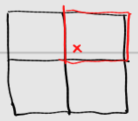
Problem 1 (3 points)

We learned about three sampling procedures that can be used to compute the pixels of an enlarged image: (1) nearest neighbour, (2) bilinear, and (3) bicubic interpolation. In the corresponding link on iCorsi, you will find four images: the original (10×10 pixels) and three enlarged (1000×1000 pixels) versions. For each of the three enlarged images, identify which sampling procedure was used and explain your reasoning.

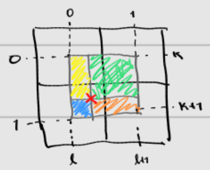
1) Nearest neighbour simply copies the pixel value of the computed equivelant under the resolution transformation. This makes transitions between sections of the image very rough and noticeble



× → computed equivelant

□ → copied pixel

2) Bilinear interpolation, unlike nearest neigbors, now considers a weighted average of the 4 surrounding pixels, making transitions between sections smoother.



$$b_{ij} = a_{k,l}(1-\mu)(1-\nu) + a_{k+1,l}(\mu)(1-\nu) + a_{k,l+1}(1-\mu)(\nu) + a_{k+1,l+1}(\mu)(\nu)$$

3) Same idea as bilinear, but instead of 2x2 kernel it uses a 4x4 kernel, resulting in higher smoothnes in transitions.

Problem 2 (3 points)

Given a set of n greyscale images A_i, $i = 1, \ldots, n$ containing uncorrelated noise, we learned how to smooth out noise by taking the average intensity of the corresponding pixels in each image, i.e., $\bar{A} = \frac{1}{n}\sum_{i=0}^{n} A_i$. Implement an equivalent formula for a set of RGB colour images B_i, $i = 1, \ldots, n$ and apply it to the set of $n = 50$ noisy images provided in the corresponding link on iCorsi. Hand in your code and your de-noised image.

Here we apply the same idea as in grayscale images, but do it for each color channel individually.

```
def reduce_noise_by_average(B):
    final_image = np.zeros_like(B[0])
    for b in B:
        final_image = final_image + b
    return final_image / len(B)
```

Problem 3 (2+2=4 points)

Given a greyscale image A with $M \times N$ pixels with intensity $A_{ij} \in \{0, \ldots, L-1\}$, $i = 0, \ldots, M-1$, $j = 0, \ldots, N-1$, let

$$p_k = \frac{\#\{(i,j) \in \{0,\ldots,M-1\} \times \{0,\ldots,N-1\} : A_{ij} = k\}}{M \cdot N}, \quad k = 0, \ldots, L-1$$

be the proportion of pixels with intensity k in A. We learned that we can equalize the histogram of A by applying the intensity transformation function

$$T\colon \{0,\ldots,L-1\} \to \{0,\ldots,L-1\}, \quad T(k) = \mathrm{round}(s_k), \quad s_k = L\sum_{j=0}^{k-1} p_j + \frac{L}{2}p_k - \frac{1}{2},$$

where "round" is supposed to round towards zero (i.e., $\mathrm{round}(-3.5) = -3$ and $\mathrm{round}(4.5) = 4$). Show that (a) T is indeed mapping into the set $\{0, \ldots, L-1\}$ and (b) T is monotonically increasing.

a) We know

(1) $\forall k \in [L-1] : 0 \leq p_k \leq 1$

(2) $\sum_{k=0}^{L-1} p_k = 1$

Then $s_k = L \sum_{j=0}^{k-1} p_j + \frac{L}{2} p_k - \frac{1}{2} \overset{(1)}{\geq} -\frac{1}{2}$ and $T(-\frac{1}{2}) = 0$, hence $T_k \geq 0$

Also $s_k < L \sum_{j=0}^{k-1} p_j + L p_k - \frac{1}{2} = L \sum_{j=0}^{k} p_j - \frac{1}{2} \leq L - \frac{1}{2}$ and $T(L - \frac{1}{2}) = L-1$, hence $T_k \leq L-1$

b) $s_{k+1} - s_k \overset{?}{\geq} 0$

$\Leftrightarrow L \sum_{j=0}^{k} p_i + \frac{L}{2} p_{k+1} - \frac{1}{2} - \left(L \sum_{j=0}^{k-1} p_i + \frac{L}{2} p_k - \frac{1}{2} \right) \geq 0$

$\Leftrightarrow L p_k - \frac{L}{2} p_k + \frac{L}{2} p_{k+1} \geq 0$

$\Leftrightarrow L p_k (1 - \frac{1}{2}) + \frac{L}{2} p_{k+1} \geq 0$

$\Leftrightarrow \frac{1}{2} L p_k + \frac{L}{2} p_{k+1} \overset{(1)}{\geq} 0$

Since $T(s_k)$ is a rounding forward zero, $s_{k+1} \geq s_k$ and $T(s_k) \geq 0$, then $T(s_k)$ is monotonically increasing.

