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

# Solutions for exam „Machine Vision“ February 27, 2018

Question 1 (1+3+4 points)

The figure below shows an array of grey level values organized in a single row of an image. Let $p_1 = 1.25$ and $p_2 = 2.5$ be two positions at subpixel accuracy.

(a) Interpolate the grey value at position $p_1$ using nearest neighbor interpolation.

(b) Interpolate the grey values at positions $p_1$ and $p_2$ using linear interpolation.

(c) If you used cubic interpolation to interpolate the grey value at $p_2$, would the resulting value be larger, smaller or equal to the linear interpolation result? Explain!

| pixel position | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| graylevel value | 83 | 75 | 75 | 83 | 95 |

## Solution

The nearest neighbor of 1.25 equals the value whose index is nearest to 1.25. In this case this is 83.

In order to interpolate $p_1$ we can use the formula in chapter 2 on slide 19.

$$\begin{aligned}
\bar{g}(x) &= g(\lfloor x \rfloor) + (x - \lfloor x \rfloor) \cdot (g(\lfloor x \rfloor + 1) - g(\lfloor x \rfloor)) \\
\bar{g}(1.25) &= g(1) + (1.25 - 1) \cdot (g(1+1) - g(1)) \\
\bar{g}(1.25) &= g(1) + (0.25) \cdot (g(2) - g(1)) \\
\bar{g}(1.25) &= 83 + (0.25) \cdot (75 - 83) \\
\bar{g}(1.25) &= 81
\end{aligned}$$

A linear interpolation for two equal values (as is the case in $p_2$) results in the same value. Therefore the value of $p_2$ is 75.

A cubic interpolation for $p_2$ would use the values $1 - 4$ in order to get a cubic equation. In this case these four values are symmetric which means that the resulting equation would consist of a parabola with min or max at 2.5. Since the value at index 1 is larger than the value at index 2 we can conclude that the parabola has a minimum at 2.5. Therefore the value would be smaller than 75.

Question 2 (8 points)

The figures below show two filter masks $F$ and $G$. The origin of both masks is in the left upper pixel. Create a new filter mask $H$ in such a way that $g * H = (g * F) * G$ for any graylevel image $g$.

F:

| | | |
|---|---|---|
| 1 | 0 | 2 |
| 0 | 1 | 1 |

G:

| |
|---|
| −1 |
| 1 |

## Solution

Since the convolution operation is associative, we can rewrite $(g * F) * G = g * (F * G)$. Hence, we obtain filter mask $H$ by convoluting $F$ and $G$. The resulting mask is depicted below.

**H:**

| | | |
|---|---|---|
| –1 | 0 | –2 |
| 1 | –1 | 1 |
| 0 | 1 | 1 |

Question 3 (2+2 points)

Assume, we want to fit a line to a set of $N \geq 2$ points in the two dimensional plane using the RANSAC algorithm. The set of points is composed out of $K$ points which are very close to the optimal plane (inliers) with $2 \leq K \leq N$ and $N - K$ points farther away from the optimal plane (outliers).

(a) How large is the probability that RANSAC picks two inliers in a single trial?

(b) How large is the probability that RANSAC picks two inliers in at least one of $L \geq 1$ trials?

### Solution

(a) RANSAC picks the first inlier with a probability of $\frac{K}{N}$ and the second inlier with probability $\frac{K-1}{N-1}$. Due to stochastic independence, the probability to pick two inliers in one trial is $\frac{K \cdot (K-1)}{N \cdot (N-1)}$

(b) Since the random selection procedure is stochastically independent between the different trials, the desired probability is equal to the complement of the probability to pick at least one outlier in each trial. With the result from part (a) we obtain

$$1 - (1 - \frac{K \cdot (K-1)}{N \cdot (N-1)})^L$$

Question 4 (8 points)

Implement a MATLAB function that takes a graylevel image, applies the dilation operator onto the image, and returns the dilated image. The dilation operator should consider the 8-neighborhoodship, i.e. it should consider the direct neighbors and the diagonal neighbors of a pixel. Your implementation should at least return correct results for all pixels of the image except the boundary pixels (topmost and bottommost row, leftmost and rightmost column). Do not use the MATLAB built-in function *imdilate*.

### Solution

```
function [ J ] = dilation ( I )
  [ rows cols ] = size(I);
  J = zeros (rows, cols);
  for v=2:rows-1
    for u=2:cols-1
      J(v,u) = max(max(I(v-1:v+1,u-1:u+1)));
    end
  end
end
```

Question 5 (6+2 points)

(a) Assume a pinhole camera and a straight line in front of the camera. The line is assumed to be orthogonal to the optical axis of the camera. Prove that the line is mapped to a straight line on the image plane by the pinhole camera.

(b) Explain briefly why this property typically does not hold for cameras with lenses.

