Institut für Mess- und Regelungstechnik
mit Maschinenlaboratorium
Karlsruher Institut für Technologie
(KIT)
Prof. Dr.-Ing. C. Stiller

Solutions for exam
“Machine Vision”
August 19, 2024

Question 1 (6 points)

Derive a filter mask to calculate the third order derivative $\frac{\partial^3 g}{(\partial u)^3}$ of a gray value function g.

Solution

There are several solutions possible. One way is to take a filter mask for $\frac{\partial g}{\partial u}$ and convolute it twice with itself. A possible filter mask is, e.g.

$$f = \frac{1}{2} \cdot \boxed{1 \mid 0 \mid -1}$$

Hence,

$$f * f = \frac{1}{4} \cdot \boxed{1 \mid 0 \mid -2 \mid 0 \mid 1}$$

and

$$f * f * f = \frac{1}{8} \cdot \boxed{1 \mid 0 \mid -3 \mid 0 \mid 3 \mid 0 \mid -1}$$

Question 2 (3+3 points)

Assume a pinhole camera with a focal length $f_{camera} = 5\ mm$ and an imager with a resolution that is descibed by its scaling factor to be $\alpha = \beta = 1000\ \frac{1}{mm}$. We are observing a plane that is orthogonal to the optical axis of the camera at depth $z = 1000\ mm$ and on which a band-bounded pattern with a cutoff frequency of $k_{cutoff} = 2\ \frac{1}{mm}$ is painted.

(a) Calculate the cutoff frequency of the pattern that is projected to the image plane.

(b) Check whether the Shannon-Nyquist sampling theorem is met.

Solution

(a) For the given pinhole camera system we obtain with the intercept theorem that a length L orthogonal to the optical axis is projected in a length l on the image plane by

$$\frac{L}{z} = \frac{l}{f_{camera}} \quad \Rightarrow \quad L \mapsto L \cdot \frac{f_{camera}}{z} \quad \Rightarrow \quad \frac{1}{L} \mapsto \frac{1}{L} \cdot \frac{z}{f_{camera}}$$

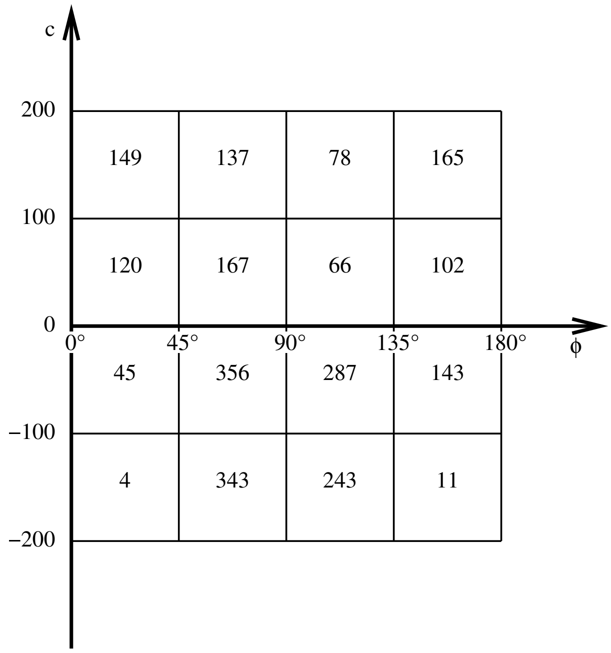
When we replace $\frac{1}{L}$ by k_{cutoff} we obtain

$$k_{cutoff,image\ plane} = k_{cutoff} \cdot \frac{z}{f_{camera}} = 2\ \frac{1}{mm} \cdot \frac{1000\ mm}{5\ mm} = 400\ \frac{1}{mm}$$

(b) We have to check whether the sampling frequency is at least twice the cutoff frequency. In our case, the sampling frequency is $\alpha = \beta = 1000\ \frac{1}{mm}$ while the cutoff frequency on the image plane is $400\ \frac{1}{mm}$. Since $1000 > 2 \cdot 400$ the sampling theorem is met.

Question 3 (2+2+2 points)

For a certain image the Hough transform was used to calculate the accumulator array in parameter space that is depicted in the figure below.



(a) Which line does the Hough transform calculate? Provide the angle ϕ of the unit normal vector and the offset of the line c.

(b) Name one advantage and one disadvantage of such a coarse partitioning of the parameter space.

Remark: The term coarse partitioning means that the parameter space is split only in a small number of large cells.

(c) Explain briefly why we can ignore angles ϕ in the interval between 180° and 360°.

Solution

(a) The maximum is found in cell in the third row and second column. The Hough transform returns the center position of that cell, i.e. $\phi = 67.5°$ and $c = -50$

(b) advantage: little memory requirements, disadvantage: little accuracy of estimated line

(c) for each line in normal form with an angle of the unit normal vector between 180° and 360° we can find an equivalent representation by rotating the unit normal vector by 180° and multiplying the offset with -1.

