Image transformation in OpenCV

Asked 10 years, 3 months ago Modified 4 years, 9 months ago Viewed 17k times

22

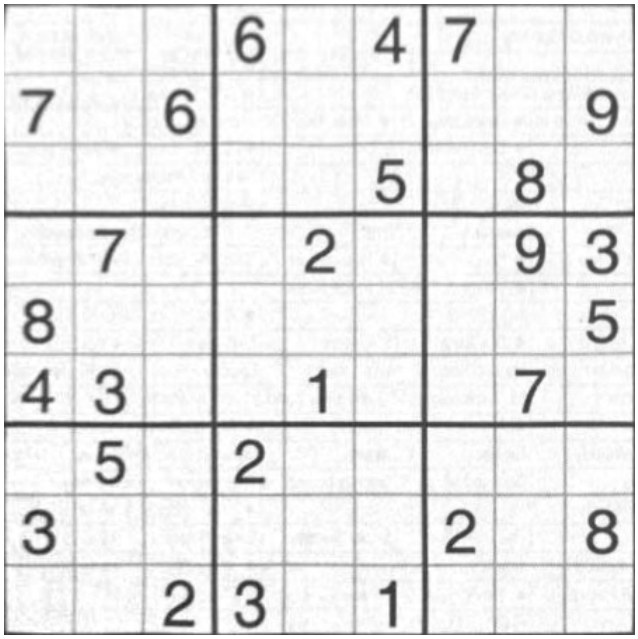
14

This question is related to this question: How to remove convexity defects in sudoku square

I was trying to implement nikie's answer in Mathematica to OpenCV-Python . But i am stuck at the final step of procedure.

ie I got the all intersection points in square like below:

Now, i want to transform this into a perfect square of size (450,450) as given below:



(Never mind the brightness difference of two images).

Question: How can i do this in OpenCV-Python? I am using `cv2` version.

Abid Rahman K
50.5k 29 144 155

2 As he suggested in his answer, just use standa ... r each individual cell ... – etarion Apr 28, 2012 at 14:35

I used warp perspective which used 4 corners ... each cell? isn't it a time consuming process? – Abid Rahman K Apr 28, 2012 at 14:52

Are you talking embedded programming? I guess not, since you're using python ... on a desktop machine, no, it should not be noticeable, since it's only 81 cells ... – etarion Apr 28, 2012 at 16:01

2 @AbidRahmanK: I haven't tested that, but I'd expect that the running time of the perspective warp transformation is proportional to the number of output pixels. So if it was fast enough to transform the whole grid, it should be fast enough transform 81 grid cells that are only 1/81 of the grid size. – Niki Apr 29, 2012 at 21:09

2 Answers

Sorted by: Trending sort available — Highest score (default)

35

Apart from etarion's suggestion, you could also use the remap function. I wrote a quick script to show how you can do this. As you see coding this is really easy in Python. This is the test image:

and this is the result after warping:

And here is the code:

```
import cv2
from scipy.interpolate import griddata
import numpy as np

grid_x, grid_y = np.mgrid[0:149:150j, 0:149:150j]
destination = np.array([[0,0], [0,49], [0,99], [0,149],
                  [49,0],[49,49],[49,99],[49,149],
                  [99,0],[99,49],[99,99],[99,149],
                  [149,0],[149,49],[149,99],[149,149]])
source = np.array([[22,22], [24,68], [26,116], [25,162],
                  [64,19],[65,64],[65,114],[64,159],
                  [107,16],[108,62],[108,111],[107,157],
                  [151,11],[151,58],[151,107],[151,156]])
grid_z = griddata(destination, source, (grid_x, grid_y), method='cubic')
map_x = np.append([], [ar[:,1] for ar in grid_z]).reshape(150,150)
map_y = np.append([], [ar[:,0] for ar in grid_z]).reshape(150,150)
map_x_32 = map_x.astype('float32')
map_y_32 = map_y.astype('float32')

orig = cv2.imread("tmp.png")
warped = cv2.remap(orig, map_x_32, map_y_32, cv2.INTER_CUBIC)
cv2.imwrite("warped.png", warped)
```

I suppose you can google and find what griddata does. In short, it does interpolation and here we use it to convert sparse mappings to dense mappings as cv2.remap requires dense mappings. We just need to convert to the values to float32 as OpenCV complains about the float64 type. Please let me know how it goes.

Update: If you don't want to rely on Scipy, one way is to implement the 2d interpolation function in your code, for example, see the source code of griddata in Scipy or a simpler one like this http://inasafe.readthedocs.org/en/latest/_modules/engine/interpolation2d.html which

fireant
13.4k 4 35 47

1 Fine, that was marvelous. This is what i was looking for. I knew i could use `cv2.remap` , but not how to use. But is it necessary to use scipy? can't it be done with numpy and cv2 only? Also can be the same method used for whole image in a single step(ie without splitting into sub-blocks)? – Abid Rahman K Apr 29, 2012 at 18:39

It was done for a part of the image to make it easier for myself typing the coordinates of the red points. In "source" you have the coordinates of the red points you have already detected and in "destination" their coordinates in the regular grid. If you'd shared your code I could show how to do it in the code, no need to split the image into subregions. Reg. no use of Scipy, see the update in the answer. – fireant Apr 29, 2012 at 22:04

`I'd like to hear how your final code solves Sudokus` . which one do you want? how to solve the sudoku grid? or the opencv version on how to do ocr etc.? (solving sudoku grid is taken from a blog , ocr part etc written by myself) – Abid Rahman K Apr 30, 2012 at 6:38

I meant an update from you when the sudoku solver is ready would be nice. If that answered your question, you could reward me by selecting as the answer :-) – fireant May 1, 2012 at 14:59

1 Hi, what is the use of creating grid_x, grid_y in this code? what does they mean? – Abid Rahman K Jan 14, 2013 at 14:20

1

if you have source points and end points (you only need 4), you can plug them into cv2.getPerspectiveTransform, and use that result in cv2.warpPerspective. Gives you a nice flat result.

Share Follow

answered Oct 2, 2017 at 3:36

john ktejik
5,627 4 48 52