

# CS419/519 Image and Flow Synthesis

## Project I

Due April 26, 2008

In this project, you will write a program to extract the edges from an image and display the edge field both as a vector field and as a tensor field.

Please submit everything necessary to compile and run your program, and a project report that contains results and discussions. Note it is impossible to obtain a good score without a well-written report. **Include images to illustrate your ideas and results.** Feel free to explore and have fun!

**Note:** the program `display_ppm` is provided for your convenience. It contains necessary components to read in a squared image and display it in an input window. Feel free to develop your own programs to manipulate images. Also, you may use other tools to construct the user interface, such as `glui`, `Microsoft API 32`, or `MFC`. Be sure to provide a `README` file so that I know how to run your program.

1. **(Image decolorization)** For an input image  $I$  of  $N \times N$  pixels, compute the intensity of each pixel based on two different formulas and compare the results. Save the result in an image. Discuss in the report which formula you like better and why?

a. 
$$L = \sqrt{\frac{R^2 + G^2 + B^2}{3}}$$

b. 
$$L = 0.30 * R + 0.59 * G + 0.11 * B$$

2. **(Image smoothing)** For an input image  $I$  of  $N \times N$  pixels and an integer  $n (\ll N)$ , compute  $I_n$ , the smooth version of  $I$  in the following way:

$$I_n(x, y) = \frac{1}{(2n+1)^2} \sum_{i=x-n}^{x+n} \sum_{j=y-n}^{y+n} I(i, j). \text{ Save the results for when } n=1, 2, 4, \text{ and } 8.$$

What do you observe? Apply the image gradient computation to the smoothed image. What do you find and what do you think are the reasons?

3. **(Image gradient and edge)** For an input image  $I$  of  $N \times N$  pixels, compute the image gradient vector for a pixel  $(x, y)$  according to the following formula:

$$G(x, y) = \begin{pmatrix} L(x+1, y+1) - L(x-1, y+1) + 2(L(x+1, y) - L(x-1, y)) + L(x+1, y-1) - L(x-1, y-1) \\ L(x-1, y+1) - L(x-1, y-1) + 2(L(x, y+1) - L(x, y-1)) + L(x+1, y+1) - L(x+1, y-1) \end{pmatrix}$$

Denote  $G(x, y) = \begin{pmatrix} G_x \\ G_y \end{pmatrix}$ . Then compute the vector-based edge field  $E_v(x, y) = \begin{pmatrix} -G_y \\ G_x \end{pmatrix}$

and the tensor-based edge field  $E_T(x, y) = \rho(x, y) \begin{pmatrix} \cos 2\theta(x, y) & \sin 2\theta(x, y) \\ \sin 2\theta(x, y) & -\cos 2\theta(x, y) \end{pmatrix}$  where

$\rho(x, y) = \sqrt{G_x^2 + G_y^2}$  and  $\theta(x, y) = a \tan 2(G_y, G_x) + \frac{\pi}{2}$ . Note, the **minor** eigenvector of  $E_T(x, y)$  is  $G(x, y)$ .

4. **(Edge field visualization)** Display  $E_V(x, y)$  using an arrow plot. You may find a need to set the maximum length of the arrows. Display  $E_T(x, y)$  using line plot.
5. **(Edge field smoothing)** Pick a user threshold  $h$ . Mark as **fixed** any vertex whose image gradient  $G(x, y)$  has a magnitude larger than  $h$ . Then perform the vector-based edge field smoothing by doing the

solving  $E_V^{new}(v_i) = E_V^{old}(v_i) + dt \sum_{j \in J} (E_V^{old}(v_j) - E_V^{old}(v_i))$  for every non-fixed vertex  $v_i$ .

Each time all non-fixed vertices have been updated is one iteration. You may need to perform many iterations depending on the value of  $dt$ . Try  $dt = 0.001, 0.1, 10, 1000$ . What do you observe? Now solve a similar but different

equation  $E_V^{new}(v_i) = E_V^{old}(v_i) + dt \sum_{j \in J} (E_V^{current}(v_j) - E_V^{old}(v_i))$ . Note

$E_V^{current}(v_j) = E_V^{new}(v_j)$  if  $v_j$  was updated before  $v_i$ . Otherwise,

$E_V^{current}(v_j) = E_V^{old}(v_j)$ . Try  $dt = 0.001, 0.1, 10, 1000$ . What do you find comparing to the other formulation? What do you think are the causes for this?

Compare the smoothed field with the original edge field. Which one do you like better? Include examples in the report.

6. **(Edge field smoothing II)** Repeat the above smoothing process for  $E_T(x, y)$ . Notice that you will need smooth the field as a tensor field. Use the equivalent of formula 2 for CS419 students and formula 2 or 3 for CS519 students. Compare the tensor-based smoothing with vector-based smoothing from the item 5. Include images to illustrate your points.