

Switching over to SimpleCV.

Copyright ©2012 SimpleCV.

June 2, 2012

SimpleCV¹, which stands for Simple Computer Vision, is an easy-to-use Python frame-work that bundles together open source computer vision libraries and algorithms for solving problems. The idea of this document is to provide a quick reference for switching from Matlab and OpenCV to SimpleCV.

Description	Matlab	OpenCV	SimpleCV
Reading an image	<code>imread('lenna.png')</code>	<code>cvLoadImage('lenna.png')</code>	<code>Image('lenna.png')</code>
Converting the image to RGB colorspace	<code>hsv2rgb(hsv_image)</code> or <code>ind2rgb(X, map)</code>	<code>CvtColor(bitmap, retVal, CV_BGR2RGB)</code>	<code>img.toRGB()</code>
Converting the image to BGR colorspace	-	<code>CvtColor(bitmap, retVal, CV_RGB2BGR)</code>	<code>img.toBGR()</code>
Converting the image to HLS colorspace	-	<code>CvtColor(bitmap, retVal, CV_RGB2HLS)</code>	<code>img.toHLS()</code>
Converting the image to HSV colorspace	<code>rgb2hsv(rgb_image)</code>	<code>CvtColor(bitmap, retVal, CV_RGB2HSV)</code>	<code>img.toHSV()</code>
Converting the image to XYZ colorspace	<code>cform = makecform('srgb2xyz'); applycform(rgb,cform);</code>	<code>CvtColor(bitmap, retVal, CV_RGB2XYZ)</code>	<code>img.toXYZ()</code>

¹References : O'Reilly Publication, Practical Computer Vision with SimpleCV by Nathan Oostendorp, Anthony Oliver, and Katherine Scott.

Description	Matlab	OpenCV	SimpleCV
Converting the image to GRAY colorspace	rgb2gray(rgb_image)	CvtColor(bitmap, retVal, CV_RGB2GRAY)	img.toGray()
Create a new, empty OpenCV bitmap	zeros(H, W, C)	SetZero(bitmap)	img.getEmpty(channels)
Full copy of the image	newimg = img	Copy(bitmap, newimg)	img.copy()
Resize the image	imresize(img, scale)	Resize(bitmap, scaled_bitmap)	img.resize(x,y)
Smooth the image	H = fspecial(<i>type</i>); imfilter(I,H)	Smooth(r, ro, algorithm, win_x, win_y, sigma, spatial_sigma)	img.smooth(algorithm_name, aperat sigma, spatial_sigma, grayscale)
Invert image	imcomplement(img)		img.invert()
Horizontally mirror an image	flipdim(img,2)	Flip(bitmap, newimg_bitmap, 1)	img.flipHorizontal()
Vertically mirror an image	flipdim(img,1)	Flip(bitmap, newimg_bitmap, 0)	img.flipVertical()
Stretch filter on a greyscale image	img(img<th_l) = 0; img(img>th_h) = 255	Threshold(grayscale_bitmap, newimg, thresh_low, 255,CV_THRESH_TOZERO)	img.stretch(thresh_low, thresh_high)
Binary threshold of the image	step(vision.Autothresher,img)	Threshold(bitmap, bitmap, thresh, maxv, CV_THRESH_BINARY_INV)	img.binarize(thresh, maxv, blocksize)
Mean color of the image	mean(reshape(im, size(im,1)*size(im,2), size(im,3)))	cv.Avg(bitmap)[0:3]	img.meanColor()
Finds the FeatureSet strongest corners first	corner(img)	GoodFeaturesToTrack(GrayscaleBitmap, eig_image, temp_image, maxnum, minquality, mindistance, None)	img.findCorners(maxnum, minqualit mindistance)
Blobs are continuous light regions	step(vision.BlobAnalysis, fg_img)		img.findBlobs(threshval, minsize, maxsize, threshblocksize, threshcons
Finding the location of a known object	-	HaarDetectObjects(EqualizedGrayscaleBitmap(), cascade.getCascade(), storage, scale_factor, use_canny)	findHaarFeatures(self, cascade, scale min_neighbors, use_canny)
Uploading the Image to Imgur or Flickr	-		img.upload(dest,api_key,api_secret,v

Description	Matlab	OpenCV	SimpleCV
Draw a circle on the Image	step(vision.MarkerInserter, img, pts)		img.drawCircle(ctr, rad, color, thickness)
Draw a line	plot(X_vector, Y_vector)		img.drawLine(pt1, pt2, color, thickness)
Size of image	[size(img,1) size(img,2)]	GetSize(bitmap)	img.size()
Split the image into a series of image chunks	-		img.split(cols, rows)
Split the channels of an image into RGB	r=img(:,:,1); g=img(:,:,2); b=img(:,:,3)	Split(bitmap, b, g, r, None)	img.splitChannels(grayscale)
Images of R,G,B channels are recombined into a single image	cat(3, r, g, b)	Merge(b,g,r,None,retVal)	img.mergeChannels(r,b,g)
Apply a color correction curve in HSL space	-		img.applyHLSCurve(hCurve, lCurve, sCurve)
Apply a color correction curve in RGB space	-		img.applyRGBCurve(rCurve, gCurve, bCurve)
Applies Intensity to all three color channels	-		img.applyIntensityCurve(curve)
Returns image representing the distance of each pixel from a given color tuple	-		img.colorDistance(color)
Apply morphological erosion to a image	imerode(img,SE)	Erode(bitmap, retVal, kern, iterations)	img.erode(iterations)
Apply morphological dilation to a image	imdilate(img,SE)	Dilate(bitmap, retVal, kern, iterations)	img.dilate(iterations)

Description	Matlab	OpenCV	SimpleCV
Histogram equalization on the image	histeq(img, hgram)	cv.EqualizeHist(GrayscaleBitmap, Equalizedgraybitmap)	img.equalize()
Returns Image of the string	-		img.toString()
Applies erosion operation followed by a morphological dilation	imerode(img, SE)	MorphologyEx(bitmap, retVal, temp, kern, CV_MOP_OPEN, 1)	img.morphOpen()
The difference between the morphological dilation and the morphological gradient	-	MorphologyEx(Bitmap, retVal, temp, kern, CV_MOP_GRADIENT, 1)	img.morphGradient()
1D histogram(numpy array) of intensity for pixels in the image	step(vision.Histogram,img)		img.histogram(numbins)
The histogram of the hue channel for the image	-		img.hueHistogram(bins)
Returns the peak hue values histogram of hues	-		img.huePeaks(bins)
Add two images	imadd(img1,img2)	Add(imgBitmap, otherBitmap, newBitmap)	img.__add__(other)
Subtract two images	imsubtract(img1,img2)	Sub(imgBitmap, otherBitmap, newBitmap)	img.__sub__(other)
Or two images	-	Or(imgBitmap, otherBitmap, newBitmap)	img.__or__(other)
Image division operation taking two images as input	imdivide(img1,img2)	Div(imgBitmap, otherBitmap, newBitmap)	img.__div__(other)