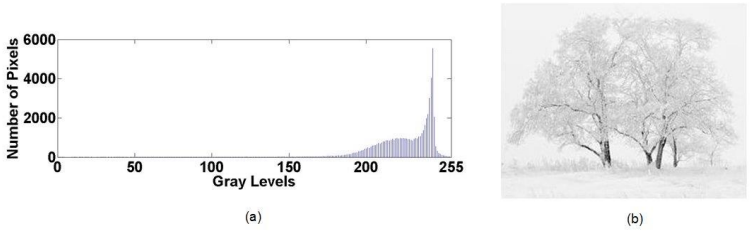


Histogram of a high-contrast image

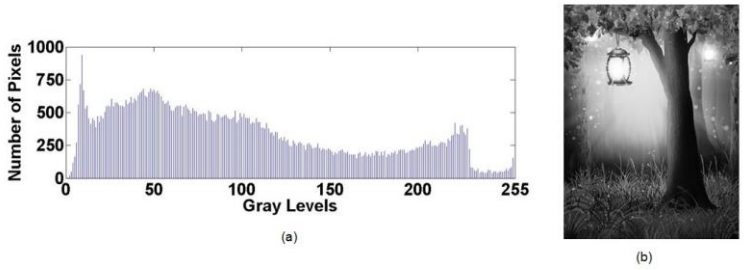
Image Processing: Histogram



$$Hist[i] = \sum_{y,x} (Image[y,x] == i)$$



Histogram of a bright image



Histogram of a high-contrast image

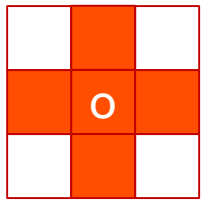
Image after histogram equalization

Image Processing: Morphology

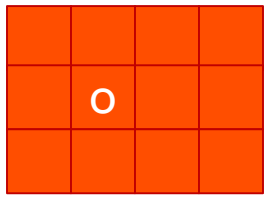
Morphology deals with shape processing.

Compares ∀ pixels in the image against its neighbors, so as to add or remove, brighten or darken such pixels.

Neighborhood is determined by structuring element (or kernel):



Cross



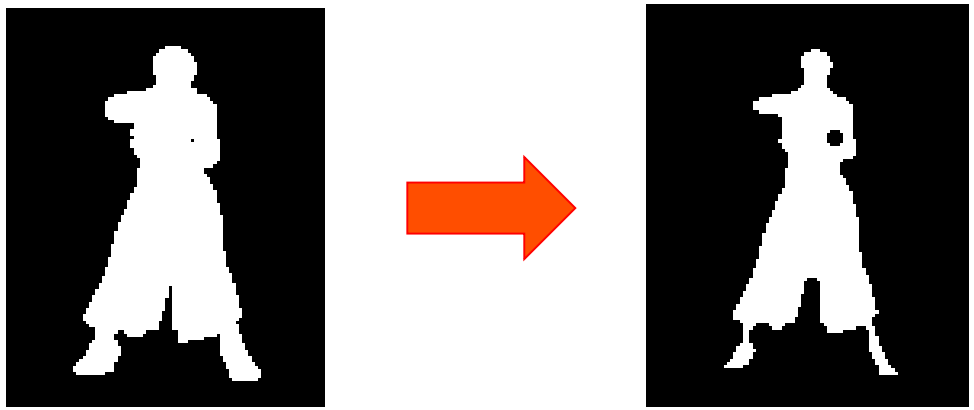
Rectangle

Can be obtained with `cv::getStructuringElement`, or just pass desired kernel in cv::Mat.

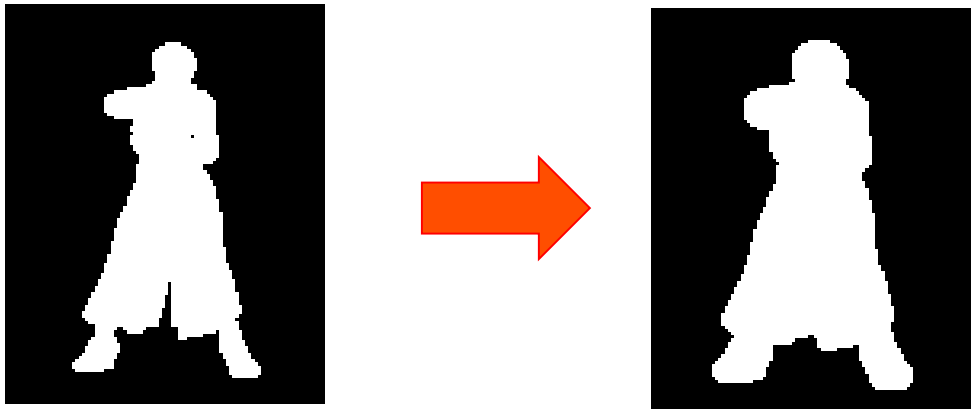
Image Processing: Morphology

Basic operations:

