Stanford CS248: Interactive Computer Graphics
Participation Exercise 7

Problem 1: Two Short Problems

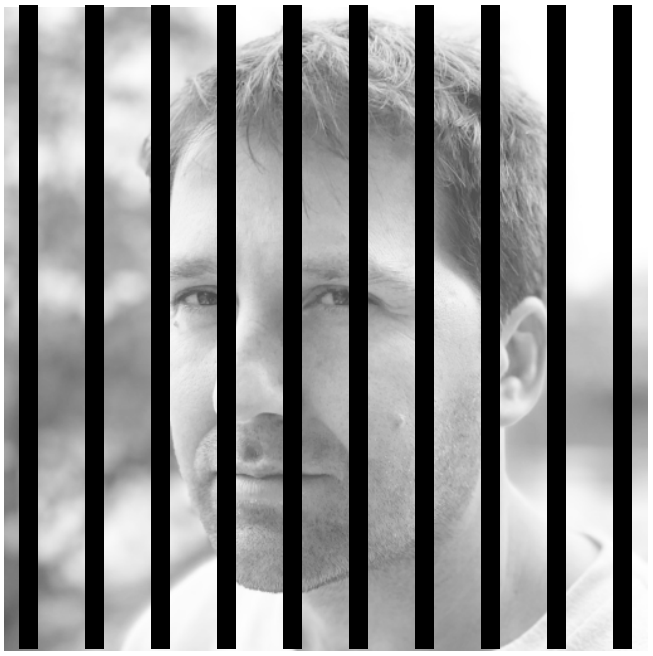
A. On your first day at your new job you are asked to implement the new autofocus algorithm for your company's new camera. Your boss tells you, "We'd like to make sure our camera chooses to focus on objects in the center of the image, and we want to make sure our autofocus algorithm doesn't get tricked by noise, which I'm sure you learned in CS248 constitutes very high frequency content in an image."

Consider the basic pseudocode of an autofocus algorithm below. If you are looking for strong edges near the center of the image, how might you use the Laplacian pyramid to determine if a distance d is a good candidate for the camera's focus? You can describe your answer without pseudocode.

```
for all possible sensor distances d:
   image = take_photograph()
   lap = compute_laplacian_pyramid(image);
```

B. From the beginning of time, students have made memes making fun of their professors. After the CS248 exam, the students feel like they want to lock their professor up, so they make the following image, and post it the web. Prof. Kayvon finds the photo on the internet, and says, "Making fun of me online is bad enough, but you could have at least done a little image blurring to smooth out my skin tones to make me look good in the photo!" Assuming that no pixel in the original photo had intensity less than 0.5, describe a bilateral filter that will blur out the Kayvon background photo using a 5×5 Gaussian blur, but not impact the sharp lines of the black bars.

Your answer give a rough expression for $\mathrm{out}(x, y) = ...$ however, feel free to use $G(i, j)$ for $(0 \leq i, j \leq 4)$ for the value of a Gaussian.



Problem 2: Deriving a New Spline

A. Consider the following linear basis functions, defined on the domain $0 \leq t \leq 1$.

$$B_0(t) = 1 - t$$
$$B_1(t) = t$$

Consider a 2D animation of the position of a bird, defined by N 2D keypoints. (We'll refer to these points as p_i's). These keypoints define $N - 1$ segments of motion. Each segment is defined by the combination of the basis functions: $f_i(t) = p_i B_0(t) + p_{i+1} B_1(t)$.

Imagine you provided the following key points: $p_0 = (0, 0)$, $p_1 = (5, 20)$, $p_2 = (10, 20)$, $p_3 = (15, 0)$. Is the resulting three-segment animation smooth (in other words, does it have continuous velocity)? Why or why not?

B. In class we talked in detail about cubic Hermite splines, where each spline segment i was defined by the positions (p_{i0}, p_{i1}) And derivatives (u_{i0}, u_{i1}) of the curve at its $t = 0$ and $t = 1$ endpoints. The matrix form of a Hermite spline is:

$$p_i(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_{i0} \\ p_{i1} \\ u_{i0} \\ u_{i1} \end{bmatrix}$$

Please give values of p_{i0}, u_{i0}, p_{i1}, and u_{i1} that yields an animation that passes through **the same keypoints as the animation from part A** (the bird is at the same location at the same time), but has continuous velocity. Your solution should begin and end the three-segment animation with ease-in, ease-out behavior. (Your solution should give a total of 12 values for the four control points for each segment.

