ENPM673 MIDTERM

Sahruday Patti

March 2022

1. **Question 1 (20 marks): Assuming the image below represents an x-ray of old coins. Read this image seen below using OpenCV taking into consideration that this is a binary image. 1. Write a program that will separate each coin from each other, so that the output image has the coins completely separated.**
   2. Write a program that will automatically count how many coins are there in the image (from step 1) . Note: You are allowed to use any built-in function in OpenCV. However, you are required to explain each step you took in your solution (Use pipeline or a block diagram).

   The Following Pipeline was followed to separate the circles in the image:

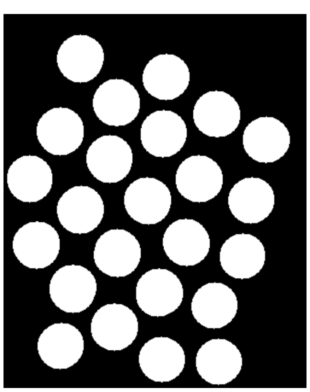
   a) Used a Ring Kernel to preserve the shape of circles while performing morphological operations. Made use of **cv2.getStructuringElement** to generate the Kernel Desired.

   b) In order to detect defects in the shape of an object in the image, first we need to remove the outer borders to get a complete object and then expand the object of interest. This is achieved by Erosion followed by Dilation.

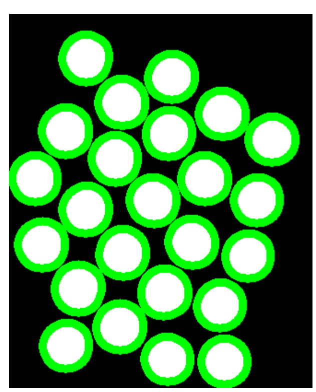
   c) Used the Canny Edge Detection to detect the edges of the image.

   d) used **cv2.findContours** to find the coordinates of the boundaries of the circle. Used **cv2.RETR-EXTERNAL** to define the parent Hierarchy, so only outer boundaries are drawn. Used **cv2.CHAIN-APPROX-NONE** so that all the points on the boundary are stored. Each boundary is considered a Contour. So, the length of the contours returned are the number of circles in the image.Later Used **cv2.drawContours** to draw the contours on the image.

   Number of Contours found = 24.



(a) Image after performing morphological operations



(b) Image with Contours Drawn

2. **Question 2: (30 marks) Given two images, imageA and imageB in this link shown below, find the matching points between these two images and stitch them together by finding the homography between them. Write the pipeline that you used to solve this problem and attach the code solution.**

   The Following Pipeline was followed to stitch the image together:

   a) To Find the Interest points and to create a descriptors for those Key points **Sift.detectAndCompute** function is used.

   b) **BFMatcher.knnMatch()** was used to get k best matches between the images.

   c) In order to find better matches, the matches of two images are compared with the distance measure and the descriptors which are under the threshold distance were used for further processing and are considered good matches.

   d) If more than 10 good matches are found, the pixel coordinates of these descriptors are taken and homography matrix has been computed between them.

   e) Then the corners of the image B in the image A has been computed using the H matrix and converted them to pixel coordinates from the homogeneous coordinates. Then the image is warped wrt the coordinates.

   f) A blank image of dimensions of B image has been created and image A has been copied into it. Then a fillpoly function is used to see and fill the corners of the B Image in the Blank image. Now, we would be left with the image that is part of A and not part of B. Now the warped image is added to the difference image(blank image initially) to obtain the stitched Image.

   g) Final Stitched image is blended by using a median filter and a smooth image is formed.

3. **Question 3 (30) Calibrate the camera (Find the intrinsic matrix K).**

   What is the minimum number matching points to solve this mathematically?

   The minimum number of matching points in world and image coordinates required to solve is 6. As the Number of Unknowns is 12 (11 defined upto scale in intrinsic). 3 Unknowns in Rotation, 3 Unknowns in Translation. These form the Extrinsic Matrix. The remaining 5 Unknowns are in the Intrinsic Parameters. The Skew, offset in x and y, the aspect ratio of the pixels. So, we have 11 unknowns so we need atleast 6 point correpondences that is 12 equations to solve this.

   What is the pipeline or the block diagram that needs to be done in order to calibrate this camera given the image above.

   a) Find the area of interest by thresholding the image in turn removing the background.
   b) Use an edge detection algorithm which has sub pixel accuracy like the Canny Edge Detector.
   c) We need points in the world coordinates with a corresponding match in the image coordinates. One of the ways to do that is by using hough transforms. Hough Transforms gives lines and intersection of these lines form corners. These points can be used as Detectors.
   Another way is to use Robert kernel, this gives us bunch points around each corner and the centroid of those can be used as a corner.
   d) Then solve for the Projection matrix using Homography we would get a 3 x 4 matrix which is a combined extrinsic and intrinsic parameters. Use any of the matrix decomposition to split it into Extrinsic and intrinsic.
   e) In order to correct for the distortion, loop through all the pixels of the undistorted image, and for each pixel, calculate the distorted location of that pixel. Can implement Bilinear Interpolation for better pixel intensity representation.