- Erosion (`cv::erode`) – "eats away" shape, reachable from any background pixel



- Dilation (`cv::dilate`) – expands shape



Images credit: http://www.imagemagick.org/Usage/morphology/

Image Processing: Morphology

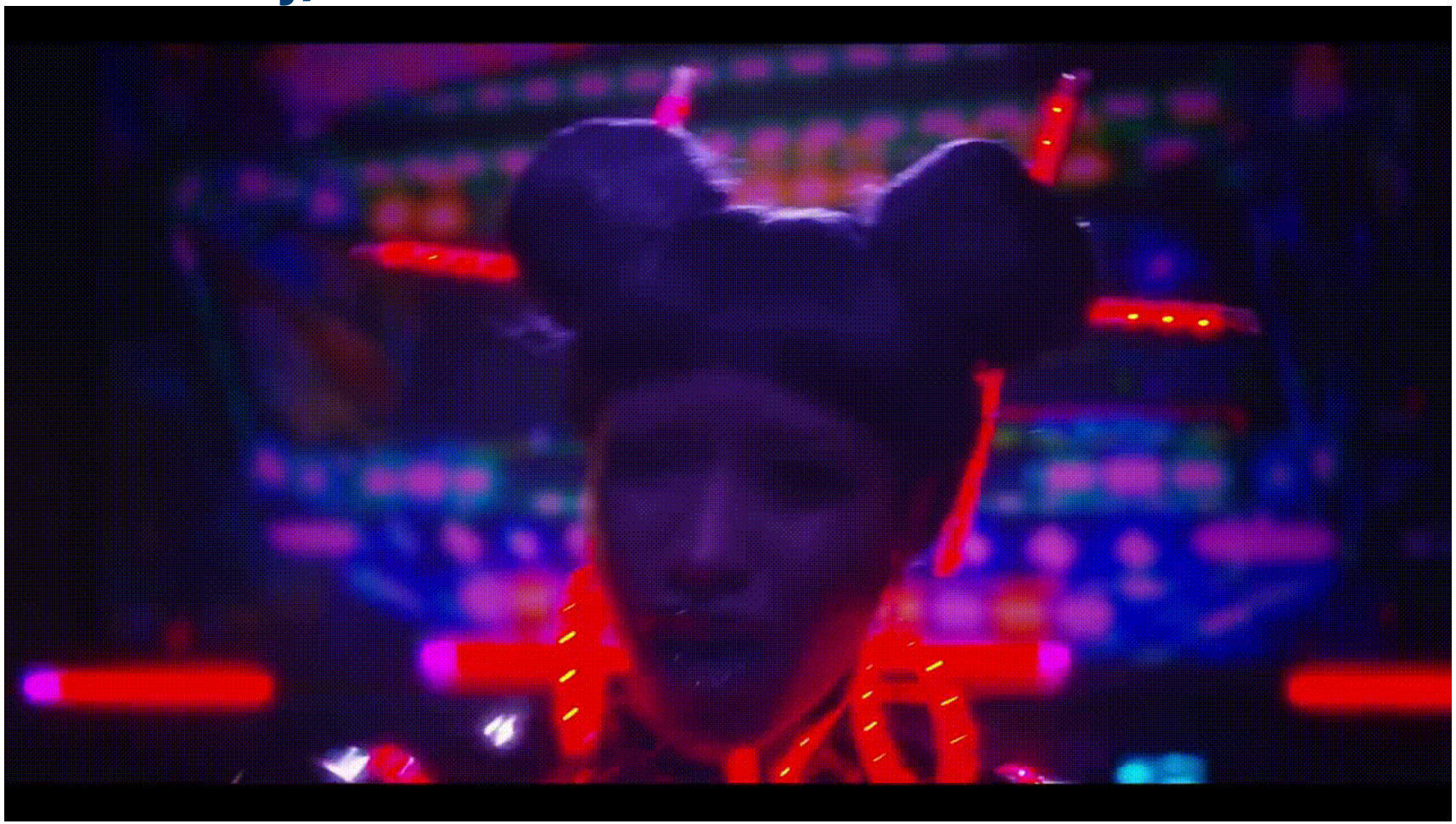
Basic operations:

- Opening (`cv::morhologyEx`) – erosion then dilation

- Closing (`cv::morhologyEx`) – dilation then erosion

Images credit: http://www.imagemagick.org/Usage/morphology/

Image Processing: Morphology (Bonus: Chun-Li by Nicki Minaj)



Images credit: https://www.youtube.com/watch?v=Wpm07-BGJnE

Image Processing: Morphology (Bonus: Chun-Li by Nicki Minaj)

Original (colors negated)

Morphological gradient

Images credit: https://www.youtube.com/watch?v=Wpm07-BGJnE

Summary

Image:

- Non-Interleaved (Planar)
- Interleaved (Packed)
- Colorspaces (Grayscale, RGB, HSV, CMYK)

Convolution:

- Low-pass filter (blur)
- High-pass filter (sharpening, gradients)

Morphology:

- Erosion (shrink), Opening (remove noise)
- Dilation (expand), Closing (fill holes)

Keypoint detection:

- Corners
- Image pyramid

Q&A

daniil.osokin@intel.com