3D Machine Vision

3D Data Processing

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Processing of Point Clouds and Depth Images

- Finding Nearest Neighbors
- Basic Statistics
- Linear Regression - Plane Fitting
- Principal Component Analysis - Linear Subspaces
- RANSAC - Random Sample Consensus
- Segmentation and Clustering
  - Split and Merge
  - Clustering/Partitioning - K-means
  - Local Clustering - Superpixels/-voxels

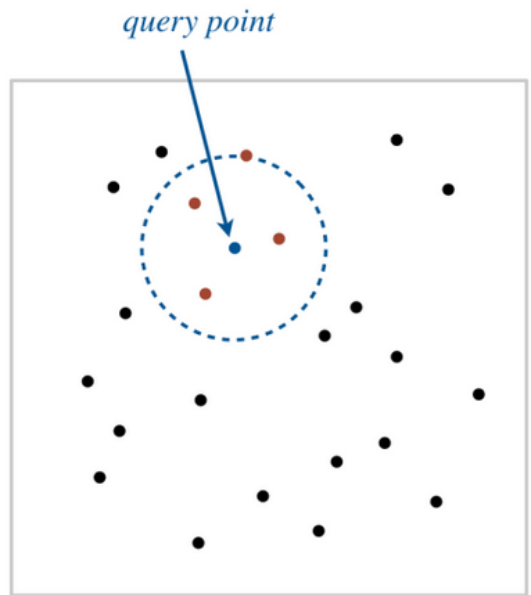
3D Data Processing

Finding K nearest neighbors using kD-tree

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Search Procedure

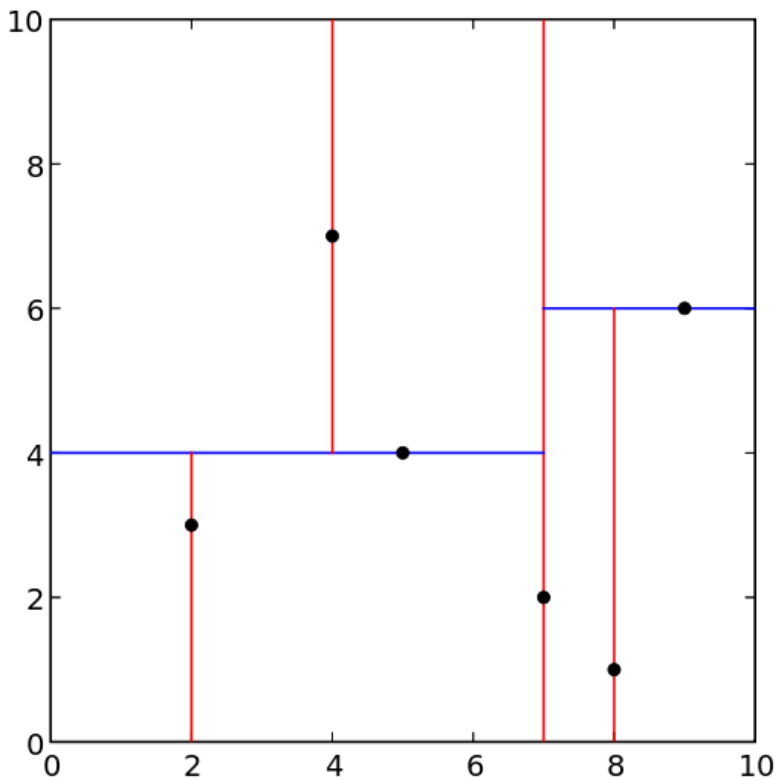
- Start at the root
- Traverse the tree to the section where the new point belongs
- Find the leaf; store it as the first element in the *Best* queue
- Traverse upward, and for each node:
  - Put it at the proper point of the *Best* queue
  - Check if there could be yet better points on the other side:
    - Fit a sphere around the point of the same radius as distance to last element in *Best* queue
    - See if that sphere goes over the splitting plane associated with the considered branchpoint
  - If there could be, go down again on the other side. Otherwise, go up another level

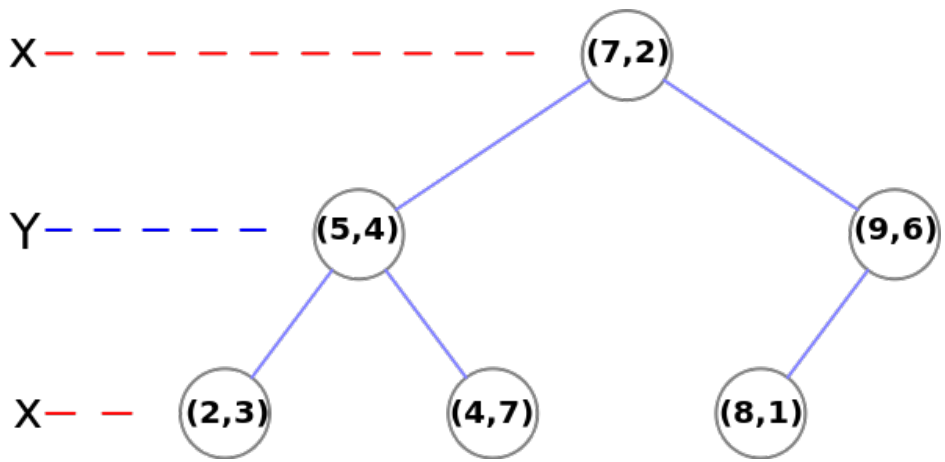


4 nearest neighbors

Finding K nearest neighbors using kD-tree

- Revisiting kD-tree example: Find 3 nearest neighbors of point $\mathbf{p} = (4, 3)$
- Use Euclidean distance $d(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_2 = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}$





- See matlab example for 3D point cloud data!

3D Data Processing

Basic Ideas

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Basic Problem

We want to efficiently find the main semantics and structure in 2D image & 3D point cloud data to make predictions about unseen data

Questions

- Are the X,Y,Z-coordinates of a point cloud and/or the attributes of the points like the color, normal etc. correlated?
- Can we find simple parametric models that compactly represent sets of 2D/3D data?
- Can we find subspaces (topological embeddings) that still represent most of the important information (structure) inherent in the data?

- See matlab example for 3D point cloud data and RGB-D image!

3D Data Processing

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Basic Ideas

Basic Problem

We want to efficiently find the main semantics and structure in 2D image & 3D point cloud data to make predictions about unseen data

Physical and Geometric Constraints

We have some certain expectation about unseen data because our world is structured and not completely random. Hence, we can imply some underlying probability density function that represents the certainty of the data

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Recap Random Variables - Expectation

The mean μ_x or expectation $\mathbb{E}\{x\}$ of 1D data is defined as:

$$\mu_x = \mathbb{E}\{x\} = \int_{-\infty}^{\infty} x p(x) dx \quad \text{bzw.} \quad \mu_X = \mathbb{E}\{X\} = \sum_{X=1}^{Q} X p(X) \,.$$

The expectation can be approximated by measuring N data points X_i and averaging measured values without knowledge of the pdf (empirical mean $\overline{\mu}$):

$$\mathbb{E}\{X\} \approx \overline{\mu}_X = \frac{1}{N} \sum_{i=1}^{N} X_i \,, \quad \mathbb{E}\{X\} = \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{N} X_i \,.$$

Recap Random Variables - Variance

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

The variance describes the deviation from the mean value μ_x:

$$\sigma_x^2 = \mathbb{E}\{(x-\mu_x)^2\} = \int_{-\infty}^{\infty}(x-\mu_x)^2 p(x)dx = \mathbb{E}\{x^2\} - \underbrace{\mathbb{E}\{x\}^2}_{\mu_x^2} \quad \text{bzw.}$$

$$\sigma_X^2 = \mathbb{E}\{(X-\mu_X)^2\} = \sum_{X=1}^{Q}(X-\mu_X)^2 p(X)\,.$$

Also, the variance can be approximated by measuring N times without knowledge of the pdf. This results in the so-called sample variance:

$$\mathbb{E}\{(X-\mu_X)^2\} \approx \overline{\sigma}_X^2 = \frac{1}{N-1}\sum_{i=1}^{N}\left(X_i - \overline{\mu}_X\right)^2.$$

Multivariate Statistics - Random Vectors

Random vectors $\mathbf{x} = (x_1, x_2, \ldots, x_N)^\top$ have a common pdf $p(\mathbf{x})$ with properties $p(\mathbf{x}) \geq 0$ and $\sum_{\mathbf{x}} p(\mathbf{x}) = 1$, where the individual random variables x_i are generally correlated and statistically dependent on each other.

$$\begin{aligned} \text{Statistical independence:} \quad p(\mathbf{x}) &= \prod_{n=1}^{N} p(x_n)\,. \\ \text{Uncorrelated random variables} \quad \mathbb{E}\{x_1 x_2\} &= \mathbb{E}\{x_1\}\mathbb{E}\{x_2\}\,. \\ \text{Orthogonal random variables:} \quad \mathbb{E}\{x_1 x_2\} &= 0\,. \end{aligned}$$

Statistical independence always implies uncorrelated data. However, the converse does not hold.

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Multivariate Statistics - Covariance Matrix

Common central moments of higher order offer the possibility of statistical dependencies between the random variables X_n of a vector $\mathbf{X}$ to be evaluated. Here, the most important tool is the covariance matrix $\boldsymbol{\Sigma}$.

$$\boldsymbol{\Sigma} = \mathbb{E}\{(\mathbf{X}-\boldsymbol{\mu})(\mathbf{X}-\boldsymbol{\mu})^\top\} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_d^2 \end{bmatrix}, \quad \boldsymbol{\mu} = \mathbb{E}\{\mathbf{X}\}.$$

This matrix is symmetric and positive semidefinite (all eigenvalues $\geq$ zero). The variances σ_i^2 are always positive, the covariances σ_{ij} can be positive or negative (correlated or anticorrelated). If the covariances are zero, then the variables are statistically independent and the covariance matrix is diagonal.

Multivariate Statistics - Covariance

Since the covariance matrix is symmetric, a coordinate system can always be found, i.e. a linear combination of the random variables, in which the covariance matrix is diagonal and thus the random variables are uncorrelated.

