Removing watermark out of an image using OpenCV

Asked 5 years, 8 months ago Active 1 year, 3 months ago Viewed 19k times

56

39

First of all I have this image and I want to make an application that can detect images like it and remove the circle (watermark) from it.

* Problem

Using Ziegler - Nichols tuning rules, design a suitable PID controller for the following system:

$$G(s) = \frac{1}{s(s+1)(s+5)} \quad , \quad H(s) = 1$$

* Sol.

- $G(s)$ has a free integrator (type-1) $\Rightarrow$ use 2nd method:
- clch eq.: $1 + k\,G(s)\cdot H(s) = 1 + \frac{k}{s(s+1)(s+5)} = 0$

$$\therefore\ s(s^2+6s+5) + k = 0$$

$$\Rightarrow\ s^3 + 6s^2 + 5s + k = 0$$

$$\therefore\ a = \frac{6*5 - k}{6}$$

s^3	1	5
s^2	6	k
s	a	0
s^0	k	

To get k_{cr}: $a = 0 \Rightarrow k_{cr} = 30$

To get ω_{cr}: $6s^2 + k_{cr} = 0 \Rightarrow -6\omega_{osc}^2 + 30 = 0$

$$\Rightarrow \omega_{cr} = \sqrt{5}\ \text{r/s} \Rightarrow P_{cr} = \frac{2\pi}{\sqrt{5}}$$

From Formulas: $G_c(s) = 0.6\,k_{cr}\left(1 + \frac{1}{0.5\,P_{cr}\,s} + 0.125\,P_{cr}\,s\right)$

$$\Rightarrow G_c(s) = 18\left(1 + \frac{0.711}{s} + 0.3512\,s\right)$$ PID Design.

```
int main(){
    Mat im1,im2,im3,gray,gray2,result;

    im2=imread(" (2).jpg");
    namedWindow("x",CV_WINDOW_FREERATIO);
    imshow("x",im2);

    //converting it to gray
    cvtColor(im2,gray,CV_BGR2GRAY);
    // creating a new image that will have the cropped ellipse
    Mat ElipseImg(im2.rows,im2.cols,CV_8UC1,Scalar(0,0,0));

    //detecting the largest circle
    GaussianBlur(gray,gray,Size(5,5),0);
    vector<Vec3f> circles;
    HoughCircles(gray,circles,CV_HOUGH_GRADIENT,1,gray.rows/8,100,100,100,0);

    uchar x;
    int measure=0;int id=0;
    for(int i=0;i<circles.size();i++){
        if(cvRound(circles[i][2])>measure && cvRound(circles[i][2])<1000){
            measure=cvRound(circles[i][2]);
            id=i;
        }
    }


    Point center(cvRound(circles[id][0]),cvRound(circles[id][1]));
    int radius=cvRound(circles[id][2]);
    circle(im2,center,3,Scalar(0,255,0),-1,8,0);
    circle(im2,center,radius,Scalar(0,255,0),2,8,0);
    ellipse(ElipseImg,center,Size(radius,radius),0,0,360,Scalar(255,255,255),-1,8);
    cout<<"center: "<<center<<" radius: "<<radius<<endl;



    Mat res;
    bitwise_and(gray,ElipseImg,result);
    namedWindow("bitwise and",CV_WINDOW_FREERATIO);
    imshow("bitwise and",result);

    // trying to estimate the Intensity  of the circle for the thresholding
    x=result.at<uchar>(cvRound(circles[id][0]+30),cvRound(circles[id][1]));
    cout<<(int)x;

    //thresholding the  output image
    threshold(ElipseImg,ElipseImg,(int)x-10,250,CV_THRESH_BINARY);
    namedWindow("threshold",CV_WINDOW_FREERATIO);
    imshow("threshold",ElipseImg);

    // making bitwise_or
    bitwise_or(gray,ElipseImg,res);
    namedWindow("bitwise or",CV_WINDOW_FREERATIO);
    imshow("bitwise or",res);

    waitKey(0);
}
```

