Low-Level Vision (Artificial)

1. Below is shown a convolution mask, H. Calculate the result of convolving this mask with (a) image I_1, and (b) image I_2.

$$H = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \qquad I_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad I_2 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Rotate the mask: $H = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$

(a)

$$H * I_1 = \begin{bmatrix} (1\times 0)+(1\times 0)+(0\times 0)+(1\times 1) & (1\times 0)+(1\times 0)+(0\times 1)+(1\times 0) \\ (1\times 0)+(1\times 1)+(0\times 0)+(1\times 0) & (1\times 1)+(1\times 0)+(0\times 0)+(1\times 0) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$

(b)

$$H * I_2 = \begin{bmatrix} (1\times 0)+(1\times 0)+(0\times 1)+(1\times 1) & (1\times 0)+(1\times 0)+(0\times 1)+(1\times 0) \\ (1\times 1)+(1\times 1)+(0\times 0)+(1\times 1) & (1\times 1)+(1\times 0)+(0\times 1)+(1\times 0) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 3 & 1 \end{bmatrix}$$

Hence, image equals template (rotated mask) at lower left corner.

2. Calculate $H * I$, padding the image with zeros where necessary, when:

$$H = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \qquad I = \begin{bmatrix} 0.25 & 1 & 0.8 \\ 0.75 & 1 & 1 \\ 0 & 1 & 0.4 \end{bmatrix}$$

Rotate the mask: $H = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

$$H * I = \begin{bmatrix} 0 & 0.25 & 1 \\ 0 & 0.75 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

3. Calculate $h * h^T$, where $h = [1, 0.5, 0.1]$. Show that this is equal to $h^T \times h$. Hence, calculate $H * I$, where:

$$H = \begin{bmatrix} 1 & 0.5 & 0.1 \\ 0.5 & 0.25 & 0.05 \\ 0.1 & 0.05 & 0.01 \end{bmatrix}, \qquad I = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Let h be the mask and h^T be the image: $h^T = \begin{bmatrix} 1 \\ 0.5 \\ 0.1 \end{bmatrix}$. *Rotate the mask:* $h = \begin{bmatrix} 0.1 & 0.5 & 1 \end{bmatrix}$

$$h * h^T = \begin{bmatrix} 1\times 1 & 0.5\times 1 & 0.1\times 1 \\ 1\times 0.5 & 0.5\times 0.5 & 0.1\times 0.5 \\ 1\times 0.1 & 0.5\times 0.1 & 0.1\times 0.1 \end{bmatrix} = \begin{bmatrix} 1 & 0.5 & 0.1 \\ 0.5 & 0.25 & 0.05 \\ 0.1 & 0.05 & 0.01 \end{bmatrix}$$

$$h^T \times h = \begin{bmatrix} 1 \\ 0.5 \\ 0.1 \end{bmatrix} \begin{bmatrix} 1 & 0.5 & 0.1 \end{bmatrix} = \begin{bmatrix} 1 & 0.5 & 0.1 \\ 0.5 & 0.25 & 0.05 \\ 0.1 & 0.05 & 0.01 \end{bmatrix}$$

Hence, $h * h^T = h^T \times h$ (more generally, the convolution of a row vector with a column vector can be written as matrix multiplication).

$I * H = I * (h * h^T) = (I * h) * h^T$

$$(I * h) = \begin{bmatrix} 1.5 & 1.6 & 0.6 \\ 1.5 & 1.6 & 0.6 \\ 1.5 & 1.6 & 0.6 \end{bmatrix}$$

$$(I * h) * h^T = \begin{bmatrix} 2.25 & 2.4 & 0.9 \\ 2.4 & 2.56 & 0.96 \\ 0.9 & 0.96 & 0.36 \end{bmatrix}$$

This is the same as $I * H$ (confirm using matlab).

4. List the categories of image features that can produce intensity-level discontinuities in an image.

- *Depth discontinuities - due to surfaces at different distances*
- *Orientation discontinuities - due to changes in the orientation of a surface*
- *Reflectance discontinuities - due to change in surface material properties*
- *Illumination discontinuities - e.g. shadows*

5. Convolution masks can be used to provide a finite difference approximation to first and second order directional derivatives. Write down the masks that approximate the following directional derivatives: (a) $-\frac{\delta}{\delta x}$, (b) $-\frac{\delta}{\delta y}$, (c) $-\frac{\delta^2}{\delta x^2}$, (d) $-\frac{\delta^2}{\delta y^2}$, (e) $-\frac{\delta^2}{\delta x^2} - \frac{\delta^2}{\delta y^2}$.