## Solution

(a) All points of the line have a representation in the camera coordinate system as $\vec{x} = (x_0 + \tau d_x, y_0 + \tau d_y, z_0)^T$ with $z_0 > 0$ and $\tau \in \mathbb{R}$. $(x_0,y_0,z_0)^T$ is an arbitrary point on the line and $(d_x,d_y,0)^T$ is its direction vector. Due to the orthogonality constraint, the third entry of the direction vector is equal to 0. If $A$ denotes the matrix of intrinsic parameters of the camera we obtain as image point for $\vec{x}$

$$z_0 \cdot \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = A \cdot \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} + \tau \cdot A \cdot \begin{pmatrix} d_x \\ d_y \\ 0 \end{pmatrix}$$

Since the second part of the RHS does not contribute to the third row, we obtain

$$\begin{pmatrix} u \\ v \end{pmatrix} = A_{1:2,1:2} \cdot \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + z_0 \cdot A_{1:2,3} + \tau \cdot A_{1:2,1:2} \cdot \begin{pmatrix} d_x \\ d_y \end{pmatrix}$$

which is a line representation in the image plane ($A_{i:j,k:l}$ used MATLAB notation to denote a submatrix of $A$ with rows from $i$ to $j$ and columns from $k$ to $l$).

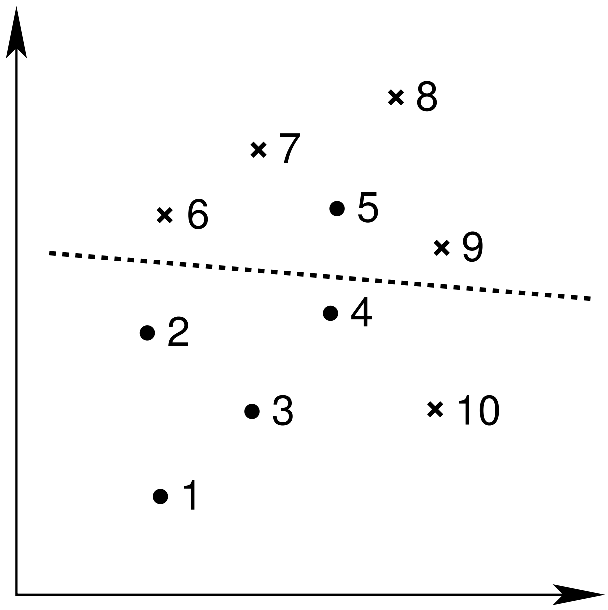
(b) Cameras with lenses typically suffer from lens distortions like radial distortion. Therefore, lines are corrupted by this kind of distortion.

Question 6 (2+4 points)

Assume a pattern classification problem in a 2-dimensional space as depicted in the figure below. There are five positive training examples No. 1–5 and five negative examples No. 6–10. Let us assume that we use AdaBoost together with linear classifiers to solve the classification task. The dashed line in the figure below shows the linear classifier that was selected by AdaBoost in the first cycle.

(a) For which patterns does AdaBoost increase its pattern weight $\gamma_i$ in the first cycle and for which patterns does it decrease $\gamma_i$?

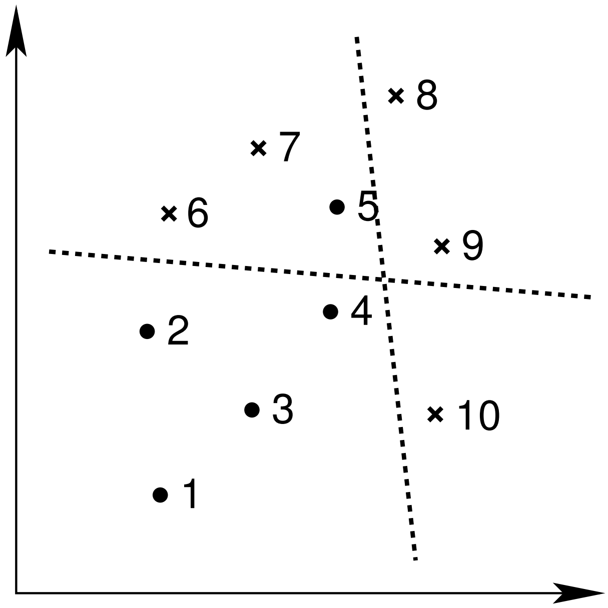
(b) Based on your results from part (a), which linear classifier will be selected by AdaBoost in the second cycle? Add a line to the figure that illustrates the decision boundary of that classifier. Justify your solution.




## Solution

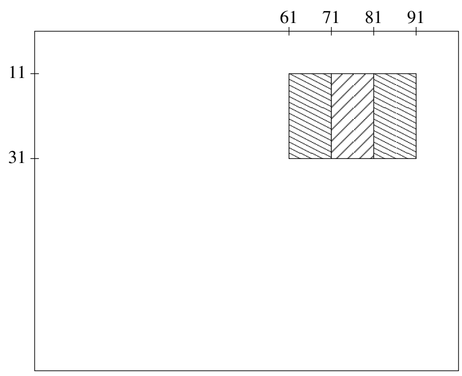
(a) AdaBoost increases the weights of misclassified pattern and decreases the weights of correctly classified patterns. Hence, the weights of pattern No. 5 and 10 are increased while the other weights are decreased.