Question 4 (6 points)

Assume an RGB color image. We want to apply the white balance procedure described in the lecture to the image. Our reference area contains four pixels with the RGB values $(0.4, 0.4, 0.2), (0.5, 0.3, 0.4), (0.5, 0.4, 0.4), (0.6, 0.5, 0.2)$. Calculate the correction factors and apply them to the RGB values $(0.5, 0.3, 0.3)$, $(1.0, 1.0, 0.9)$.

Solution

We calculate the average red, green, and blue values as

$$\bar{R} = 0.5,\ \bar{G} = 0.4,\ \bar{B} = 0.3$$

Hence, the correction factors are

$$c_R = \frac{\bar{G}}{\bar{R}} = \frac{4}{5},\ c_B = \frac{\bar{G}}{\bar{B}} = \frac{4}{3}$$

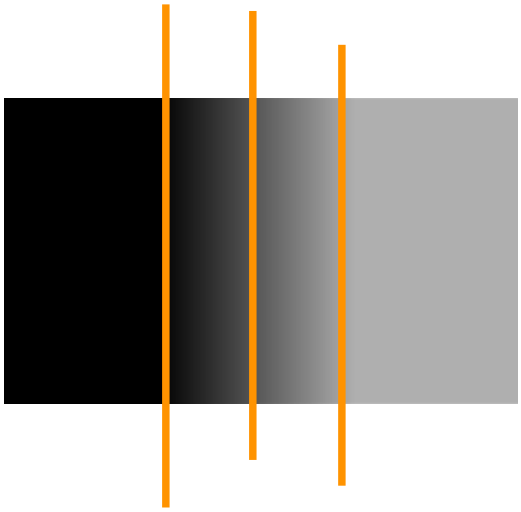
We apply the white balance to the RGB values and obtain

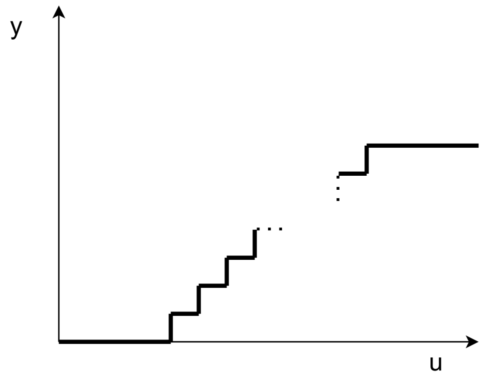
$$\begin{aligned}(0.5, 0.3, 0.3) &\mapsto (0.4, 0.3, 0.4)\\(1.0, 1.0, 0.9) &\mapsto (0.8, 1.0, 1.0)\end{aligned}$$

Mind that RGB values never exceed 1.

Question 5 (2+2+2+3 points)

Consider a gray value image as the one shown below on the left. It consists of a black area on the left, a light gray area on the right, and a gray value ramp between, i.e. the gray value increases linearly from left to right with a gray value difference of $d > 0$ between neighboring pixels. In the right image, the gray values (y) from one row are shown.





(a) Which partitioning do we obtain when we apply the connected components labeling algorithm and treat gray values as similar as long as their gray values do not differ more than $\frac{d}{2}$?

(b) Which partitioning do we obtain when we apply the connected components labeling algorithm and treat gray values as similar as long as their gray values do not differ more than $2d$?

(c) Which partitioning do we obtain when we apply k-means-clustering with $k = 2$? Assume that k-means-clustering finds the global optimum.

(d) For the case discussed in part (c) calculate the gray values of the two prototype gray values. Assume that the gray value of the dark area is 0, the gray value of the light gray area is 240, the width of the image is 300, the width of the black area is 100, and the width of the gray value ramp is 100. The height of the picture is 180.

Solution

(a) The black area, the light gray area, and each column of the gray value ramp will become individual segments.

(b) The whole image will become a single segment.

(c) The black area plus the dark half of the gray value ramp will become one segment, the light gray area together with the bright half of the gray value ramp will become the second segment.

(d) The average gray value of the dark segment is

$$\frac{100 \cdot 0 + 50 \cdot 60}{150} = \frac{3000}{150} = 20$$

The average gray value of the brighter segment is

$$\frac{50 \cdot 180 + 100 \cdot 240}{150} = \frac{9000 + 24000}{150} = \frac{33000}{150} = 220$$

Question 6 (4 points)

Given the camera matrix $\mathbf{A}$. Project point $(\mathbf{x}, \mathbf{y}, \mathbf{z}) = (-0.5, 1, 2)$ into image space.

$$\mathbf{A} = \begin{pmatrix} 600 & 0 & 300 \\ 0 & 400 & 300 \\ 0 & 0 & 1 \end{pmatrix}$$

Solution

$$\begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 600 & 0 & 300 \\ 0 & 400 & 300 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} -0.5 \\ 1 \\ 2 \end{pmatrix} = \begin{pmatrix} 150 \\ 500 \\ 1 \end{pmatrix}$$

Question 7 (8 points)

Assume a binary classification task with the three training examples

$$\vec{x}^{(1)} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \ d^{(1)} = -1$$
$$\vec{x}^{(2)} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, \ d^{(2)} = +1$$
$$\vec{x}^{(3)} = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \ d^{(3)} = +1$$

Calculate the hard margin support vector machine with dot product kernel. Provide the weight vector $\vec{w}$, the bias weight b and the margin ρ.

