Final Project, due June 5, 2020

1. We consider the following procedure that removes outlying intensities before performing linear stretching to fix overexposed and underexposed images. Let f be a given overexposed or underexposed image. Given percentages $0 \leq p_{dark}, p_{light} \leq 1$, with $p_{dark} + p_{light} < 1$, let c_{dark} be an intensity such that p_{dark} percent of the image has intensities $\leq c_{dark}$. Similarly, let c_{light} be an intensity such that p_{light} percent of the image has intensities $> c_{light}$. With c_{dark}, c_{light}, define linear stretching function $g : \mathbf{R} \to \mathbf{R}$ as the line satisfying $g(c_{dark}) = 0, g(c_{light}) = 1$. Finally let the fixed image be $g \circ f$.

   Numerically, we consider the discrete image function f_{ij}. To approximate c_{dark}, c_{light}, first choose $m + 1$ evenly spaced intensities ranging from f_{min} to f_{max}; then form the piecewise linear interpolant $P(c)$ for the cumulative histogram function using these intensities as nodes; and finally solve $P(c_{dark}) = p_{dark} \cdot width \cdot height$ and $P(c_{light}) = p_{light} \cdot width \cdot height$ for c_{dark}, c_{light}. Note: if c_{dark} does not exist, set it to be f_{min}.

   Apply this procedure to the following images for the given values of p_{dark}, p_{light}, m. Plot the initial image and the final result, and write down the intensity of the final image at pixel $(64, 64)$.

   (a) "bricks.bmp" for $p_{dark} = 0$, $p_{light} = 0.1$, and $m = 3$

   (b) "flowers.bmp" for $p_{dark} = 0.1$, $p_{light} = 0$, and $m = 7$

   Then:

   (c) Apply the procedure to another either overexposed or underexposed image of your choice and to a regular image of your choice. For each, choose good p_{dark}, p_{light}, m, plot the initial image and final result, and write down the parameters you chose.

2. We consider the following procedure for performing histogram equalization on a color image. Suppose we have a color image with f_r, f_g, and f_b as its red, green, and blue channels, respectively. We seek to perform histogram equalization to each individual channel, then put these results together from the different channels to get the final color image. (see "setupfinal.C" for basic functions for reading the three channels and writing the color image from the channels)

   Numerically, choose m. Then for channel f_r, take $m + 1$ evenly spaced intensities ranging from the minimum intensity of f_r to the maximum intensity of f_r. Now let $g_r(c)$ be such that $g_r(f_r)$ is the histogram equalization of f_r. Approximate this instead by $P_r(f_r)$, where $P_r(c)$ is the piecewise linear interpolant of g_r using the $m + 1$ evenly spaced intensities as nodes. Similarly, get $P_g(f_g)$ and $P_b(f_b)$ using the same m, and combine all the channels $P_r(f_r), P_g(f_g), P_b(f_b)$ to form the final image.

   Apply this procedure to the following images for the given values of m. Plot the initial image and the final result, and write down the red, green, and blue intensity values of the final image at pixel $(64, 64)$.

(a) "brickscolor.bmp" for $m = 7$

(b) "flowerscolor.bmp" for $m = 15$

Then:

(c) Apply the procedure to two more color images of your choice with also your choice of good m. For each, plot the initial image and final result, and write down the m you chose.

3. We consider the following procedure for resizing a color image. Suppose we have a color image with f_r, f_g, and f_b as its red, green, and blue channels, respectively. We simply resize each individual channel and put together the results to get the resized color image. (see "setupfinal.C" for basic functions for reading the three channels and writing the color image from the channels)

   Apply this procedure to resize the following images to the following target sizes. Plot your final images and write down the red, green, and blue intensity values of the final image at pixel $(64, 64)$.

   (a) "rainbowcolor.bmp", with target size 200×400

   (b) "peonycolor.bmp", with target size 750×750

   Then:

   (c) Apply the procedure to two more color images of your choice with also your choice of good m. For each, plot the initial image and final result, and write down the m you chose.

