CS6640 24 Sept '20

P1 Grades

P2: Must have quantitative ground truth + performance results

---

Texture: see Texture-Shapiro-Stockman.pdf

Laws Energy Measures

* uses local masks (filters, kernels)
* 2D kernels generated from 1D

1D Kernels: 4 of them

L5	[1 4 6 4 1]	Level
E5	[-1 -2 0 2 1]	Edge
S5	[-1 0 2 0 -1]	Spot
R5	[1 -4 6 -4 1]	Ridge

2D Kernels: 16 of them

L5L5	E5L5	S5L5	R5L5
L5E5	E5E5	S5E5	R5E5
L5S5	E5S5	S5S5	R5S5
L5R5	E5R5	S5R5	R5R5

E.g.) L5L5 = L5' L5;

Laws Process

step 0 Compute 15x15 average and subtract from image

step 1 Apply 16 masks to get F_k, $k = 1, 16$

step 3 Get texture energy maps E_k $k = 1, 16$

$$E_k(r,c) = \sum_{j=-7}^{7} \sum_{i=-7}^{7} |F_k(i,j)|$$

step 4 Get 9 energy maps

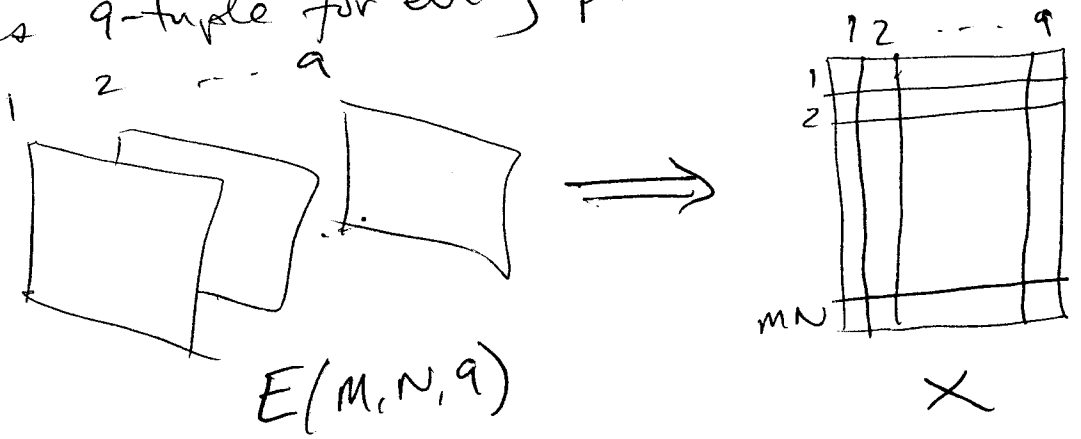
$\frac{L5E5 + E5L5}{2}$, $\frac{L5R5 + R5L5}{2}$, $\frac{E5S5 + S5E5}{2}$,

S5S5, R5R5, $\frac{L5S5 + S5L5}{2}$, E5E5,

$\frac{E5R5 + R5E5}{2}$, $\frac{S5R5 + R5S5}{2}$

2,5 4,13 7,10, 11, 16, 3,9 .. 6 8,14 12,15

produces 9-tuple for every pixel



→ convert to vectors

classify: [cidx, ctrs] = kmeans(X, 6)
↑ # classes

✗ see map!

A3 Issues

Q1 Prewitt

* definition of 1D vectors for k

$$\begin{bmatrix} ? \\ ? \end{bmatrix} \begin{bmatrix} ? & ? \end{bmatrix}$$

$f_{1,k}$ $f_{2,k}$

State $f_{1,k}$ = give precise description

$f_{2,k}$ = " " "

* Give formulas for # of *'s + +'s for both 1D + 2D filters

e.g.) consider finding max value in a vector with n elements where operation is comparison (i.e., $\leq$)

* a formula:

$$f_{max}(n) = n - 1$$

expressed as quadratic:

$$f_{max}(n) = 0n^2 + 1n + (-1)$$

$$\downarrow \quad \downarrow \quad \downarrow$$

$$an^2 + bn + c$$

for our case, MN is a constant and k the variable

* Implement (e.g. 3):

CS6640_Prewitt(k)
returns f_1, f_2, F

write yourself to ensure details of * + †

† CS6640_Prewitt1D(im, k)
returns image result of
conv(conv(im, f_1), f_2)

† CS6640_Prewitt2D(im, k)
returns conv(im, F)

CS6640_A3_driver
* varies k from 3 to 100
* times CS6640_Prewitt1 t_1
* times CS6640_Prewitt2 t_2
* returns $\begin{matrix} & k & t_1 & t_2 \\ 1 & & & \\ 2 & & & \\ \vdots & & & \\ 98 & & & \end{matrix}$

Issues: How to perform convolution?

* imfilter
* conv, conv2
* your own

How to force MN computations?

* Pad image with 0's so there's always M x N where k x k fits

How to account for OS + HW?

* for each k, run multiple times and average

What is role of M + N?

* must be large enough to show effect

P2. Issues?

P3. Issues: size of smoothing