Institut für Mess- und Regelungstechnik
mit Maschinenlaboratorium
Universität Karlsruhe (TH)
Prof. Dr.-Ing. C. Stiller

Solutions for exam
„Machine Vision“
February 24$^{\text{th}}$, 2016

Contents

Question 1 (6 points)

Let $f : \mathbb{R} \to \mathbb{R}$ be a function and $g(x) = f(x+1)$. Prove the following property of the Fourier transform:

$$\hat{g}(k) = e^{2\pi ik}\hat{f}(k)$$

Note, that $\hat{f}$ denotes the Fourier transform of the function f

Solution

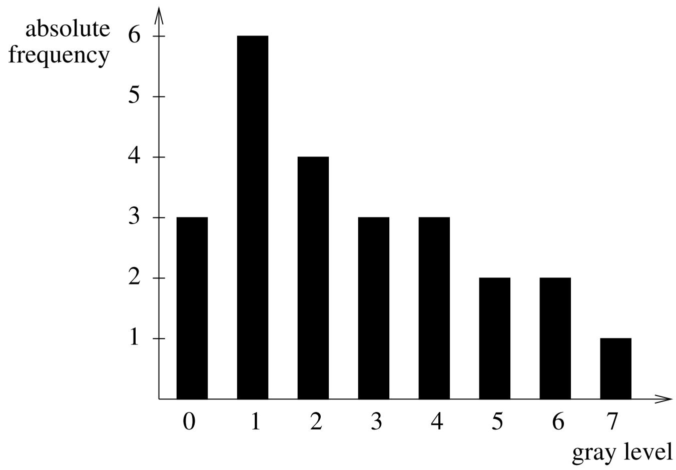
$$\hat{g}(k) = \int_{-\infty}^{\infty} g(x)e^{-2\pi ikx}dx = \int_{-\infty}^{\infty} f(x+1)e^{-2\pi ikx}dx = \int_{-\infty}^{\infty} f(y)e^{-2\pi ik(y-1)}dy$$
$$= e^{2\pi ik}\int_{-\infty}^{\infty} f(y)e^{-2\pi iky}dy = e^{2\pi ik}\hat{f}(k)$$

Question 2 (8 points)

The figure on the right depicts a small gray level image. The numbers represent the gray value of the respective pixel. Gray values range from 0 to 7. Plot the gray level histogram for this image. Do not forget to annotate the axes.

1	1	4	6	6	7
1	2	3	5	4	5
0	1	2	3	4	3
1	0	0	2	2	1

Solution



Question 3 (6 points)

The image on the right shows a gray level ramp, i.e. the gray value linearly increases from left to right. Assume, we apply the Harris corner detector. Would it classify the pixel in the center of the image as corner pixel, edge pixel, or as a pixel in a homogeneous area? Justify your answer briefly.

Solution

The partial derivatives $\frac{\partial g}{\partial v}$ are all zero while $\frac{\partial g}{\partial u} = c > 0$ for all pixels. Hence, the matrix S of the Harris detector would be

$$S = \begin{pmatrix} n \cdot c^2 & 0 \\ 0 & 0 \end{pmatrix}$$

with n the number of pixels which are considered. The Eigenvalues of S are $n \cdot c^2$ and 0. Hence, the pixel will be classified as edge pixel if $n \cdot c^2$ is greater than the threshold θ. Otherwise, it will be classified as homogeneous area.

Question 4 (8 points)

Which of the following properties are necessary, desirable, undesirable, or unsuitable for the error function $\rho(d)$ of an M-estimator? Assign one of the adjectives to each property and explain your answers briefly.

(a) $\rho(d) \geq 0$ for all values of d

(b) $\rho(d)$ takes its minimum for $d = 2$

(c) $\rho(d)$ is decreasing for large values of d

(d) $\rho(d)$ grows more slowly than d^2

Solution

(a) This property is necessary. $\rho(0)$ should be zero and $\rho(d)$ should be larger than zero for all other values of d

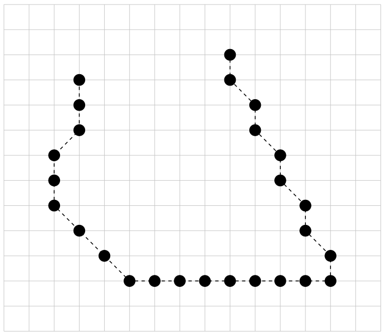
(b) This property is unsuitable since it would favor lines that have a distance of 2 to the set of points.

(c) This property is unsuitable/undesirable. $\rho(d)$ should not be decreasing for large values of d since such a function would favor large deviations from the fitted line over small deviations.

(d) This property is desirable since the influence of gross outliers should be reduced compared to the least-sum-of-squares estimator. However, if ρ would grow faster than d^2 the influence of gross outliers would even be increased.

