INUSTRCTIONS:

This assignment is worth 2% of your grade. Upload a single **pdf** onto LumiNUS by **Tuesday Sept. 7, 2021, 23h59**. Late assignments are accepted (with a 0.3% deduction per day) until the solution release on Sept. 10. Please post any questions you have directly on the Forum. Solutions will be discussed in tutorial on Sept. 10 (9-11h) and Sept. 14 (15-17h).

**Lecture 1**

1. Suppose a flat area with center at $(x_0, y_0)$ is illuminated by a light source with the following intensity distribution

$$i(x, y) = Ae^{-((x-x_0)^2+(y-y_o)^2)}$$

   Assume for simplicity that the reflectance of the area is constant and equal to 1.0, and let *A* = 255. If the intensity of the resulting image is quantized using *k* bits, and the eye can detect an abrupt change of six intensity levels between adjacent pixels, what is the lowest value of *k* that will not have visible false contouring?

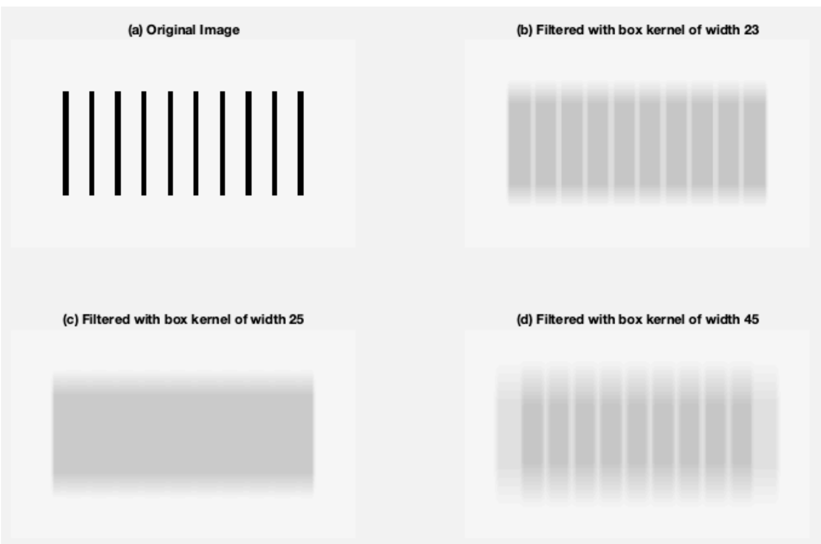
   Now consider *k=2*. Sketch the resulting image, considering (x, y) are continuous in this part.

**Lecture 2**

2. Suppose we have a 3-bit image (8 grey levels) of size 32x32 with the following intensity distribution. Fill in the following table to estimate the mapped intensity values.

| grey-value | number of pixels | proportion | Cumulative proportion | Mapped intensity (w/out rounding) | Mapped intensity (with rounding) |
|---|---|---|---|---|---|
| 0 | 164 | 0.160 | 0.160 | 1.12 | 1 |
| 1 | 230 | | | | |
| 2 | 210 | | | | |
| 3 | 156 | | | | |
| 4 | 114 | 0.111 | 0.854 | 5.97 | 5 |
| 5 | 72 | | | | |
| 6 | 60 | | | | |
| 7 | 18 | | 1.000 | | 7 |

3. You have an image of vertical bars 5 pixels wide, 100 pixels high and separated by 20 pixels. This image gets blurred using a square box kernel of sizes 23, 25, and 45 elements respectively for the results in (b), (c) and (d). Note how the bars have merged together in image (c), even though the blur kernel used is smaller than the kernel used to produce (d). Explain why.



4. A *separable* filter kernel is a matrix that can be expressed as the outer product of 2 vectors. Conveniently, the product of a column vector and a row vector is the same as the 2-D convolution of the vectors applied as kernels. For example, a separable kernel $\boldsymbol{w}$ can be decomposed vectors $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$, where $\boldsymbol{w} = \boldsymbol{v_1}\boldsymbol{v}_2^T = \boldsymbol{v_1} * \boldsymbol{v}_2^T$.

   Now suppose we are given a kernel $\boldsymbol{w}$ of size $m \times n$ which can be further decomposed into kernels $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$. Answer the following questions.

   (a) What are the dimensions of the kernels $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$?
   (b) For input image $\boldsymbol{I}$, prove that $\boldsymbol{w} * \boldsymbol{I} = \boldsymbol{v}_2^T * (\boldsymbol{v_1} * \boldsymbol{I})$.
   (c) If input image $\boldsymbol{I}$ of size $M \times N$, compare the computational complexity of filtering with kernel $\boldsymbol{w}$ instead of instead of its separable component kernels $\boldsymbol{v_1}$ and $\boldsymbol{v_2}$. How does this compare for a kernel of 3×3? 11×11? 25×25 (using division to compare)?
   (d) Are the four kernels used as gradient operators on slide 12 separable? What about the the 2D Lapacian kernel on slide 25 or its extended version on slide 26? If so, write the decomposed kernels.

LECTURE 3

5. Below you find an image of a photolithography mask used in wire-bonding for making connections for integrated circuits on printed circuit boards.

(a) Design 3x3 kernels to detect 1-pixel wide horizontal, vertical and +/-45 degree lines on the mask. Assume that angles are considered in counter-clockwise direction and that a vertical line is 0 degrees.

(b) Design a set of 3x3 kernels that can be used to detect one-pixel breaks in the four types of lines that you outlined above. Assuming that the intensities of the lines are 1 and the background is 0.

LECTURE 4

6. (a) Find the normal representation of the line y = -5x + 2.

   (b) Show that the number of operations required to implement the Hough-transform algorithm as outlined in slides 14 (slope-intercept form) and 19 (normal form) are both linear in *n*, the number of edge points present in the image.