Midterm Practice

Quiz 1: Image Formation, Camera and Lenses

Question 1

Question 1: Which of the following statements are **true** of a pinhole camera? Which are **false**?

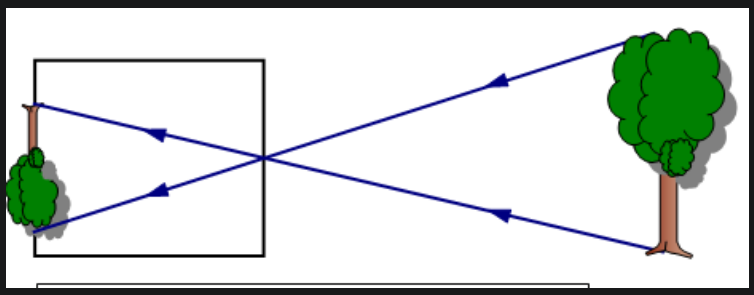
(A) A pinhole camera is a box with a small hole in it.

(B) Images in a pinhole camera are upside down.

(C) A pinhole camera has a fixed focal length, f.

(D) Images in a pinhole camera are a perspective projection.

- A) this is True, that's what the pinhole camera is
- B) this is also True, perspective projections are inverted



- C) this is False, you are able to move the focal length (how far the pinhole is from the wall)
- D) this is True, perspective projection means to create a two-dimensional representation of a three-dimensional scene which is what all cameras are doing
    - it trying to capture 3D world into a 2D surface (film or some photosensitive material)

Question 2

Question 2: Which of the following statements are **true** of an actual pinhole camera. Which are **false**?

(A) It takes too long to acquire an image.

(B) It uses an orthographic projection.

(C) It works only for black and white (B&W), not for colour.

(D) It has too small a depth of field (i.e., too small a range of object distances for which the image is in sharp focus).

- A) True - this was said in class
    - pinhole cameras are slow, because only a very small amount of light from a particular scene point hits the image plane per unit time
    - since a pinhole camera has a very small aperture, it lets in very little light so it requires a much longer exposure time to gather enough light to form a properly exposed image

- B) False - it uses perspective projection (i.e depths and stuff are preserved)
- C) False - whether it can capture colours or not is based on the sensor it uses
  - the pinhole itself lets light of all colours in
- D) False - we can use some formulas we used in class
  - we know **smaller aperture = bigger depth of field**
  - and since we have a tiny hole as our aperture - we have a huge depth of field (nearly infinite in fact)

Question 3

Question 3: Lens vignetting is a type of image distortion. Which of the statements, (A)–(D), are **true** of lens vignetting and which are **false**?

(A) Vignetting is a slight curvature of straight lines away from the center of the image.

(B) Vignetting is a shift in colour owing to the varying refraction of light at different wavelengths.

(C) Vignetting is a darkening of an image towards its edges.

(D) Vignetting makes it difficult to bring all parts of an image into focus at the same time.

- A) False
  - vignetting is about colour differences at the corners - has nothing to do with how straight lines look
- B) False
  - this is chromatic aberrations
- C) True - the only thing we know about vignetting is the following
  - common photographic effect that occurs when the corners or edges of an image appear darker or less illuminated than the center
- D) False

Question 4

Question 4: Consider Snell's law. Which of the following statements are **true**? Which are **false**?

(A) It describes how light bends when passing from one material into another.

(B) It describes how fast light travels in one material compared to another.

(C) It describes the angle at which light bounces off a mirror surface.

(D) It describes how much light is reflected and how much passes through the boundary between two materials.

- A) True - this is the definition of Snell's law
  - Snell's law, also known as the law of refraction, explains how light changes direction (bends) when it passes from one material into another with a different refractive index

- B) False - velocity is not involved
  - **corrections: The answer is TRUE**
  - the bending angle is dependant on the refraction - the reason it bends is because it moves slower
    - the light bends because it changes speed
    - so Snell's law is somewhat related to how fast light travels from one material to another
  - the law relates to the indices of refraction of the two materials, which determine the speed of light in each medium
- C) False
  - Snell's law does not describe the reflection of light off a mirror surface. It deals with the refraction of light as it passes through material boundaries, not with reflection
- D) False - again, does not deal with reflection

Quiz 2: Linear Filtering

Question 1

Question 1: Which of the following statements are **true** of a 2D (circular) pillbox filter? Which are **false**?

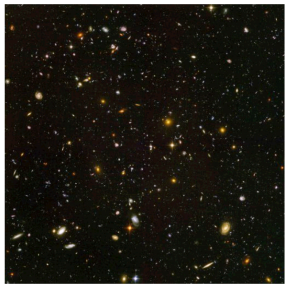
(A) A 2D (circular) pillbox filter is separable.

(B) A 2D (circular) pillbox filter is rotationally invariant.

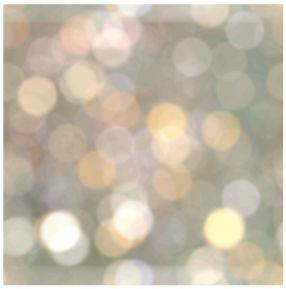
(C) Smoothing with a 2D (circular) pillbox filter acts like a "low-pass" filter. This reduces artifacts owing to sub-sampling when we construct a pyramid scaled representation.

(D) Smoothing with a 2D (circular) pillbox models "blurring" that occurs when the lens is out of focus.

- A) False - pillbox filter is NOT separable
- B) True - pillbox filter is rotationally invariant (because it's circular)
- C) True - it is does blurring - so it is a bit like a low-pass filter
  - **answer is FALSE**
    - it is a low-pass filter (so first statement is correct), but second statement is false because we use Gaussian because it better
    - pillbox kinda creates artifacts in the artifacts
- D) True - see this example



Hubble Deep View



With Circular Blur

Question 2

(a) Give a 3×3 linear filter that shifts an image 1 pixel downwards and also reduces the image brightness by 50%. Assume the filter is to be implemented as a correlation.

(b) Using your answer to part (a), what is the 3×3 linear filter if it is to be implemented as a convolution?

- A) First, let's make the filter for the correlation

$$\frac{1}{2}\begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}$$

  - the 1/2 makes it so that we reduce the brightness by 50%
  - the position of the 1 in the filter itself is a bit more complicated
    - let's imagine the filter was centered on point `(4, 4)` if the original image
      - note that the current coordinate system is `(row, column)`
    - the convolution would be `1 * image[3][4] + (0 * every_other_coordinate)` - so the result of the convolution would be the intensity that's present at pixel `(3 ,4)`, hence we've shifted the image down 1 pixel
- B) If we want it to be implemented as a convolution, we just have to flip it row wise and column wise

$$\frac{1}{2}\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 1 & 0\end{bmatrix}$$

Question 3

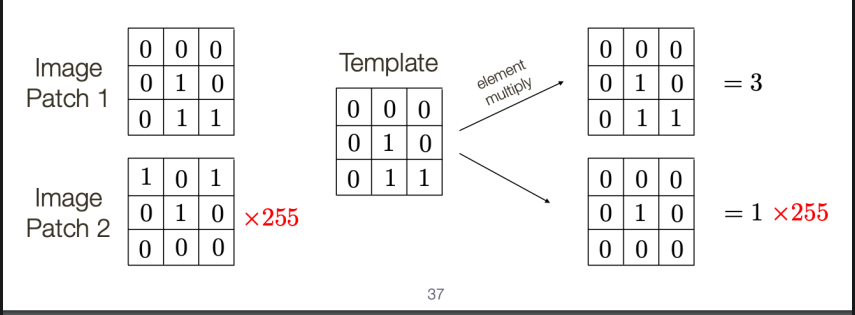
Question 3: Digital filtering in computer vision often includes some form of normalization of the filter values. In a sentence or two for each, describe a situation (or task) where the filter values are

(a) normalized to sum to one.

(b) normalized to sum to zero.

(c) normalized so that the sum of the squared values is one (i.e., the filter has magnitude one).

- A) when we are doing blurring because we want to preserves the mean of the filtered image
  - sum to 1 guarantees that the overall brightness level of the image is unchanged
- B) when we are doing derivative filters
  - **sum to zero means that the output is zero when the input image is constant** (which is what we want because we want to detect change)
- C) when we're doing template matching
  - because different regions might have different intensity, by not normalizing them we might get wildly different scores when matched with a template when it doesn't necessarily mean that one region is much more a like than the other to the template



Question 4

(a) Let S be the number of multiplication operations required to convolve a Gaussian filter with an image of size $n \times n$ pixels. Assume that we use two separable 1D filters, each of length $6\,\sigma$ as in Assignment 2. Give an expression for S in terms of σ and n. For simplicity, express S as a real number without accounting for integer roundoff in filter length or any special treatment near image boundaries.

(b) Let R be the number of multiplications operations required to convolve the equivalent single 2D Gaussian filter with an image of size $n \times n$ pixels, rather than using the two separable 1D filters. Give an expression for R in terms of σ and n. Again, express R as a real number without accounting for integer roundoff in filter length or any special treatment near image boundaries.

- A) let $m = 6\sigma$ (size of the 1D filters)
  - at each point, we are doing m multiplication (convolve with first filter) and then another m multiplication (convolve with second filter) → $2m$ operations at every pixel
  - we do this for ever pixel → there are $n \times n$ pixels
  - total is $(n \times n) \times 2m = n^2 \times 12\sigma$
  - another way of looking at it: you're basically doing 2 separate convolution (on the intermediate results of course)
    - first convolution: $n^2 \times 6\sigma$
    - second convolution: $n^2 \times 6\sigma$
    - total: $12\sigma \times n^2$
- B) again, let $m = 6\sigma$ → but the filter is now m^2 in size
  - at each point we are doing 1 convolution - which is m^2 multiplications
  - do this for every pixel
  - so in total, $(n \times n) \times m^2 = n^2 \times 36\sigma^2$

Question 5

Part 1

Question 5:

(a) We can smooth an image by convolving the 1D vector $[0.25, 0.5, 0.25]$ with the rows and then the columns of the image. Instead, give the 2D matrix that combines these row and column operations into a single 3×3 filter that can be convolved with the image. Hint: The calculation is easier if you represent the 1D vector with fractions, as in $[1/4, 1/2, 1/4]$.

