Final Project

1. In this problem, we use the cumulative histogram to remove outlying intensities before stretching intensities to use all of $[0, 1]$. Let $f$ be an image and $P$ be a chosen percentage. Let $a, b$ be intensities such that $P$ percent of the image has intensities $\leq a$ and $P$ percent of the image has intensities $\geq b$. Then form $g \circ f$, where $g : [a, b] \to [0, 1]$ such that $g(a) = 0$, $g(b) = 1$, and the graph of $g$ is a line.

To approximate $a, b$, choose $N$ evenly spaced intensities ranging from $m$ to $M$ (where $m = \min f$, $M = \max f$), and form the piecewise linear interpolant $Q$ for the cumulative histogram function using these intensities as nodes. Then set $a$ to be the intensity where $Q$ reaches $P$ percent of the total number of pixels, and similarly for $b$. Note, we do assume the number of pixels of intensity $m$ is $\leq P$ percent of the total number of pixels.

(a) Apply this procedure to “bright.bmp”, but change $f[1][0] = 1$ and $f[0][1] = 0$, so that $f$ has at least one pixel that is white and one that is black. Use $P = 10\%$ and $N = 8$ and submit the final image and write down the intensity at pixel $(17, 17)$.

(b) Apply this procedure to “dark.bmp”, but change $f[1][0] = 1$ and $f[0][1] = 0$, so that $f$ has at least one pixel that is white and one that is black. Use $P = 10\%$ and $N = 8$ and submit the final image and write down the intensity at pixel $(17, 17)$.

2. Let $f$ be a color image, and let $f_r, f_g$, and $f_b$ be its red, green, and blue channels, respectively. For channel $f_r$ (and similarly for the other channels), let the minimum intensity be $m$ and maximum intensity be $M$, and let

$$g(c) = \frac{CH_{f_r}(c) - CH_{f_r}(m)}{CH_{f_r}(M) - CH_{f_r}(m)}.$$

Then choose $N$ evenly spaced intensities ranging from $m$ to $M$ and let $Q_{f_r}$ be the piecewise linear interpolant for $g$ using these intensities as nodes. Finally, apply histogram equalization to $f_r$ through $Q_{f_r} \circ f_r$, and build the final image from the red, green, and blue channels of $Q_{f_r} \circ f_r$, $Q_{f_g} \circ f_g$, and $Q_{f_b} \circ f_b$, respectively.

Apply this procedure to the color image “fruit.bmp” with $N = 6$ and submit the final image. Also, write down the red intensity, green intensity, and blue intensity at pixel $(17, 17)$.

3. 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. Let $x_i < z < x_{i+1}$ be a location where we want to approximate the value of $g$. The ENO polynomial of degree 1 is the Lagrange interpolation polynomial for the nodes $x_i$ and $x_{i+1}$. The ENO polynomial of degree 2 is the Lagrange interpolation polynomial for either the nodes $x_i, x_{i+1}, x_{i+2}$ or $x_{i-1}, x_i, x_{i+1}$, depending on whether $|f[x_i, x_{i+1}, x_{i+2}]|$ or $|f[x_{i-1}, x_i, x_{i+1}]|$ is smaller. The ENO 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 right or one to the extreme left, depending on the absolute value of the coefficient of the $x^3$ term for each case.
(a) Use ENO interpolation polynomials of degree 3 and first order von Neumann boundary conditions to resize “bricks.bmp”, swapping the width and height of the image. Submit the final image and write down the intensity at pixel (17,17).

(b) (Math 279) Use ENO interpolation polynomials of degree 8 and first order von Neumann boundary conditions to resize “glass.bmp”, swapping the width and height of the image. Submit the final image and write down the intensity at pixel (17,17).

4. Let \( f : [-1,1] \times [-1,1] \to [0,1] \) and suppose the image is corrupted in the domain \( D \subseteq [-1,1] \times [-1,1] \). Consider inpainting by replacing intensity values in \( D \) with those from the steady state heat equation:

\[
f_{xx} + f_{yy} = 0,
\]

where we use von Neumann boundary conditions at the boundary of \([-1,1] \times [-1,1]\). Now consider the following procedure:

- Replace values of \( f \) in \( D \) by 0.5.
- Form the linear system of equations (for the unknown vector, pixels locations \((i,j)\) should be ordered nondecreasing in \( i \) and, for those that have the same \( i \), increasing in \( j \)) that has equation associated to pixel \((i,j)\):

\[
\frac{f_{i+1,j} - 2f_{i,j} + f_{i-1,j}}{h^2} + \frac{f_{i,j+1} - 2f_{i,j} + f_{i,j+1}}{h^2} = 0,
\]

using a coordinate list format on the sparse matrix.
- Use SOR, with \( \omega = 1.9 \), on the linear system (for initial guess, use the corrupted intensities 0.5), until

\[
||f^{(n)} - f^{(n-1)}||_\infty < 10^{-6}.
\]

Apply the procedure to “ocean.bmp”, where the corrupted domain \( D \) includes all pixels \((i,j)\) satisfying \( i \leq 100 \) and \( j \leq 100 \); or \( i + j \) or \( i + j + 1 \) or \( i + j + 2 \) divisible by 9. Submit your final image and write down the number of iterations and the final infinity norm error. Compare this to the number required by Jacobi method using the same stopping condition.

