Contents

- Overview
- Filters for texture description
- Clustering using K-means
- Texton histograms
- Orientation energy
- Cue combination

1 Overview

In this lecture, we discuss the use of vectors of filter responses for texture analysis. We provide on operational definition of textons. Lastly, we present an algorithm for performing image segmentation using both texture and contour cues. Please refer to the paper *Textons, Contours and Regions: Cue Integration in Image Segmentation* by Malik, Belongie, Shi, and Leung for the figures used in this lecture. Throughout these lecture notes, the above paper will be referred to as "the paper."

2 Filters for texture description

2.1 Gaussian derivative filter set

We will use vectors of filter responses to represent texture information. Consider a gaussian derivative filter set consisting of 6 orientations, 3 scales, and 2 phases (odd and even) for a total of 36 filters. Figure 1 in the paper shows such a filter set.

Let the matrix I denote the image with element (x, y) denoting an individual pixel in the image. Let the $f_i, i = 1 \ldots 36$ represent the gaussian filters. Now define a mapping from pixels in the image to points in $\mathbf{R^{36}}$ as follows.

$$\begin{aligned} r_1(x,y) &= I * f_1 \\ r_2(x,y) &= I * f_2 \\ &\vdots \\ r_{36}(x,y) &= I * f_{36} \end{aligned}$$

We have that each pixel (x, y) gets mapped to the vector $\vec{r}(x,y) = (r_1(x,y), r_2(x,y), \ldots, r_{36}(x,y))^T$ in $\mathbf{R^{36}}$.

2.2 Problems with linear filters

There is one serious problem with linear filters: they generally respond everywhere. A naive solution to this problem is to simply threshold the the filter outputs. However, this is an unsatisfactory solution since for low threshold, we have too much firing, and for high threshold, important image features are lost.

3 Clustering using K-means

As an improvement over simply thresholding the filter outputs, we perform vector quantization (clustering) in the vector space of filter responses to find prototypes. We use an operational definition a textons where the textons are defined to consist of the prototypes found by performing clustering.

We use the K-means algorithm to find the cluster centers. K-means clustering is performed as follows:

Choose the value of K to use. Initialize the cluster centers by choosing K filter response vectors at random. Then perform the two-step iterative procedure: (1) Assign each response vector to the closest cluster center using the Euclidean distance measure. (2) Re-compute the cluster means using

$$\mu_j = \frac{1}{N_j} \sum_{n \in S_j} x_n$$

4 Texton histograms

After clustering is performed, we have a mapping from each pixel to a texton. Supposing that $K = 25$ textons are used, then each pixel in the image will get mapped to exactly one texton. The set of all pixels that get mapped to the i^{th} texton is called the i^{th} texton channel. Figure 2a in the paper shows a polka-dot image. Figure 2b the textons found using K-means with $K = 25$. Figure 2c shows the texton channels for the image in 2a.

4.1 Measuring texture similarity

The texture similarity for two pixels can be computed by comparing the texton histograms of the two pixels the chi-square test. Each histogram has K bins. The k^{th} histogram bin for pixel i is given as

$$h_i(k) = \sum_{j \in N(i)} I[T(j) = k]$$

where $N(i)$ is the set of pixels in the box centered at i. $I[\cdot]$ is the indicator function. $T(j)$ is the texton assigned to pixel j.

The chi-square test can then be used to compare the histograms. It is given by

$$\chi^2(h_i, h_j) = \frac{1}{2} \sum_{k=1} \frac{(h_i(k) - h_j(k))^2}{h_i(k) + h_j(k)}$$

5 Orientation energy

Orientation energy (OE) can be thought of as a generalization of the term "edge detection." Orientation energy can be used to compute contour information in an image. Let f_o be an odd-symmetric gaussian filter (edge- detector) and let f_e be an even-symmetric gaussian filter (bar-detector). Then the orientation energy , $R(x,y)$ is defined to be

$$R(x,y) \stackrel{\text{def}}{=} (I * f_o)^2 + (I * f_e)^2$$

The orientation energy can then be computed for several different orientations. The resulting orientation energies can then be added or the maximum can be chosen. Now define OE_{ij} as the maximum orientation energy along the line segment from pixel i to pixel j.

6 Cue combination

6.1 Naive cue combination algorithm

We would like to be able to integrate information from both contours and textures when assigning a weight (similarity measure) between two pixels. A naive way of doing this (which doesn't work) is to simply multiply the contribution from contour by the contribution from texture.

$$W_{ij} = W_{ij}^{IC} W_{ij}^{TX}$$

$W_{ij}^{IC} = exp(-max_{x \in l_{ij}} \frac{OE(x)}{\sigma_{con}})$ is the contribution from the contour and l_{ij} is the line connection pixels i and j.

$W_{ij}^{TX} = exp(-\chi^2(\frac{h_i, h_j}{\sigma_{tex}}))$ is the contribution from the texture using the chi-square distance between texton histograms.

6.2 Texture as a problem for contour

The texture weighting function, W_{ij}^{TX}, can cause spurious edges to be found in the texture region. This is undesirable.

6.3 Contour as a problem for texture

The contour weighting function, W_{ij}^{IC}, can cause contours to be segmented into separate regions. See Figures 5 and 6 for an example of this.

6.4 A simple solution

A solution to the above problems has been proposed by Malik, Belongie, Shi, and Leung in the paper *Textons, Contours and Regions: Cue Integration in Image Segmentation* Their strategy is to label every pixel as texture or non-texture using local isotropy estimates. The W_{ij}^{TX} and W_{ij}^{IC} are weighted by a discrete version of a signmoidal function of the local isotropy. Normalized cuts is then performed to segment the image. See Figures 9 and 10 for segmentation results obtained by using this method.