- basically, you need to do a matrix multiply between the 2 1D filter

$$A = \begin{bmatrix} \frac{1}{4} \\ \frac{1}{2} \\ \frac{1}{4} \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}$$

$$= \begin{bmatrix} (1/4)(1/4) & (1/4)(1/2) & (1/4)(1/4) \\ (1/2)(1/4) & (1/2)(1/2) & (1/2)(1/4) \\ (1/4)(1/4) & (1/4)(1/2) & (1/4)(1/4) \end{bmatrix}$$

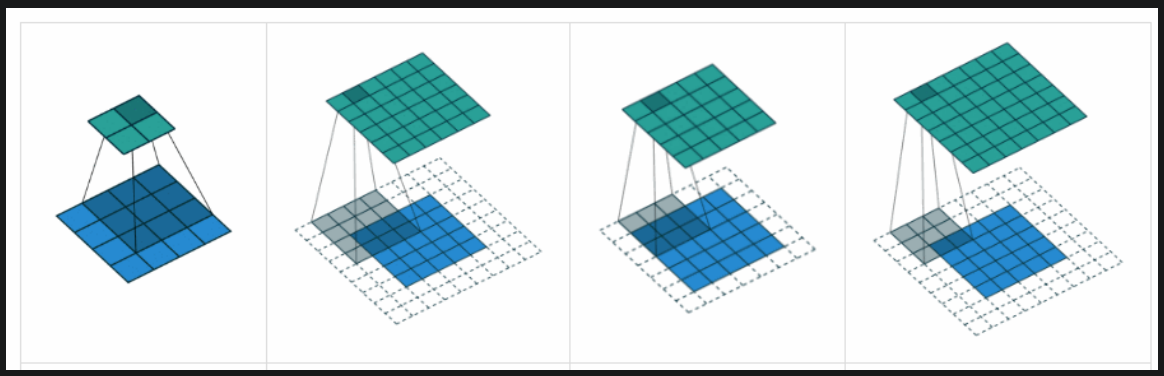
$$= \begin{bmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{bmatrix}$$

  - note: the ordering 100% matters
    - $(3 \times 1) \cdot (1 \times 3) = (3 \times 3)$ but $(1 \times 3) \cdot (3 \times 1) = (1 \times 1)$

Part 2

(b) Following this smoothing, we wish to use the 1D filter $[-1, 0, 1]$, applied to the rows of the image, to calculate the first central difference in the horizontal direction and thus estimate image gradients corresponding to vertical edges. Combine this 1D filter with your answer from part (a) to generate a single 2D filter that estimates the horizontal derivative of the smoothed image. Hint: Be sure to indicate whether you intend your final filter to be implemented as a correlation or as a convolution.

- I guess above I was cheating a bit - what you're supposed to do was to perform a "full convolution" between them



  - first one is "valid" convolution, there is no padding
  - third one is "same" padding, there is enough padding so that the result image is the same size as the original image
  - fourth one is full padding, blue square is padded as much as possible - so the result image (green) is even bigger
    - `row_padding = filter_row - 1` (do it twice, at the top and the bottom)
    - `col_padding = filter_col - 1`
    - in the last case, the filter is `(3x3)` so we pad the top row by 2
  - note: we're also supposed to do a convolution - so the filter actually becomes `[1, 0, -1]`
  - we do full padding because we want to preserve the values at the end
- doing the padding (we flipped the filter here because it's a convolution)

$$A = \begin{bmatrix} 1/16 & 1/8 & 1/16 \\ 1/8 & 1/4 & 1/8 \\ 1/16 & 1/8 & 1/16 \end{bmatrix} \circledast [1 \quad 0 \quad -1]$$

$$= \begin{bmatrix} 0 & 0 & 1/16 & 1/8 & 1/16 & 0 & 0 \\ 0 & 0 & 1/8 & 1/4 & 1/8 & 0 & 0 \\ 0 & 0 & 1/16 & 1/8 & 1/16 & 0 & 0 \end{bmatrix} \circledast [1 \quad 0 \quad -1]$$

$$= \begin{bmatrix} (0 \cdot 1 + 0 \cdot 0 + 1/16 \cdot -1) & (0 \cdot 1 + 1/16 \cdot 0 + 1/8 \cdot -1) & (1/16 \cdot 1 + 1/8 \cdot 0 + 1/16 \cdot -1) & \dots & \dots \\ & \ddots & & & \\ & & \ddots & & \\ & & & \ddots & \end{bmatrix}$$

$$= \begin{bmatrix} -\frac{1}{16} & -\frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{16} \\ -\frac{1}{8} & -\frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{8} \\ -\frac{1}{16} & -\frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{16} \end{bmatrix}$$

  - note that this would be the filter for correlation, if you wanted it for convolution, just flip it

- **TODID: how would you do this**
  - what I did is correct
- solution:

As correlation

$$\begin{bmatrix} -\frac{1}{16} & -\frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{16} \\ -\frac{1}{8} & -\frac{1}{4} & 0 & \frac{1}{4} & \frac{1}{8} \\ -\frac{1}{16} & -\frac{1}{8} & 0 & \frac{1}{8} & \frac{1}{16} \end{bmatrix}$$

As convolution

$$\begin{bmatrix} \frac{1}{16} & \frac{1}{8} & 0 & -\frac{1}{8} & -\frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & 0 & -\frac{1}{4} & -\frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & 0 & -\frac{1}{8} & -\frac{1}{16} \end{bmatrix}$$

Quiz 3: Edges and Corners

Question 1

Question 1: We have considered two approaches to edge detection, one based on extrema of a 1st derivative operator and the other based on zero crossings of a 2nd derivative operator. Which of the statements, (A)–(D), are **true** when designing a digital filter to use for differentiation and which are **false**?

(A) Filter values are normalized to sum to one.

(B) Filter values are normalized to sum to zero.

(C) Filter values are normalized so that the sum of the squared values is one (i.e., so that the filter has magnitude one).

(D) Filter values are whatever they are. Normalization is not required.

- A) False
- B) True
- C) False
- D) False
- Overall point: derivative filters are arranged in a way such that they sum to 0. This is because we want **the filter to output 0 when the input image is constant** (which is what we want because we want to detect change)

Question 2

Question 2: Two thresholds are used when linking edge points in Canny edge detection. Which of the statements, (A)–(D), are **true** of Canny edge detection and which are **false**?

(A) Different thresholds are needed to select edge points when linking edges forward or backward from the starting location.

(B) The detection of edge points is more accurate when two thresholds are used.

(C) The use of two thresholds prevents gaps that would otherwise appear in the linked edge points.

(D) The X and Y directional derivatives each require a threshold when linking to new edge points.

- A) False
  - we have 2 thresholds, but that is not what is asked for the question
  - we use the same T_{high} and T_{low} for both edge linking forward or backwards
- True

    - **the answer is FALSE**
        - it's because edge POINTS themselves don't need the thresholds - we just want to link them together better with the thresholds
        - doesn't affect accuracy - only help finding weak edges
    - (this has been said by the prof to be a shitty question as accuracy here is subjective)
- True - this is what the 2 thresholds are for
- False - we need a threshold on the magnitude of the gradient (not on the partial derivatives themselves)

Question 3

Question 3: The Harris corner detector is stable under some image transformations. For which of the image transformations, (A)–(D), is it **true** that the Harris corner detector is stable? For which is it **false**? Hint: Features are considered stable if the same locations on an object are typically selected in the transformed image.

(A) Image scaling.

(B) Image translation.

(C) Image rotation.

- A) False
    - scale invariant definition: if you scale the image, you would still find the corner at that point (scaling coordinate system as well)
    - you can think of it as because the algorithm depends on the size of the filter (i.e the window that we're sliding around) so scaling the object you might not find the corner anymore with that same filter size
    - the operations (computing gradient) requires smoothing, and the size of the smoothing matters
        - if you had image at different scale, you'd find a different gradient
- B) True - if you shift the corner around, the window will get slid over there eventually and we'll still be able to detect it
- C) True - the harris algorithm does not depend on orientation (i.e a 45 corner would also be detected, so you can rotate them and still find it)

Question 4

Question 4: Name four scene properties that would cause an edge (brightness discontinuity) in an image.

- note: I read the question wrong - I thought it was why edges are broken sometimes
    - but in actually they just want what will cause an edge

- solution:
    - a depth discontinuity (i.e., a foreground/background segmentation)
    - a surface orientation discontinuity (e.g., two intersecting planar surfaces)
    - a reflectance discontinuity (i.e., a change in surface colour/material on an otherwise smooth surface)
    - illumination boundaries (e.g., cast shadows, light sources, specularities)
- chatGPT
    - **Object Boundaries**
        - edges often occur at the boundaries of objects in a scene, where there is a change in the intensity or color between the object and its background
        - so the edge we're detecting is between an object and its background
    - **Texture Changes:**
        - differences in textures, patterns, or surface properties within a scene can result in edges
        - when one area of an image has a different texture or pattern than its neighbouring regions, an edge can be observed where the transition occurs
    - **Depth Discontinuities:**
        - in 3D scenes, changes in depth or disparities between objects can cause edges
        - these depth changes are often reflected as brightness or intensity discontinuities in two-dimensional images
    - **Material Changes:**
        - when different materials or substances with varying reflective properties are present in a scene, they can lead to brightness or colour changes and, consequently, edges in the image
    - **Shadows:**
        - shadows cast by objects can result in intensity changes, leading to the presence of edges in the image
    - **Illumination Changes:**
        - variations in lighting conditions, such as gradients in illumination, can create brightness discontinuities or edges

Question 5

Question 5: Consider the matrix, $\mathbf{M}$, defined at each image point where

$$\mathbf{M} = \begin{bmatrix} {I_x}^2 & I_x I_y \\ I_x I_y & {I_y}^2 \end{bmatrix}$$

Note that $\mathbf{M}$ can also be written as the outer product of the image gradient, $[I_x, I_y]$, with itself. That is,

$$\mathbf{M} = \begin{bmatrix} I_x \\ I_y \end{bmatrix} [I_x, I_y]$$

(a) Assuming I_x and I_y are not both zero, what is the rank of $\mathbf{M}$?

(b) Write expressions for the eigenvalues, λ_1 and λ_2, of $\mathbf{M}$.

(c) Is the computation of $\mathbf{M}$ at each image point a linear operation? Is it shift invariant?

