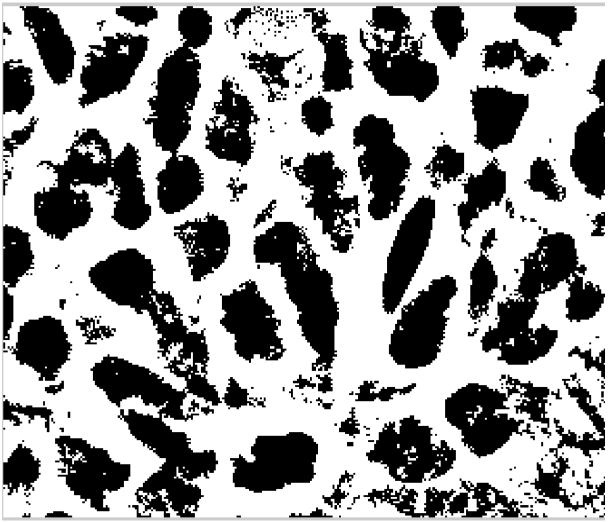
Question 1 . . . . $(\mathbf{\Sigma = 13})$

(a) What is thresholding? **(2 pts)**

(b) Name two use-cases of thresholding. **(2 pts)**

(c) Describe Otsu's thresholding algorithm. **(4 pts)**

(d) You have used a thresholding algorithm and obtained the following image:



i. We want to clean up the image using morphological operators. Explain how the morphological operators work. **(3 pts)**

ii. Which morphological operator(s) would you use on this image and why? Remember: Foreground are the black cells. **(2 pts)**

Name: Matr.-No.:

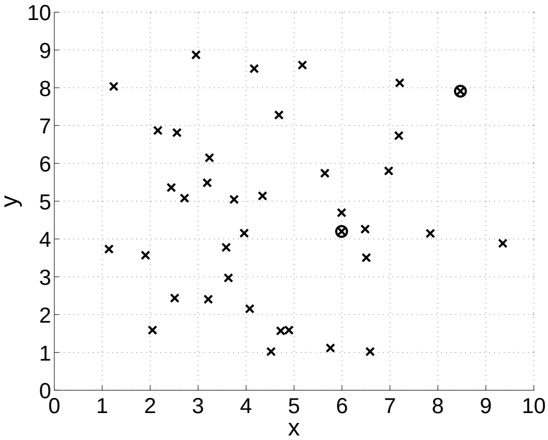
Question 2 .... $(\mathbf{\Sigma = 16})$

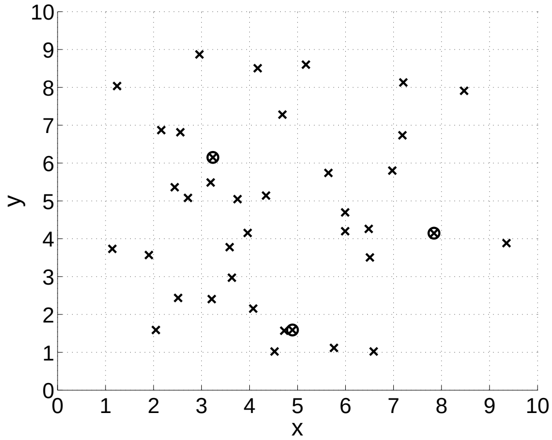
(a) List the steps of the k-Means algorithm. **(4 pts)**

(b) Properties of k-Means **(3 pts)**

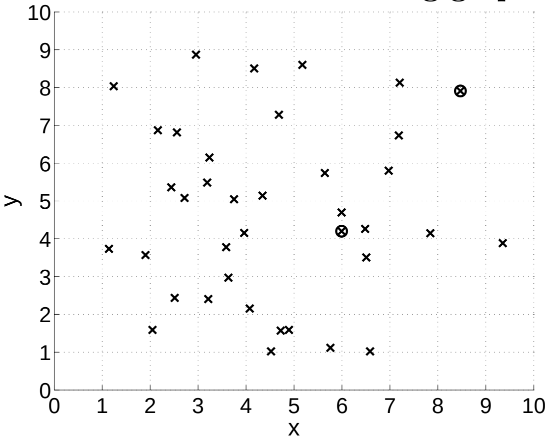
Will k-Means always converge?	◯ Yes	◯ No
Does k-Means always find the best solution with respect to its objective function?	◯ Yes	◯ No
Is the problem of finding the optimal solution NP-complete?	◯ Yes	◯ No

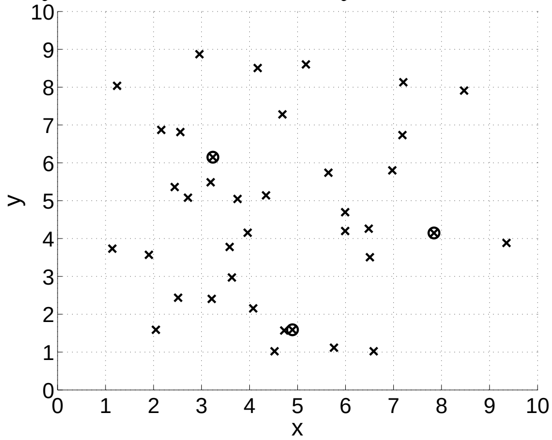
(c) Sketch the (approximate) cluster boundaries and their means k-Means would give for the following dataset for $k = 2$ (left) and $k = 3$ (right). The circled points are the initial means. **(4 pts)**





Please use the following graphs if you want to correct yourself.