C. Instead of specifying points and velocities, an alternative spline specification defines all $N-1$ spline segments using a sequence of N points p_i $(0 \leq i < N)$, where segment i interpolates points p_i and p_{i+1}. In this specification, we'll assume the derivative at each point p_i is set to $\frac{p_{i+1}-p_{i-1}}{2}$. **You can ignore boundary cases for segments 0, and N-1, just answer for segments 1...N-2.** Given this definition you should be able to specify a 4×4 matrix that converts the new control points $p_{i-1}, p_i, p_{i+1}, p_{i+2}$ into the inputs for a Hermite spline. Fill in that matrix below.

$$f_i(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \quad & \quad & \quad & \quad \\ & & & \\ & & & \\ & & & \end{bmatrix} \begin{bmatrix} p_{i-1} \\ p_i \\ p_{i+1} \\ p_{i+2} \end{bmatrix}$$

D. A correct answer to the previous problem would yield the overall spline matrix:

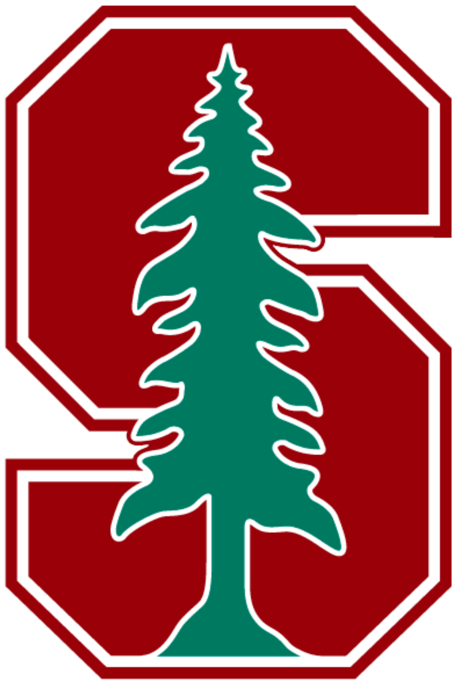
$$f_i(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -\frac{1}{2} & \frac{3}{2} & \frac{-3}{2} & \frac{1}{2} \\ 1 & -\frac{5}{2} & 2 & -\frac{1}{2} \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} p_{i-1} \\ p_i \\ p_{i+1} \\ p_{i+2} \end{bmatrix}$$

Write the four basis functions $B_i(t)$ for this new spline, such that:

$$f_i(t) = p_{i-1}B_0(t) + p_iB_1(t) + p_{i+1}B_2(t) + p_{i+2}B_3(t)$$

Problem 3: More Short Problems (THIS PROBLEM IS OPTIONAL AND NOT REQUIRED FOR PARTICIPATION CREDIT)

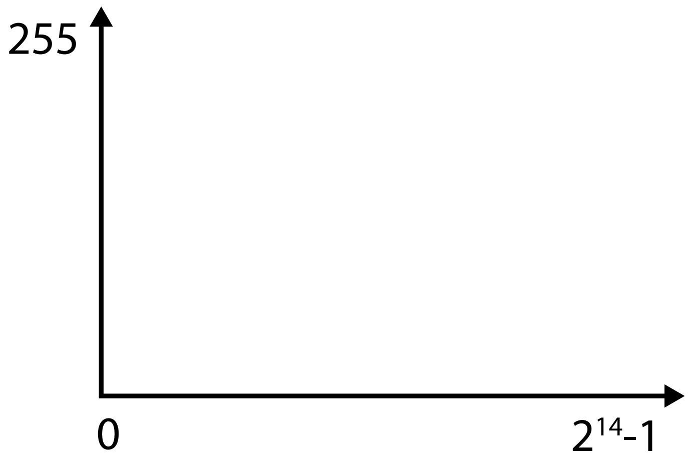
A. Consider compressing the image below using the JPG compression scheme as discussed in class. Please point to one region of the image that should compress extremely well using JPG compression. Point to one region of the image that JPG should struggle to compress well (by well, I mean retaining high accuracy while significantly reducing the amount of information that must be retained.) In both cases please explain why the region does or does not compress well?



B. Imagine if all three types of cone cells in your eye had *the same spectral response function*. If this were the case, would you have color vision (the ability to differentiate different colors)? Why or why not?

C. You are asked to implement an auto-exposure algorithm for the new smart phone. Your input is a grayscale image with 14-bit pixel values (in the range 0 to $2^{14} - 1$). You need to output 8-bit pixel values in the range 0 to $2^8 - 1$. You already know from class that it is efficient to run the input values through a non-linear function (like $f(x) \propto x^{.45}$) prior to saving to the 8-bit image.