4. We consider the replacement of the Jacobi iterative method by SOR for better convergence when solving the steady state heat equation for inpainting. This involves using SOR with a given $0 < \omega < 2$, along with the coordinate list format for sparse matrices, to approximately solve the steady state heat equation on a given discete image function f_{ij} in a given corrupted region. In application, the image is corrupted using intensities of 0.5 in the corrupted region, von Neumann boundary conditions are used at the boundary of the image, and the stopping condition is at the first iteration k satisfying

$$||f^{(k)} - f^{(k-1)}||_\infty = \max_{ij} |f_{ij}^{(k)} - f_{ij}^{(k-1)}| \leq 10^{-11}.$$

   (a) Apply this procedure to "wires.bmp", where the the corrupted region is composed of all (i, j) satisfying i odd, or j or $j + 1$ or $j + 2$ divisible by 6, and with $\omega = 1$. Plot the initial image, with intensities 0.5 in the corrupted region, and the final image.

   (a) Use $\omega = 1$. Plot the initial image, with intensities 0.5 in the corrupted region, the final image, and write down the time step k satisfying the stopping condition.

(b) Try three other $1 < \omega < 2$ of your choice, with at least one of them with faster convergence than in the $\omega = 1$ case. Write down your chosen ω and, for each, the time step k satisfying the stopping condition.

(c) Use the ω you found with the fastest convergence in the previous part to handle a different image of your choice with the same region of corruption: i odd, or j or $j+1$ or $j+2$ divisible by 6. Plot the initial image, with intensities 0.5 in the corrupted region, the final image, and write down the time step k satisfying the stopping condition.

5. We consider mean curvature flow of level-sets with some elements coming from fitting terms for denoising. For this, we combine the result of descent flow of the fitting term with the regularization of mean curvature flow of level-sets:

$$f_t = w_r \left(\Delta f - \frac{\nabla f \cdot (\nabla^2 f \nabla f)}{|\nabla f|^2 + \epsilon^2} \right) - w_f (f - f_{noisy}),$$

where f_{noisy} is a given noisy image.

Numerically, we use Euler's method in time and 2nd order central differencing for each derivative term $f_x, f_y, f_{xx}, f_{xy}, f_{yy}$, von Neumann boundary conditions at the boundary of the image, and the stopping condition of a given iteration k. For the CFL condition, we use $\Delta t = \frac{h^2}{4w_r + w_f \frac{h^2}{2}}$.

(a) Apply this procedure to "housenoisy.bmp" using $h = w_f = 1$, $\epsilon = 10^{-11}$, and $w_r = 100$ for 1000 iterations. Plot the given noisy image and your final result. Also, write down the intensity of the final image at pixel $(64, 64)$.

(b) Continuing, try fewer iterations to determine the number of iterations needed before the image is not really changing visually. Write this number down.

(c) Apply the procedure to "housenoisy.bmp" using $h = w_f = 1$, $\epsilon = 10^{-11}$, and $w_r = 10$ until the image is not really changing visually. Plot the your final result and write down the number of time steps you ran.

(d) Apply the procedure to two other noisy images using $h = w_f = 1$, $\epsilon = 10^{-11}$, and good choices of w_r and stopping iterations. For each, plot the given noisy image and your final result, and write down your choice of w_r and the number of time steps you ran.

6. We consider backwards heat flow, regularized by regular (forwards) heat flow, for the process of deblurring. Suppose an original image has been blurred by a numerical heat flow, using time stepsize $\Delta t = h^2/4$, for N steps, from time 0 to time $N\Delta t$.

To deblur, we can continually iterate:

- run a few steps, N_{back}, of numerical backwards heat flow, using time stepsize Δt_{back};

- run a few steps, $N_{forward}$, of numerical (forwards) heat flow, using time stepsize $\Delta t_{forward}$;

until we get back to time 0. Thus, a few unstable steps of backwards heat flow are performed and then regularized by a few stable steps of (forwards) heat flow, until we get back to the original image.

(a) Apply this procedure to "coconutblurry.bmp", which was blurred using heat flow for 25 steps under $\Delta t = \frac{h^2}{4}$. Use $h = 1$ and the choices of $N_{back} = 1$, $\Delta t_{back} = \frac{h^2}{4}$ and $N_{forward} = 1$, $\Delta t_{forward} = \frac{2\Delta t_{back}}{3}$. Plot the given blurry image and the final image, and write down the intensity at pixel $(64, 64)$.