Problem 4 (3 points)

Given an image f with 1000×1000 pixels and three 5×5 kernels g, h, and i, consider convolving the image with the formula

$$(i \star (g \star f)) + (i \star (f \star h)).$$

Rewrite this in an equivalent manner such that it minimizes the number of computations required.

$$
\begin{aligned}
& i \star (g \star f) + i \star (f \star h) \\
&= i \star [(g \star f) + (f \star h)] && \text{distributivity} \\
&= i \star [(f \star g) + (f \star h)] && \text{commutativity} \\
&= i \star (f \star (g + h)) && \text{distributivity}
\end{aligned}
$$

Problem 5 (2 points)

Given an 8-directional compass (N, NE, E, SE, S, SW, W, NW), in which direction must an edge point for the kernel $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$ to produce the strongest response?

Let's consider a 2x2 image, where the compass direction of an edge is given by the direction of largest gradient change. In example, a NE edge would be identified by the matrix $\begin{bmatrix} x+1 & x+2 \\ x & x+1 \end{bmatrix}$

Cases and response to kernel $\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$:

N:

- $\begin{bmatrix} x+1 & x+1 \\ x & x \end{bmatrix}$
- response = $x+1-x = 1$

NE:

- $\begin{bmatrix} x & x+1 \\ x-1 & x \end{bmatrix}$
- response = $x+1-(x-1) = 2$

NW:

- $\begin{bmatrix} x+1 & x \\ x & x-1 \end{bmatrix}$
- response = $x-x = 0$

E:

- $\begin{bmatrix} x & x+1 \\ x & x+1 \end{bmatrix}$
- response = $x+1-x = 1$

S:

- $\begin{bmatrix} x & x \\ x+1 & x+1 \end{bmatrix}$
- response = $x-(x+1) = -1$

SE:

- $\begin{bmatrix} x-1 & x \\ x & x+1 \end{bmatrix}$
- response = $x-x=0$

SW:

- $\begin{bmatrix} x & x-1 \\ x+1 & x \end{bmatrix}$
- response = $x-1-(x+1) = -2$

W:

- $\begin{bmatrix} x+1 & x \\ x+1 & x \end{bmatrix}$
- response = $x-(x+1) = -1$

If we consider the above compass mapping, we can se that the absolute strongest responses come from NE and SW edges.

Problem 6 (2 points)

When detecting lines using the Hough line transform, the (ρ, θ) representation of lines is typically chosen over the slope-intercept form (i.e., $y = mx + c$). Why is (ρ, θ) favourable?

The slope-intercept form of a line cannot represent vertical lines (this would equate to the parameter m going to infinity). On the other hand, we can represent every line considering it's distance to the origin (ρ) and it's angle with the x-axis (θ), thus enabling a discretization of the space of possible lines given an edge image.

Problem 7 (1 point)

Given the lines $\boldsymbol{l} = (1, 2, 3)^T$ and $\boldsymbol{m} = (-3, 2, 3)^T$ in homogeneous coordinates, compute their intersection in Cartesian coordinates.

$p = l \times m$, where $\underline{\times}$ is the cross-product operator.

Hence we have:

$$x = \begin{vmatrix} e_1 & e_2 & e_3 \\ 1 & 2 & 3 \\ -3 & 2 & 3 \end{vmatrix} = e_1 \begin{vmatrix} 2 & 3 \\ 2 & 3 \end{vmatrix} - e_2 \begin{vmatrix} 1 & 3 \\ -3 & 3 \end{vmatrix} + e_3 \begin{vmatrix} 1 & 2 \\ -3 & 2 \end{vmatrix} =$$

$$= e_1 (6-6) - e_2 (3+9) + e_3 (2+6) =$$

$$= 0 \times \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} - 12 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + 8 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -12 \\ 8 \end{bmatrix}$$

homogenous coordinates!

Conversion to cartesian:

- devide by last component
- Extract first two components

$$x' = \frac{1}{8}(0, -12) = \left(0, -\frac{3}{2}\right)$$

Problem 8 (3 points)

Show that an affine transformation can map a circle to an ellipse, but cannot map an ellipse to a hyperbola or parabola.

$H_A(x) = Ax+t$

$x \in \{(x,y) : (x-a)^2 + (y-b)^2 = r^2\}$

$x^2 - 2xa + a^2 + y^2 - 2yb + b^2 - r^2 = 0$

$Ax+t = \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} ax+by+t_1 \\ cx+by+t_2 \end{bmatrix}$

$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0$

$\begin{cases} A = 1 \\ B = 0 \\ C = 1 \\ D = -2a \\ E = -2b \\ F = a^2+b^2-r^2 \end{cases}$ $\quad B^2 - 4AC < 0$

$(ax+by+t_1)^2 + (cx+dy+t_2)^2 = r^2$

$= a^2x^2 + 2abxy + 2axt_1 + b^2y^2 + 2byt_1 + t_1^2$

$+ c^2x^2 + d^2y^2 + t_2^2 + 2cxdy + 2cxt_2 + 2dyt_2 \quad = r^2$

$(\Leftrightarrow)\ (a^2+c^2)x^2 + (2ab+2cd)xy + (b^2+d^2)y^2 + (t_1^2+t_2^2-r^2) = 0$