(d) List one advantage and two disadvantages of k-Means. **(3 pts)**

(e) Describe in detail how k-Means can be used for image segmentation. **(2 pts)**

Question 3 .... **(Σ = 12)**

(a) Please fill in the following Matlab code fragment to complete the Hessian detector. (Pseudo-code is sufficient, as long it is unambiguously clear what is meant.) **(4 pts)**

```matlab
function [px, py] = computeHessian(filename, sigma, thresh)
  % ------------------------------------------
  % Preprocessing
  % ------------------------------------------
  I               = loadImage(filename);
  Ig              = gaussianfilter(I, sigma); % Gaussian filter
  [Ix, Iy]        = gaussderiv(I, sigma);     % first derivatives
  [Ixx, Ixy, Iyy] = gaussderiv2(I, sigma);    % second derivatives

  % ------------------------------------------
  % Compute Hessian score for each pixel
  % ------------------------------------------
  [height, width] = size(I);
  score = zeros(height, width);
  for y = 1:height
    for x = 1:width
      % Compute Hessian score for pixel I(y, x) and store it in score(y, x)
```

```matlab
    end
  end

  % ------------------------------------------
  % Postprocessing
  % ------------------------------------------
  % Extract the interest points from the computed score map.
  [py, px] = find(score > thresh)
end
```

(b) The above code is still not fully correct. There are 2 steps missing. Please point them out (A verbal explanation is sufficient). **(2 pts)**

(c) Details to the Hessian detector.

i. What are the differences between the Hessian and the Harris detector? Please briefly describe. **(2 pts)**

ii. Properties of Hessian keypoints **(2 pts)**

Is the Hessian detector scale-invariant? ◯ Yes ◯ No

Is the Hessian detector translation-invariant? ◯ Yes ◯ No

iii. How can Harris be extended to detect key points with arbitrary scale automaticaly. **(2 pts)**

Question 4 .... $\left(\mathbf{\Sigma = 15}\right)$

(a) The Adaboost *training* algorithm.

i. What is the input and what is the output of this algorithm? **(2 pts)**

ii. Briefly explain the steps of the Adaboost *training* algorithm (no formulas required). **(3 pts)**

(b) Which property has to be fulfilled by the weak classifiers? **(1 pt)**

(c) How is a test point classified? Give the equation. **(2 pts)**

(d) What are the weak classifiers that are used for Viola-Jones face detection? (You may sketch to support your answer.) **(2 pts)**

(e) Integral Images

i. What is an integral image? **(1 pt)**

ii. Why and how are integral images used for Viola-Jones face detection? (You may sketch to support your answer.) **(2 pts)**

(f) Briefly explain how cascading classifiers for detection works. **(2 pts)**

Question 5 .... $(\boldsymbol{\Sigma} = \mathbf{13})$

(a) Briefly explain the following concepts

i. Fundamental Matrix **(1 pt)**

ii. Epipolar plane **(1 pt)**

(b) Eight-point algorithm **(4 pts)**

i. Fill in the first row of the following matrix in order to complete the Eight-point algorithm. Assume that the point correspondence is called $(\mathbf{x}, \mathbf{y})$ where $\mathbf{x} = (x_1, x_2, 1)$ is located in the left image and $\mathbf{y} = (y_1, y_2, 1)$ in the right image. (**Hint**: Use the derivation of the algorithm).

$$\begin{bmatrix} \boxed{} & \boxed{} & \boxed{} & \boxed{} & \boxed{} & \boxed{} & \boxed{} & \boxed{} & \boxed{} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \end{bmatrix} \cdot \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

ii. How do we solve this equation? What exactly is the solution? **(1 pt)**

iii. Solving this equation directly usually leads to very inaccurate results in presence of noise. Please explain why. **(1 pt)**

iv. What can we do to about this issue in order to get more accurate results, and how? **(1 pt)**

(c) Rank constraints of the Fundamental Matrix.

i. What is the rank of the Fundamental matrix? Why? **(2 pts)**

ii. What would happen if F had full rank? **(1 pt)**

iii. How do we enforce the rank constraint when estimationg F? **(1 pt)**