The covariance of two random variables X and Y indicates to what extent the random variables are related to each other.

$$\sigma_{XY} = \mathbb{E}\{(X - \mu_X)(Y - \mu_Y)^\top\} = \mathbb{E}\{XY\} - \mathbb{E}\{X\}\mathbb{E}\{Y\}\,.$$

The correlation coefficient is the covariance normalized by the variances:

$$\varrho_{XY} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y}\,, \quad \text{mit} \quad |\varrho_{XY}| \leq 1\,. \quad \sigma_{XY}^2 \leq \sigma_X^2 \sigma_Y^2 \quad \text{(Chauchy-Schwartz Inequality)}\,.$$

Uncorrelated random variables satisfy the following relationships:

$$\varrho_{XY} = 0 \;\Leftrightarrow\; \sigma_{XY} = 0 \;\Leftrightarrow\; \mathbb{E}\{XY\} = \mathbb{E}\{X\}\mathbb{E}\{Y\}\,.$$

Example: Correlations in an RGB color image

$\overline{\mu}_R = 182$, $\overline{\sigma}_R = 42$

$\overline{\mu}_G = 124$, $\overline{\sigma}_G = 46$

$\overline{\mu}_B = 99$, $\overline{\sigma}_B = 44$

3D Data Processing

Example: Correlations in an RGB color image

$\overline{\varrho}_{RG} = 0.93$

$\overline{\varrho}_{RR} = 1$

$\overline{\varrho}_{RB} = 0.85$

$\overline{\varrho}_{RR_{inv}} = -1$

$\overline{\varrho}_{GB} = 0.98$

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

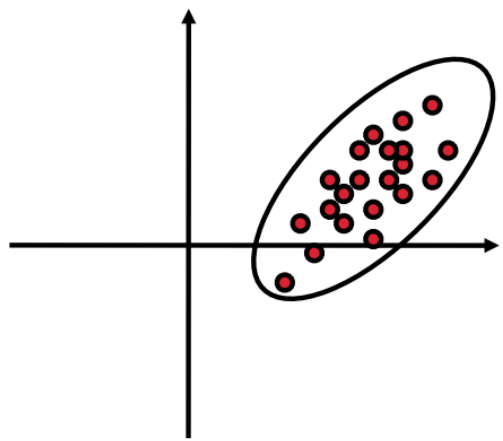
Basic Statistics - Data Matrix & Mean Vector

Data matrix of N data points $\mathbf{x}_i \in \mathbb{R}^D$:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_N^\top \end{bmatrix}$$

Sample set mean of N data points $\mathbf{x}_i$:

$$\mathbf{m} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i = \frac{1}{N} \mathbf{X}^\top \mathbf{1}$$



thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

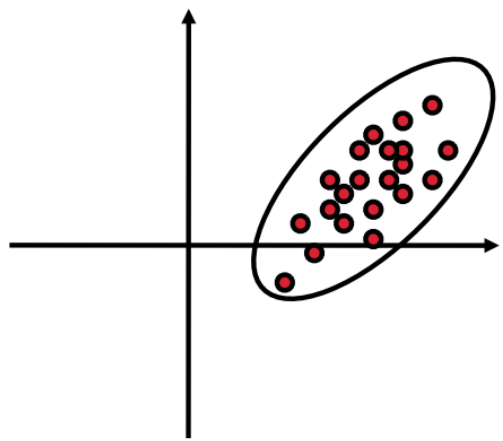
Basic Statistics - Data Covariance Matrix

Data matrix of N data points $\mathbf{x}_i \in \mathbb{R}^{\mathbb{D}}$:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \\ \vdots \\ \mathbf{x}_N^\top \end{bmatrix}$$

Data covariance matrix of N data points $\mathbf{x}_i$:

$$\mathbf{S} = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^\top = \frac{1}{N} \mathbf{X}^\top \mathbf{X} - \mathbf{m}\mathbf{m}^\top$$



Please derive matrix notation!

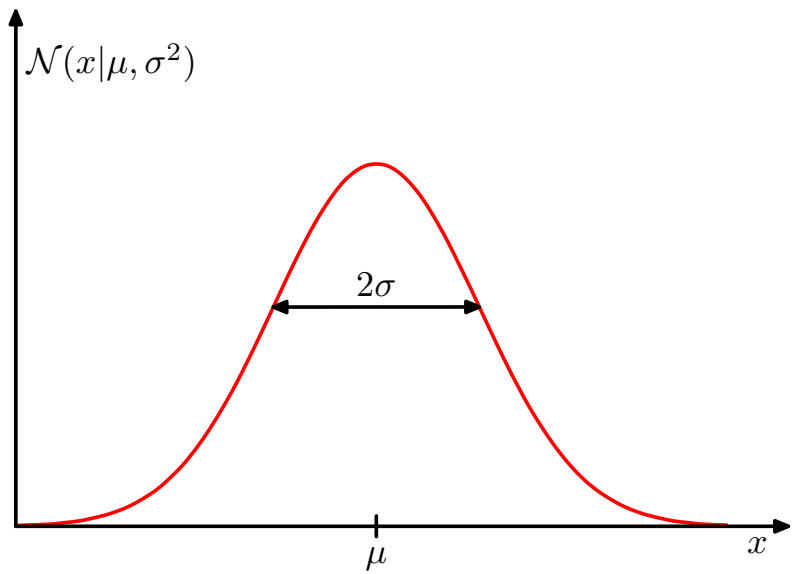
3D Data Processing

Basic Statistics - 1D Gaussian Distribution

The 1D Gaussian or the nD multivariate Gaussian distribution is the simplest way to model noisy data, data distributions and model uncertainty.

The one-dimensional normal distribution or Gaussian distribution $\mathcal{N}(x|\mu, \sigma^2)$ over the random variable x with mean μ and variance σ^2 (or precision $\beta = 1/\sigma^2$) has the form:

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x-\mu)^2} .$$

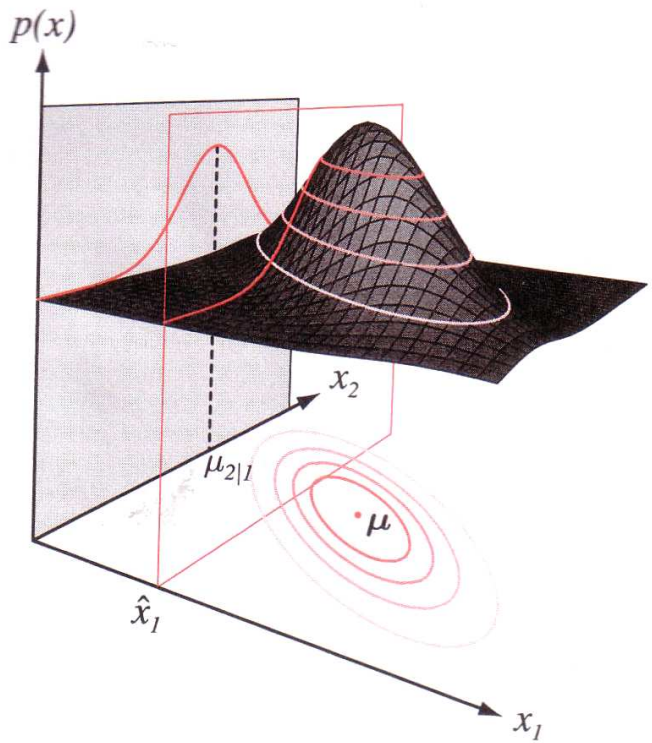


Basic Statistics - Multivariate Gaussian Distribution

The multivariate Gaussian distribution $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$ over a multiple D-dimensional random variable $\mathbf{x}$ with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ or the precision matrix $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^{-1}$ has the form:

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{(2\pi)^D |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \boldsymbol{\Lambda}(\mathbf{x}-\boldsymbol{\mu})},$$

where $|\boldsymbol{\Sigma}|$ denotes the determinant of the covariance matrix. The exponent contains the so-called Mahalanobis distance $\Delta^2 = (\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Lambda}(\mathbf{x} - \boldsymbol{\mu})$ which corresponds to the Euclidean distance when the covariance matrix is equal to the identity matrix $\boldsymbol{\Lambda} = \mathbf{I}$.



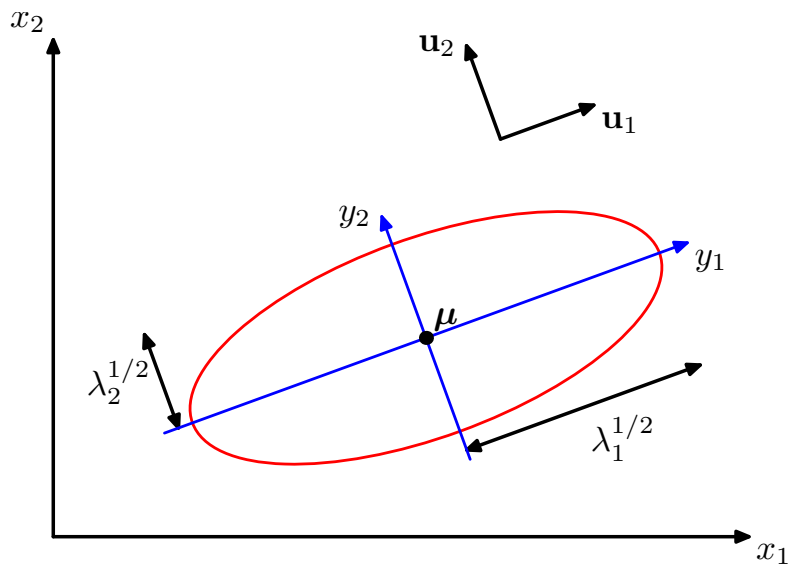
thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Basic Statistics - Multivariate Gaussian Distribution

The covariance matrix can always be decomposed into a weighted sum of the following separable matrices:

$$\boldsymbol{\Sigma} = \sum_{i=1}^{D} \lambda_i \mathbf{u}_i \mathbf{u}_i^\top , \quad \boldsymbol{\Lambda} = \sum_{i=1}^{D} \frac{1}{\lambda_i} \mathbf{u}_i \mathbf{u}_i^\top .$$

The vectors $\mathbf{u}_i$ correspond to the eigenvectors of $\boldsymbol{\Sigma}$ and form an orthonormal basis. The weights λ_i correspond to the eigenvalues of $\boldsymbol{\Sigma}$.