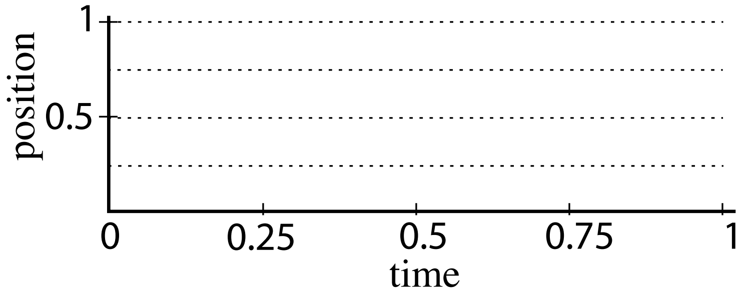
But in addition to compressing the dynamic range you wanted to implement a setting where you simulate an *under-exposed* photograph. In other words, you need to spread the top 50% of measured intensity values over the range of values representable by 8-bit numbers, clipping the bottom 50% to black. On the figure below, sketch a function that meets the exposure constraint AND also makes efficient use of all representable output values.



Problem 4: Splines (THIS PROBLEM IS OPTIONAL AND NOT REQUIRED FOR PARTICIPATION CREDIT)

A. In class we talked about the animation principle of ease-in, ease out, where a motion starts slowly, accelerates, then decelerates to a stop. Consider specifying the spline control points for an animation of a ball, where the spline's value at time t determines the ball's Z coordinate in the scene. Assume that you are to supply control point values at times $t_0 = 0$, $t_1 = 0.25$, $t_2 = 0.75$, $t_3 = 1.0$ (let's call these values C_i's). The spline *interpolates* the control points at t_0 and t_3, and the difference between C_1 and C_0 as well as C_3 and C_2 is used to compute curve tangents at the spline's endpoints.

Please draw control points on the figure below that result in an ease-in, ease-out animation where the ball begins at Z=0.25 and ends at Z=0.5.



B. Now assume that the interior control points C_1 and C_2 can be placed at any t. In other words, there are interior control points at (t_1, C_1) and (t_2, C_2). Write down the four constraint equations for the spline $at^3 + bt^2 + ct + d$ in terms of the (t_i, C_i)'s. Keep in mind that the spline interpolates (t_0, C_0) and (t_3, C_3) and the spline tangent at t_0 is the slope of the line between (t_1, C_1) and (t_0, C_0) (similarly for the tangent at t_3).

C. Now turn your four constraints into a linear system. (Fill out the coefficients for the matrix $\mathbf{M}$, as well as the vector on the right-hand side of the equation below.)

$$\mathbf{M}\begin{bmatrix}a\\b\\c\\d\end{bmatrix} = \begin{bmatrix} & & & \\ & & & \\ & & & \\ & & & \end{bmatrix}\begin{bmatrix}a\\b\\c\\d\end{bmatrix} = \begin{bmatrix} \\ \\ \\ \\ \end{bmatrix}$$

D. Imagine that instead of ease-in, ease-out, you instead want to make sure the position of the ball begins at Z=0.25, is located at Z=0.5 at t_1, at Z=0.75 at t_2 and ends at Z=0.5. What is the name of a cubic spline that provides a good representation for this problem? (Hermite? B-spline? Catmull-Rom? Bezier?) Why?

E. Let's return to the spline you derived in part C, which can be written in terms of the input control vector $\mathbf{c}$ (your right hand side in part C) and matrix $\mathbf{M}$, which you provided in part C as well. **(NOTE: a correct answer to part C is not necessary to do this problem.)**

$$Z(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \mathbf{M}^{-1}\mathbf{c}$$

Now consider a different spline representation, such as a Bezier spline, which can represent the same spline curve, but using different basis matrix and different input control points ($\mathbf{c}_{\text{bez}}$)

$$Z(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \mathbf{B}\mathbf{c}_{\text{bez}}$$

Imagine you only had an implementation of Bezier splines, but you wanted to implement the same curve that was specified in part B. How could you convert your spline control points $\mathbf{c}$ from part B to a vector of Bezier control points that result in the same curve?

F. One of the benefits of a Bezier curve is that the control points form *a convex hull* for the actual spline curve. Consider the challenge of determining if two spline curves intersect. How precisely might you use the convex hull property to accelerate intersection tests?