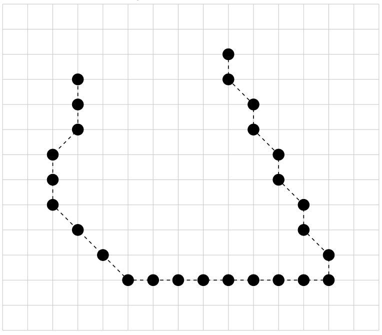
Question 5 (6 points)

The black spots in the figure on the right show a set of points on a rectangular grid. The points are ordered. The ordering is indicated by the dashed line, i.e. each point is connected to its predecessor and its successor.
Apply the Ramer-Douglas-Peucker algorithm to subdivide the set of points into subsets and create a polyline. The lateral deviation of points from the resulting polyline should be at most the width of one grid cell. Draw the initial polyline in the figure labeled with “initial polyline” and draw the intermediate polylines that are generated after each cycle of the algorithm in the figures which are labeled respectively. After how many cycles does the algorithm terminate?

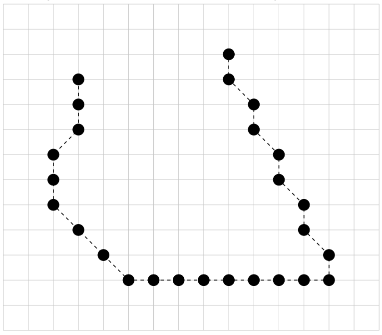


Remark: If the algorithm terminates earlier than after 8 cycles, leave the remaining figures blank.

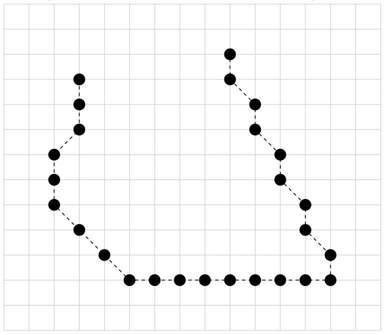
initial polyline:



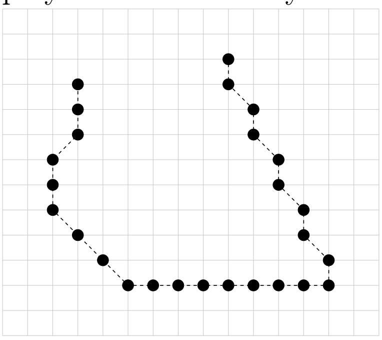
polyline after first cycle:



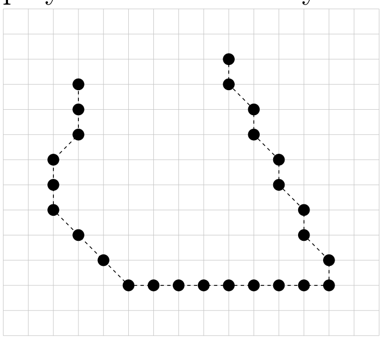
polyline after second cycle:



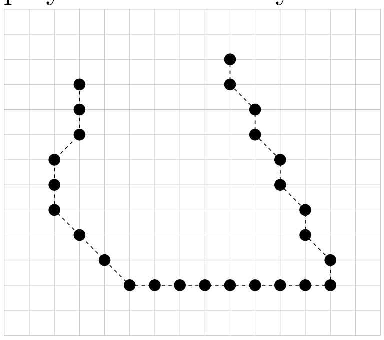
polyline after third cycle:



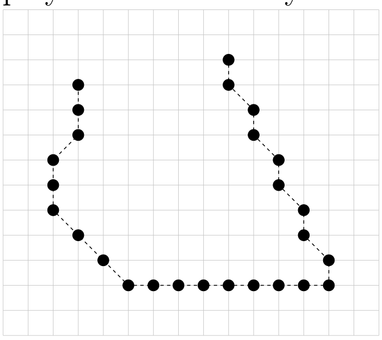
polyline after fourth cycle:



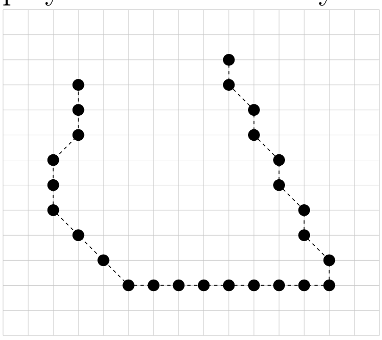
polyline after fifth cycle:



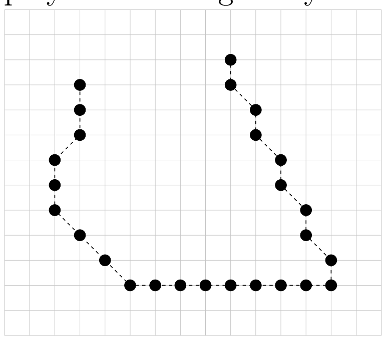
polyline after sixth cycle:



polyline after seventh cycle:

