Image Processing (imgproc module)

In this section you will learn about the image processing (manipulation) functions inside OpenCV.

- **Basic Drawing**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  We will learn how to draw simple geometry with OpenCV!

- **Random generator and text with OpenCV**

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  We will draw some *fancy-looking* stuff using OpenCV!

- **Smoothing Images**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  Let's take a look at some basic linear filters!

Linear filtering is one of the most powerful image enhancement methods. It is a process in which part of the signal frequency spectrum is modified by the transfer function of the filter. In general, the filters under consideration are linear and shift-invariant, and thus, the output images are characterized by the convolution sum between the input image and the filter impulse response

high-pass filtering usually sharpens the edges of an image. They can even be used for edge detection, which is used in image analysis algorithms.

- **Eroding and Dilating**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  Let's *change* the shape of objects!

- **More Morphology Transformations**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  Here we investigate different morphology operators

- **Hit-or-Miss**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.4

  Author: Lorena García

  Learn how to find patterns in binary images using the Hit-or-Miss operation

- **Extract horizontal and vertical lines by using morphological operations**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Theodore Tsesmelis

  Here we will show how we can use different morphological operators to extract horizontal and vertical lines

- **Image Pyramids**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

Author: Ana Huamán

What if I need a bigger/smaller image?

- **Basic Thresholding Operations**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  After so much processing, it is time to decide which pixels stay

- **Thresholding Operations using inRange**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Rishiraj Surti

  Thresholding operations using inRange function.

- **Making your own linear filters!**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  Where we learn to design our own filters by using OpenCV functions

- **Adding borders to your images**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  Where we learn how to pad our images

- **Sobel Derivatives**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  Where we learn how to calculate gradients and use them to detect edges

- **Laplace Operator**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  Where we learn about the *Laplace* operator and how to detect edges with it

- **Canny Edge Detector**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  Where we learn a sophisticated alternative to detect edges

- **Hough Line Transform**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

Author: Ana Huamán

Where we learn how to detect lines

- **Hough Circle Transform**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  Where we learn how to detect circles

- **Remapping**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  Where we learn how to manipulate pixels locations

- **Affine Transformations**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  Where we learn how to rotate, translate and scale our images

- **Histogram Equalization**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  Where we learn how to improve the contrast in our images

- **Histogram Calculation**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  It plots the number of pixels for each tonal value

  Where we learn how to create and generate histograms

- **Histogram Comparison**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  Where we learn to calculate metrics between histograms

- **Back Projection**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  Where we learn how to use histograms to find similar objects in images

- **Template Matching**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

Author: Ana Huamán

Where we learn how to match templates in an image

- **Finding contours in your image**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  Where we learn how to find contours of objects in our image

- **Convex Hull**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  Where we learn how to get hull contours and draw them

- **Creating Bounding boxes and circles for contours**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  Where we learn how to obtain bounding boxes and circles for our contours

- **Creating Bounding rotated boxes and ellipses for contours**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  Where we learn how to obtain rotated bounding boxes and ellipses for our contours

- **Image Moments**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  Where we learn to calculate the moments of an image

- **Point Polygon Test**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Ana Huamán

  Where we learn how to calculate distances from the image to contours

- **Image Segmentation with Distance Transform and Watershed Algorithm**

  Languages: C++, Java, Python

  Compatibility: > OpenCV 2.0

  Author: Theodore Tsesmelis

  Where we learn to segment objects using Laplacian filtering, the Distance Transformation and the Watershed algorithm.

- **Out-of-focus Deblur Filter**

  Languages: C++

  Compatibility: > OpenCV 2.0

Author: Karpushin Vladislav

You will learn how to recover an out-of-focus image by Wiener filter.

- **Motion Deblur Filter**

  Languages: C++

  Compatibility: > OpenCV 2.0

  Author: Karpushin Vladislav

  You will learn how to recover an image with motion blur distortion using a Wiener filter.

- **Anisotropic image segmentation by a gradient structure tensor**

  Languages: C++

  Compatibility: > OpenCV 2.0

  Author: Karpushin Vladislav

  You will learn how to segment an anisotropic image with a single local orientation by a gradient structure tensor.

- **Periodic Noise Removing Filter**

  Languages: C++

  Compatibility: > OpenCV 2.0

  Author: Karpushin Vladislav

  You will learn how to remove periodic noise in the Fourier domain.

Loading [MathJax]/extensions/TeX/AMSsymbols.js

Generated on Fri Dec 21 2018 18:02:53 for OpenCV by doxygen 1.8.12