$(\Leftrightarrow)\ A'x^2 + B'xy + C'y^2 + \underset{0}{D'}x + \underset{0}{E'}y + F = 0$

$B'^2 - 4A'C' = (2ab+2cd)^2 - 4(a^2+c^2)(b^2+d^2) =$

$= 4(a^2b^2 + 2abcd + c^2d^2) - 4(a^2b^2 + a^2d^2 + c^2b^2 + c^2d^2)$

$= 4(2abcd - a^2d^2 - c^2b^2) = -4(ad-bc)^2 \leq 0$ $\Bigg\}$ it's an elipse or parabola

For A to be an affine transformation then A is non singular, that is $\det(A) \neq 0 \Leftrightarrow \begin{vmatrix} a & b \\ c & d \end{vmatrix} \neq 0 \Leftrightarrow ad-bc \neq 0$

Then $-4(ad-bc)^2 < 0$, which means it's an ellipse.

Problem 9 (3 points)

Consider the camera with projection matrix

$$P = \begin{pmatrix} 1 & 2 & -1 & 4 \\ -2 & 0 & 3 & 1 \\ 3 & 1 & -3 & 0 \end{pmatrix}.$$

What are the homogeneous world coordinates of the camera centre?

The camera center C are the points in 3D mapped onto the origin under P, that is

$$PC = \bar{0}, \quad \text{with} \quad C = (X, Y, Z, W)$$

$$\begin{bmatrix} 1 & 2 & -1 & 4 \\ -2 & 0 & 3 & 1 \\ 3 & 1 & -3 & 0 \end{bmatrix} \to \begin{bmatrix} 1 & 2 & -1 & 4 \\ 0 & 4 & 1 & 9 \\ 0 & -5 & 0 & -12 \end{bmatrix} \to \begin{bmatrix} 1 & 2 & -1 & 4 \\ 0 & 20 & 5 & 45 \\ 0 & -20 & 0 & -48 \end{bmatrix}$$

$$\to \begin{bmatrix} 1 & 2 & -1 & 4 \\ 0 & 20 & 5 & 45 \\ 0 & 0 & 5 & -3 \end{bmatrix} \to \begin{bmatrix} 5 & 10 & -5 & 20 \\ 0 & 20 & 0 & 48 \\ 0 & 0 & 5 & -3 \end{bmatrix} \to \begin{bmatrix} 5 & 10 & 0 & 17 \\ 0 & 20 & 0 & 48 \\ 0 & 0 & 5 & -3 \end{bmatrix} \to$$

$$\to \begin{bmatrix} 5 & 0 & 0 & -7 \\ 0 & 10 & 0 & 24 \\ 0 & 0 & 5 & -3 \end{bmatrix} = \begin{cases} 5X = 7W \\ 10Y = -24W \\ 5Z = 3W \end{cases}$$

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = W \begin{bmatrix} 7/5 \\ -24/10 \\ 3/5 \\ 1 \end{bmatrix}, \quad \text{for } W = 1 \text{ we have } \tilde{C} = \left(\frac{7}{5}, -\frac{24}{10}, \frac{3}{5} \right)$$

Problem 10 (3 points)

Let P^1 and P^2 be the projective matrices of a camera pair and $(\boldsymbol{x}_i^1, \boldsymbol{x}_i^2)$, $i = 1, \ldots, n$ be pairs of corresponding image points in homogeneous camera coordinates (i.e., the projections of certain world points $\boldsymbol{X}_i$ into the respective camera image planes). Given the $n = 10$ pairs of image points that you can find through following the corresponding link on iCorsi, use the *8-point algorithm* with the rank-2-approximation according to the Eckart–Young–Mirsky theorem to estimate the fundamental matrix F of the camera pair (P^1, P^2). Submit your code and the computed F.

```
def get_A_row((x1, y1), (x2, y2)):
    return np.array([x1*x2, x1*y2, x1, y1*x2, y1*y2, y1, x2, y2, 1])

def get_F_rank2(X):

    n = len(X)
    A = np.zeros((n, 9))

    for i, (p1, p2) in enumerate(X):
        A[i] = get_A_row(p1, p2)

    _, _, V = np.linalg.svd(A)

    F = V[-1].reshape((3,3))

    U, S, V = np.linalg.svd(F)
    S[-1] = 0

    F_rank2 = np.dot(U, np.dot(np.diag(S), V))
    # normalize to ensure epipolar constraint
    F_rank2 = F_rank2 / np.linalg.norm(F_rank2)

    return F_rank2
```

Bonus Problem (3 points)

The image on slide #60 in Lecture #18 (on "Poles and Polars") shows the geometric construction of the polar line $\boldsymbol{l}$ for a point $\boldsymbol{x}$ outside an ellipse. Give a geometric construction for the polar when the point is inside. Hint: start by choosing any line through $\boldsymbol{x}$. The pole of this line is a point on the polar of $\boldsymbol{x}$.

- For a point x inside the elipse choose 2 arbitrary lines that pass throug this point.
- For each line, draw tangent lines from points of intersection with elipse
- See where lines intersect and mark point pi.
- Draw the line l that passes through points pi. This is the polar of x, while x is the pole l.