(a) $-\frac{\delta}{\delta x} \approx \begin{bmatrix} -1 & 1 \end{bmatrix}$

(b) $-\frac{\delta}{\delta y} \approx \begin{bmatrix} -1 \\ 1 \end{bmatrix}$

(c) $-\frac{\delta^2}{\delta x^2} \approx \begin{bmatrix} -1 & 2 & -1 \end{bmatrix}$

(d) $-\frac{\delta^2}{\delta y^2} \approx \begin{bmatrix} -1 \\ 2 \\ -1 \end{bmatrix}$

(e) $-\frac{\delta^2}{\delta x^2} - \frac{\delta^2}{\delta y^2} \approx \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} or \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

6. Convolve the mask $\begin{bmatrix} -1 & 1 \end{bmatrix}$ with itself.

The "image" padded with zeros is $\begin{bmatrix} 0 & -1 & 1 & 0 \end{bmatrix}$ *and the rotated mask is* $\begin{bmatrix} 1 & -1 \end{bmatrix}$.
Hence, the result is:
$\begin{bmatrix} (0 \times 1 + -1 \times -1) & (-1 \times 1 + 1 \times -1) & (1 \times 1 + 0 \times -1) \end{bmatrix} = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$

i.e., $-\frac{\delta}{\delta x} * -\frac{\delta}{\delta x} = \frac{\delta^2}{\delta x^2}$

7. Write down a mathematical expression describing the effect of convolving an image I with a Laplacian mask (i.e. $L = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$). Hence, write down a mathematical expression describing the effect of convolving an image I with the following mask: $L' = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$

$I' = I * L \approx -\left(\frac{\delta^2 I}{\delta x^2} + \frac{\delta^2 I}{\delta y^2}\right)$
$I'' = I * L' \approx I - c\left(\frac{\delta^2 I}{\delta x^2} + \frac{\delta^2 I}{\delta y^2}\right)$

Note: $(h_1 * I) + (h_2 * I) = H * I$ *where* $H = h_1 + h_2$
Whereas: $h_1 * (h_2 * I) = (h_1 * h_2) * I = H * I$ *where* $H = h_1 * h_2$

8. For edge detection, a Laplacian mask is usually "combined" with a Gaussian mask to create a Laplacian of Gaussian (or LoG) mask. (a) How are these masks "combined"? (b) Why is this advantageous for edge detection? (c) What mathematical function is usually used to approximate a LoG mask?

(a) How are these masks "combined"?
Using convolution.

(b) Why is this advantageous for edge detection?

The Laplacian is sensitive to noise as well as other intensity-level discontinuities.

The Gaussian is a smoothing mask that suppresses noise.

The combination of the two produces a mask that is sensitive to intensity-level discontinuities that are image features rather than noise.

(c) What mathematical function is usually used to approximate a LoG mask?

A Difference of Gaussians (DoG) mask:

$$\frac{1}{2\pi\sigma_1^2}\exp\left(\frac{-(x^2+y^2)}{2\sigma_1^2}\right)-\frac{1}{2\pi\sigma_2^2}\exp\left(\frac{-(x^2+y^2)}{2\sigma_2^2}\right)$$

where $\sigma_2 > \sigma_1$

9. To perform multiscale feature analysis, it would be possible to either (1) keep the image size fixed and vary the size of the mask, or (2) keep the mask size fixed and vary the size of the image. (a) Why is the latter preferred? (b) Give and explicit example of the advantage of method (2) assuming that we have a 100 by 100 pixel image and a 3 by 3 pixel mask and we want to detect features at this scale and at double this scale.

(a) It is computationally cheaper.

(b) Convolving a $m \times m$ *pixel mask with an* $n \times n$ *pixel image requires* $m^2 n^2$ *multiplication operations.*

For both methods we need to convolve a 100×100 *pixel image with a* 3×3 *pixel mask, which requires* $100^2\times3^2 = 90000$ *multiplications. In addition:*

- *For method (1) we need to convolve a* 100×100 *pixel image with a* 6×6 *pixel mask, which requires* $100^2\times6^2 = 360000$ *multiplications.*
- *For method (2) we need to convolve a* 50×50 *pixel image with a* 3×3 *pixel mask, which requires* $50^2\times3^2 = 22500$ *multiplications. (i.e.,* 2^4 *fewer multiplications than above.)*

10. What is aliasing and how is this avoided when down-sampling images to create an image pyramid?

Aliasing refers to the distortion, or misrepresentation, that can occur due to the sampling of an image.

To avoid aliasing images are smoothed (by convolving with a Gaussian mask) prior to down-sampling.

11. Briefly describe what is meant by (a) a Gaussian image pyramid, and (b) a Laplacian image pyramid.

(a) a Gaussian image pyramid is a multiscale representation of a single image at different resolutions obtained by iteratively convolving an image with a Gaussian filter and down-sampling.

(b) a Laplacian image pyramid is a multiscale representation of a single image that highlights intensity discontinuities at multiple scales. It is obtained by iteratively convolving an image with a Gaussian filter, subtracting the smoothed image from the previous one, and down-sampling.