First write down the mathematical formation for your answer including steps that needs to be done to find the intrinsic matrix K

After finding the point correspondences from Two Different Planes, The Solution is given by AX = 0 where,

$$A = \begin{bmatrix} x1 & y1 & z1 & 1 & 0 & 0 & 0 & 0 & -x1*xp1 & -y1*xp1 \\ -z1*xp1 & -xp1 \\ 0 & 0 & 0 & 0 & x1 & y1 & z1 & -1 & -x1*yp1 & -y1*yp1 \\ -z1*yp1 & -yp1 \end{bmatrix} \tag{1}$$

Note: The bottom two elements are part of the same row. The problem with scaling hasn't allowed me to correct it.

x1,y1,z1 are world coordinates and xp1,yp1 are corresponding image coordinates.

$$X = \begin{bmatrix} H_{11} \\ H_{12} \\ H_{13} \\ H_{14} \\ H_{21} \\ H_{22} \\ H_{23} \\ H_{24} \\ H_{31} \\ H_{32} \\ H_{33} \\ H_{34} \end{bmatrix} \tag{2}$$

The Homogeneous linear equation system given by the expression Ax = 0, we have 12 unknowns in x coefficients. In order to find the Non trivial solution, which is approximate we constrain the vector x magnitude to $\|x\| = 1$. This becomes a quadratic optimization problem.But for a simple quadratic problem in matrices, we can solve for Ax= 0 subject to constraint $\|x\| = 1$ by Singular Value Decomposition of matrix A.

This is a constrained optimization problem $min_{\|x\|=1}\|Ax\|$. This solution is approximate because the minimum-norm solution for homogeneous systems is not always unique.

So, we are minimizing the $E = \|Ax\|$ subject to the constraint $\|x\| = 1$. Here E is the error measure, which is the Total least square i.e we are minimizing the squared error of shortest distance between the point and the vector x.
so to find the solution we should minimize $\frac{\|Ax\|}{\|x\|}$.

SVD factors the matrix into a two orthonormal matrices U and V and a diagonal matrix Σ such that $A = U\Sigma V^T$.

Now, lets say the solution to the above optimization problem is $\hat{x}$.

$$\hat{x} = argmin_x \|U\Sigma V^T x\|^2 \tag{3}$$

Constrain is carried throughout the development of the solution,so not mentioning it throughout.
For an orthonormal vector set, the magnitude of vectors is 1 and are orthogonal to each other. So the

orthonormal basis set just rotates the data but doesn't have any effect on the magnitude. we have constrained the above optimization problem with a magnitude i.e $\|x\|^2 = 1$. Therefore, we can write the above equation as,

$$\hat{x} = argmin_x \|\Sigma V^T x\|^2 \tag{4}$$

Now, say $y = V^T x$, therefore the equation becomes,

$$\hat{y} = argmin_y \|\Sigma y\|^2 \tag{5}$$

Now, if we have sorted diagonal entries in descending order, minimum deviation from the points in image plane would give us the best-fit solution. The above equation is minimum when all the other diagonal entries are zero and we multiply with the least singular value. i.e $y = [0, 0, ..., 1]^T$

we know x = Vy (since for an orthogonal matrix, its transpose is equal to inverse). Now if we multiply y with V we get the last column of V as our Solution.

Now, the obtained matrix is a 3x4 matrix. We know that the intrinsic matrix is a upper triangular matrix and Extrinsic matrix is a 3x3 matrix which is an orthogonal matrix.

The QR decomposition of a matrix is a decomposition of the matrix into an orthogonal matrix and a triangular matrix. A QR decomposition of a real square matrix A is a decomposition of A = QR. where Q is an orthogonal matrix and R is an upper triangular matrix. This Decomposition is called Gram-Schmidt Process.

An orthogonal matrix has row and column vectors of unit length. Consider a matrix A with n column vectors such that: A=[a1 a2.....an]. The Gram-Schmidt process is a way of finding an orthogonal projection V for each column vector of A and then subtracting its projections onto the previous projections. The resulting vector is then divided by the length of that vector to produce a unit vector. Thus forming an orthonormal matrix set.

$$v1 = a1; e1 = \frac{v1}{\|v1\|} \tag{6}$$

$$v2 = a2 - (a2.e1)e1 \tag{7}$$

$$\tag{8}$$

(The projection of a2 on a1 is subtracted to get rid of the common part of the two vectors)
When you project a vector onto a subspace, and subtract the result, you get a vector orthogonal to the subspace.
similarly,

$$vn = an - (an.e1)e1.....(an.e_{n-1})e_{n-1}, en = \frac{vn}{\|vn\|} \tag{9}$$

$$A = \begin{bmatrix} e1 & e2 & e3....en \end{bmatrix} \begin{bmatrix} a1.e1 & a2.e1... & an.e1 \\ 0 & a2.e2...... & an.e2 \\ . & . & . \\ . & . & . \\ 0 & 0 & an.en \end{bmatrix} \tag{10}$$

The unit vector matrix is our Extrinsic camera Matrix, the mapping between world coordinates and the camera coordinates. The Upper triangular matrix is our Intrinsic Matrix.

4. **Question 4: (20 marks) Implement K-means algorithm to separate this image below, based on color, into 4 classes. Note: You are NOT allowed to use any built-in function, implement your code from scratch. Explain each step you take.**

   The pipeline used to implement this algorithm on the given image is as follows:

   a)Select (K=4) random data points, in our case they are the pixel Intensity. These four initial points are the four centroids in initial iteration.

   b)Calculate the distance between each centroid and all the other image pixels and assign the pixels to the closest centroid based on the Euclidean distance.

   c)The New centroid is calculated, and it is compared with the old centroid. If the distance is less than 100 then the loop is broken. Else, the samples are again clustered based on new centroids and this process is iterated till the number of iterations.

   d) After the iterations, the clusters are returned. Using the centroids and the clusters, the segmented image is plotted.

   e) Predict function is where the loop is iterated till max˙iters. Get˙Cluster˙Labels Classifies samples as the index of their clusters. Create˙Cluster function Assign the samples to the closest centroids to create clusters. Get˙Centroids calculates the mean and creates the centroids.

   In this we are segregating into 4 clusters only. But we have 5 different colours. As the intensity of colors goes as VIBGYOR, the yellow and red are assigned to the orange intensities as seen in the final output image.