Remark: All three training examples are support vectors. You may use this assumption in your answer without proving it.

Solution

Since all three examples are support vectors it must hold

$$\begin{aligned}\langle \vec{x}^{(1)}, \vec{w} \rangle + b = d^{(1)} \\ \langle \vec{x}^{(2)}, \vec{w} \rangle + b = d^{(2)} \\ \langle \vec{x}^{(3)}, \vec{w} \rangle + b = d^{(3)}\end{aligned}$$

So we get the system of linear equations

$$\begin{aligned} w_1 + w_2 + b &= -1 \\ w_1 + 2w_2 + b &= +1 \\ 3w_1 + w_2 + b &= +1 \end{aligned}$$

Subtracting the first equation from the second and third one we obtain

$$\vec{w} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}, b = -4$$

We obtain the margin as

$$\rho = \frac{1}{||\vec{w}||_2} = \frac{1}{\sqrt{1^2 + 2^2}} = \frac{1}{5}\sqrt{5}$$

Question 8 (8 points)

Complete the Python function below to calculate an integral image. The input to the function is a gray value image with shape $[v,\, u]$ and the output of the function is expected to have the same shape.

Note: An integral image I_{int} based on image I is defined as:
$I_{int}[v{,}u] = \sum_{i=0}^{i \le v} \sum_{j=0}^{j \le u} I[i, j]$.

```
def integral_image(image:  np.ndarray):
  integral = np.zeros_like(image, dtype=np.float64)
```

Solution

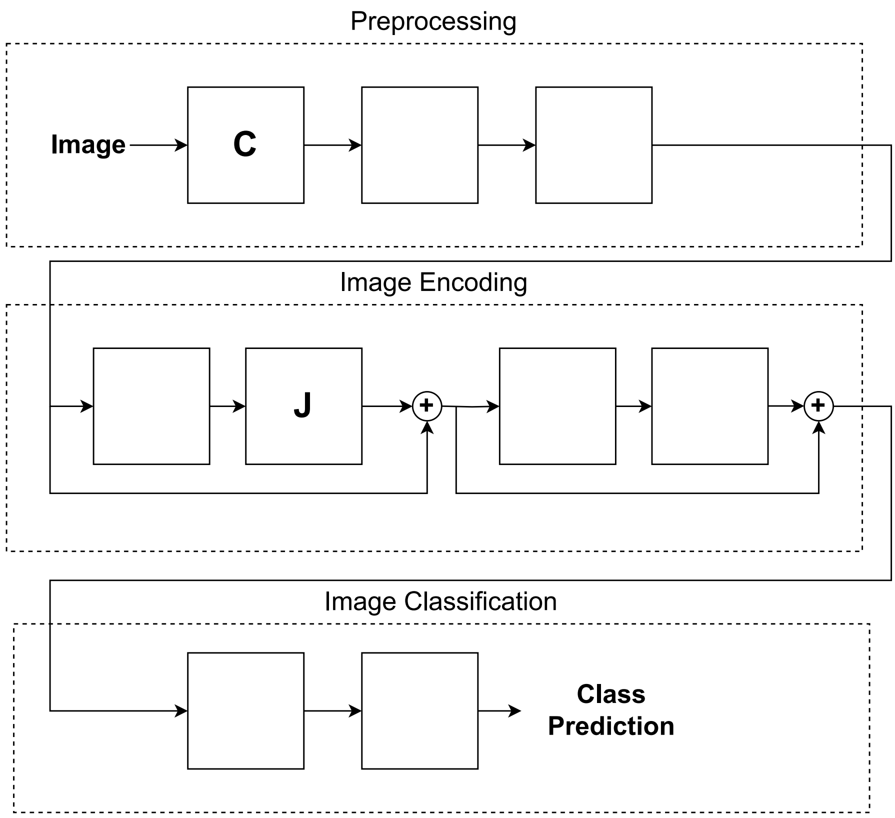
```
def integral_image(image: np.ndarray):

  integral = np.zeros_like(image, dtype=np.float64)

  for i in range(image.shape[0]):
    for j in range(image.shape[1]):
      integral[i, j] = image[i, j]
      if i > 0:
        integral[i, j] += integral[i - 1, j]
      if j > 0:
        integral[i, j] += integral[i, j - 1]
      if i > 0 and j > 0:
        integral[i, j] -= integral[i - 1, j - 1]
  return integral
```

Question 9 (7 points)

The graph below represents the major computation steps of a transformer-encoder-based architecture for image classification.
Use the provided computation steps and fill in their corresponding letters into the box.

Note: You may use some boxes multiple times, and it's NOT necessary to use all of them.

A: Positional Embedding	**B**: MLP	**C**: Patch Extraction
D: Decision-Tree	**E**: Argmax/Softmax	**F**: CNN
G: Transformer Encoder	**H**: LSTM	**I**: Patch Embedding
J: Multi-Head-Attention	**K**: Norm	



Solution

C, I, A,
K,J,K,B
B, E

Gesamtpunkte: 60