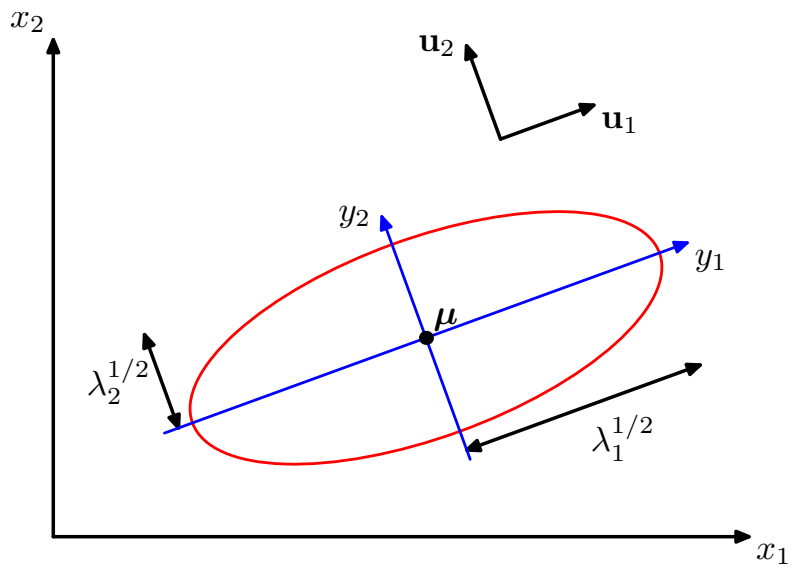
Basic Statistics - Multivariate Gaussian Distribution

Previous decomposition can also be written as follows:

$$\Sigma = \mathbf{U}^\top \mathbf{S} \mathbf{U}, \quad \Lambda = \mathbf{U}^\top \mathbf{S}^{-1} \mathbf{U},$$

where $\mathbf{S} = \text{diag}(\lambda_i)$, $i = 1, \ldots, D$. The row vectors $\mathbf{u}_i^\top$ generate the rotation matix $\mathbf{U}$ and the eigenvalues λ_i the diagonal matrix $\mathbf{S}$. Thus, the Mahalanobis distance can also be calculated in a new coordinate system $\mathbf{y} = \mathbf{U}(\mathbf{x} - \boldsymbol{\mu})$ shifted by $\boldsymbol{\mu}$ and rotated around $\mathbf{U}$:

$$\Delta^2 = \mathbf{y}^\top \mathbf{S}^{-1} \mathbf{y}.$$



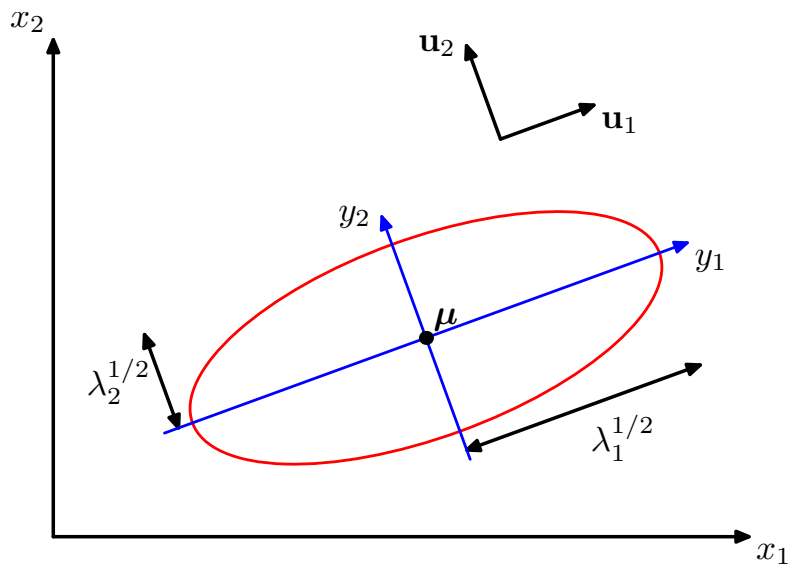
thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Basic Statistics - Multivariate Gaussian Distribution

Thus, each multivariate Gaussian distribution in the coordinate system **x** corresponds to to a mean-free multivariate normal distribution in the transformed coordinate system **y** with the principal axes of the distribution along the coordinate axes

$$\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) = \mathcal{N}(\mathbf{y}|\mathbf{0}, \mathbf{S}) ,$$

where the multivariate normal distribution in the transformed coordinate system corresponds to a product of *D* one-dimensional normal distributions.



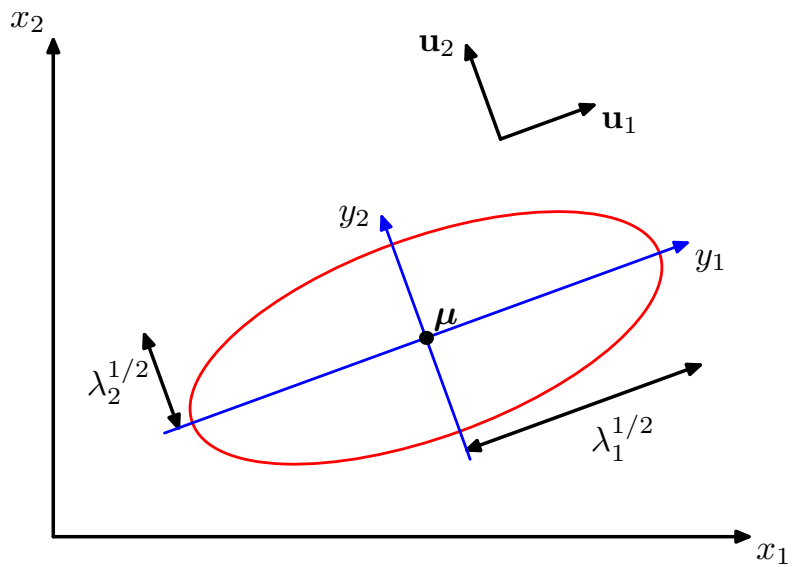
Basic Statistics - Multivariate Gaussian Distribution

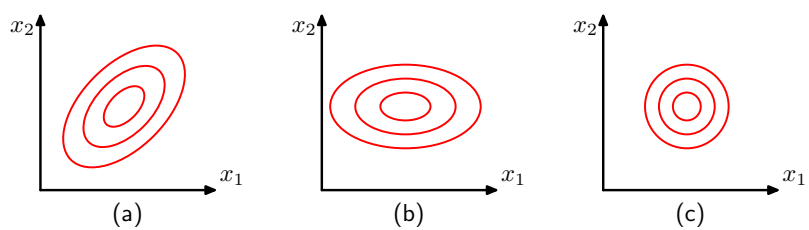
Product of univariate normal distributions:

$$\mathcal{N}(\mathbf{y}|\mathbf{0},\mathbf{S}) = \prod_{i=1}^{D} \frac{1}{\sqrt{2\pi\lambda_i}} e^{-\frac{y_i^2}{2\lambda_i}} .$$

Thus, there are three categories of multivariate Gaussian distributions:

- a) General anisotropic Σ,
- b) separable anisotropic $\Sigma = \text{diag}(\sigma_i^2)$,
- c) separable isotropic $\Sigma = \sigma^2\mathbf{I}$.



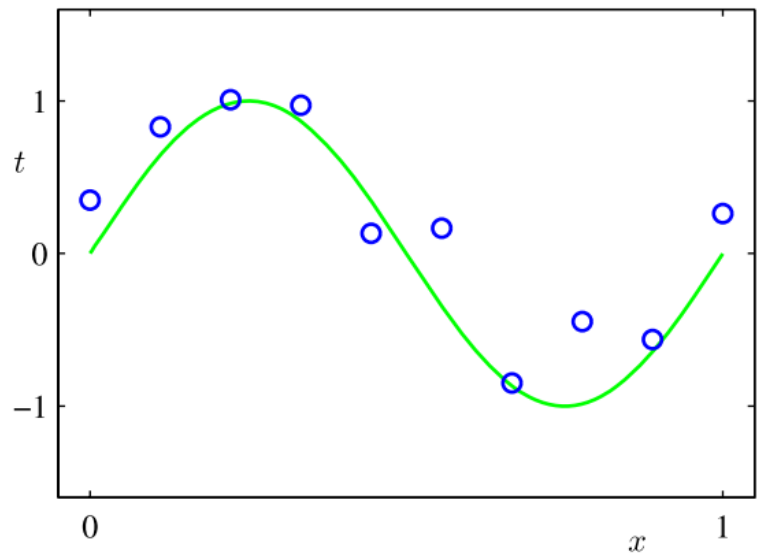


Regression

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Overall Idea

- Fitting parametric models $f(\mathbf{x}_i, \mathbf{w})$ to data points $(\mathbf{x}_i, t_i)$
- Belongs to supervised learning
- The training step tries to find a model that maps a given set of input data $\{\mathbf{x}_i\}$ to some known corresponding target values $\{t_i\}$
- Once a model is found, then we can predict the output for new input data
- linear models are linear in the parameters $\mathbf{w}$



