Assignment A3: Frequency Domain Filtering

CS 6640
Fall 2018

Assigned: 24 September 2018

Due: 18 October 2018

For this problem, handin a report (PDF file) as required below as well as the Matlab .m files for the functions described by the headers below, and any help functions you write.

None of the functions should write to the interpreter, draw, etc. unless explicitly required by the header.

1. Develop a texture feature analysis tool based on the 2D FFT power spectrum. For every 5x5 region in the image, compute the 2D FFT, compute the power spectrum, and use that as a 25-element feature vector. Produce a texture feature array, M*N by 25, and then use kmeans as in A2 to explore the usefulness of this for semantic region analysis (based on texture) in our videos.

```
function T = CS6640_FFT_texture(im)
% CS6640_FFT_texture - compute FFT texture parameters
% On input:
%     im (MxN array): input image
% On output:
%     T (M*Nx25 array): texture parameters
%        each texture parameter is a column vector in T
% Call:
%     T = CS6640_FFT_texture(im);
% Author:
%     <Your name>
%     UU
```

```
%     Fall 2018
%
```

2. Develop another FFT texture analysis approach based on radial segments of the power spectrum. A radial feature is defined by a radius pair:

$$f_{r_1,r_2} = \sum\sum \mid F(u,v) \mid^2$$

where $r_1^2 \leq u^2 + v^2 < r_2^2$. Suppose there are 10 such pairs, specified by 10 radii; i.e.: $R = \{0, r_1, r_2, \ldots, r_{10}\}$ Produce a texture feature array, M*N by 10, and then use kmeans as in A2 to explore the usefulness of this for semantic region analysis (based on texture) in our videos.

```
function T = CS6640_FFT_radial(im)
% CS6640_FFT_radial - compute FFT radial texture parameters
% On input:
%     im (MxN array): input image
% On output:
%     T (M*Nx10 array): texture parameters
%        each texture parameter is a column vector in T
% Call:
%     T = CS6640_FFT_radial(im);
% Author:
%     <Your name>
%     UU
%     Fall 2018
%
```

3. Develop another FFT texture analysis approach based on angular segments of the power spectrum. An angular feature is defined by a angle pair:

$$f_{\theta_1,\theta_2} = \sum\sum \mid F(u,v) \mid^2$$

where $\theta_1 \leq tan^{-1}(v,u) < \theta_2$, and $u^2 + v^2 < L^2$ for some appropriate L. Suppose there are 8 such segments, specified by 9 angles; i.e.: $A = \{0, \Delta, 2\Delta, \ldots, 8\Delta\}$, where $\Delta = 45$ degrees. Produce a texture feature array, M*N by 8 (where these cover $\Delta\theta$ between the following angles: $\{0, 45, 90, 135, 180, 225, 270, 315, 360\}$, and then use kmeans as in A2 to explore the usefulness of this for semantic region analysis (based on texture) in our videos.

```
function T = CS6640_FFT_angular(im)
% CS6640_FFT_angular - compute FFT angular texture parameters
% On input:
%     im (MxN array): input image
% On output:
%     T (M*Nx8 array): texture parameters
%        each texture parameter is a column vector in T
% Call:
%     T = CS6640_FFT_angular(im);
% Author:
%     <Your name>
%     UU
%     Fall 2018
%
```

4. Develop a Fourier shape descriptor function as described in the Zahn and Roskies paper (in 'lectures' link from class web page, weeks5-6). I.e., given a shape as a set of connected boundary points:

$$P = \{(x_i, y_i)\}$$

where $i = 1 : N$, of a shape (going clockwise), then the angle at each point is computed as:

$$\theta(t) = atan2(y(t+w) - y(t), x(t+w) - x(t))$$

where w is a distance along the curve. θ is better produced by taking the previous direction, and adding to it the amount of change in angle to get the current angle. Then produce the function:

$$\phi(t) = (\theta(t) - \theta(0)) mod 2\pi$$

and

$$\psi(a) = (\phi(\frac{La}{2\pi}) + a) mod \pi$$

where ψ is computed over the interval $[0, 2\pi]$. Take the FFT of ψ, and if there are 2p Fourier coefficients, use the power of coefficients 2 to p as the shape descriptor (a vector).

```
function X = CS6640_FFT_shape(Z,w)
% CS6640_FFT_shape - compute Fourier shape descriptors for a curve
% On input:
%     Z (Nx2 array): input curve (should be closed)
%     w (int): distance along curve to determine angles
```

```
% On output:
%     X ((N/2-1)x1 vector): the Fourier coefficients for the curve
% Call:
%     X = CS6640_FFT_shape(curve,2);
% Author:
%     <Your Name>
%     UU
%     Fall 2018
%
```