Module 2

Name:

---

(1) How would you write a nested for loop to iterate through all the pixels in an image? (10 pts)

1. for i in range(0, height * width):
2. for row in range(0, height):
   for pixel in range(0, width):
3. for col in range(0, width):
   for row in range(0, height):
4. for pixel in range(0, width):
   for row in range(0, height):
5. None of the above

Answer:

---

(2) What is the purpose of thresholding in image processing? (10 pts)

1. To search for values that exactly match black or white.
2. To select all pixels above a certain brightness value.
3. To convert an RGB image to greyscale.
4. To perform a filter on the image using a fixed value as the dividing point.
5. None of the above

Answer:

---

(3) The following pseudo code will detect what feature? (10 pts)

For each row in the image:
  For each pixel in the row:
    Subtract the current pixel's RGB values with the pixel to the left
    Set the current pixel to this value.

1. Colors
2. Edges
3. Noise
4. Circles
5. None of the above

Answer:

Answer:

---

(4) To remove noise in an image what type of filter should you implement? (10 pts)

1. Sharpening
2. Blur
3. Horizontal Edge Detecting
4. Circle Detecting
5. None of the above

Answer:

---

(5) Which of the following statements is true regarding interpolation in image processing? (10 pts)

1. It is a method of deleting pixels from an image when it is downsized.
2. It is a method of adding noise to an image.
3. It is a method of constructing new data points based on known data points to fill gaps in pixel data during image resizing.
4. It is a method of converting an image from color to black and white.
5. None of the above

Answer:

---

(6) Which type of image interpolation has the fastest execution time? (10 pts)

1. Bilinear
2. Distorted
3. Bicubic
4. Nearest Neighbor

Answer:

---

(7) How many neighboring pixels are referenced in bilinear interpolation to estimate the value of a new pixel?? (10 pts)

1. 1
2. 16
3. 4
4. 8

Answer:

---

(8) Which of the following is the correct algorithm for performing image convolution? (10 pts)

1. For each pixel in the image frame, multiply it with its corresponding kernel value, sum the products, and set the current pixel to this value.
2. For each pixel in the kernel, visit each pixel in the image, multiply their values, sum the products, and set the current pixel to this value.
3. For each row in the image, visit each pixel in the row, apply the kernel values to the neighboring pixels, and set the current pixel to this value.
4. For each row in the kernel, visit each pixel in the row, apply the kernel values to the neighboring pixels, and set the current pixel to this value.

Answer:

---

(9) Which one of the following kernels can be used to blur an image? Explain Why? (10 pts)

$$(A)\begin{bmatrix}1 & 0 & -1\\ 1 & 0 & -1\\ 1 & 0 & -1\end{bmatrix}\quad(B)\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix}\quad(C)\begin{bmatrix}1/9 & 1/9 & 1/9\\ 1/9 & 1/9 & 1/9\\ 1/9 & 1/9 & 1/9\end{bmatrix}\quad(D)\begin{bmatrix}0 & -1 & 0\\ -1 & 5 & -1\\ 0 & -1 & 0\end{bmatrix}$$

Answer:

|

|

|

---

(10) Fill in the missing comments for each OpenCV function in the Sobel pipeline, explaining the purpose of each step. The (10 pts - 1 pt each)

Comment = #

```
#
#
img = cv2.imread('Graphics/face_conv.png')


#
#
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)


#
#
kernelOne = np.array([[1, 2, 1], [2, 4, 2], [1, 2, 1]],dtype=np.float32) / 16


#
#
filterOneImage = cv2.filter2D(KernelOne, cv2.CV_64F, gray)


#
#
kernelX = np.array([[-1, 0, 1], [-2, 0, 2], [-1, 0, 1]], dtype=np.float64)


#
#
KernelY = np.array([[-1, -2, -1], [0, 0, 0], [1, 2, 1]], dtype=np.float64)


#
#
imageX = cv2.filter2D(filterOneImage, cv2.CV_64F, kernelX)

#
#
imageY = cv2.filter2D(filterOneImage, cv2.CV_64F, kernelY)


# Compute the magnitude of the gradients of imageX and imageY
mag = cv2.magnitude(imageX, imageY)


#
#
val, thresh = cv2.threshold(mag, 100, 200, cv2.THRESH_BINARY)

#
#
plt.imshow(thresh, cmap='gray')
```