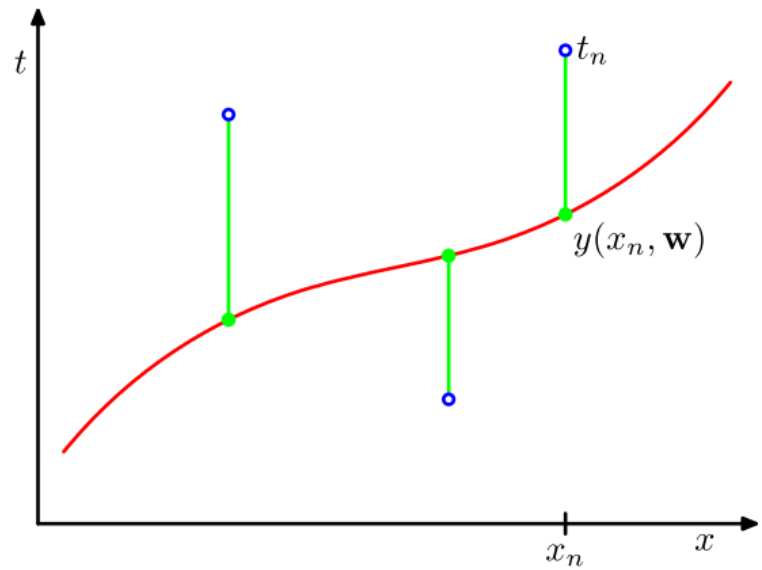
thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Regression

Error Function

- To find a model that fits best to some data, we first have to define some error function to be minimized
- One of the simplest objectives is the sum-of-squares error function:

$$E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}(f(\mathbf{x}_i, \mathbf{w}) - t_i)^2$$



- The optimal parameter set $\mathbf{w}^*$ can be found by the following optimization:

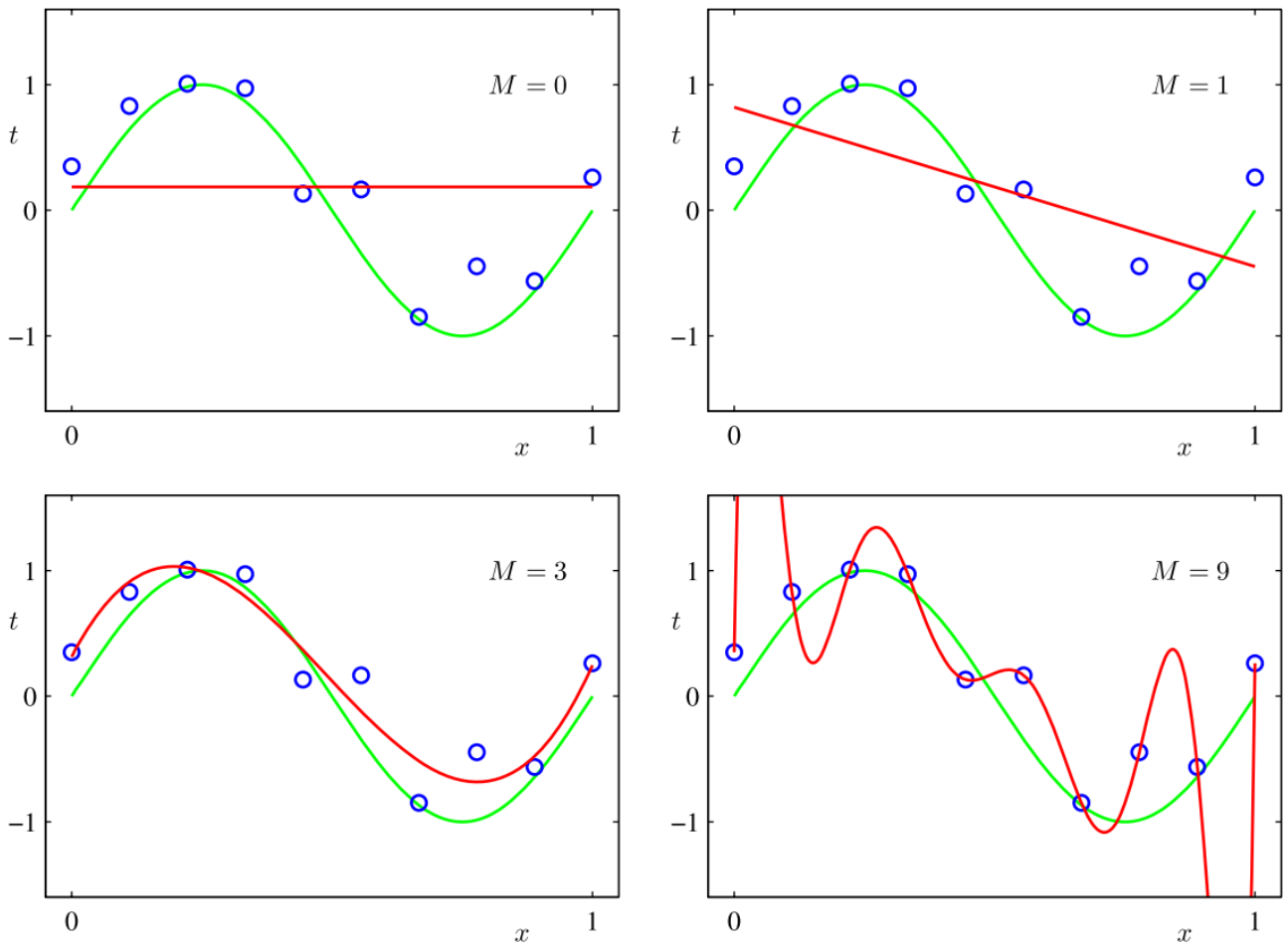
$$\mathbf{w}^* = \text{argmin}_{\mathbf{w}} E(\mathbf{w})$$

- The quality of the fit is given by the root-mean-square error: $\sqrt{2E(\mathbf{w}^*)/N}$

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Regression - Finding the right model complexity

- prevent under- and over-fitting

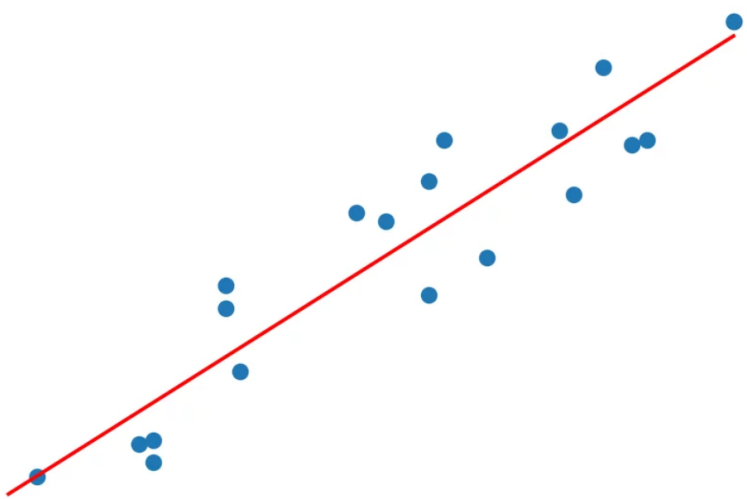


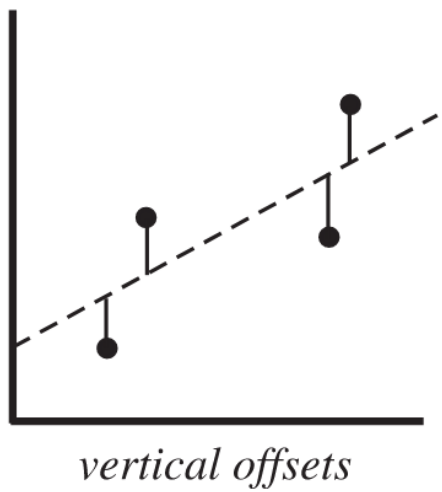
Regression - Line Fitting to 2D Point Clouds

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

2D Line Fitting

- Apply linear regression to find the best fitting line to some 2D point cloud minimizing the sum-of-squares error function
- Rewrite your solution in matrix notation
- Test your solution in matlab





Regression - Summary Line Fitting to 2D Point Cloud

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

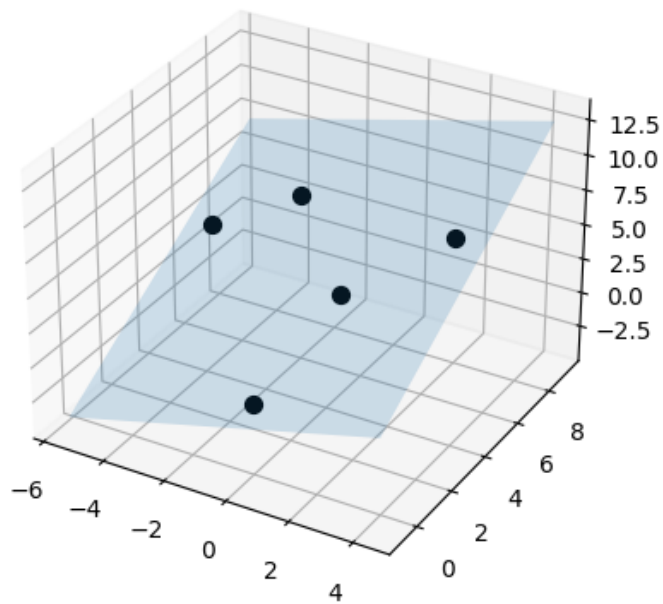
- line equation: $y = ax + b$
- defining data $\mathbf{x}_i = x_i$, target $t_i = y_i$ and parameters $\mathbf{w} = [a, b]^\top$
- normalized error function: $E(\mathbf{w}) = \frac{1}{2}\sum_{i=1}^{N}(f(\mathbf{x}_i, \mathbf{w}) - t_i)^2 = \frac{1}{2N}\sum_{i=1}^{N}(ax_i + b - y_i)^2$
- optimization leads to: $\begin{bmatrix} \frac{1}{N}\sum_i x_i^2 & \frac{1}{N}\sum_i x_i \\ \frac{1}{N}\sum_i x_i & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \frac{1}{N}\sum_i x_i y_i \\ \frac{1}{N}\sum_i y_i \end{bmatrix}$
- solution:
  - $a = \frac{\frac{1}{N}\sum_i x_i y_i - m_x m_y}{\frac{1}{N}\sum_i x_i^2 - m_x^2} = \frac{\sigma_{xy}}{\sigma_x^2} = \varrho_{xy}\frac{\sigma_y}{\sigma_x}$ (regression coefficient)
  - $b = m_y - \frac{\sigma_{xy}}{\sigma_x^2} m_x$
- quality measure: ϱ_{xy} (correlation coefficient)
  If $\varrho_{xy} = 1$, then the data forms a perfect line.
  The smaller ϱ_{xy} the less the data fits to the line.

