OpenCV Notes

1 Camera Distortion

The camera on the quad has a slight radial distortion.

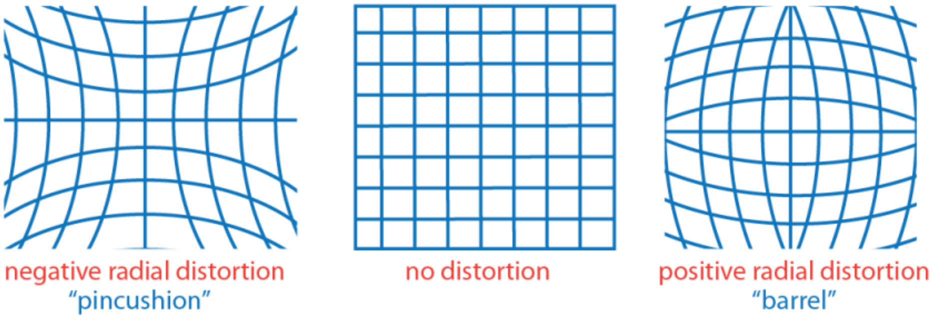


Figure 1: Radial Distortion

$$x_d = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$
$$y_d = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

Undistort using the OpenCv `cv::undistort` function, about which you can read further here:

`https://docs.opencv.org/4.2.0/d4/d94/tutorial_camera_calibration.html`
`https://docs.opencv.org/2.4.13.7/doc/tutorials/calib3d/camera_calibration/camera_calibration.html`

2 Topics

Topic tutorials can be found here:

`https://docs.opencv.org/4.2.0/d9/df8/tutorial_root.html`

1. Color spaces
2. Threshholds
3. Geometric Transforms
4. Smoothing
5. Morphological Transforms
6. Gradients
7. Canny Edges
8. Contours
9. Template Matching

3 API Lookup

`https://docs.opencv.org/4.2.0/index.html`

4 Color Spaces

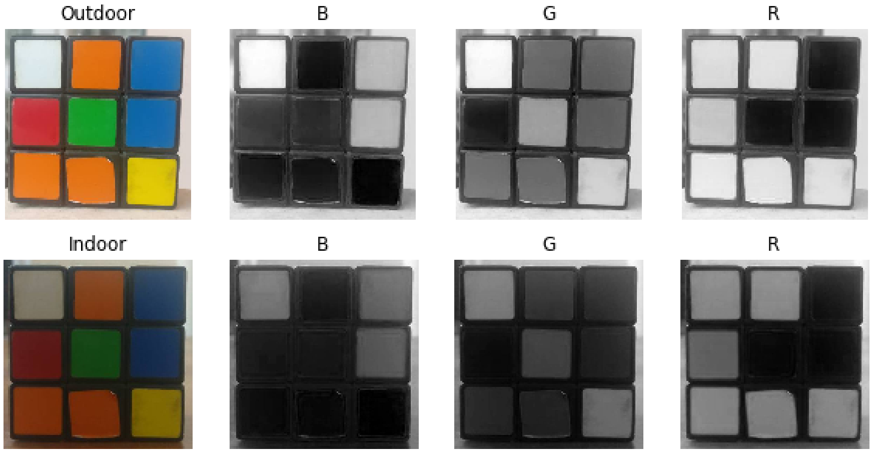
What qualities do color spaces encode?

1. Color (three channels?)
2. Brightness
3. What else?

Over 150 color space conversions in OpenCV. Here are 4 of the most popular color spaces.