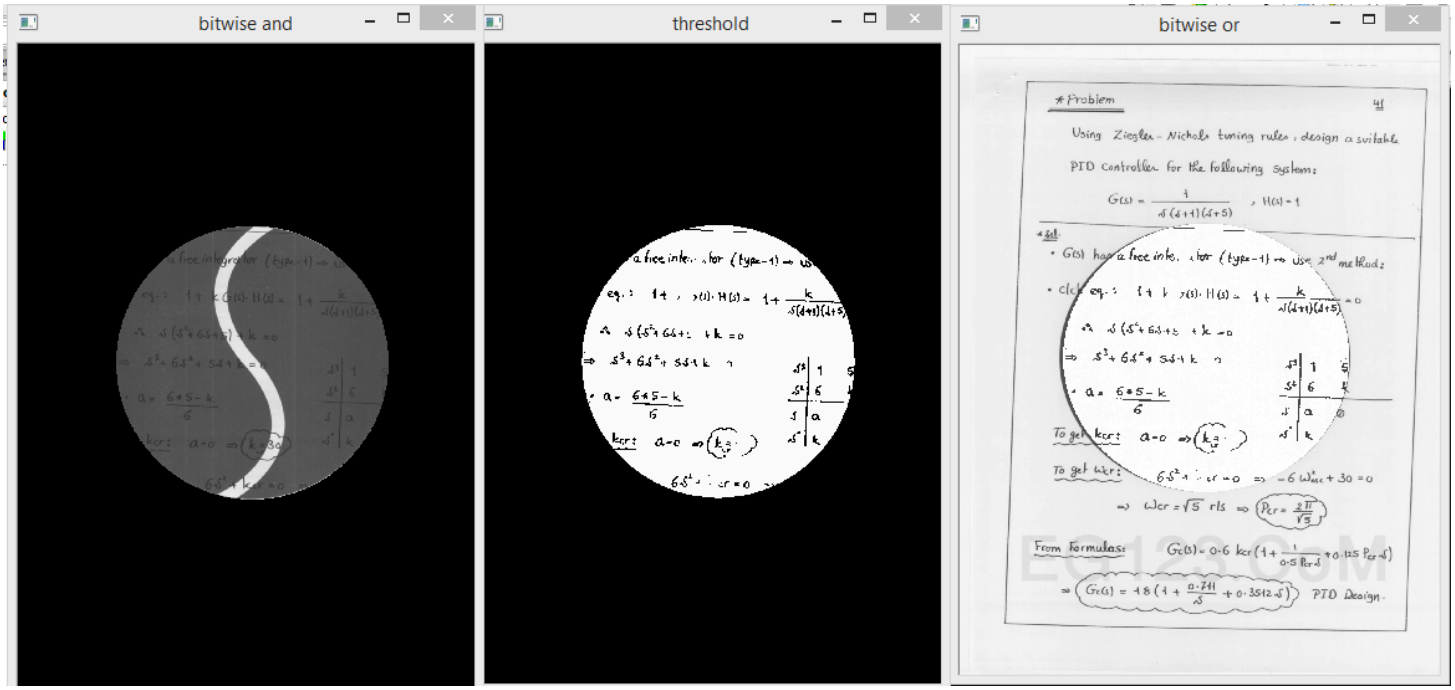
So far what I made is:

1. I convert it to grayscale
2. I detect the largest circle using Hough circles and then make a circle with same radius in a new image

5. `bitwise_or` the result of the threshold

My problem is that any black text on the curved white line inside this circle didn't appear. I tried to remove the color by using the pixel values instead of threshold, but the problem is the same. So any solutions or suggestions?

These are the results:



c++ opencv image-processing watermark

Share Improve this question Follow

edited Feb 4 '20 at 8:34
Joel G
21 2 7

asked Aug 20 '15 at 18:03
Ahmed Ramzy
724 1 9 17

2 Answers

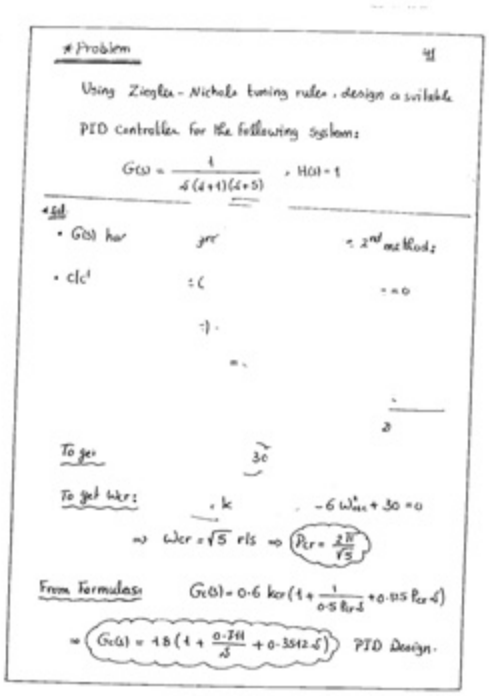
Active | Oldest | Votes

I'm not sure if the following solution is acceptable in your case. But I think it performs slightly better, and doesn't care about the shape of the watermark.