- it's easiest to start with B
- B)

$$\begin{aligned} \det(A - \lambda I) = \lambda^2 (I_x^2 \cdot I_y^2) - (I_x I_y)^2 = 0 \\ (I_x^2 - \lambda)(I_y^2 - \lambda) - (I_x I_y)^2 = 0 \\ \lambda^2 - I_x^2 \lambda - I_y^2 \lambda = 0 \\ \lambda(\lambda - I_x^2 - I_y^2) = 0 \\ \lambda = 0,\ I_x^2 + I_y^2 \end{aligned}$$

  - since there is only 1 non-zero eigenvalue, this tells us that the rank is 1
  - another way: can get rank from determinant ($\det(A) = 0$ implies there's a rank of 1 or less)
- A) the rank is 1 (see above)
- C)
  - squaring operator itself is a non linear
  - but since the same operation is applied at every pixel - so position does not matter, in that case it is shift invariant

Quiz 4: Textures

Question 1

Question 1: Texture representation is hard. Which of the following statements are **true** of texture? Which are **false**?

(A) Texture depends on scale, illumination and viewpoint.

(B) To date, texture analysis has proven more tractable than texture synthesis.

(C) The "spots" and (oriented) "bars" approach to texture representation described in Forsyth and Ponce is motivated, in part, by properties of human vision.

(D) The Laplacian pyramid provides no explicit representation of orientation. But, if we process each layer of the Laplacian pyramid further with a set of oriented filters then we can represent energy at distinct scales and orientations as an "oriented pyramid."

- A) True
- B) False - texture synthesis is more tractable (easier) than texture analysis
- C) True
- D) True
    - this makes sense because Laplacian allows us to represent image at scale
    - oriented filter banks allow us to represent image in different orientations
    - by applying a set of oriented filters to each layer of the pyramid, one can capture the orientation information at multiple scales, creating an "oriented pyramid"

Question 2

Question 2: The Efros and Leung texture synthesis method uses a degree of randomization to select a match from among the good patch matches. What can be expected if we **increase** the degree of randomization for selecting patches? (Indicate which of the following statements are **true** and which are **false**).

(A) Unrealistic repeating patterns may appear in the texture.

(B) The accuracy of selected patches from the sample texture may decrease, leading to unrealistic textures.

(C) We will need to use a larger training sample of the texture to maintain similar performance.

(D) The method can run faster since we no longer need to compute the actual best match.

- A) False - very similar to below above but the important part is repeating
    - extra randomization means that the pattern is less likely to be **repeating**

    - this is not to say that the result won't be worse - hence why the one below is true
- B) True - as the degree of randomization increases, the accuracy of the selected patches from the sample texture may indeed decreases
    - so the resulting texture will be unrealistic overall
    - the algorithm is more likely to select patches that are not the closest match to the surrounding texture, potentially leading to less accurate and less realistic textures
- C) TODID: how would a larger training sample help if we're still picking randomly?
    - don't worry too much for this one
- D) TODID: why is this false - is this why there's randomization in the first place?
    - point of randomization is not for speed, but for variety
    - the algorithm still needs to compute the similarity between patches to determine which ones are "good" matches before randomization is applied

Question 3

Question 3: It is common to use normalization of image patches when they are being matched for stereo correspondence. For the Efros and Leung texture synthesis method (as implemented in Assignment 3) would it further improve the results also to normalize patches in the matching step? Explain your answer with just one or two sentences.

- No, this will not work
    - in template matching - we want to match so that the SHAPE matches and not the INTENSITY (i.e a bright face should still match with a dark face - because they are both face shaped)
    - but in texture synthesis, brightness matters as well
    - so if we normalize the patches ("filters") - we will sometimes match dark patches with light patches, causing a unrealistic results
-

Past Midterms

- question 7

Question 7: [6 marks]
A rectange in the plane $Y = 1$ is defined by the points

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ a \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 1 \\ a \end{bmatrix}$$

Compute the mapping of the points to the image plane under the projection equation

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

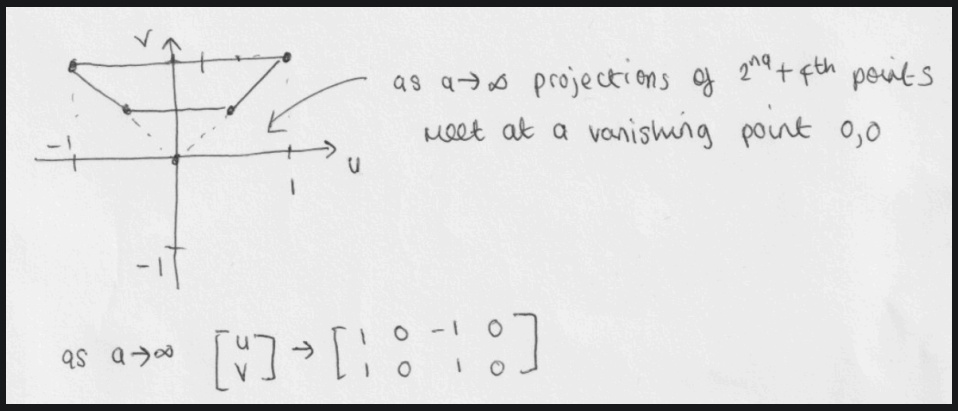
Sketch the appearance of the projected rectange for $a = 2$. Describe what happens as $a \to \infty$.

- from the projection equation, we can say that

$$\begin{aligned} su &= X \\ sv &= Y \\ s &= Z \end{aligned}$$

$$\therefore u = X/Z, \;\; v = Y/Z$$

- compute all the points (u, v) like so (divide by the z value)
- we get $(1, 1), (1/a, 1/a), (-1, 1), (-1/a, 1/a)$
- if we draw this out for $a = 2$, we can see that they are shrinking towards a point, if we make a bigger, it's approaching a vanishing point



- question 9

Below are a 3×3 filter and a 6×6 image. Your task is to apply the filter to the image as a correlation.

Filter (3×3):

-1	0	1
-2	0	2
-1	0	1

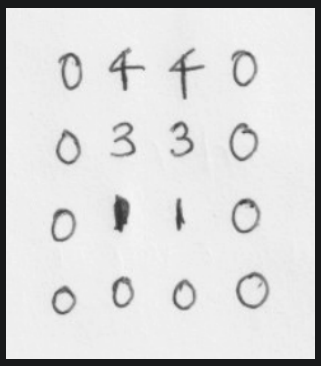
Image (6×6):

0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1
1	1	1	1	1	1

- part a

(a) [4 marks] Let the result of the correlation be a 4×4 image (defined only at locations where the filter fits entirely within the original image). Show your result here:

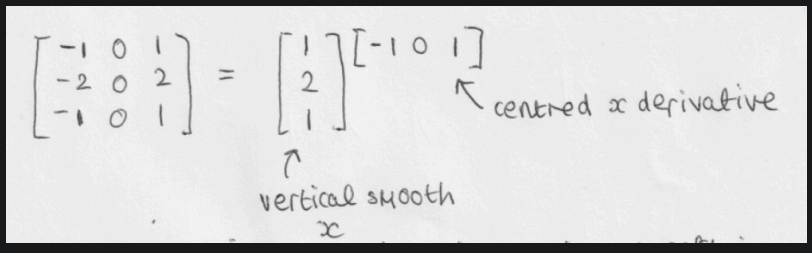
  - so no padding, we can just do it and we can see



- part b

(b) [4 marks] What sort of structure does this filter detect? The filter can be written as a product of row and column filters. Write down these filters and give an interpretation for their action if individually applied.

  - just by doing the correlation, we can see that a pattern is that we're taking a point in front (to the right of the center), and subtracting it by a point backwards → a bit like taking differences/derivative
  - we can also see that this is a Sobel filter - which we know can be broken up
  - again, see that it's forward minus backwards - so we know it's a horizontal derivative - so the horizontal component of the filter will be $[-1, 0, 1]$, and the other part is just the blurring



  - this can be used for edge detection (for Sobel's algorithm you'd need both the horizontal and vertical gradient and their magnitude though)

- question 11

Question 11: [6 marks]

(b) [4 marks] A location in an image has Harris matrix

$$\mathbf{H} = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$$

Is this likely to represent a corner? Explain why or why not. By computing eigenvalues or otherwise, deduce what kind of image structure is likely.

$$\begin{aligned} H - I\lambda &= \begin{bmatrix} 1-\lambda & 2 \\ 2 & 4-\lambda \end{bmatrix} \\ det(H - I\lambda) &= 2(2) - (1-\lambda)(4-\lambda) \\ &= 4 - 4 - 5\lambda - \lambda^2 = 0 \\ \lambda^2 - 5\lambda &= 0 \\ \lambda &= 0, 5 \end{aligned}$$

- since one of the eigenvalue is small, while the other is big → it's likely to be an edge

Final Practice

Quiz 5: Fitting Data to a Model

- (couldn't do the first 2 question because they were about Hough Transforms for some shit like that)
- question 3

Short Answer Questions.

Question 3 The title of the Efros and Leung paper that formed the basis for Assignment 4 is, "Texture synthesis by non-parametric sampling."

(a) In the context of CPSC 425, what does the term non-parametric mean?

(b) In the context of the Efros and Leung paper, is the use of the term non-parametric in the title appropriate? (Briefly justify your answer).

1. It means we are not assuming any underlying distribution about the sample
    - **solution**: close but not quite → "Non-parametric means that no assumptions are made about the particular functional form a model (or representation) takes"
2. Yes, since we are doing an exhaustive search over the sample patch every time to find the closest match to our generated texture
    - solution: correct → "Efros and Leung sample texture directly from windows in the image. They make no assumptions about the functional form an analytic or synthetic model of the texture might take"

- question 4

Question 4 Suppose we want to fit a circle to a set of points using RANSAC. Assume that 75% (i.e., 3/4) of the points are outliers. How many random samples of 3 points are needed to detect the circle with 95% probability? (Note: In an exam setting, you wouldn't need to compute an actual number, but just show how it would be computed if you had a scientific calculator available).

- first, not that you require 3 points to fit a circle (basically extrapolate the circle from a triangle)
- then you can define

$$P(\text{success for 1 trial}) = (1/4)^3 = \frac{1}{64}$$
$$P(\text{failure for 1 trial}) = 1 - \frac{1}{64} = \frac{63}{64}$$

$$\begin{aligned} P(\text{no success in } k \text{ cycles}) &= \left(\frac{63}{64}\right)^k \\ \left(\frac{63}{64}\right)^k &< 0.05 \\ k \cdot \ln\left(\frac{63}{64}\right) &< \ln(0.05) \\ k &\geq \frac{\ln(0.05)}{\ln\left(\frac{63}{64}\right)} \\ &\geq 190.225 \end{aligned}$$

- thus, 191 samples are required
  - solution: this is correct

Quiz 6: Stereo, Motion and Optical Flow

- question 1

Question 1. Consider conditions under which an epipolar constraint used in stereo matching holds between images from two cameras. Which of the following conditions are **true**? Which are **false**? Note: You can assume that the cameras perform standard perspective projection.