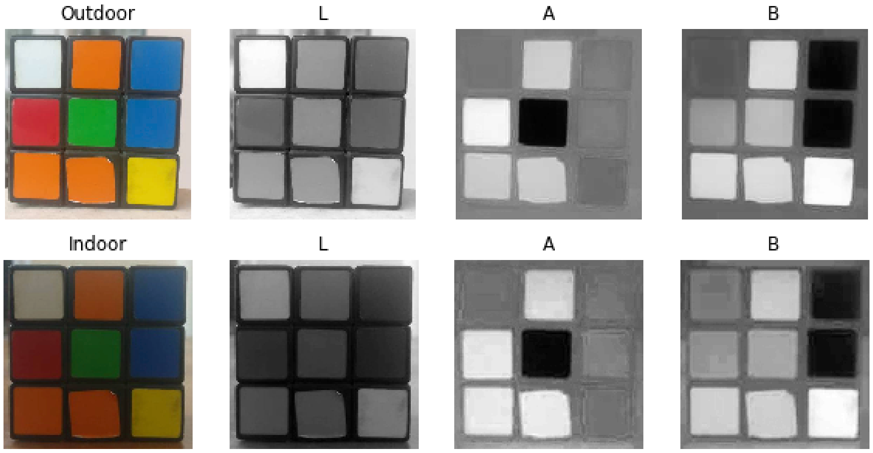
4.1 RGB

1. Additive. Linear combination of Red, Blue, and Green



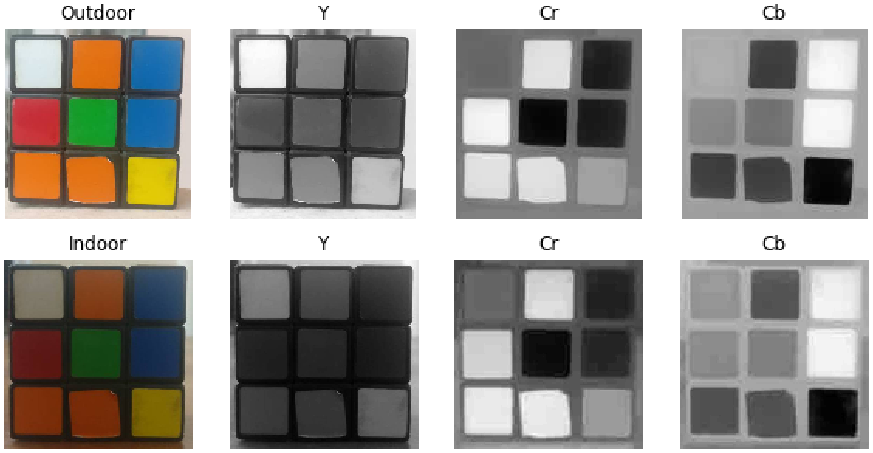
4.2 LAB

1. Lightness (intensity), AB color spaces (green to magenta; blue to yellow)



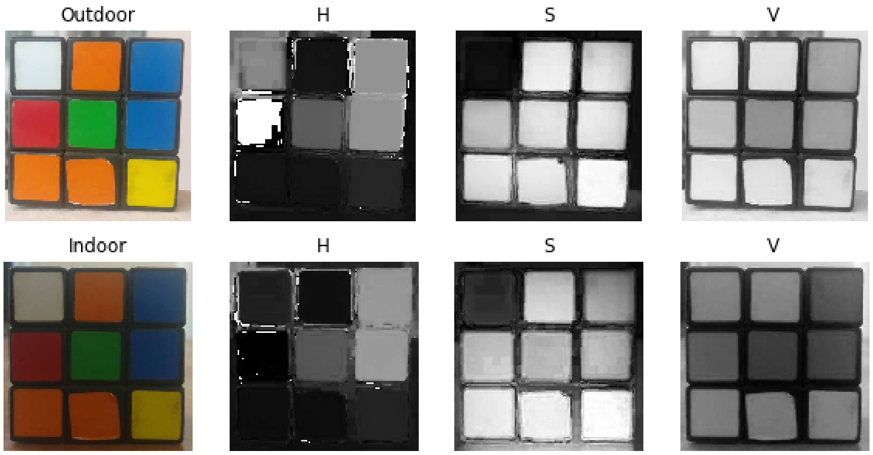
4.3 YCrCb

1. Lumenance, (Red - Yellow), (Blue - Yellow)



4.4 HSV

1. Hue, Saturation, Value
2. Hue: dominant wavelength
3. Saturation: purity/shades of color
4. Value: intensity
5. Encodes color in only one channel (hue)
6. Hue is represented as a circle that starts at 0 and ends at 179, then aliases modulo 180 (so that 185 corresponds to 5). The color red is centered at 0.



4.5 References

1. https://www.learnopencv.com/color-spaces-in-opencv-cpp-python/

5 Smoothing/Blurring

Why?

