Chapter 6: Texture

A homogeneous visual motif: (see A7 script)

- gray level
- color
- shape

Early proposals: captured by 1st + 2nd order statistics of texture features

Texture representations

* Local texture: how region looks near pixel
* Projected texture: description of image domain
* Data-driven texture: synthesize texture

Can do shape from texture

Local texture using filters (Texture parameters)

Texton (texel): texture element

E.g., woven fabric, pebbles, grass

Describe:
1. Texton
2. Pattern
use basic elements: spots + bars

use filters to find sub-elements

spots: combinations of symmetric Gaussian filters

- **Spot 1**: 3 filters:
  - $G_1$: $\sigma_1^2 = 0.62$
  - $G_2$: $\sigma_2^2 = 1$
  - $G_3$: $\sigma_3^2 = 1.6$

  $S_1 = G_1 - 2G_2 + G_3$ (i.e., weights 1, -2, 1)

- **Spot 2**: 2 filters
  - $G_1$: $\sigma_1^2 = 0.71$
  - $G_2$: $\sigma_2^2 = 1.44$

  $S_2 = G_1 - G_2$ (i.e., weights 1, -1)
2D Gaussian filter can have different variances in $x \neq y$.

\[ G_{\sigma_1, \sigma_2}(x, y) = e^{-\frac{(ax+by)^2}{2\sigma_1^2}} - \frac{(cx+dy)^2}{2\sigma_2^2} \]

How to compute this:

Input: $x_0, y_0$, center of filter; usually 0, 0

$\min_x, \max_x$, range of $x$ values; usually $-c_x, c_x$

$\min_y, \max_y$, range of $y$ values; usually $-c_y, c_y$

$dx$, step in $x + y$

$\alpha, \beta, \gamma, \delta$, orientation of bar

$\sigma_x$, variance in $x$

$\sigma_y$, variance in $y$

Show examples of CS5320 oriented Gaussian
bar filters

basic bar:

\[ G_1 \equiv \sigma_x = 2 \quad \sigma_y = 1 \quad x_0 = 0 \quad y_0 = 1 \quad x_{\text{max}} = 5 \quad y_{\text{max}} = 5 \]

\[ G_2 \equiv \sigma_x = 2 \quad \sigma_y = 1 \quad x_0 = 0 \quad y_0 = 0 \]

\[ G_3 \equiv \sigma_x = 2 \quad \sigma_y = 1 \quad x_0 = 0 \quad y_0 = -1 \]

\[ B = -G_1 + 2G_2 - G_3 \]

then use 6 rotated versions of this:

\[ B_{45} = \text{imrotate} \left( B, 45 \right) \]

other possible features: edge info: max gradient dir

in some window around pixel variance, mean, etc. of windows over filter outputs

\Rightarrow texture vector

\[ R_1 = B \times I \quad (\text{e.g., } R_1 = \text{filter2}(B, \text{im-tex}) \]

\[ B = \text{imresize}(B, [11, 11]) \]

Now, rectify: produce 2 maps

\[ \max(0, R_1) \]

\[ \max(0, -R_1) \]

Finally, compute Gaussian at twice scale of 2 maps
K-means

Show k-means in MATLAB.

Use Alg. 6.3 to find 20 clusters from im.txt and see how well it performs.