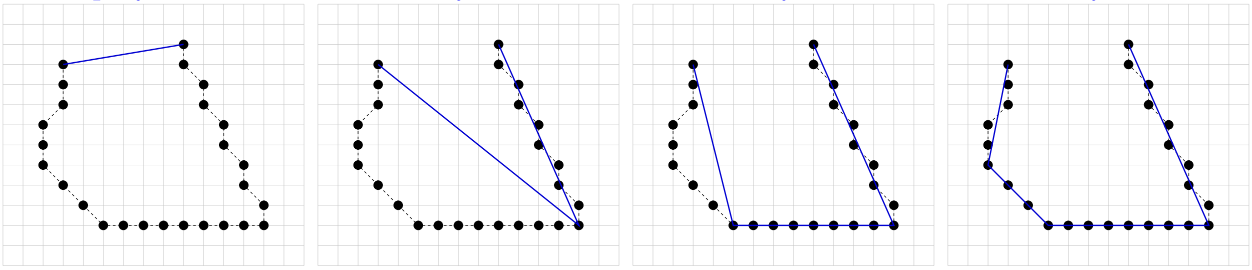


polyline after eighth cycle:



Solution

initial polyline: after 1st cycle: after 2nd cycle: after 3rd cycle:



The algorithm terminates after 3 cycles.

Question 6 (8 points)

Assume a level set evolution in which the boundary of the foreground segment should locally be expanded proportionally to the gray value of the pixel on the boundary, i.e. the boundary should be expanded quickly in bright areas and slowly in dark areas.

(a) Provide an evolution term $\frac{\partial \mathbf{x}}{\partial t}$ that implements this idea.

(b) Based on part (a) derive $\frac{\partial \phi}{\partial t}$. ϕ denotes the signed distance function.

Solution

(a) let I be the gray value of the image and $\alpha > 0$ a constant that controls the amount of expansion. Then, the idea is modeled by

$$\frac{\partial \mathbf{x}}{\partial t} = \alpha \cdot \frac{\nabla \phi}{||\nabla \phi||} I$$

(b) based on (a) we obtain

$$\frac{\partial \phi}{\partial t} = -\nabla \phi \cdot \frac{\partial \mathbf{x}}{\partial t} = -\alpha \frac{\nabla \phi \nabla \phi}{||\nabla \phi||} I = -\alpha ||\nabla \phi|| I$$

Question 7 (6 points)

Assume a pinhole camera with intrinsic parameters $\alpha' = 800$, $\beta' = 900$, $(u_0, v_0) = (500, 400)$ and skewing angle $\theta = 90°$. The camera images are 1000 pixels wide and 800 pixels high. Check which of the following points are mapped onto the camera image by the camera. The points are provided in camera coordinates. Justify your answers briefly.

(a) $(x, y, z) = (1, -8, 4)$

(b) $(x, y, z) = (1, -1, 10)$

(c) $(x, y, z) = (-1, 8, -10)$

Solution

Points with negative z-value are not mapped to the image because they are located behind the camera. For points with positive z we obtain their image position by $(u,v) = (800 \cdot \frac{x}{z} + 500, 900 \cdot \frac{y}{z} + 400)$. If $0 \leq u < 1000$ and $0 \leq v < 800$ the point is in the image, otherwise it is outside of the image. Hence,

(a) This point is in front of the camera, however, its v-coordinate is less than zero so that it is not mapped into the image

(b) This point is in front of the camera and it is mapped to the position $(u,v) = (580,310)$ which is within the image

(c) This point is behind the camera and therefore not mapped onto the image.

Question 8 (6 points)

Answer the following questions and justify your answers briefly.

(a) In the hard margin case of a support vector machine (SVM), what happens if we remove the last non-support vector from the set of training examples and retrain the SVM?

(b) In the soft margin case of a support vector machine, which training examples become support vectors?

(c) Assume we have trained a support vector machine on a set of 100 training examples. 20 training examples have become support vectors. How often do we have to retrain the support vector machine at least if we want to calculate the leave-one-out error?

Solution

(a) The support vector machine does not change since non-support vectors do not have any influence on it.

(b) Examples which are misclassified or which have a distance from the decision hyperplane that is less or equal to the margin become support vectors. All the other examples do not become support vectors.

(c) To calculate the leave-one-out-error we would have to check for each example what happens if we train the SVM on all other examples and evaluate its performance on the remaining example. Due to (a), the SVM does not change if we leave out a non-support vector, i.e. in 80 cases we do not need to retrain the SVM while for the 20 support vectors we have to retrain it. Hence, we need to retrain it 20 times.

Question 9 (6 points)

Create a MATLAB function that implements the Histogram intersection kernel. The function should take as input two vectors `x` and `y` of the same length which represent the two histograms. The signature of the function should be:

```
function value = histogram_intersection_kernel ( x, y )
```

Solution

```
function value = histogram_intersection_kernel ( x, y )
   value = 0;
   for i=1:length(x)
      value = value+min(x(i),y(i));
   end
end
```

Gesamtpunkte: 60