CMSC 733, Computer Processing of Pictorial Information
Homework 3: Road from Blobs to Faces to Seams!
Due on: 11:59:59PM on Wednesday, May 10 2017

Prof. Yiannis Aloimonos,
Nitin J. Sanket

April 28, 2017

In this homework, you will implement various small algorithms which are useful for a variety of tasks in computer vision.

1 Filtering without blurring edges! - 20Pts

While de-noising an image especially when we do gaussian blur we're not only removing the noise but are also blurring the edges which is a key information in the image. This is because gaussian function is an isotropic function, i.e., affects both the dimensions of the image to the same extent (circular contours). One can imagine that by distorting this circular gaussian we can tailor the filter to affect only a certain orientation. If you implemented the last part of HW0 where we used a 3D Gaussian, you would've observed that we implemented exactly this but in 3D. The way a gaussian is distorted is by changing the value of Σ in the equation. As we are working with grayscale image in this case, we have a 2D gaussian. Hence, the Σ will have 4 values. The principal diagonal elements are a function of maximum amplitude in x and y. The off-diagonal elements depict how inter-linked are the dimensions x and y (changes the eccentrcity of the ellipse; the circular gaussian we saw earlier is now an ellipse).

Conceptually, if you can do this dynamically on an image, you can tailor a kernel function (filter) to filter along the direction of the edge and not normal to it (this is the 'bleeding' of the edge). This would mean that we can denoise the image still retaining edge sharpness. A very simple method which works very well was formulated by Pietro Perona and Jitendra Malik in Ref. [1] and it was called Anisotropic Diffusion (AD).

To make your life simple (and I being an awesome TA), I have broken down the equations in the paper for you (feel free to read the paper if you are interested).

AD is an iterative algorithm. Let t be the current iteration number and I be the image under consideration. x, y represent the index of the current pixel.

$$I_{i,j}^{t+1} = I_{i,j}^{t} + \lambda \left[c_N \nabla_N I + c_S \nabla_S I + c_E \nabla_E I + c_W \nabla_W I \right]_{i,j}^{t}$$

Here λ is called the integration constant and affects what percentage of update you do between iterations. Keep this value between 0 and 0.25 for numerical stability.

The letters N, S, E and W signify North, South, East and West neighbours of the current pixel. Note that for this problem 4-connectivity is assumed, **but you need to implement 8-connectivity neighborhood**. You can assume that the pixels outside the image are zeros (zero-padding) or if you want to be fancy you can perform Mirror Padding (http://www.mathworks.com/help/images/ref/padarray.html).

The function ∇ here is the nearest neighbour difference given by:

$$\nabla_N I_{i,j} = I_{i-1,j} - I_{i,j}$$

$$\nabla_S I_{i,j} = I_{i+1,j} - I_{i,j}$$

$$\nabla_E I_{i,j} = I_{i,j+1} - I_{i,j}$$

$$\nabla_W I_{i,j} = I_{i,j-1} - I_{i,j}$$

As I mentioned earlier, we need to dynamically change the filter strength depending upon the presence/absence of an edge. This is done by changing the conduction co-efficient using the following equations:

$$c = \exp - \left[\frac{\|\nabla I\|}{\kappa} \right]^2$$

Use the above equation for variant 1 of the code. This function previlages high contrast edges over low contrast ones.

The second function to be used for variant 2 of the code is given below. This previlages wide regions over small ones.

$$c = 1 / \left(1 + \left[\frac{\|\nabla I\|}{\kappa} \right]^2 \right)$$

Note that κ here is the conduction coefficient. Use `Brain.jpg` image from `Images` folder to test your algorithm.

2 EigenFaces - Face Recognition using Eigen Vectors and Values! - 30Pts

This is one of the most trivial ways to do face recognition. Follow Ref. [2] and implement this method of Face Recognition. You are given images from a database called the

ORL dataset which is taken from http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html. The database is split into train and testing sets and can be found in the folder `Images/Dataset/Train` and `Images/Dataset/Test` re3spectively. In the training set the subjects are named from `s1` to `s40` (with 4 images per subject and 40 subjects). In the testing set the subjects are named from `s1` to `s40` (with 6 images per subject and same 40 subjects).

Your task is to code up for the whole pipeline (Training and Testing). Calculate Regognition Rate (RR) as the percentage of correctly classified images in the Train and Test Set. Report both Training Set and Testing set accuracies.