(A) The two cameras must have coplanar projection planes.

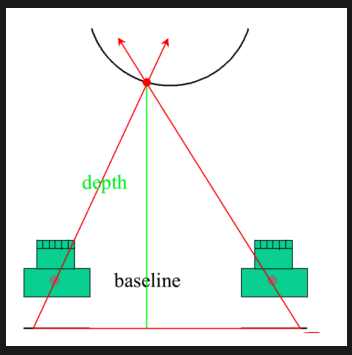
(B) The two cameras must face in the same direction (i.e., have parallel optical axes).

(C) The two images must be rectified.

(D) There are no restrictions on camera locations or orientations, an epipolar constraint always applies.

1. True. For the stereo and epipolar line thing to hold, two cameras must be on the same plane

    - **solution**: False. While usually, 2 cameras being on the same plane is preferred, it is not required. If they are on the same plane, then the epipolar line is horizontal so it simplifies the math a bit
2. True. Because if not then you'd have multi view problem instead
    - **solution**: False. Stereo is usually used to capture the same scene from different views. Even in our stereo camera, the 2 cameras are angled towards a point - much like human vision



    - TODO: double check this → also what is parallel optical axes
3. False. You can rectify the images so that they are on the same plane so that the math simples out, but this is not required, the epipolar constraints will hold regardless.
4. False. We know there's the planar constraint
    - **solution**: True. We know that they don't have to be on the same plane so the cameras can be oriented however they want as long as they are pointing at the same scene

- note: the big problem with this question was that I operated under the assumptions that the cameras must be on the same plane for stereo vision → THIS IS NOT THE CASE

- question 2

Question 2. Stereo matching can be performed by correlating windows of pixels between the two images. But, it is difficult to know what window size to use. The following statements identify problems when the selected window size is too large. Which are **true**? Which are **false**?

(A) There will be more false matches due to ambiguity and image noise.

(B) The exact location of correct matches will be known with less accuracy.

(C) Places where depth is discontinuous will be poorly matched.

(D) The epipolar constraint is not as effective to limit the number of matches.

1. False. We know that bigger windows mean less noise
    - it's less impacted by noise because it's averaging over more pixels
2. True. Larger windows means that there are less details
    - since we're averaging, this affects the precision of feature localization
3. False. Not sure why

    - **solution**: it's because we'll include pixels from different depths in the same window, which will cause the correlation measure to fail, so harder to find matches across the window
4. True. There will be more matches
    - **solution**: False. This is not asking about the number of matches, but asking about the effectiveness of the epipolar constraint. The epipolar constraint holds regardless of window size

- question 3

Question 3. The Lucas–Kanade method makes several assumptions about motion and optical flow. Which of the assumptions, (A)–(D), are **true** of Lucas–Kanade and which are **false**? Note: This is a question about the Lucas–Kanade method, not about assumptions that may or may not be true, in general, about the world.

(A) Corresponding points in a sequence of images of a moving object have exactly the same brightness values.

(B) Sampling in x, y and t is frequent enough that the partial derivative, I_x, I_y and I_t, are well-defined

(C) The motion, $[u, v]$, is constant in the selected window about each image point, $[x, y]$.

(D) The matrix

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} \sum I_x{}^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y{}^2 \end{bmatrix}$$

has rank 2 in the selected window about each image point.

1. True. This is the brightness constraint → made in every algorithm we learned, not just LK
2. True. We say that all the delta are small to make the derivatives more accurate
3. False. Do not think this was a requirement
    - **solution**: True. This is literally the assumption that LK method makes to make the equations works. The method assumes that the motion of the pixels within the window is approximately uniform
4. True. The matrix have to have this property for the series of equations to be solvable (hence why this algorithm works best when there are interesting features in the patch)

- question 4

Question 4. The second edition of Pat Winston's textbook, *Artificial Intelligence*, published by Addison-Wesley, contains a discussion of stereo vision. Included is an extended example based on a stereo pair of images shown in the text as a figure. The figure caption reads, in part, "The two pictures are arranged so that you can see depth yourself with the aid of a stereoscopic viewer." At the last minute, prior to printing, a graphic designer at Addison-Wesley made the artistic decision that the stereo pair looked better arranged above and below (i.e., top to bottom) rather than left to right. Accordingly, that is how the initial press run was printed – a left/right stereo pair printed with the left image above and the right image below.

Winston was not amused and insisted that Addison-Wesley reprint the entire book again, at its cost, with the figure in question corrected. Aside: This is a true story.

Briefly describe why Winston would insist that the figure be corrected.

- assuming the set up of the experiment was one where 2 cameras are on the same plane
- having 2 pictures side-by-side helps illustrate the fact that their epipolar lines are parallel to each other
- also this is more representative of human vision, since human eyes are side-by-side
- (this stereoscopic experiment was likely about fusing the 2 pictures together organically using your eyes - like a 3D type thing)
- in cases where the left image is placed on the right and the right image on the left, the brain can still fuse these images into a single 3D image, but the depth perception will be reversed (what should appear near will appear far and vice versa) because the disparity between corresponding points is inverted

- question 5

Question 5. As we have seen, determining corresponding points in the left image and in the right image is the hardest part of stereo vision. A variety of things can go wrong in stereo matching. In a sentence or two for each, give a specific example of a scene where

(a) there are not enough locally distinct features that match

(b) there are too many locally distinct features that match

(c) locally distinct features match incorrectly

1. If you are looking at a blank wall at different perspectives. There might not be enough unique features about the scene to match
2. When there are closely spaced, visually similar features - like a brick wall.
3. When you have repeating textures, like a wall of flowers. Even from different perspectives it's hard to say which flower match to which flower

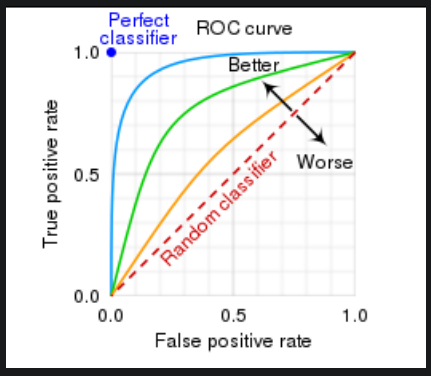
- question 6
  - simply can't do this problem because we didn't learn Hough

Quiz 7: Segmentation and Classification

- this quiz looks a little strange, I think a lot of stuff we didn't cover in this version of the class
- question 1

> **Question 1.** A Receiver Operating Characteristic (ROC) curve plots true positive rate (TPR) on the y-axis and false positive rate (FPR) on the x-axis. Which of the following statements about an ROC curve are **true**? Which are **false**?
>
> (A) The diagonal represents random guessing.
>
> (B) A good classifier lies near the upper left.
>
> (C) An ROC curve is useful for tuning a given classifier.
>
> (D) ROC curves are useful for comparing 2 classifiers.

(this is what an ROC curve looks like)



1. True. This diagonal represents a random classifier - meaning it's randomly assigning classes
2. True. TPR of 1.0 with FPR of 0 is perfect
3. True. You can use it to balance between TPR on FPR of your model
4. True. Same reasoning as above

- question 2
  - we didn't learn this kind of clustering
- question 3

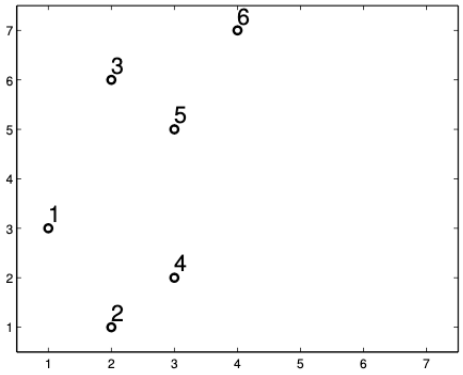
> **Question 3.** The K-means algorithm can converge to different solutions depending on which points are selected at random as the initial cluster centers. If we ran the algorithm 10 separate times with different random selections and wanted to select the best solution, how would we determine which solution is the best? Give your answer using one or two sentences.

  - you can try to minimize the inter-cluster distance
  - that is look at the distances between the points in the clusters to their cluster center → we sum them all up

- the model with the lowest sum is the best one

- question 4

Question 4. For this question, the task is to perform K-means clustering of the six labelled points at locations (1,3), (2,1), (2,6), (3,2), (3,5), (4,7), shown as follows:



We will attempt to find two clusters, A and B, and will start, as usual, by selecting two points at random as the initial cluster centers. Suppose the two random points initially selected are (1,3) for A and (3,5) for B.

(a) After the first iteration of the algorithm, what points will be assigned to cluster A? What points will be assigned to cluster B? What will be the new, adjusted locations of the cluster centers?

(b) Will subsequent iterations of the algorithm change the assignment of the points to clusters? (Briefly justify your answer).

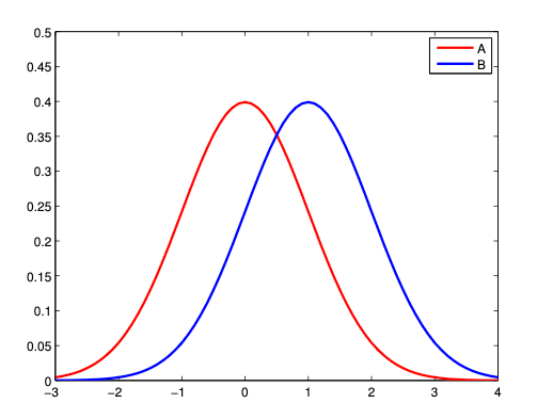
1. You can do the math but it's also pretty obvious. `(1,2,4)` will be cluster A and `(3,5,6)` will be the other cluster B. The new cluster mean is

$$
\begin{aligned}
A &= \{(1,3),(2,1),(3,2)\} \\
\text{new cluster center of A} &= \left(\frac{1+2+3}{3}, \frac{3+1+2}{3}\right) \\
&= (2,2)
\end{aligned}
$$

$$
\begin{aligned}
B &= \{(2,6),(3,5),(4,7)\} \\
\text{new cluster center of B} &= \left(\frac{2+3+4}{3}, \frac{6+5+7}{3}\right) \\
&= (3,6)
\end{aligned}
$$

2. No, if you re-ran the algorithm, you would still assign the same points to the same cluster center

- question 5
  - A) basically draw the same curve but with the center (peak) slightly shifted to $x = 1$ for distribution B



- B) with the given curve
  - you basically want $P(A \mid x) > P(B \mid x)$ or vice versa - we're talking about the y-values here
  - so you can see that where the curve intersect (call it 0.5), everywhere to the right of that, y value of B is bigger than y-value of A, and vice versa for the left of 0.5
  - so we can see that the boundary is at 0.5
- C) this is basically saying that the cost of predicting B when it's actually A is 10, and the cost of predicting A when it's B is 1

> (c) Now, assume that we have a loss function L and that $L(A \rightarrow B) = 10$ and $L(B \rightarrow A) = 1$. The Bayes estimator incorporates the loss function to reflect the cost of errors. In which direction will the decision boundary move to reflect the cost of errors?