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Regression - Plane Fitting to 3D Point Clouds

3D Plane Fitting

- Use plane equation: $z = ax + by + c$
- Solve the regression via the Pseudoinverse
- Test your solution in matlab



3D Data Processing

Regression - Plane Fitting to 3D Point Clouds

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Recap: Eigenvalue Decomposition

- Given a squared matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$
- $$\underbrace{\mathbf{M}}_{(n \times n)} = \underbrace{\mathbf{V}}_{(n \times n)} \underbrace{\mathbf{D}}_{(n \times n)} \underbrace{\mathbf{V}^\top}_{(n \times n)} = \sum_{i=1}^{n} \lambda_i \mathbf{v}_i \mathbf{v}_i^\top = \lambda_1 \mathbf{v}_1 \mathbf{v}_1^\top + \lambda_2 \mathbf{v}_2 \mathbf{v}_2^\top + \cdots + \lambda_n \mathbf{v}_n \mathbf{v}_n^\top ,$$
- Orthogonal matrix $\mathbf{V} = \left[\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\right]$ (rotation matrix)
- Diagonal squared matrix $\mathbf{D} = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$
- Sorted eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \sigma_n$

Regression - Plane Fitting to 3D Point Clouds

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Recap: Singular Value Decomposition (SVD)

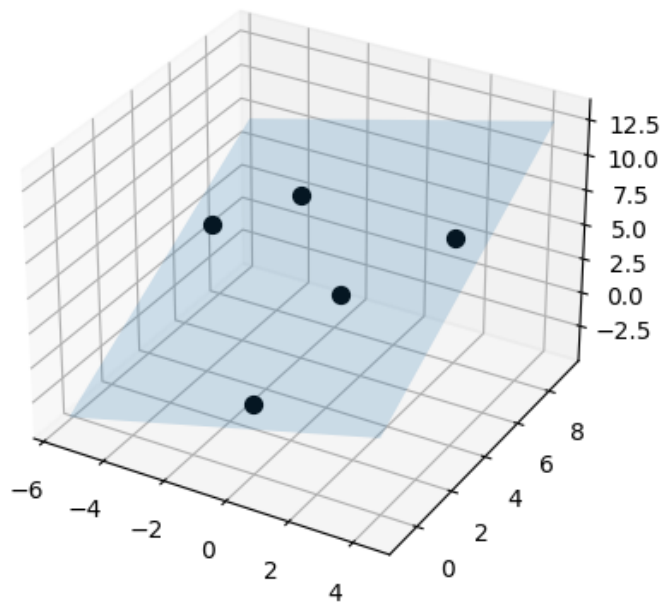
- SVD of a matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$ if $m \geq n$
- $$\underbrace{\mathbf{M}}_{(m \times n)} = \underbrace{\mathbf{U}}_{(m \times m)} \underbrace{\mathbf{S}}_{(m \times n)} \underbrace{\mathbf{V}^\top}_{(n \times n)} = \sum_{i=1}^{n} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top = \sigma_1 \mathbf{u}_1 \mathbf{v}_1^\top + \sigma_2 \mathbf{u}_2 \mathbf{v}_2^\top + \cdots + \sigma_n \mathbf{u}_n \mathbf{v}_n^\top ,$$
- Orthogonal matrices $\mathbf{U} = \left[\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m\right]$ and $\mathbf{V} = \left[\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\right]$
- Diagonal non-squared matrix $\mathbf{S} = \left\{\sigma_1, \sigma_2, \ldots, \sigma_n\right\}$
- Sorted singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$

Regression - Plane Fitting to 3D Point Clouds

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

3D Plane Fitting: Efficient Numeric Calculation

- Let's have a look at the relation between the pseudoinverse and the singular value decomposition
- Use the relation to solve the plane fitting problem
- Test your solution in matlab



3D Data Processing

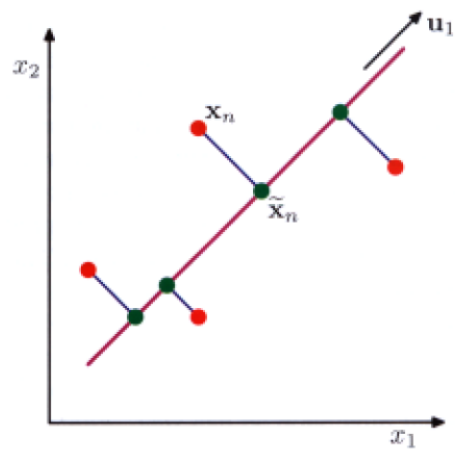
Linear Subspaces - Line and Plane Fitting

Technical University of Applied Sciences Würzburg-Schweinfurt

Overall Idea

- Finding a $D-1$ dimensional linear subspace of a D dimensional point cloud
- Input data are the D-dimensional coordinates of the point cloud
- Find a coordinate transformation that maximizes the variance of the data along the first dimension of the transformed data
- The principal components of PCA (Principal Component Analysis) define the basis vectors of this coordinate transformation

PCA seeks a space of lower dimensionality, known as the principal subspace (magenta line), such that the orthogonal projection of the point cloud (red dots) onto this subspace maximizes the variance of the projected points (green dots). That is the same as minimizing the sum-of-squares of the projection errors (blue lines).
`C.M. Bishop`

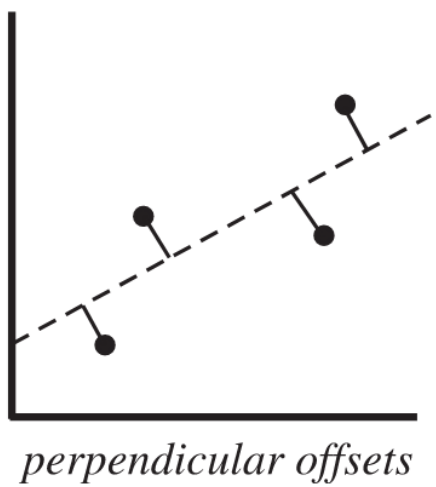


Linear Subspaces - 2D Line Fitting

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

2D Line Fitting via Principal Component Analysis

- Find the subspace by minimizing the perpendicular error
- Solve it by eigenvalue decomposition of the covariance matrix of mean value free data
- Solve it by singular value decomposition of the data matrix
- Use the inverse of the condition number to quantify the error

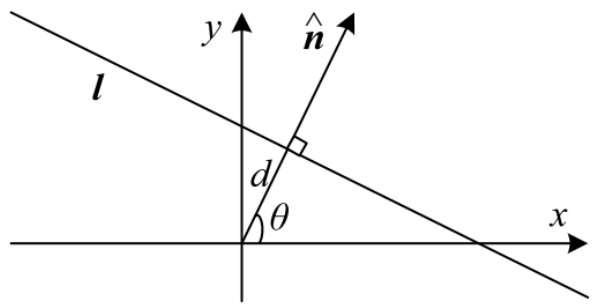


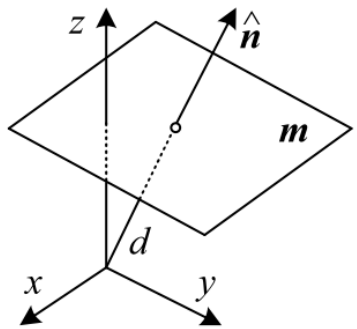
thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Linear Subspaces - Line and Plane Fitting

Recap: Hesse Normalform

- $\mathbf{n}^\top \mathbf{x} - d = 0$
- Distance to origin: $d \geq 0$
- Normal vector of line/plane: $\mathbf{n}^\top$
- Distance a of point $\mathbf{p}$ to line/plane: $\mathbf{n}^\top \mathbf{p} - d = a$



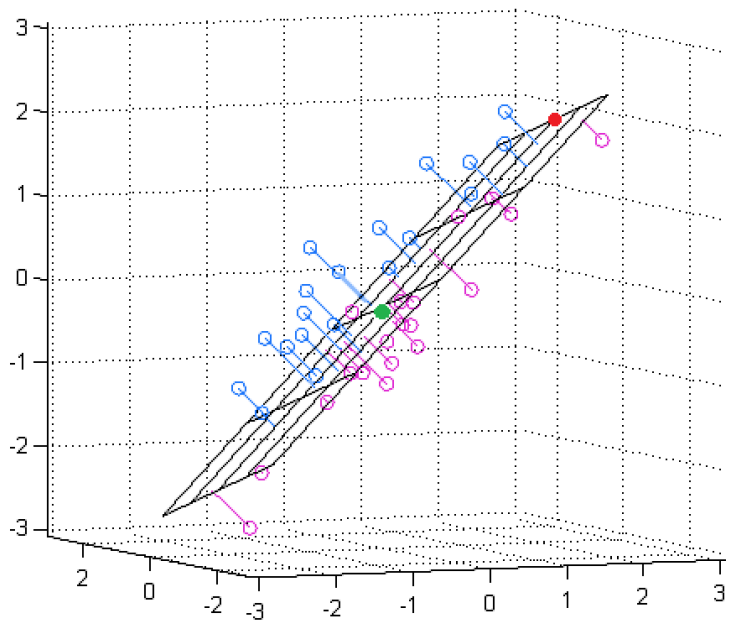


Linear Subspaces - 3D Plane Fitting

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

3D Plane Fitting via Principal Component Analysis

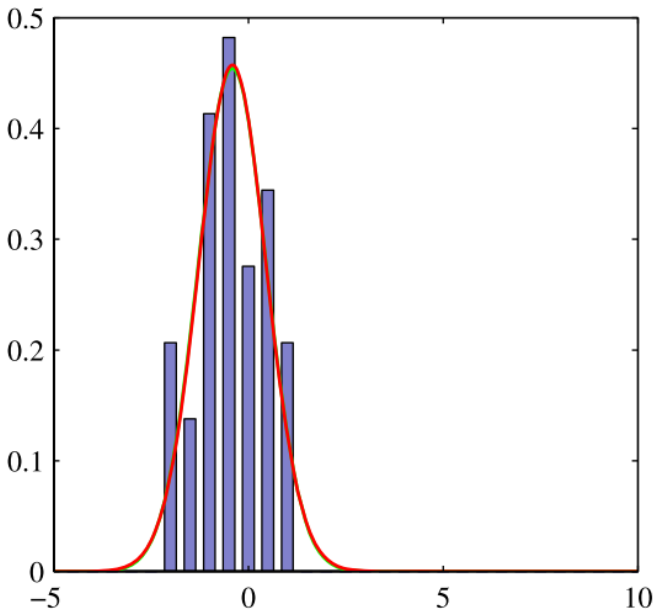
- Use SVD of the data matrix to find the plane
- Test it with matlab



