Unsupervised Texture Segmentation Using Gabor Filter

Aman Purwar Arkajyoti Pal Subham Rajgaria

1 Outline

We are given a mosaic of 44 mosaic consisting of 16 textures collected from the texture database of the link :Texture Database .Our job is Texture Segmentation. Texture Segmentation is a problem in Image Processing where a certain region having similar statistic is clustered. So basically Our Problem reduces to cluster properly the 16 textures in the mosaic.

2 Libraries and Frameworks

1. Python
2. opencv
3. argparse
4. glob
5. scipy
6. sklearn
7. numpy

3 Algorithm

3.1 Unsupervised Texture Segmentation using Gabor Filters

The Steps are as followed :

1. Subjecting an input image to a series of Gabor filters to produce a set of filtered images.

2. Applying a nonlinear transducer to the set of filtered images to get a series of response images, whereby the nonlinear transducer behaves as a blob detector. Note that a blob is just some region where some property is constant, or in common with nearby pixels . The size of the window used to apply the transduction technique is determined with respect to the radial frequency that the Gabor filter in the previous step is tuned to.

3. Applying a local energy computation to the smooth the response of the activation function.

4. Combining feature images and applying a clustering scheme on the feature images to present a segmented image.

3.2 Getting Gabor Filters and Producing Filtered Images

1. In the realms of image processing and computer vision, Gabor filters are generally used in texture analysis, edge detection, feature extraction, disparity estimation (in stereo vision), etc. Gabor filters are special classes of bandpass filters, i.e., they allow a certain band of frequencies and reject the others.

2. When a Gabor filter is applied to an image, it gives the highest response at edges and at points where texture changes.

3. According to paper, we have taken the values of Φ to be 0, 45 , 90 and 135 and varied the lambda to $4\sqrt{2}$ to width $4\sqrt{2}$ in the multiples of 2 and generated the the filters to generate the filtered Images.

3.3 Applying Nonlinear transduction and Gaussian Smoothing

1. We get a bunch of Filtered Images after Applying Gabor filter.Now our job is to throw a certain number of filtered images to reduce the error in reconstruction in the image from those Filtered Images.

2. $S(x, y)$ is said to be the image obtained by adding all of the filtered images together into a single image. $\hat{S}(x, y)$ is then the array obtained by using a single filter on the input image. The error of using $\hat{S}$ instead of just using S is given as: $SSE = \sum_{x,y} \left\| S(x, y) - \hat{S}(x, y) \right\|$, $R^2 = 1 - \frac{SSE}{SSTOT}$, $SSTOT = \sum_{x,y} \|S(x, y)\|^2$

3. The Algorithm is as follows :

   (a) Find the filtered image resulting in the smallest SSE value.

   (b) Try to add another image with the smallest one found, if it approximates $S(x, y)$ the best, then keep it.

   (c) Try picking up another filtered image by repeating the second step. We can continue this until $R^2 \geq 0.95$.

4. Now the Activation Function is used and the activation function used is tanh with an activation window.

5. For the filtered input images, features are exaggerated once this transducer is applied, since it is nonlinear. Due to the threshold-like nature of the sigmoidal function, more prominent qualities are exaggerated, and lesser prominent ones become even less important. For this reason, it is known as an activation function. This assists in the clustering of features because 3 more weight will be associated with prominent features, and a smaller weight will be associated with less prominent features.

6. Now a Gaussian Smoothing filter is used to each filtered Image.

3.4 Clustering

1. At this point, all of the feature images necessary for texture segmentation have been acquired. Features are arranged such that the value of each pixel in a single feature image counts as a feature for that pixel coordinate. So, for some pixel at location (i,j), with N different

feature images we can construct a feature vector, as introduced by, and given by: $T_{i,j} = \langle F_1(i,j), F_2(i,j), ..., F_n(i,j) \rangle$

2. Now we apply K-means Clustering algorithm to cluster the pixels with almost similar features.

References

[1] A. Jain and F. Farrokhnia, ”Unsupervised texture segmentation using Gabor filters”, in *IEEE International Conference on Systems, Man, and Cybernetics Conference Proceedings*, 1990.