  - I'm assuming the model will be a lot more inclined to predict A
  - so the decision boundary should shift right so that we will predict A more often → that is $x > 0.5$
- D) this would be a False Negative, we predicted negative (B) when it is actually positive (A)

Quiz 8: Colour

- question 1

Question 1. Which of the following statements about colour are **true**? Which are **false**?

(A) Modern, high quality computer monitors can display all the visible colours of the CIE XYZ colour system.

(B) Two light mixtures with different spectral power distributions will be perceived as the same colour by a (normal) human observer if they have the same coordinates in an RGB colour space.

(C) A fluorescent light bulb is more efficient than an incandescent light bulb because its output is tuned to the spectral response of human scotopic (nighttime) vision.

(D) In the human eye, the rods are cells that are used for sensing very low light levels and the cones are used for sensing under normal illumination. Only cones contribute to the perception of colour.

- A) False. Pretty sure modern displays RGB
  - but technically you could, instead of using RGB lights/dots, you could you XYZ primaries instead, but there are compatibility issues so we don't
- B) True. I know this is true for `[L M S]` responses, but not sure how those exactly map to RGB
  - yea so this is true, this is the very core principle of metamerism - where two different spectral power distributions can produce the same colour sensation and thus look identical to the human eye, even though the physical light spectra are different
- C) Literally no idea
  - **solution**: False. Fluorescent bulbs are more efficient than incandescent bulbs **but** the statement is incorrect in attributing this efficiency to the tuning of the bulb's output to the spectral response of human scotopic vision (means vision under low-light levels). Both lights are designed for photopic (daylight vision)
- D) True. We have more rods then we do cones and rods are used for night time

- question 2

Question 2. We would like to generate all the colours that humans can perceive by varying the intensity of 3 coloured lights and combining their colours. Which of the following statements about properties these 3 lights must have are **true**? Which are **false**?

(A) The lights must have colours that lie at the corners of the CIE colour diagram.

(B) The lights must be chosen so that each stimulates a single colour receptor of human vision.

(C) The lights can not be monochromatic. Instead, the individual output of each light needs to span the full visible spectrum, albeit in differing relative amounts.

(D) It is impossible to find 3 light sources that can do this.

- A) True. Just like how for RGB the colours at the corners of the triangle are primaries, same with CIE - the things at the corners are their primaries
  - **solution**: False. The three lights should ideally cover a wide range of the colour space and be able to mix to match as many perceivable colours as possible. The corners of CIE are just the most saturated colours humans can perceive - doesn't have anything to do with this
  - another way to think of it is the fact that we do RGB mixing where RGB are the primaries - but RGB does not lie on the corner of CIE colour diagram
- B) False. CIE primaries are a combination of colours - not like RGB where individually they are R G B which stimulates a single colour receptor
  - also all three types of cones (which are sensitive to different ranges of wavelengths) have overlapping ranges of sensitivity, and a single light source will typically stimulate more than one type of cone
  - key is that the three lights must be chosen such that their combined stimulations can produce the range of colours that can be perceived by these overlapping cone responses
- C) False. Just like RGB, you can just have Red or just X in the CIE case
- D) False. you probably can
  - **solution**: True. It is indeed impossible to find three light sources that can exactly match all the colours perceivable by the human eye
  - TODO: I thought CIE XYZ can cover all perceivable lights
    - you can't shine lights that are the XYZ primaries to create colors in the same way you can with RGB lights
    - The X, Y, and Z primaries are not physical colors; they are mathematical components in the CIE XYZ color space so you can't shine and then mix and match them

- question 3

Question 3. Which of the following are **true** statements about goals of a colour constancy algorithm? Which are **false**?

(A) Correct for changes in the illumination in different parts of the scene so that the entire image is made consistent with what would be seen with a single, uniform light source.

(B) Correct the colours of an image so that the brightest patch is normalized to white.

(C) Normalize the colours in an image so that all colours have the same average brightness.

(D) Correct the colours of an image taken under a coloured light so that they appear the same as if seen under a standard white light.

(not sure if we actually learned this in class - I think we did it for gamma correcting or something like that, the thing about snow actually being blue but appearing white to humans)

- Color constancy is the ability of the human visual system to perceive the colors of objects as relatively stable and consistent under varying lighting conditions
- The purpose of color constancy algorithms is to replicate this perceptual phenomenon in computer vision and image processing
- These algorithms aim to ensure that the colours of objects appear consistent, or close to their true colours, regardless of changes in illumination
- given all of this, only D is the correct answer

- question 4

Question 4. If you were asked to develop a computer vision system to help a paint store match the colours of paints for customers, would it be best to use the standard CIE XYZ colour space or a uniform colour space? Explain your answer with just one or two sentences.

Solution : You should use a uniform colour space because it measures which differences in colour will be perceptible to human vision.

- TODO: better explanation
- ChatGPT: In simpler terms, a uniform colour space is better for matching paint colours because it's designed to match the way we see colours. This means that if two colors look slightly different to us, they will also look slightly different in the uniform color space. The CIE XYZ colour space doesn't match our vision this way, so it's not as good for making sure paint colours look the same to our eyes.

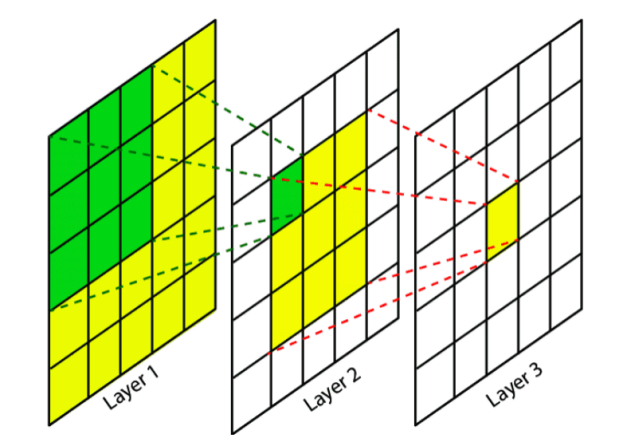
Quiz 9: Neural Networks

- question 1
  - A) False. I think it would have less information since we are just taking the max over a neighbourhood
    - **solution**: True. By taking the max, it makes the neurons in subsequent layers have a 'larger' receptive field, meaning they are affected by a larger part of the input image

    - B) False. Doing max pooling reduces the output dimension. If your neural network was a FC NN - this would actually greatly reduce the number of parameters for later layers
    - C) False. Since we are doing max-pooling, the network is less sensitive to positions
        - so we're making it less sensitive to translations in the input image → because it takes the maximum value within a window, small changes in the position of the feature within that window will not change the output of the max-pooling operation
- question 2
    - A) Not sure
        - **solution**: False.
            - scale invariance means that if you enlarge or reduce the size of an input image, the network would still recognize the objects in the image with the same accuracy
            - basic CNN without additional scaling mechanisms does not inherently possess this property; its filters are designed to detect patterns at a scale they were trained on
        - **TODO: how to determine if something is scale/rotation/etc invariant or not**
    - B) False. Rotating the input image gives different results
        - if you rotate the input image, the patterns that the convolutional filters have learned to detect may not be recognized if they appear at a new, untrained orientation
        - recall from template matching, convolutions can't detect features when they are rotated
    - C) True. Shifting the image slightly will give same result
        - TODO: why, chatGPT say it's because of max-pooling (also don't say "Convolutions are shift-invariarnt" - WHY are they shift invariant)
    - D) False
- question 3
    - forward pass: to compute the linear combination and activiation function
    - backward pass: to compute the partial derivatives for all the parameters with respect to the loss
- question 4
    - λ in this case is the step size or learning rate
    - setting this to be too large means that you converge too quickly, but it's possible to overshoot the optimal solution and never getting a solution (or getting a bad one)
    - setting this to be too small mean that you converge very slowly, more likely to get a good solution but there's a possibility that you end up in a sub-optimal minima
- question 5
    - note: the question is asking "You want to map **every possible image** of size 64 x 64 to a binary category (cat or non-cat)" and not "how many bits are needed to map **one** image"
    - with `64x64` image and `3` channel you have `64x64x3` total cells that you need to fill
        - the first cell can take 256 values
        - 2nd cell can take 256 values

        - 3rd cell can take 256 values, etc
        - so overall the combinations are $256^{64 \times 64 \times 3}$
    - TODO: better explanation
- question 6
    - A) assuming we're talking FC NN here
        - TODO: see 19.4 in the notes - the dimensions are backwards, how are they going to handle that
        - hidden layer
            - 100 hidden layers means that we have 100 rows in W_1
            - all the neurons in hidden layer is fully connected with the input which is all the pixels in the image or $64 \times 64 \times 3$
            - so W_1 is 100×12288
        - output layer
            - you will only need **1 neuron in the output layer** (because it's binary) → so W_1 have 1 row
                - TODO: is this a special feature because we have binary, say we have 4 mutually exclusive classes, how many neurons do we need? ChatGPT says yes
            - there are 100 neurons in the hidden layer which is fully connected to the output layer, so we have 100 columns
            - so W_2 is 1×100
    - B) do we have 1 bias per neuron? if so then
        - $b_1 = 100 \times 1$
        - $b_2 = 1 \times 1$
    - C) it should be the sum of all the elements in W_1, W_2, b_1, b_2
        - so it's $(64 \times 64 \times 3 \times 100) + (1 \times 100) + (100) + (1)$
    - D) you have the number of parameters from C), each of which takes 64 bits to represent, so you just multiply the value above by 64
        - $64 \times [(64 \times 64 \times 3 \times 100) + (1 \times 100) + (100) + (1)]$
- question 7
    - A) reasonable idea
        - filter size of 11×11 since that's the biggest a car can be
            - note that we don't need to make a 3D filter because the picture is in grayscale
        - activation layer (max-pool) has no parameters
            - but the size of the output map is $(1024 - 10) \times (1024 - 10)$ because we don't pad
            - (it cannot slide over the last 10 pixels of the image's border, which explains the subtraction of 10)
        - we'd have $(11 \times 11) + 1$ weights to learn so 122
            - the extra 1 is the bias term