3 Telekinetic object re-targeting! - 20Pts

Seam Carving is an algorithm for content aware resizing or image re-targeting, i.e you don't always want to just shrink the image as it affects the whole image. Instead you remove some redundant details so that the image is resized and the key features stay the same. This is extremely useful when you want to retain the important information but you want to adjust an image to fit from a huge TV to a phone's screen.

This wikipedia link - https://en.wikipedia.org/wiki/Seam_carving explains the concept very well. However, the paper which explains this idea is given in Ref. [3].

The image `SeamCarving.jpg` in the `Images` folder is of size 968×1428. You need to remove 100 horizontal seams and 200 vertical seams (remove 1 horizontal seam and 2 vertical seams per iteration). Include a figure comparing the original image with the seam carved image in the pdf. The output image should be a color image.

4 Test your understanding! - 30Pts

Answer the following questions regarding the three algorithms. This section will not be evaluated if you have not implemented the three algorithms.

1. In Anisotropic Diffusion, What is the effect of changing the value of κ (both increasing κ and decreasing κ, present these outputs in your writeup)?

2. In Anisotropic Diffusion, What is the effect of changing the value of λ (both increasing λ and decreasing λ, present these outputs in your writeup)?

3. In EigenFaces, Is it obvious that as M' is increased the Recognition Rate (RR) (percentage of correct classifications) increases and then decreases? What is the value of M' for which RR is the highest?

4. What does this trend signify?

5. What are the drawbacks of using Eigen-Faces for Face Recognition?

6. Is the Eigen-Faces method invariant to the following changes (individually) and explain why?
(a) Illumination (Dark and bright version of the same face give similar results in terms of RR).
(b) Rotation (2D rotation of image gives same results in terms of RR).
(c) Scale (The same face can be bigger or smaller in the image and the RR will be the same.
(d) Pose (Different face poses give the same result).

7. Look at 2 particular images (present these images in your writeup) which were misclassified (failed cases) and describe why you think they failed. What assumptions in the paper or the concept did they violate?

8. The energy function used in Ref. [3] is the sum of absolute values of gradients along x and y. This is a very simple function but may not be the best. Formulate a different energy function for seam carving.

5 Submission Guidelines

Answer the questions in a pdf file with the naming convention `Answers.pdf`.

Submit your codes (`.m` files) and answer to questions (`Answers.pdf`) in a zip file with the naming convention
`YourDirectoryID_hw1.zip` onto ELMS/Canvas. DO NOT INCLUDE any sub-directories or images in the zip folder. If your code does not comply with the above guidelines, you'll be given **ZERO** credit.

6 Starter Code

There is no starter code given for this project.

7 Submission Guidelines

Typeset all the answers in LaTeXusing the IEEETran format given to you in `Draft` folder. The output file should be (**pdf and pdf ONLY**). Submit your codes (whatever you write, `.m` files) with the naming convention `YourDirectoryID_hw3.zip` onto ELMS/Canvas (**Please compress it to .zip and no other format**). Your `DirectoryID` is the username to your UMD e-mail ID. If your email ID is `ABCD@terpmail.umd.edu` or `ABCD@umd.edu`, your `DirectoryID` is `ABCD`. Your zip file should have the following things:

- Outputs of the codes as requested in the document you typeset. For this homework, you don't have to explain the algorithms. The following outputs are requested:

  - Output of Anisotropic Diffusion on the `Brain.jpg` image from `Images` folder. Present outputs for different parameter values and talk about them.
  - Recognition Rates for both Training and Testing sets.
  - Color Output image in Seam carving after removing 100 horizontal seams and 200 vertical seams.

- Answers to the questions asked with respective images.

If your code does not comply with the above guidelines, you'll be given **ZERO** credit.

8 Allowed Matlab functions

All general MATLAB functions expect which complete a majority of the homework.

9 Collaboration Policy

You are restricted to discuss the ideas with at most two other people. But the code you turn-in should be your own and if you **DO USE** (try not it and it is not permitted) other external codes/codes from other students - do cite them. For other honor code refer to the CMSC733 Spring 2017 website here https://www.cs.umd.edu/class/spring2017/cmsc733/.

DON'T FORGET TO HAVE FUN AND PLAY AROUND WITH IMAGES!.

References

[1] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7):629–639, 1990.

[2] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

[3] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. In *ACM Transactions on graphics (TOG)*, volume 26, page 10. ACM, 2007.