(b) Apply the procedure to two more blurry images (note "houseblurry.bmp", "airplaneblurry.bmp", and "fruitblurry.bmp" are all blurred using heat flow for 25 steps under $\Delta t = \frac{h^2}{4}$). Use $h = 1$ and good choices for $N_{back}, \Delta t_{back}$ and $N_{forward}$, $\Delta t_{forward}$. For each, plot the given blurry image and the final image, and write down the parameters you chose.

7. We study the geodesic active contour algorithm on the image "tuna.bmp". Use a continuous level-set function ϕ to initialize a curve surrounding the fish in the image, and flow

$$\phi_t = \frac{1}{1 + K|\nabla f|^2}\left(\Delta\phi - \frac{\nabla\phi^T \nabla^2 \phi \nabla\phi}{|\nabla\phi|^2 + \epsilon^2}\right),$$

Numerically, use Euler's method in time and 2nd order central differencing on the derivative terms $\phi_x, \phi_y, \phi_{xx}, \phi_{xy}, \phi_{yy}$ and f_x, f_y. Use $h = 1, \epsilon = 10^{-11}$, and a good choice of parameter K. Plot the initial curve over the image (see "setupfinal.C" for basic functions for plotting the zero level-set curve over an image), then plot the curves over the image at three different times of your choice that shows the flow of the curve to a final result. Also, write down the time steps associated to each of the times you chose.

8. We study the Mumford-Shah binary level-set algorithm, which uses flips of the sign of a binary level-set function to flow and decrease an energy composed, for simplification, of just a fitting term describing best approximation by a two-intensity image. The algorithm stops when there exist no flips that decrease the energy. Use, for the initial binary level-set function, $\phi_{width-1,height-1} = -1$ and $\phi_{ij} = 1$ for all other i, j.

(a) Apply the procedure to the image "earthworm.bmp". Plot the final curve over the image and write down the final two intensities of the approximate two-intensity image.

(b) Apply the procedure to two more images of your choice. For each, plot the final curve over the image and write down the final two intensities of the approxiamte two-intensity image.

9. (Math 279) We consider the replacement of the linear interpolation, in bilinear interpolation, and the cubic interpolation, in bicubic interpolation, by a higher degree interpolation polynomial that respects edges. This can then be used for resizing an image with less harm to the edges in the image.

   Let $x_0 < x_1 < \ldots < x_n$ be nodes and let $g(x)$ be a function whose values, $g(x_0)$, $g(x_1)$, $\ldots, g(x_n)$, are known at the nodes. The ENO interpolation polynomial of degree 1 is the (Lagrange) interpolation polynomial for the nodes x_i and x_{i+1}. The ENO interpolation polynomial of degree 2 is the (Lagrange) interpolation polynomial for either the nodes x_{i-1}, x_i, x_{i+1} or x_i, x_{i+1}, x_{i+2}, depending on whether $|f[x_{i-1}, x_i, x_{i+1}]|$ or $|f[x_i, x_{i+1}, x_{i+2}]|$ is smaller, respectively. The ENO interpolation polynomial of degree 3 is the (Lagrange) interpolation polynomial taking the nodes used for the degree 2 case and adding either one node to the extreme left or one to the extreme right, depending on the absolute values of the coefficients of the x^3 term of each polynomial.

   (a) Use ENO interpolation polynomials of degree 3 and von Neumann boundary conditions to resize "rocks.bmp" to target size 500×250. Plot the final image and write down its intensity at pixel $(64, 64)$.

   (b) Use ENO interpolation polynomials of degree 3 and von Neumann boundary conditions to resize another image of your choice to a target size of your choice. Plot the final image. Plot the final image and write down its intensity at pixel $(64, 64)$.

   (c) Use ENO interpolation polynomials of degree 9 and von Neumann boundary conditions to resize an image of your choice to a target size of your choice. Plot the final image and comment on whether the result is good for such a large choice of polynomial degree interpolation.