- B) I'm guessing
  - you add more filters in hopes that they will pick up different parts of a car
  - another thing you can do is blur and downsample the image
  - or just add more layers
  - **solution**: something to do with receptive field



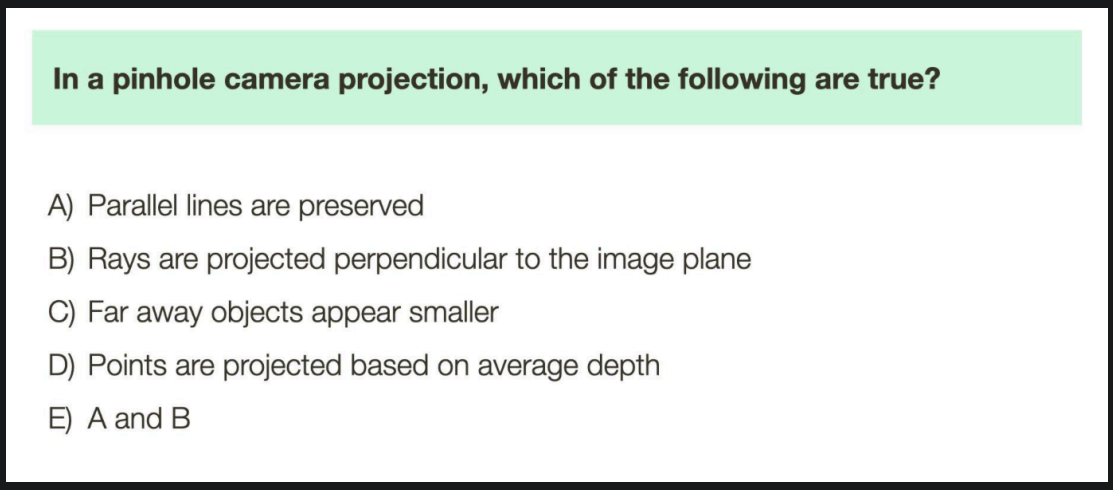
    - A single layer with a 3x3 filter has a receptive field of 3x3. When you stack another 3x3 filter layer on top, the second layer's neurons are not just looking at a 3x3 area from the original image; they're looking at a 3x3 area from the first layer's feature map - which looks at the 3x3 feature map from the original image → the overlap shown here say that it's seeing 5x5 neighbourhood in the original image
    - so first layer you see 3x3
    - second layer you see 5x5
    - third layer you see 7x7
    - fourth layer you see 9x9
    - fifth layer you see 11x11
    - so you want to do 5 consecutive layers of 3x3 filters → this is $5(3 \times 3) + 5(1) = 50$ parameters including biases
- question 8
  - since all of the filters are in 1 convolution layer, the stride doesn't really affect the number of parameters, we have 48 weight matrices, each of which are $(7 \times 7 \times 128) + 1$ so in total it's $48 \times (7 \times 7 \times 128 + 1)$
  - the output size after applying will be affected by the stride
    - if stride was 1, we'd have $(512 - 6) \times (512 - 6) \times 48$
    - so if the stride is 2, make sense we'd have $\frac{512 - 6}{2} \times \frac{512 - 6}{2} \times 48$
- question 9
  - I would apply max-pooling, max-pooling reduces the dimensions and is a way of down smapling
  - do multiple convolutional layers so that it can learn multiple abstract features (i.e nose, eyes, mouths, etc)
  - in my assignment, I created my own filter (i.e using a picture of a face), in the NN case, the weights/filters will be learned during the training process

        - **TODO: why is the answer so different**
    - question 10
        - A) so there are 8 filters, each are $5 \times 5 \times 6$ - so they're $(8 \times (5 \times 5 \times 6) + 8)$ parameters
            - extra 8 for the outputs
        - B) multiply it all together $(6 \times 20 \times 20) \times (8 \times 10 \times 10)$
            - we'll have $6 \times 20 \times 20$ input nodes
            - we need $8 \times 10 \times 10$ neurons in the hidden layer to make it match the dimensions specified in the question

In-class Quizzes

Quiz 1

- question 1

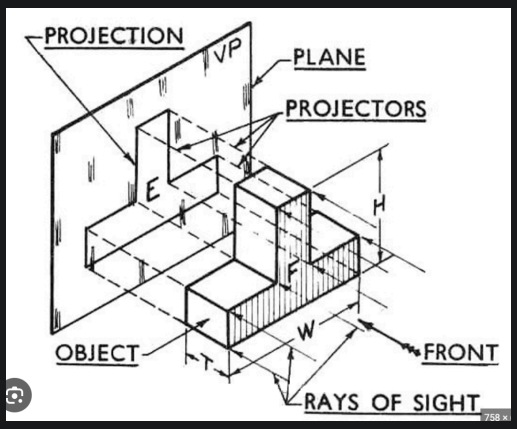


    - A) False - because cameras uses perspective projection, parallel lines will be altered to show depth
    - B) False - rays are projected through the pinhole onto the image plane
    - C) True - because it simulates depth (i.e human perception) - far away objects appear smaller
    - D) False - points are projected based on their individual depths (distance from the image plane), not their average depth (this would be scaled ortho → simulates depth perception)
- question 2

In an orthographic projection, which of the following are true?

A) Parallel lines are preserved
B) Rays are projected perpendicular to the image plane
C) Far away objects appear smaller
D) Points are projected based on average depth
E) A and B

- A) True - orthographic perception ignores depth, so parallel lines appear parallel
- B) True - it's a bit confusing but since it ignores depth, the rays that are projected only comes from the "face" that's parallel to the image plane, thus the rays from that is perpendicular to the image plane



- C) False - there's no depth so things don't appear "smaller"
- D) False - again, no relation to depth
- E) True since both A and B is true

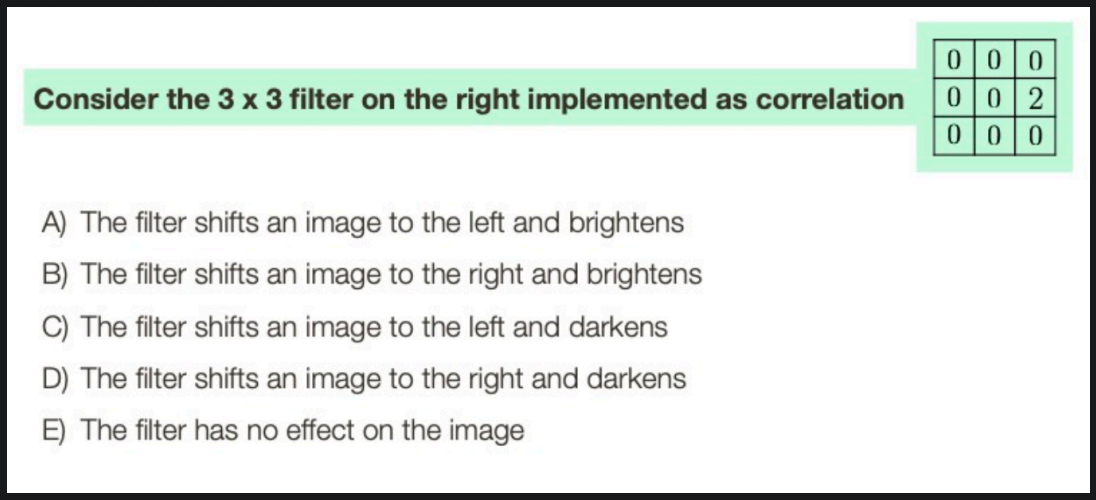
- question 3

The apparent brightness of a pure diffuse / Lambertian surface depends on:

A) The angle between the viewer and the reflection direction
B) The angle between the light source and the surface
C) The shininess of the surface
D) A and B
E) A, B and C

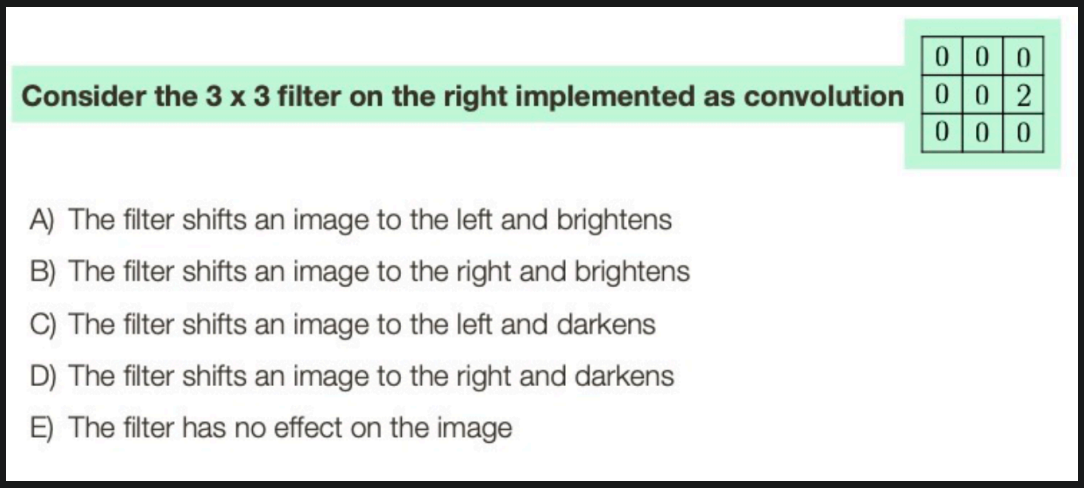
- for diffuse surface, light is reflected everywhere → key point is that the observed intensity is independent of the viewer's position, but depends on the angel between the light source and the surface
- hence only B is true

- question 4



- so the only interesting value is to the right - so it is actually shifting left
  - (run it on a 5x5 image with only 1 in the middle)
- the value is also 2 so that means it's doubling the value at a point - so it's actually brightening the image
- so the answer is A

