Chapter 6 Texture

a homogeneous visual motif: (see A7 script)

gray level
color
shape

early proposals: captured by 1^{st} + 2^{nd} order statistics of texture features

texture representations

* local texture: how region looks near pixel
* pooled texture: description of image domain
* data-driven texture: synthesize texture

Can do shape from texture

Local texture using filters (texture parameters)

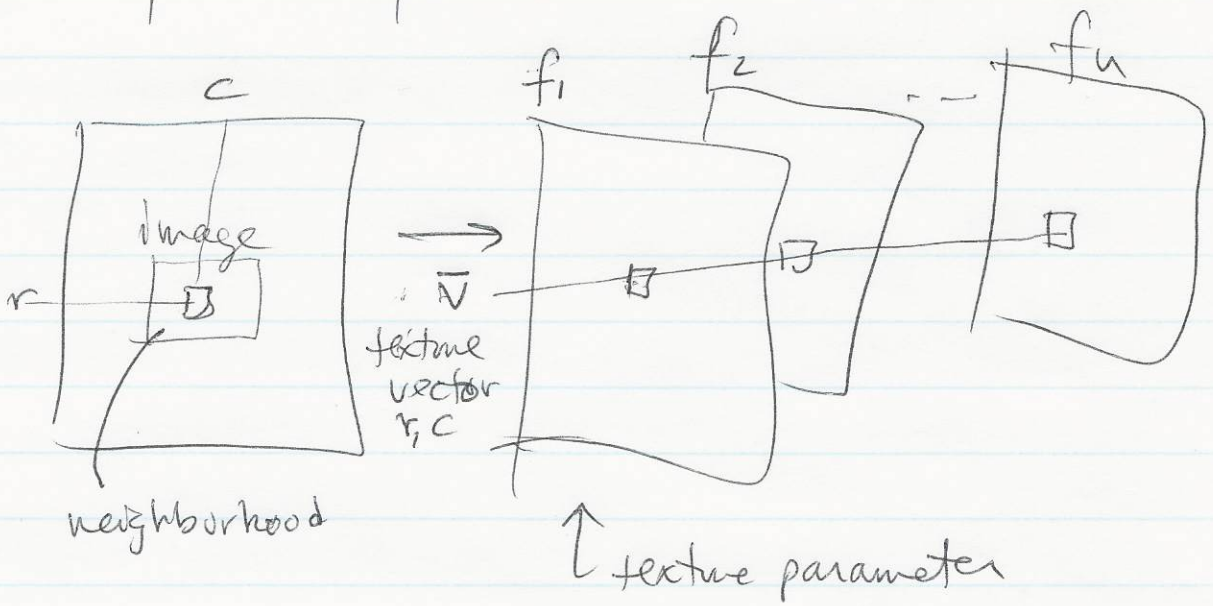
texton (texel): textural element

e.g., woven fabric, pebbles, grass

describe: (1) texton
(2) pattern

use basic sub-elements: spots + bars

use filters to find sub-elements



spots: combinations of symmetric Gaussian filters

e.g.,

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 : $\sigma_2^2 = 1.14$

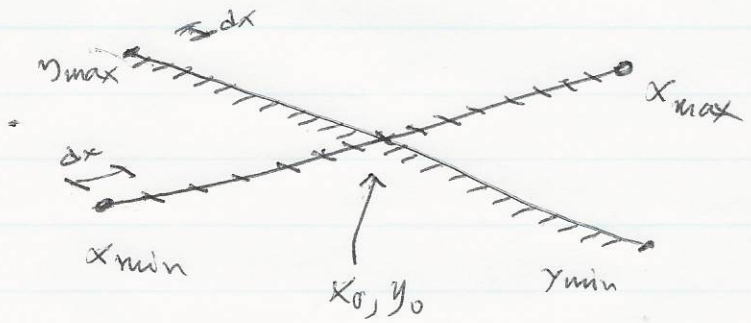
$S_2 = G_1 - G_2$ (i.e., weights 1, -1)

bar filters

2D Gaussian filter can have different variances in x & 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

x_{min}, x_{max} range of x vals; usually $-c_x, c_x$

y_{min}, y_{max} " " y " ; " $-c_y, c_y$

dx step in x & y

a, b, c, d orientation of bar [a b] [c d]

sigma1 variance in x

sigma2 variance in y

show examples of CS5320_oriented-Gaussian

bar filters

basic bar;

$G_1 \equiv \sigma_x = 2 \quad \sigma_y = 1 \quad x_0 = 0 \quad y_0 = 1 \quad x_{min} = -5 \quad x_{max} = 5$
y_{min} " y_{max} "

$G_2 \equiv \sigma_x = 2 \quad \sigma_y = 1 \quad x_0 = 0 \quad y_0 = 0$ "

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

$$B = -G_1 + 2G_2 - G_3$$

then use 6 rotated versions of this:

B45 = imrotate(B, 45)
⋮

other possible features: edge info: max gradient dir in some window around pixel
variance, mean, etc. of windows over filter outputs
⇒ texture vector

$R_1 = B ** I$ (e.g., R1 = filter2(B, im-tex)

B = imresize(B, [11, 11]);

Now, rectify: produce 2 maps max(0, R1)
max(0, -R1)

Finally, compute Gaussian at twice scale of 2 maps [Alg. 6.1*]

K-means

show kmeans in Matlab

Use Alg. 6.3 to find 20 clusters from im_tex & see how well it performs.