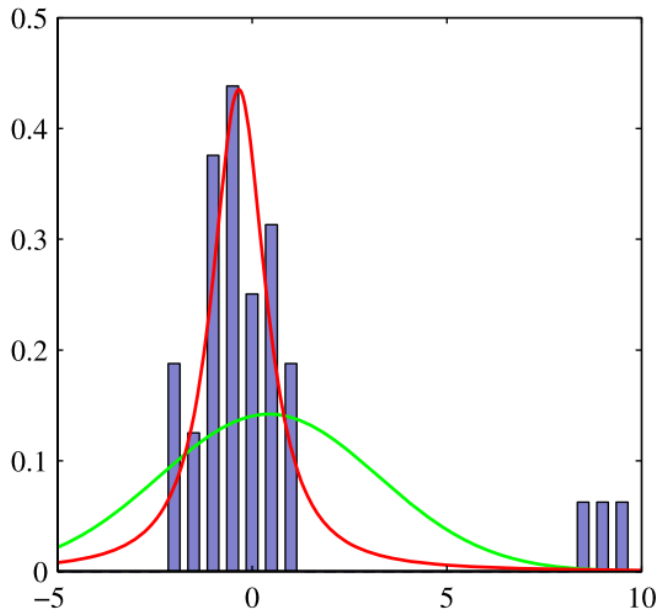
thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Robust Line and Plane Fitting to Point Clouds

How do Least Squares Solution work for Outliers?



Error statistics without outliers



Error statistics with outliers

Robust Line and Plane Fitting to Point Clouds

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

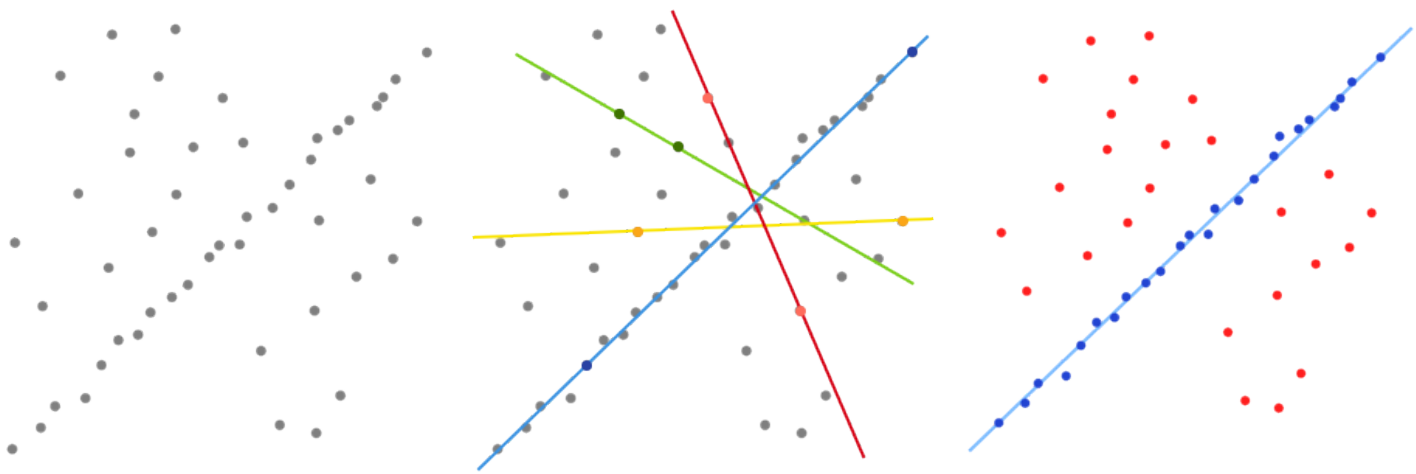
Robust Solution: RANSAC (Random Sample Consensus) Algorithm

1. Randomly select the minimum number of points from the data points that are needed to calculate the parameters of the model. This is done with the expectation that this set is free of outliers.
2. Using the selected points, determine the model parameters.
3. Determine the subset of measured values whose distance from the model curve is smaller than a certain threshold (this subset is called the consensus set). If it contains a certain minimum number of values, a good model was probably found and the consensus set is saved.
4. Repeat steps 1-3 several times until the consensus set exceeds a certain number or some defined maximum iteration number is reached.

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Robust Line and Plane Fitting to Point Clouds

Robust Solution: RANSAC (Random Sample Consensus) Algorithm

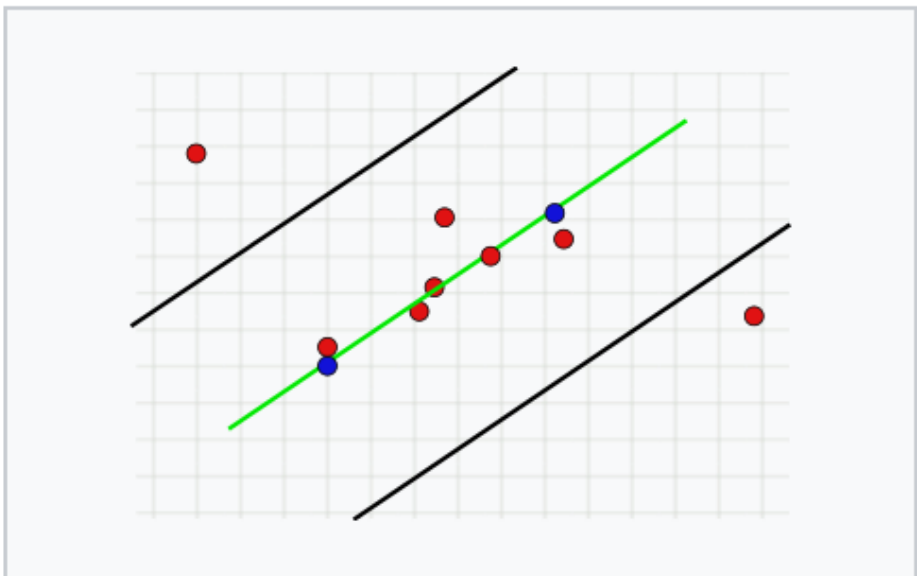


Robust Line and Plane Fitting to Point Clouds

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Robust Solution: RANSAC (Random Sample Consensus) Algorithm

- Correct Threshold

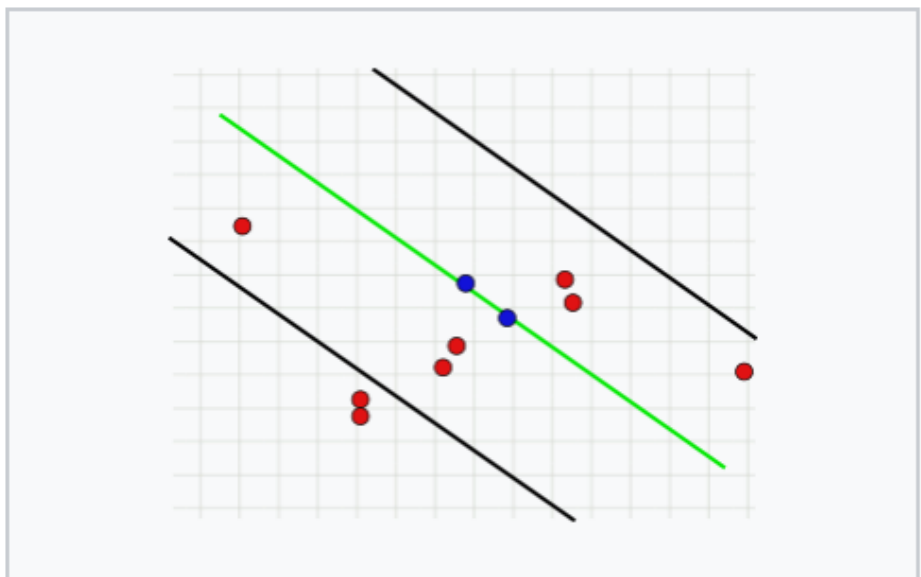


Robust Line and Plane Fitting to Point Clouds

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Robust Solution: RANSAC (Random Sample Consensus) Algorithm

- Threshold too large

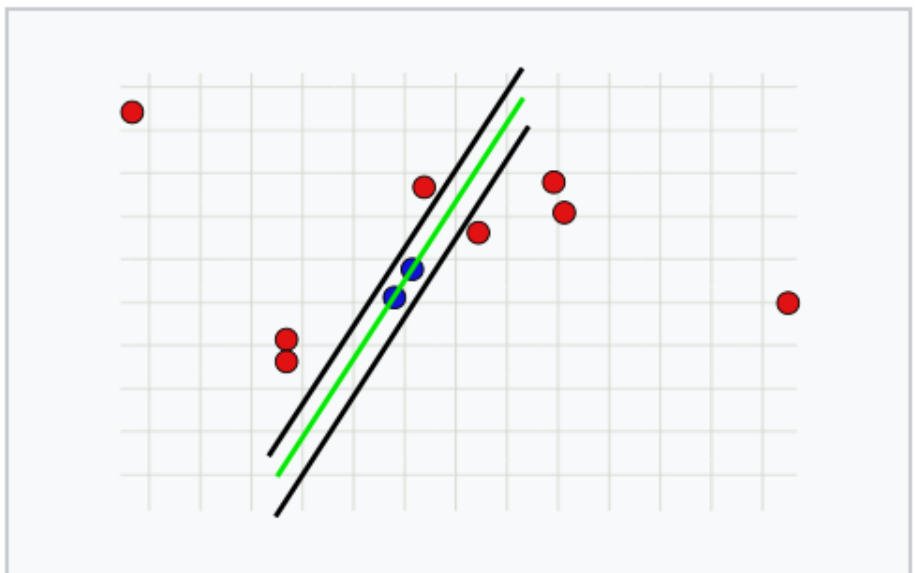


Robust Line and Plane Fitting to Point Clouds

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Robust Solution: RANSAC (Random Sample Consensus) Algorithm

- Threshold too small



3D Data Processing

Matrix Decomposition/Factorization

Technical University of Applied Sciences Würzburg-Schweinfurt

Generative Model: $\underbrace{\mathbf{X}}_{N\times D} \approx \underbrace{\mathbf{R}}_{N\times K}\underbrace{\mathbf{M}}_{K\times D}$, $\mathbf{x}_n \approx [\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_K]\,\mathbf{r}_n$ with $K \ll N$.