(b) AdaBoost will select a linear classifier that passes between the patterns No. 4, 5, and 7 on the left and No. 8, 9, and 10 on the right as depicted below. In this examples, every linear classifier makes at least two mistakes. Since the patterns are weighted and the weights of patterns No. 5 and 10 are larger than those of the other patterns, AdaBoost prefers a classifier that makes two mistakes but is correct for patterns No. 5 and 10.




Question 7 (8 points)

Assume a Haar feature that is composed out of three rectangles as depicted in the figure on the right. The top left pixel of the leftmost rectangle has the image coordinate $(u,v) = (61,11)$, the top left pixel of the rectangle in the middle has the image coordinate $(u,v) = (71,11)$, and the top left pixel of the rightmost rectangle has the image coordinate $(u,v) = (81,11)$. All rectangles are 10 pixels wide and 20 pixels high. The Haar feature calculates the difference between the average gray value of the central area and the average of the joint left and right areas. Let $I(u,v)$ be the integral image. Provide a formula that calculates the feature value of the Haar feature given the integral image $I$. Simplify the formula as good as possible.




**Solution**

To calculate the sum of gray values of a rectangular area with top left pixel $(u,v)$, a width of $w$ and a height of $h$, we calculate $I(u+w-1,v+h-1)+I(u-1,v-1)-I(u+w-1,v-1)-I(u-1,v+h-1)$. Hence, we obtain the sum of gray values of the left, central, and right areas as

$$\begin{aligned} S_{left} &= I(70,30) + I(60,10) - I(70,10) - I(60,30) \\ S_{center} &= I(80,30) + I(70,10) - I(80,10) - I(70,30) \\ S_{right} &= I(90,30) + I(80,10) - I(90,10) - I(80,30) \end{aligned}$$

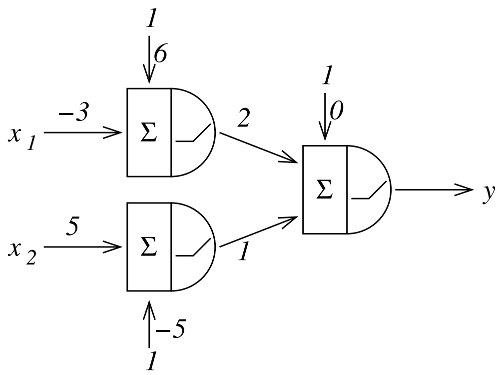
The area of each rectangle is $w \cdot h = 200$ pixels large. Hence, we obtain as value of the

Haar feature

$$\frac{1}{200}S_{center} - \frac{1}{400}(S_{left} + S_{right}) = \frac{1}{400}\Big( - I(60{,}10) + 3I(70{,}10) - 3I(80{,}10) + I(90{,}10) + I(60{,}30) - 3I(70{,}30) + 3I(80{,}30) - I(90{,}30)\Big)$$

Question 8 (3+4 points)

Assume a multi layer perceptron with a two-dimensional input $(x_1, x_2)$, two perceptrons in a single hidden layer, and one output perceptron. All perceptron use the ReLU activation function. The network structure is depicted in the figure on the right. Weights are shown as numbers next to the arrows. Bias weights are shown next to arrows that begin at the symbol "*1*".




(a) Calculate the output of the multi layer perceptron for the input vector $(x_1,x_2) = (1, -1)$

(b) Calculate for which part of the $(x_1,x_2)$-plane the network returns 0. You might illustrate this area in a diagram, if necessary.

## Solution

(a) The upper perceptron in the hidden layer calculates $\max\{1 \cdot (-3) + 6, 0\} = 3$. The lower perceptron in the hidden layer calculates $\max\{-1 \cdot 5 - 5, 0\} = 0$. The output perceptron calculates $y = \max\{3 \cdot 2 + 0 \cdot 1 + 0{,}0\} = 6$

(b) The output of the two hidden perceptrons is always nonnegative. Furthermore, the upper hidden perceptron returns a value different from 0 only if $-3 \cdot x_1 + 6 > 0$, i.e. for values of $x_1 < 2$. The lower hidden perceptron returns a value different from 0 only if $5 \cdot x_2 - 5 > 0$, i.e. for values $x_2 > 1$. The output perceptron calculates a weighted sum with positive weights of the output of the two hidden perceptrons. Hence, it returns 0 only if both hidden perceptrons return 0. This happens if $x_1 \geq 2$ and $x_2 \leq 1$ at the same time.

Question 9 (1+2 points)

(a) How many perceptrons does an autoencoder network have in its output layer compared to its input layer?

(b) Why does an autoencoder network typically have less perceptrons in each hidden layer than in the input layer?

## Solution

(a) Input and output layer of an autoencoder are equally large.

(b) An autoencoder should learn to compress the information that is contained in the input. If the hidden layer would be as large as the input layer or even larger there would not be the need to compress the information of the input.

Gesamtpunkte: 60