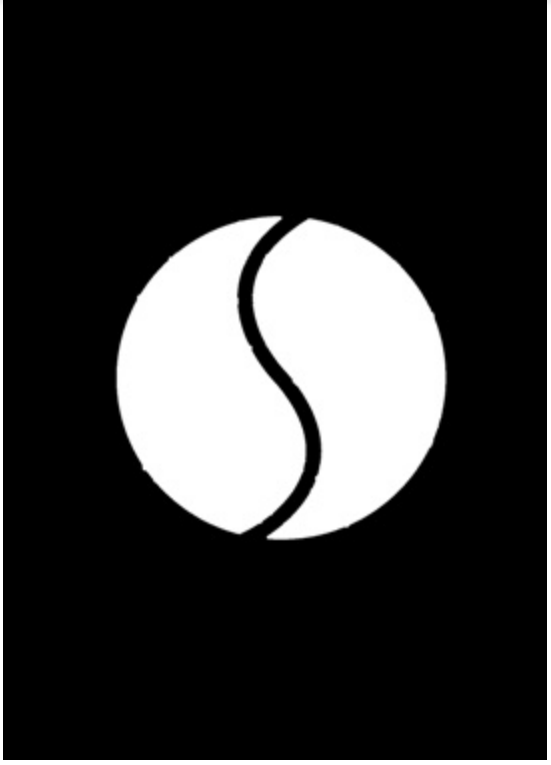
44

- Remove the strokes using morphological filtering. This should give you a background image.

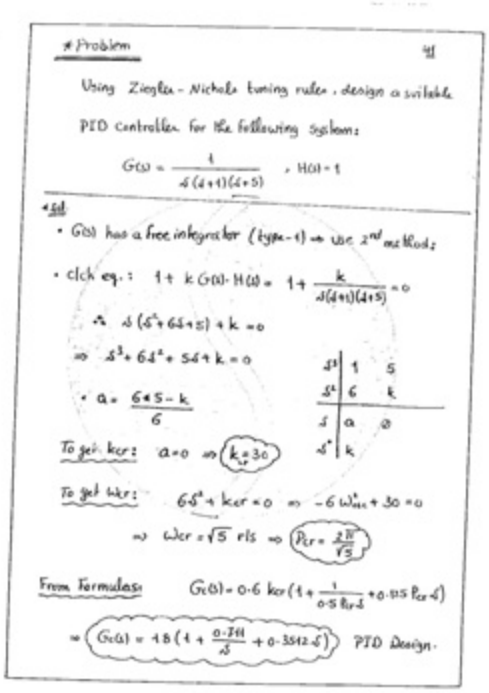
- Calculate the difference image: difference = background - initial, and threshold it: binary = threshold(difference)



- Threshold the background image and extract the dark region covered by the watermark



- From the initial image, extract pixels within the watermark region and threshold these pixels, then paste them to the earlier binary image



Above is a rough description. Code below should explain it better.

```
Mat im = [load the color image here];

Mat gr, bg, bw, dark;

cvtColor(im, gr, CV_BGR2GRAY);
```

```
    Mat kernel2 = getStructuringElement(MORPH_ELLIPSE, Size(2*r+1, 2*r+1));
    morphologyEx(bg, bg, CV_MOP_CLOSE, kernel2);
    morphologyEx(bg, bg, CV_MOP_OPEN, kernel2);
}

// difference = background - initial
Mat dif = bg - gr;
// threshold the difference image so we get dark letters
threshold(dif, bw, 0, 255, CV_THRESH_BINARY_INV | CV_THRESH_OTSU);
// threshold the background image so we get dark region
threshold(bg, dark, 0, 255, CV_THRESH_BINARY_INV | CV_THRESH_OTSU);

// extract pixels in the dark region
vector<unsigned char> darkpix(countNonZero(dark));
int index = 0;
for (int r = 0; r < dark.rows; r++)
{
    for (int c = 0; c < dark.cols; c++)
    {
        if (dark.at<unsigned char>(r, c))
        {
            darkpix[index++] = gr.at<unsigned char>(r, c);
        }
    }
}
// threshold the dark region so we get the darker pixels inside it
threshold(darkpix, darkpix, 0, 255, CV_THRESH_BINARY | CV_THRESH_OTSU);

// paste the extracted darker pixels
index = 0;
for (int r = 0; r < dark.rows; r++)
{
    for (int c = 0; c < dark.cols; c++)
    {
        if (dark.at<unsigned char>(r, c))
        {
            bw.at<unsigned char>(r, c) = darkpix[index++];
        }
    }
}
```