$\mathbf{x}_n$

$\approx r_{1n}$  $\mathbf{m}_1$

$+r_{2n}$  $\mathbf{m}_2$

$+r_{3n}$  $\mathbf{m}_3$

$+r_{4n}$  $\mathbf{m}_4$

$+\ldots$

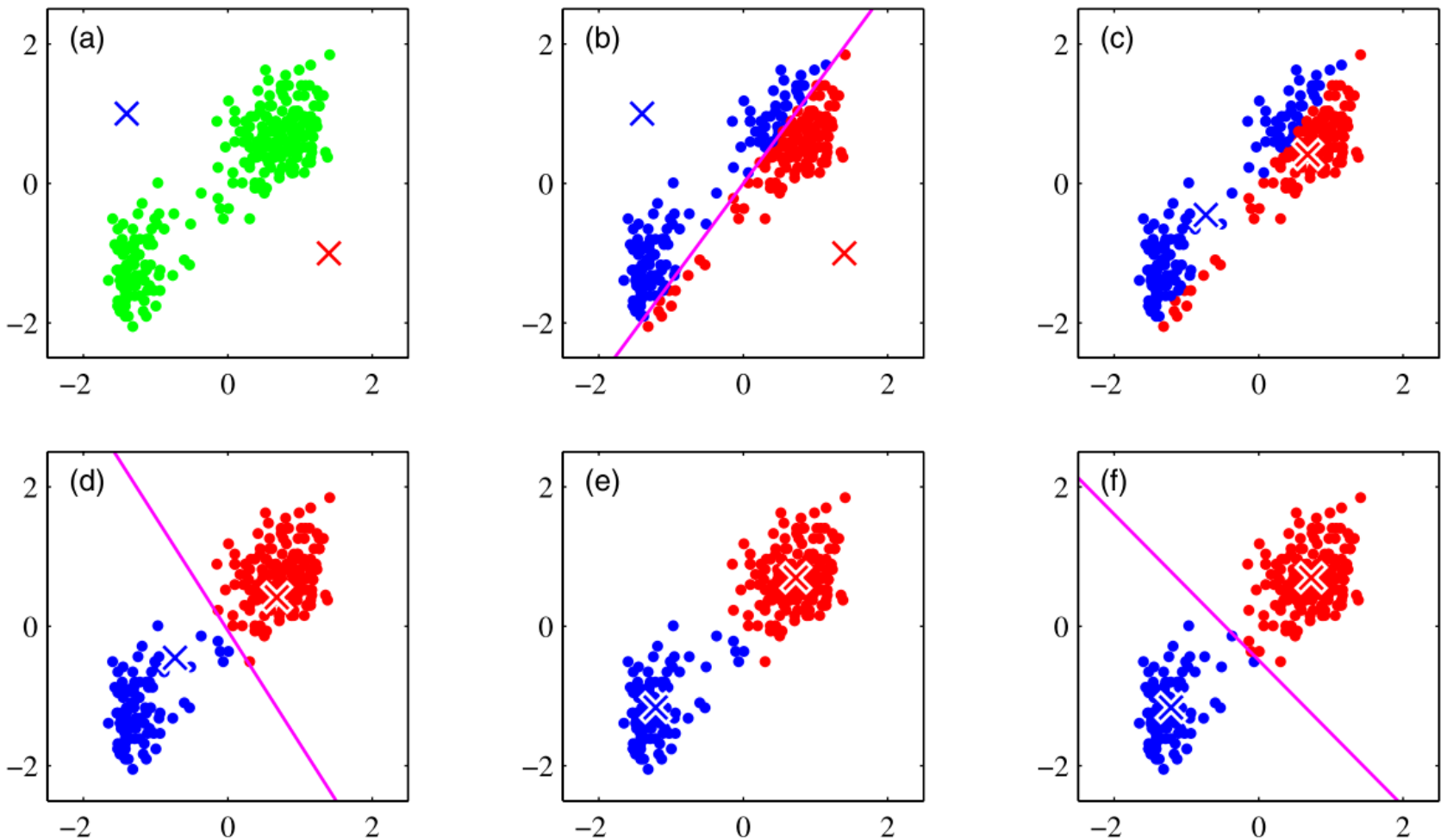
Objective: $J(\mathbf{R}, \mathbf{M}) = \|\mathbf{X} - \mathbf{RM}\| + \mathit{constraints}(\mathbf{R}, \mathbf{M})$

- data matrix **X**
- basis vectors **M**
- activities **R**

Applications

- PCA, ICA, NMF, ...
- K-Means, Gaussian Mixtures, Fuzzy-C-means, ...
- Codebook/Representation learning, Classification, ...

Recap: K-means Clustering Algorithm



Recap: K-means Clustering Algorithm

Technical University of Applied Sciences Würzburg-Schweinfurt

Let's have a look at

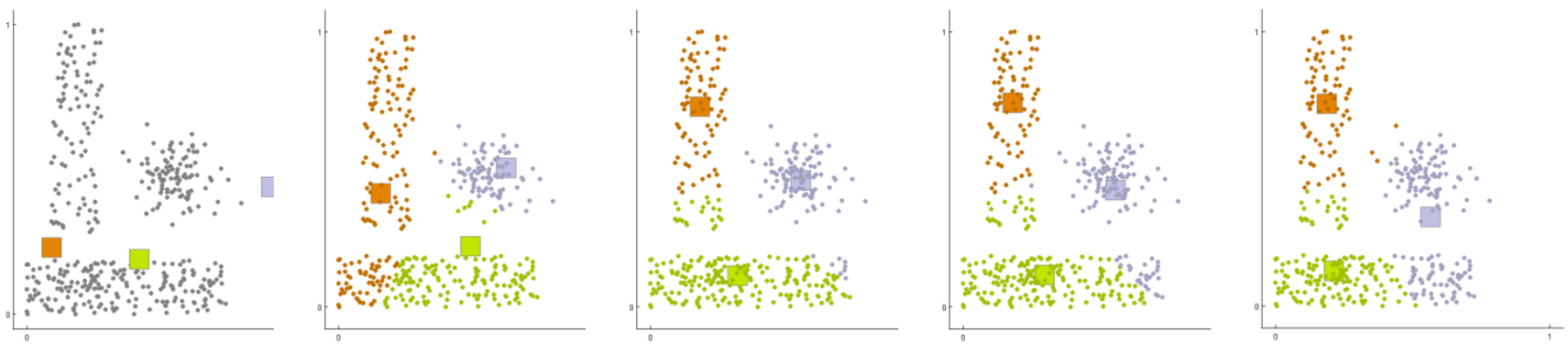
- the algorithm and
- the underlying optimization problem

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

When does K-means fail?

Disadvantages of K means:

- it can get stuck in a local minimum or on a saddle point
- it is dependent on the initialization of the cluster centers
- the objective is not robust against outliers in the data
- the data cannot be devided by hyperplanes



3D Data Processing

Example: RGB-D Clustering

Technical University of Applied Sciences Würzburg-Schweinfurt

Applying K-means clustering to RGB-D images

- 27648 Pixel, 8 Cluster & 76800 Pixel, 16 Cluster

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Superpixels

Let's do some paper review of the SLIC algorithm

- Find out something about conference/journal and authors
- What are superpixels?
- What are the main properties of the algorithm?
- How is it related to and what is the difference to k-means?
- How is the distance measure defined?

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

SLIC: Simple Linear Iterative Clustering

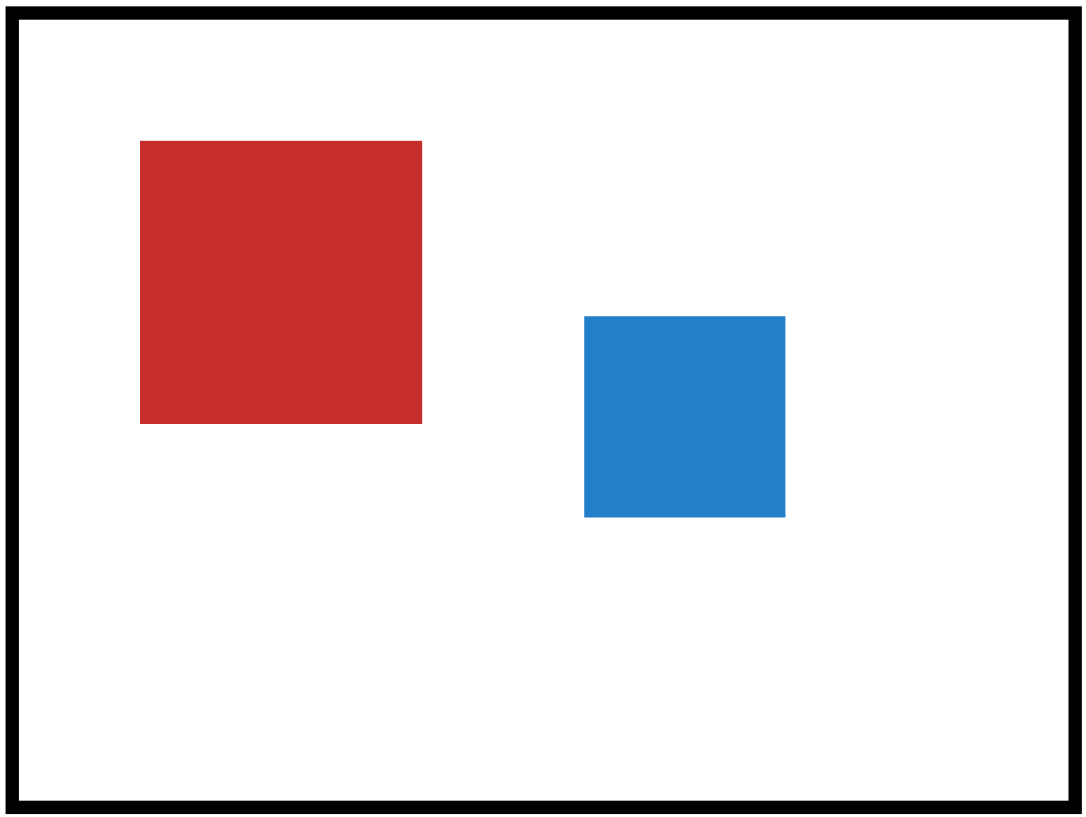


Figure: Initial image.