- question 5



- note: this question is about convolution instead
- you can flip the kernel twice, it'll look like

$$\begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

- it still brightens but it's shifting it right now
- so B is right

- question 6

A circular / pillbox filter is

A) A type of low-pass filter
B) Rotationally symmetric
C) Expressible as a product of row and column filters
D) A and B
E) A, B and C

- A) True, it's blurring is it's low-pass
- B) True, it's circular so it's rotationally invariant (if you want to think about it like that)
- C) No, the function for the pillbox is super messy - not separable
- thus D is true

Quiz 2

- question 1

A bilateral filter

A) Computes the median of nearby pixel values
B) Involves domain and range kernels
C) Combines Gaussian and Pillbox Filters
D) Smooths the image whilst preserving edges
E) B and D

- A) False - that's the median filter
- B) True - it assigns Gaussian weight based on how far a point is from the center point (domain) and how far apart their values are (range)
- C) False - this just isn't true
- D) True - this is a hallmark of Bilinear - it smooths while preserving edges
- so E is correct

- question 2

Aliasing can be avoided by

A) Increasing the sampling rate
B) Blurring the input
C) Using a low-pass filter
D) A and C
E) A, B and C

- A) True - you sample more and you won't lose as much information
- B) True - blurring the input "smooths" the info - so when you sample, you're not losing as much
- C) True - same thing as blurring above
- so E is true

- question 3

The Nyquist sampling criterion can be written as

A) $f_s > f_{max}$
B) $f_s < f_{max}$
C) $f_s > 2 \times f_{max}$
D) $f_s < 2 \times f_{max}$
E) $f_{max} > 2 \times f_s$

- point: you have to sample at a rate f_s that's 2 as much as the maximum frequency $f_{\max}$ of a signal
- so C is True

- question 4

The 2D Fourier Transform of an image

A) Shows the magnitude and direction of spatial frequencies
B) Is used for Gamma correction
C) Can be used to perform convolution
D) A and C
E) A, B and C

- (I would personally not worry too much about this question)
- A) True - FTT transforms from spatial domain into frequency domain
  - B) False - Gamma correction uses the power rule to make things brighter
- C) True - the 2D Fourier transform can be used to perform convolution in the frequency domain
- so D is True

- question 5

Template matching can be made more robust using

A) A Laplacian Pyramid
B) An edge preserving filter
C) A large template
D) Normalized correlation
E) A separable filter

- A) False - we use Gaussian, using a Laplacian might highlight the edge and stuff more, which in this case (looking for faces) isn't what we want
- B) False - same reason as above, we're not looking for edges here
- C) False - it'll still be sensitive to scale (now can't detect small faces)
- D) True, for a couple of reasons
  - it's less sensitive to brightness
  - produces an interpretable score that's between -1 and 1
- E) False - just makes things more efficient, not robust
- so D is true

- question 6

A Laplacian of Gaussian filter is correctly described as

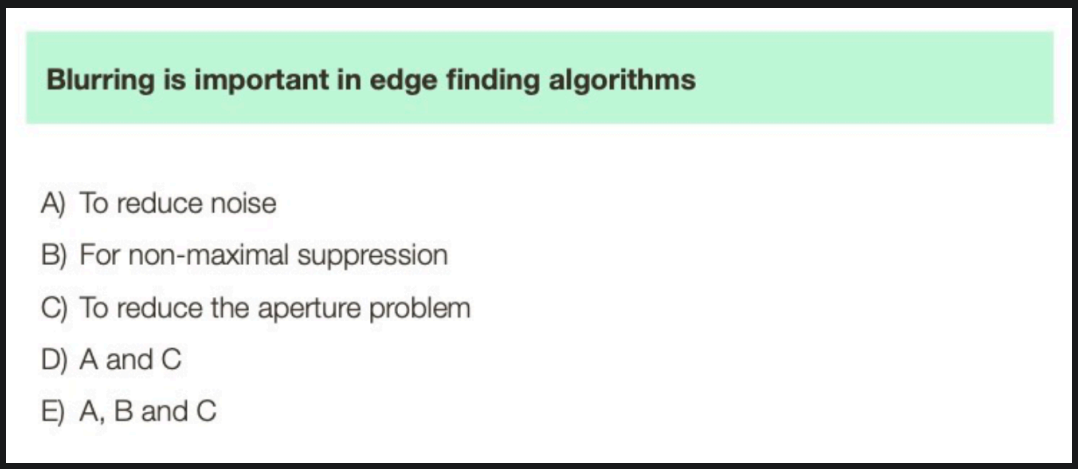
A) A band pass filter
B) A separable filter
C) A high pass filter
D) A low pass filter
E) An edge preserving filter

- Gaussian FILTER is a low pass filter

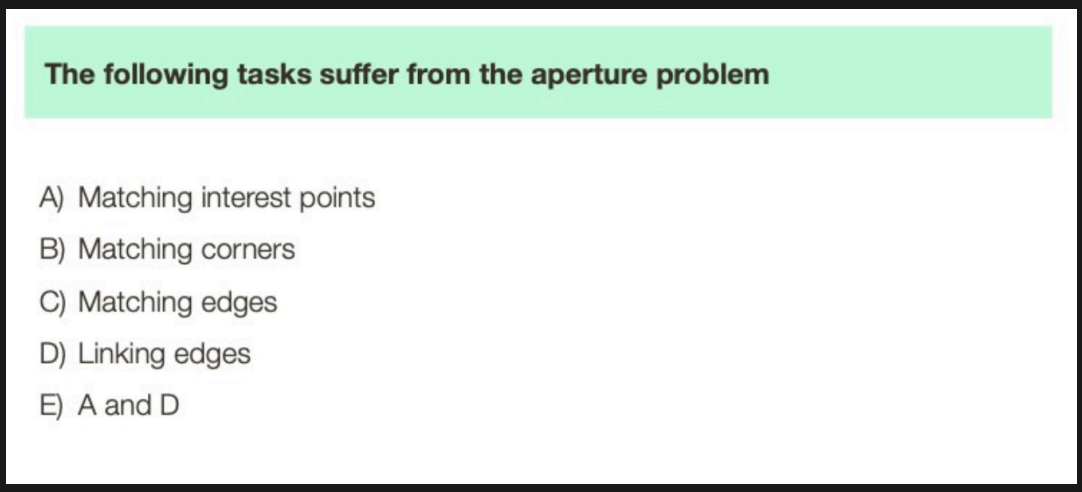
- Laplacian FILTER is a high pass filter
- when you combine the Gaussian smoothing with the Laplacian operator, you obtain a filter that passes a certain band of spatial frequencies while attenuating both low and high frequencies

Quiz 3

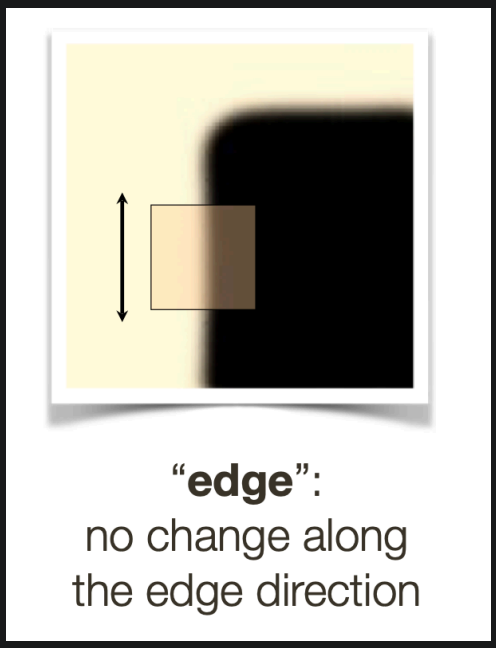
- question 1



  - A) True - this is why we blur
  - B) False - this is not anything to do with blurring, it's a technique you have to do
  - C) False - blurring doesn't help this
    - think of the window experiment, blurring does not fix this at all
  - so A is correct
- question 2

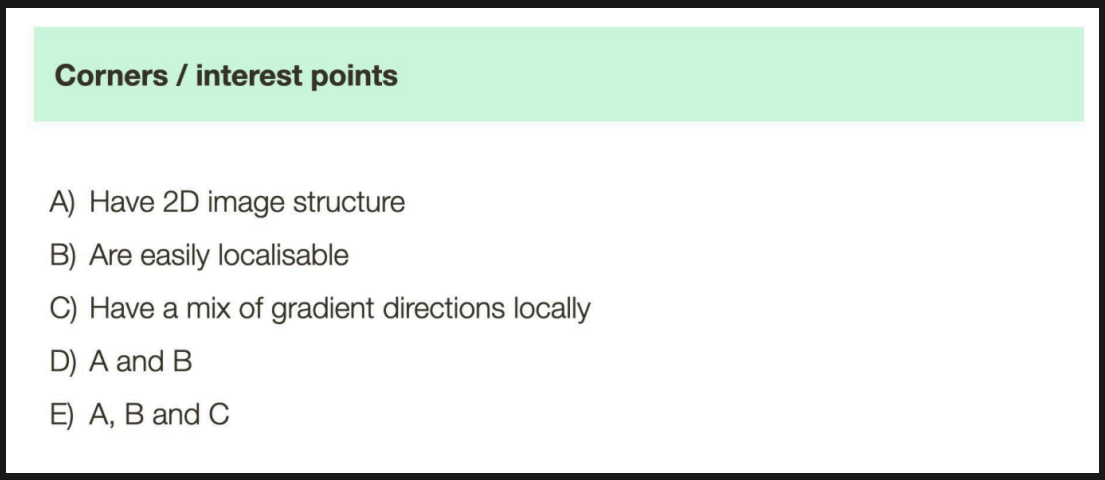


  - aperture problem: when things are hard to localize - if you slide a window up and down an vertical edge, it's hard to tell where exactly in the edge you are - because it all looks the same



- corners can be localized very reliably - so aperture problem is NOT an issue
- so it's matching edges that's the problem
- so the answer is C

- question 3



- A) True - 2D structure means they can be localized in both direction (again, window thought experiment)
- B) True - same as above
- C) True - corners are when 2 edges meet (so not parallel) - thus they must have a mix of gradients
- so the answer is E

- question 4

Given an image $I(x, y)$ a centred x derivative is given by

A) $I(x+1, y) - I(x, y)$
B) $I(x, y+1) - I(x, y-1)$
C) $I(x+1, y+1) - I(x-1, y-1)$
D) $I(x, y) - I(x-1, y)$
E) $I(x+1, y) - I(x-1, y)$

- they ones involving current and immediately adjacent pixels (i.e $x \text{ and } x+1$) will suffer from shift
- so the answer is E
  - C is also centered but it's in both directions - we just want the x direction

- question 5

Filter banks for texture classification might include

A) Laplacian of Gaussian filters
B) Bilateral filters
C) Oriented Edge Filters
D) A and C
E) A, B and C

- A) True - texture filter banks often include Laplacian of Gaussian (LoG) filters because LoG filters are effective at capturing and enhancing the details and fine structures present in textures
- B) False - this blurs, doesn't do anything interesting
- C) True - this is the core, combinations of edges is usually how we detect textures
- so the answer is D

- question 6

The following could be applications of texture synthesis

A) Distinguishing between natural and man-made materials
B) Outpainting to increase the size of an image
C) Interpolating between the samples of a Nyquist sampled signal
D) A, B and C
E) B and C

- A) False - that's analysis, we're not creating anything here
- B) True - we are creating more of a texture
- C) False - I believe the key here is that Nyquist sampled signal are do not require interpolating (they can be reconstructed perfectly without loss)
  - ChatGPT also agrees that this is False but argues that it's because "This refers to signal processing, where interpolation is used to estimate new data points within the range of a discrete set of known data points. Texture synthesis does not directly relate to signal interpolation; it's more about spatial pattern extension in images"
  - not sure if I 100% agree with this as in class it was said that images can be seen as signals that can be sampled - but to satisfy Nyquist you need to sample once per pixel (hence you just have the entire picture)

Quiz 4

- question 1

Matching patches via correlation is invariant to which transforms?