Share Improve this answer Follow

answered Aug 21 '15 at 13:02

dhanushka

9,622 2 30 44

Awesome , it worked very well but i have problem with darker pages - darker watermark - it just copy the whole watermark to the bw image so it's like nothing done at the end , how can i deal with something like that ? – Ahmed Ramzy Aug 21 '15 at 15:39

1 Check the intermediate images: the **difference**, **background**, the **watermark mask** and the **intermediate binary**. Here we use Otsu method, so the images subjected to thresholding had better be bimodal. You can check if the letters inside the watermark are segmented as expected by cropping a part of watermark that contains text and applying Otsu thresholding to it. It could also be a matter of CV_THRESH_BINARY vs CV_THRESH_BINARY_INV. – dhanushka Aug 22 '15 at 10:42

@dhanushka Please Can someone help with the Java Code for the vector Looping Part.I cannot find something equivalent in Java. I posted Question here --> answers.opencv.org/question/130997/... – Vishal Nair Mar 2 '17 at 21:30

@VishalNair I'm not very familiar with opencv java interface. But a simple google search pointed here and here. So, basically you can use `Mat::get` and `Mat::put` methods. The `darkpix` will have to be an opencv `Mat` if java interface does not operate on java vectors. – dhanushka Mar 3 '17 at 2:33

5

A Python version of dhanushka's answer

```
# Import the necessary packages
import cv2
import numpy as np


def back_rm(filename):
    # Load the image
    img = cv2.imread(filename)

    # Convert the image to grayscale
    gr = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Make a copy of the grayscale image
    bg = gr.copy()

    # Apply morphological transformations
    for i in range(5):
        kernel2 = cv2.getStructuringElement(cv2.MORPH_ELLIPSE,
                                            (2 * i + 1, 2 * i + 1))
        bg = cv2.morphologyEx(bg, cv2.MORPH_CLOSE, kernel2)
        bg = cv2.morphologyEx(bg, cv2.MORPH_OPEN, kernel2)

    # Subtract the grayscale image from its processed copy
    dif = cv2.subtract(bg, gr)

    # Apply thresholding
    bw = cv2.threshold(dif, 0, 255, cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)[1]
    dark = cv2.threshold(bg, 0, 255, cv2.THRESH_BINARY_INV | cv2.THRESH_OTSU)[1]

    # Extract pixels in the dark region
    darkpix = gr[np.where(dark > 0)]

    # Threshold the dark region to get the darker pixels inside it
    darkpix = cv2.threshold(darkpix, 0, 255, cv2.THRESH_BINARY | cv2.THRESH_OTSU)[1]

    # Paste the extracted darker pixels in the watermark region
    bw[np.where(dark > 0)] = darkpix.T

    cv2.imwrite('final.jpg', bw)


back_rm('watermark.jpg')
```

Here is the final result:
The processing time is very short using numpy

```
time python back_rm.py

real    0m0.391s
user    0m0.518s
sys     0m0.185s
```

* Problem

Using Ziegler - Nichols tuning rules, design a suitable PID controller for the following system:

$$G(s) = \frac{1}{s(s+1)(s+5)} \quad , \quad H(s) = 1$$

*** Sol.**

- $G(s)$ has a free integrator (type-1) $\Rightarrow$ use 2nd method:
- clch eq.: $1 + k\,G(s)\cdot H(s) = 1 + \frac{k}{s(s+1)(s+5)} = 0$

$$\therefore\ s(s^2+6s+5) + k = 0$$

$$\Rightarrow\ s^3 + 6s^2 + 5s + k = 0$$

- $a = \frac{6*5 - k}{6}$

s^3	1	5
s^2	6	k
s	a	0
s^0	k	

To get k_{cr}: $a = 0 \Rightarrow \boxed{k_{cr} = 30}$

To get ω_{cr}: $6s^2 + k_{cr} = 0 \Rightarrow -6\,\omega_{osc}^2 + 30 = 0$

$$\Rightarrow \omega_{cr} = \sqrt{5}\ \text{rls} \Rightarrow \boxed{P_{cr} = \frac{2\pi}{\sqrt{5}}}$$

From Formulas: $G_c(s) = 0.6\,k_{cr}\left(1 + \frac{1}{0.5\,P_{cr}\,s} + 0.125\,P_{cr}\,s\right)$

$$\Rightarrow \boxed{G_c(s) = 18\left(1 + \frac{0.711}{s} + 0.3512\,s\right)}\ \text{PID Design.}$$

answered Sep 24 '19 at 11:29

singrium
1,886 4 16 36