1. Average out noise
2. Remove sharp corners. Useful for contours (later).

How? With a convolution.

$$K = \frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$$

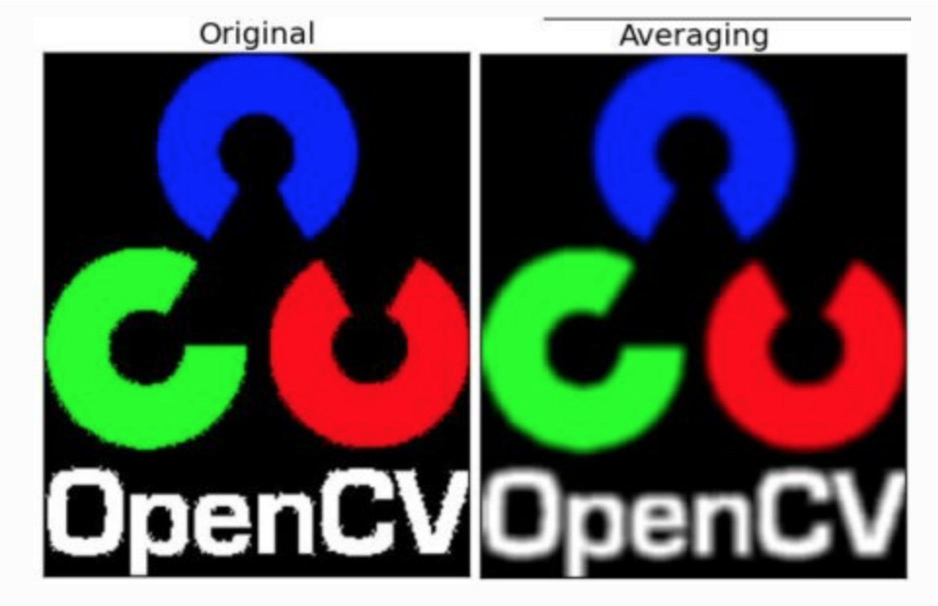


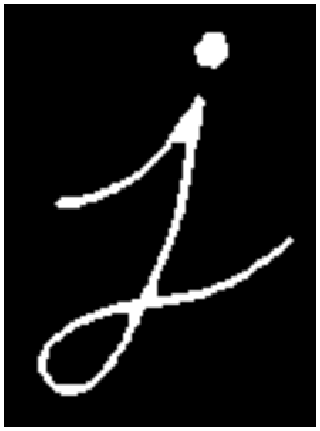
Figure 2: Box Blur

Other blurs include a Gaussian Blur using a Gaussian kernel.

6 Morphological Transforms

6.0.1 Erosion

Erodes the foreground from the background. Applies a kernel over an image. New pixel will be white only if all kernel pixels are white.

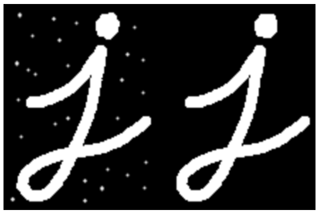


6.0.2 Dialation

Opposite of erosion. A new pixel is white if any of the pixels in the kernel are white.

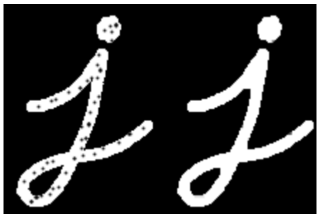
6.0.3 Opening

Erosion followed by dialation. Useful for removing noise.



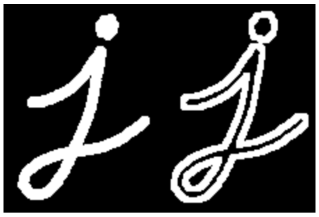
6.0.4 Closing

Dialation followed by erosion. Useful for closing holes.



6.0.5 Morphological Gradient

Difference between opening and closing. Good for outlines.



7 Contours

Contours can be explained simply as a curve joining all the continuous points (along the boundary), having same color or intensity. The contours are a useful tool for shape analysis and object detection and recognition.

For better accuracy, use binary images. Before finding contours, apply threshold or canny edge detection. The findContours function modifies the source image, so if you want source image even after finding contours, store it to some other variables. In OpenCV, finding contours is like finding a white object from black background: object to be found should be white and background should be black.

1. Moments
2. Area
3. Perimeter
4. Convex hull
5. Bounding rectangle
6. Minimum enclosing circle

8 Pattern Matching

8.1 Pattern Matching

Tries to find a template in an image. Takes an input template (image) and uses that as a convolution over a new image.

8.2 Hough Circle

Finds circles within a specified radius.