A) Rotation
B) Translation
C) Scaling
D) Skew
E) B and C

- (I'm using a lot of examples from the face-template matching portion of the course)
- A) False - recall to template matching, if your template is a different orientation than your picture, it won't match, thus it is not invariant to rotation
- B) True - if you're simply shifting the face in the picture, your template would still pick it up
- C) False - if the face is too big or too small compared to the template, it won't be detected → this is why we do it with a pyramid
- D) False - again, it's not invariant to rotation

- question 2

Lowe's SIFT determines keypoint locations and calculates an associated keypoint descriptor. Which of the following statements about **keypoint location** is false?

A) Keypoint location is robust to changes in translation, rotation, scale, 3D pose and illumination

B) Keypoint location includes the image coordinates, (X,Y), at which the keypoint was detected

C) Keypoint location includes the spatial scale, σ, at which the keypoint was detected

D) Keypoint location includes the dominant orientation of local image gradients where the keypoint was detected

E) Keypoint location includes an estimate of the principal axes of affine deformation

  - (this is basically asking what do we need to know about the keypoint location to calculate the descriptor)
  - A) True - this is why SIFT is widely used
    - only weird one was 3D pose, apparently by using the gradient and and descriptors - you're able to be insensitive to small 3D pose change
  - B) True - SIFT algorithm provides the (X, Y) coordinates where each keypoint is detected in the image
  - C) True - SIFT detects keypoints across multiple scales, and each keypoint is associated with the scale (σ) at which it was detected (Gaussian blur level)
  - D) True - at each keypoint, SIFT computes the gradient magnitudes and orientations in its neighborhood and assigns a dominant orientation based on these gradients
    - allows for rotational invariance
  - E) False - nothing about SIFT says anything about axes of affine deformation
  - So the answer is E

- question 3

A 2D transformation is represented by following the matrix $\begin{bmatrix}1 & 2 & 5\\3 & 4 & 6\\0 & 0 & 1\end{bmatrix}$ **Which of the following are true about this transformation?**

A) Parallel lines are preserved

B) It is a pure rotation

C) It is a pure translation

D) It is an affine transformation

E) A and D

  - (we can see that we have `[0,0,1]` on the last row - this means that it's an affine transformation, and we know that affine transformations preserve parallel lines)
  - answer is E)

- question 4

Quiz 4, Question 4

Recall Hartley and Zisserman's table for RANSAC (p = 0.99)

Sample size n	Proportion of outliers						
	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177

The task is to fit a circle to a set of points in the plane using RANSAC. Assume that 40% of the points are outliers. How many random samples are needed to detect the circle with 99% probability?

A) 11 B) 19 C) 34 D) 57 E) Can't tell

- you actually can do the math here, but it's not necessary
- the point is that you need to know how many points you need to fit a circle → it's 3, you basically extrapolate from a triangle; so $n = 3$
- now we know our outlier percentage is 40%, we can look up $n = 3, \text{prop of outlier} = 40\%$ in the table and that gives us 19 samples
- so answer is B

- question 5

Stereo algorithms involve the following

A) Using LIDAR to find depth
B) 1D search along epipolar lines
C) 2D search in the epipolar plane
D) Minimizing disparity between images
E) B, C and D

- A) False - we use stereo cameras (not sure what LIDAR is - we didn't learn it)
- B) True - since points appear along a epipolar line in corresponding image, it's a 1D linear search
- C) False - opposite of the above
- D) False - disparity tells us information about the depth of the image, we're not trying to minimize it

- question 6

In a rectified stereo configuration, points which are close to the camera have

A) Zero disparity
B) Small disparity
C) Large disparity
D) Infinite disparity

- this was something kinda mentioned in class, but far away objects have small disparity while objects close to the cameras have larger disparity
- so answer is C)
- (experiment: hold a pencil close to your face and shut one of your eyes, do the same for the other one; repeat this for different distances)

Quiz 5

- question 1

Which of the following are true about the optical flow constraint equation:

$$I_x u + I_y v + I_t = 0$$

A) It can be derived from a brightness constancy assumption
B) It is a solution to the aperture problem
C) Solution for (u,v) from a video sequence is underconstrained
D) A and B
E) A and C

- A) True - we need the brightness constancy assumption to even get this equation
- B) False - optical flow (tracking brightness) itself succumb to the aperture problem
- C) True - we have 1 equation and 2 unknown, this is under-constrained
- answer is E

- question 2

The Lucas-Kanade algorithm assumes that

A) Nearby pixels have similar flow vectors
B) The scene is static and rigid
C) Motion is perpendicular to image edges
D) A and B
E) A, B and C

    - LK algorithm simply assumes that pixels within the same neighbourhood are moving in the same direction (i.e have similar optical flow vectors)
    - so the answer is just A
    - in fact, for B) - the LK algorithm would actually perform very poorly because it prefers patches with interesting textures and features
- question 3

In multiview matching, RANSAC might be used

A) To remove incorrect feature matches
B) To jointly solve for camera and structure parameters
C) To find a set of correctly matching images
D) A and C
E) A, B and C

    - A) True - RANSAC is used for outlier removal
    - B) False - RANSAC itself doesn't solve for camera and structure parameter - we need SVD or LSE
    - C) True - in multi-view matching we can do RANSAC between pairs of images to establish which one is the best match??
    - answer is D
- question 4

Global alignment / bundle adjustment

A) Corrects for accumulated error in pairwise matching
B) Uses non-linear least squares optimization
C) Helps to close gaps in panoramic stitching
D) A and B
E) A, B and C

    - A) True - this is the reason for global alignment
    - B) True - you need some complex algorithm to solve
        - Levenberg-Marquardt is the most popular NLLS algorithm (not that he mentioned this in class)
    - C) True - it does help close gaps and make panoramic pictures look better
    - answer is E
- question 5

Visual word histograms / bag of words are useful for

A) Object instance recognition
B) Object category recognition
C) Performing the k-means algorithm
D) A and B
E) A, B and C

- we've seen in class the BOW is used to identify something like "Car" or "Dog" but not "Porsche" and "Golden Retriever" - this is object category
  - this is why we needed to pick out features and assign histograms, so that objects with similar feature frequency have similar histograms
- so the answer is B) only

- question 6

Suppose we have a codebook of 1,000 SIFT visual words (recall that SIFT is a 128-dimensional feature). Now we are given a new image and we extract 2,000 SIFT descriptors from it. What is the dimensionality of a bag of words descriptor?

A) 128
B) 1,000
C) 2,000
D) 128,000
E) It is not possible to construct a bag of words because there are more SIFT descriptors in the image than visual words

- code book here means our dictionary → so we have 1000 words in our dictionary
- this means that no matter what the new image is, it'll get assigned to a histogram with 1000 words (bins) that aligns with our dictionary, so the answer is B
  - basically the image gets turned into a vector with 1000 columns, each column corresponding to one of our SIFT visual word

Quiz 6

- question 1

The decision boundary of a nearest neighbour classifier is

A) Linear
B) Quadratic
C) Undefined
D) Piecewise Linear
E) Maximum margin

- from class, it's kinda jagged → it's piecewise linear

- question 2

The following are properties of the RELU function

A) It is a linear function
B) It is an invertible function
C) It is equivalent to f(x) = max(0, x)
D) A and C
E) A, B and C

    - A) False - it is non-linear, that's the entire point
    - B) False (through process of elimination)
        - we know C is for sure True and A is for sure False
        - this can't be True because there's no option for B and C
        - invertible also means that "unique input maps to a unique output" (one-to-one) but you can see that $f(-1) = f(-2) = f(-\inf) = 0$ so it's not one-to-one (multiple output can map to 0)
    - C) True - this is the formula
    - answer is C
- question 3

You are building a linear classifier to classify 10x10 colour images in 5 classes. What is the size of the weight matrix W?

A) 10 x 5
B) 100 x 5
C) 30 x 15
D) 300 x 5
E) 300 x 15

    - (in my mind the number of rows is the number of classes we have, but here it's opposite but it's fine)
    - we have 5 rows because we have 5 classes
    - in linear classification, we also mention throwing away spatial information and just vectorizing all the pixels as 1 big vector, so now we have $10 \times 10 \times 3 = 300$ columns
    - answer should be D (flip row and column)
- question 4

Deep neural networks are typically trained using

A) Gauss-Newton iteration
B) Linear least squares
C) Batches of input data
D) 2D image gradients
E) All of the above

- A) False - didn't learn it, also not used for NN
- B) False - since ReLu is non-linear, we can't solve this using LSE
- C) True - SGD perform gradient descent on a batch of data for speed
- D) False - it doesn't use the image gradients, it rather uses the gradient of the Loss function of the NN

- question 5

Backpropagation makes use of

A) The chain rule for differentiation
B) Finite difference approximation of derivatives
C) Forward mode auto-differentiation
D) A, B
E) A, B and C

- A) True - this is the backbone of backprop
- B) False - not sure what this is and we didn't learn it
- C) False - it uses backward mode auto-diff
- answer is A