5. In this problem, we try curvature flow of level-sets with fitting term,

\[
\frac{\partial f}{\partial t} = \left( \Delta f - \frac{\nabla f \cdot (\nabla^2 f \nabla f)}{|\nabla f|^2 + \epsilon^2} \right) - 2\lambda(f - f_0),
\]

where \( f_0 \) is the original noisy image, to denoise an image.

Apply this procedure to “pebblesnoisy.bmp” using \( h = 1 \), \( \epsilon = 10^{-9} \), \( \lambda = 0.1 \), \( k = h^2/(4 + h^2\lambda) \), and 500 steps. Submit your final image and write down the intensity at pixel (17,17).
6. In this problem, we try to deblur an image using forwards and backwards heat flow. Suppose an original, sharp image has been blurred by heat flow, using time stepsize \( k = h^2/4 \), for a number of steps \( N \).

To deblur, we can continually iterate:

- run a few steps, \( N_1 \), of backward heat flow, using time stepsize \( k_1 \);
- run a few steps, \( N_2 \), of (forward) heat flow, using time stepsize \( k_2 \);

until we get back to the initial time.

Apply this procedure to “okapiblurry.bmp”, which was blurred from heat flow for 40 steps with \( h = 1 \), choosing \( N_1 = 1 \), \( k_1 = h^2/4 \) and \( N_2 = 1 \), \( k_2 = 2k_1/3 \). Submit the final image and write down the intensity at pixel (17,17).

7. Capture the fish in “tuna.bmp” by starting with the rectangle encompassing pixels \([30,210] \times [20,110]\), using a continuous \( \phi \) level set function, and flowing by

\[
\frac{\partial \phi}{\partial t} = \frac{1}{1 + K|\nabla f|^2} \left( \Delta \phi - \frac{\nabla \phi \cdot (\nabla^2 \phi \nabla \phi)}{|\nabla \phi|^2 + \epsilon^2} \right).
\]

Use \( h = 1 \), \( \epsilon = 10^{-9} \), \( k = h^2/4 \), \( K = 10000 \), and run for the following number of steps. Submit your images, with the final red curve attempting to circle the fish.

(a) 10000 steps.
(b) 20000 steps.
(c) 40000 steps.

8. (Math 279) Let \( f \) be an image, and let \( \vec{f} \) be a vector containing all the \( f_{i,j} \) as its components. Also let \( f_0 \) be the original noisy image.

(a) For

\[
E(\vec{f}) = \sum_{i=0}^{width-1} \sum_{j=0}^{height-1} \sqrt{(f_{i+1,j} - f_{i,j})^2 + (f_{i,j+1} - f_{i,j})^2 + \epsilon^2 + \\
\lambda \sum_{i=0}^{width-1} \sum_{j=0}^{height-1} (f_{i,j} - (f_0)_{i,j})^2},
\]

find \( \partial E/\partial f_{ij} \) and, from this, write down the gradient descent flow \( (f_t)_{i,j} = \ldots \).

(b) Consider Euler’s method to solve for the gradient descent flow. Run this algorithm on “birdnoisy.bmp”, with time stepsize \( k = 10^{-4} \), \( \lambda = 4 \), and \( \epsilon = 10^{-9} \), for 2000 or 3000 steps. Submit your final image and write down the intensity at pixel (17,17).

(c) Try larger time stepsizes \( k \), running to the same final time of 0.2 or 0.3. What can you get \( k \) up to, roughly, while still remaining stable?
9. (Math 279) Consider the binary level set function $\phi$ that only returns values of $\pm 1$. Consider the energy

$$E(\phi, C_{in}, C_{out}) = \sum_{i=0}^{width-1} \sum_{j=0}^{height-1} (f_{ij} - C_{in})^2 H(-\phi) + \sum_{i=0}^{width-1} \sum_{j=0}^{height-1} (f_{ij} - C_{out})^2 H(\phi),$$

where $H$ is the heaviside function:

$$H(x) = \begin{cases} 0, & x < 0 \\ 1, & x > 0 \end{cases},$$

whose minimizer forms the best 2-intensity approximation of a given image $f$.

(a) Viewing $C_{in}, C_{out}$ as fixed constants, what is the change in energy if, at pixel $(i, j)$, we flip the sign of $\phi$, changing $\phi_{ij}$ into $-\phi_{ij}$?

(b) Get expressions for (optimal) $C_{in}$ and $C_{out}$ when $\partial E/\partial C_{in} = 0$ and $\partial E/\partial C_{out} = 0$.

(c) Consider the algorithm:

- Start with a binary level set function with values of all 1’s, except at pixel $(0, 0)$, where there is a value of $-1$;
- Calculate optimal $C_{in}$ and $C_{out}$;
- Going through all the pixels of the image, record the energy change from flipping the sign of $\phi$, using the $C_{in}, C_{out}$ calculated in the last step;
- Going through all the pixels of the image, flip the sign of $\phi$ if the energy change recorded in the last step is $< 0$;
- Go back to the second step (calculating optimal $C_{in}, C_{out}$) unless no flips were made in the previous step.

i. Apply this procedure to “tuna.bmp” and form and submit the final 2-intensity image. Also, write down the values of the two intensities.

ii. Apply this procedure to “chameleon.bmp” and form and submit the final 2-intensity image. Also, write down the values of the two intensities.

iii. Apply this procedure to “chameleonnoisy.bmp” and form and submit the final 2-intensity image. Also, write down the values of the two intensities.