

Introduction to Computer Vision

Alvin Lin

August 2018 - December 2018

Texture

Texture is defined by patterns of structure from changes in surface albedo, surface shape, or many small surface patches. It is hard to concretely define, but texture tells us what a surface is, and may sometimes tell us what the object is or what shape the surface is. Often they are very complex to recognize. Synthesizing textures from examples is also a complex problem in computer graphics.

The core idea is that textures consist of a set of elements repeated in some way. Ideally, we should be able to identify the elements and summarize the repetition.

Filter Based Texture Representations

1. Choose a set of filters, each representing a pattern element, such as spots and bars.
2. Apply all the filters to an image at a variety of scales.
3. Rectify the filtered images. We should avoid averaging contrast reversals. We do not want to average dark spots on light backgrounds and vice versa.
4. Compute summaries of rectified filtered images.
5. Describe each pixel by a vector of summaries.

Alternatively, we can compute the responses of blobs and edges at various orientations and scales. Since textures are made of repeated local patterns, we need to find the patterns and describe their statistics within each local window. Generally this involves using filters that look like the patterns as described in the steps above and plotting a histogram of “prototypical” feature occurrences.

Choosing the Filters

When choosing the filters, we can measure responses to over-complete filter banks and describe textures using clusters of responses. Alternatively, we can build a vocabulary of pattern elements from pictures and describe textures using this vocabulary. There are two steps to building a pooled texture representation for texture in an image domain.

1. Build a dictionary representing the range of possible pattern elements, using a large number of texture patches.
2. Vectorize the patches in images using the clusters learned.

To build the dictionary:

1. Collect many training example textures.
2. Construct the vectors \vec{x} for the relevant pixels. Either reshape a patch around a pixel or compute a vector of filter outputs at the pixel.
3. Obtain k cluster centers c for these examples.
4. For each relevant pixel i in the image, compute the vector representation \vec{x}_i for that pixel. Obtain j , the index of the cluster center c_j closest to that pixel and insert j into a histogram for that domain.

Clustering can be performed with k-means, representing each patch with either an intensity vector or a vector of filter responses over the pixel/patch. To summarize the high dimensional clusters, we can find the nearest cluster center in the dictionary and use that to represent the entire cluster.

You can find all my notes at <http://omgimanerd.tech/notes>. If you have any questions, comments, or concerns, please contact me at alvin@omgimanerd.tech