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

SLIC: Simple Linear Iterative Clustering

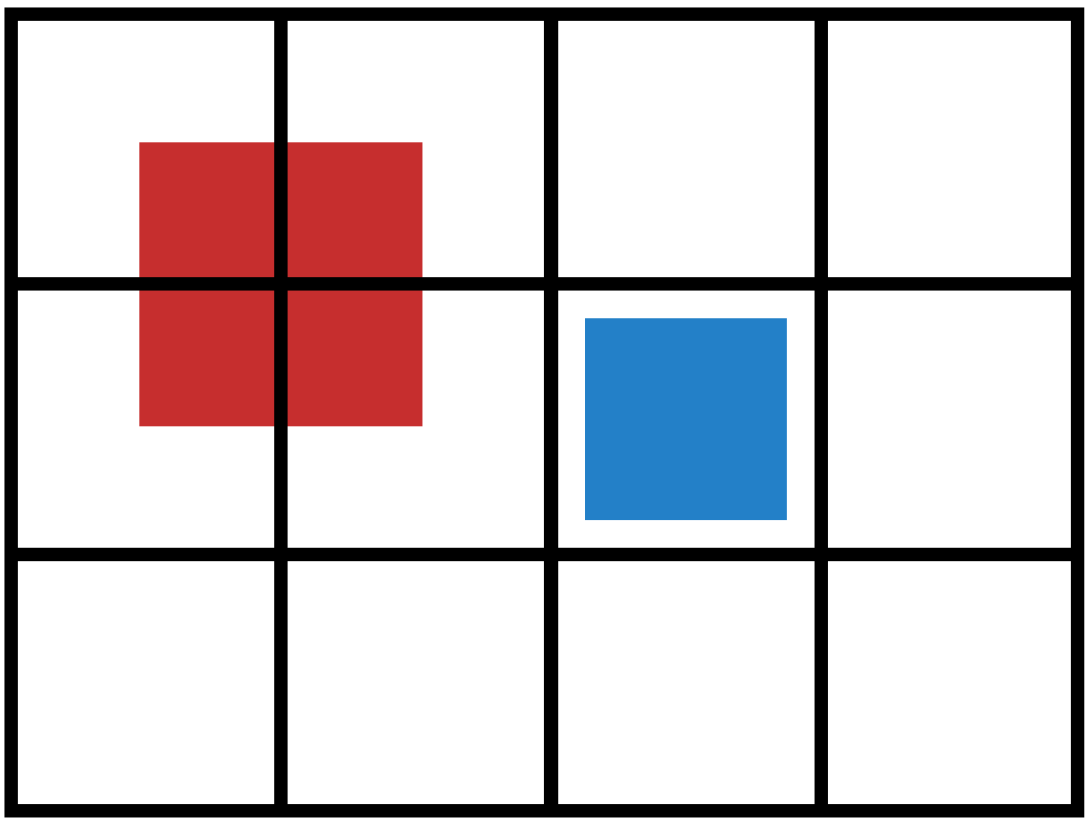


Figure: Seeding grid placed.

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

SLIC: Simple Linear Iterative Clustering

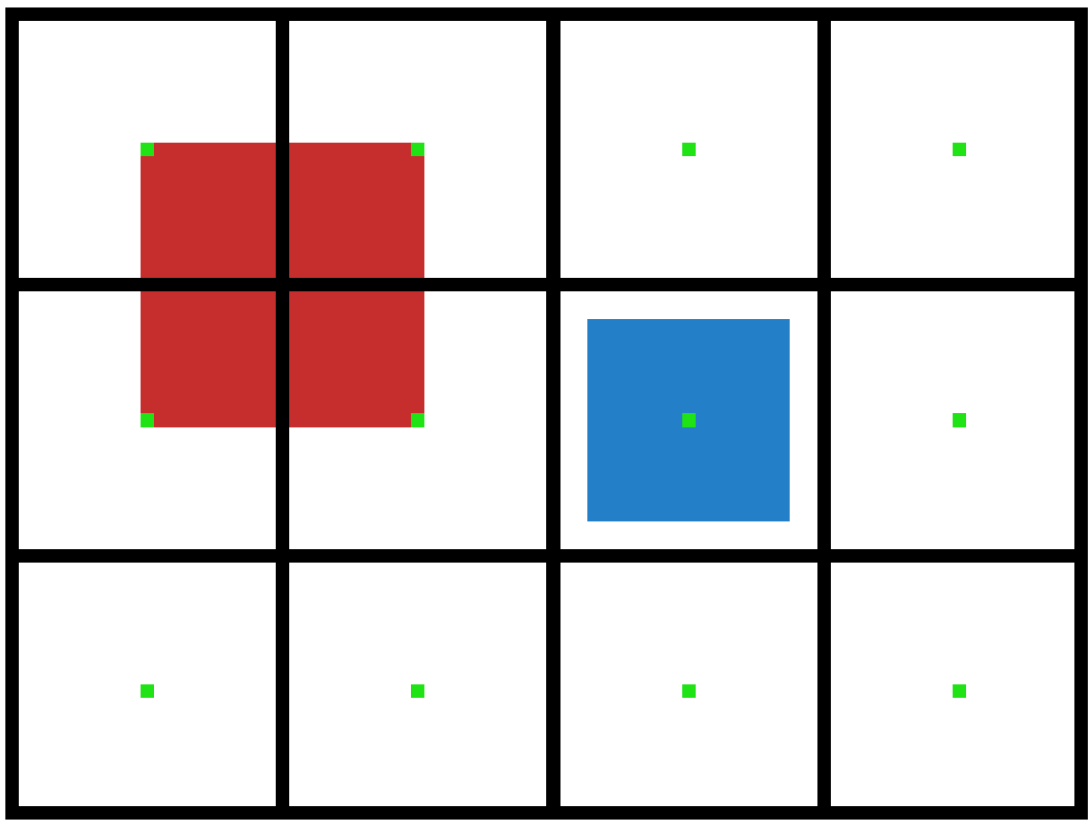


Figure: Seeding centers placed.

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

SLIC: Simple Linear Iterative Clustering

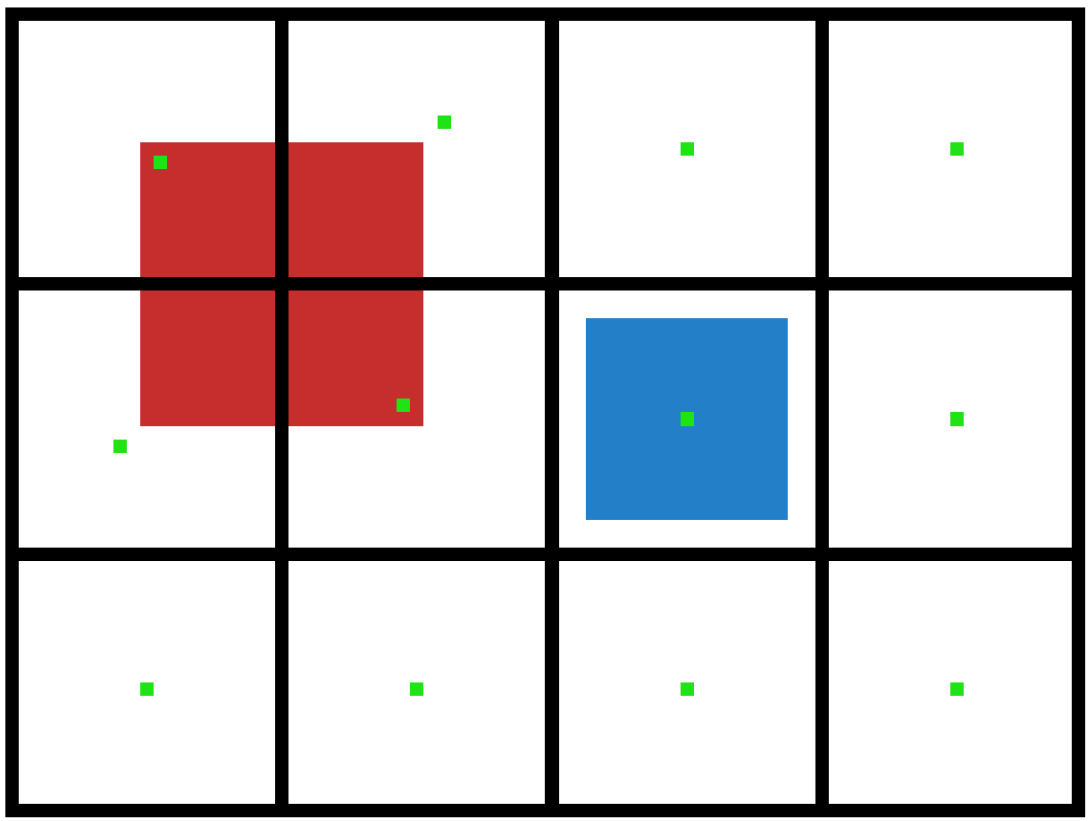


Figure: Centers moved away from edges and corners.

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

SLIC: Simple Linear Iterative Clustering

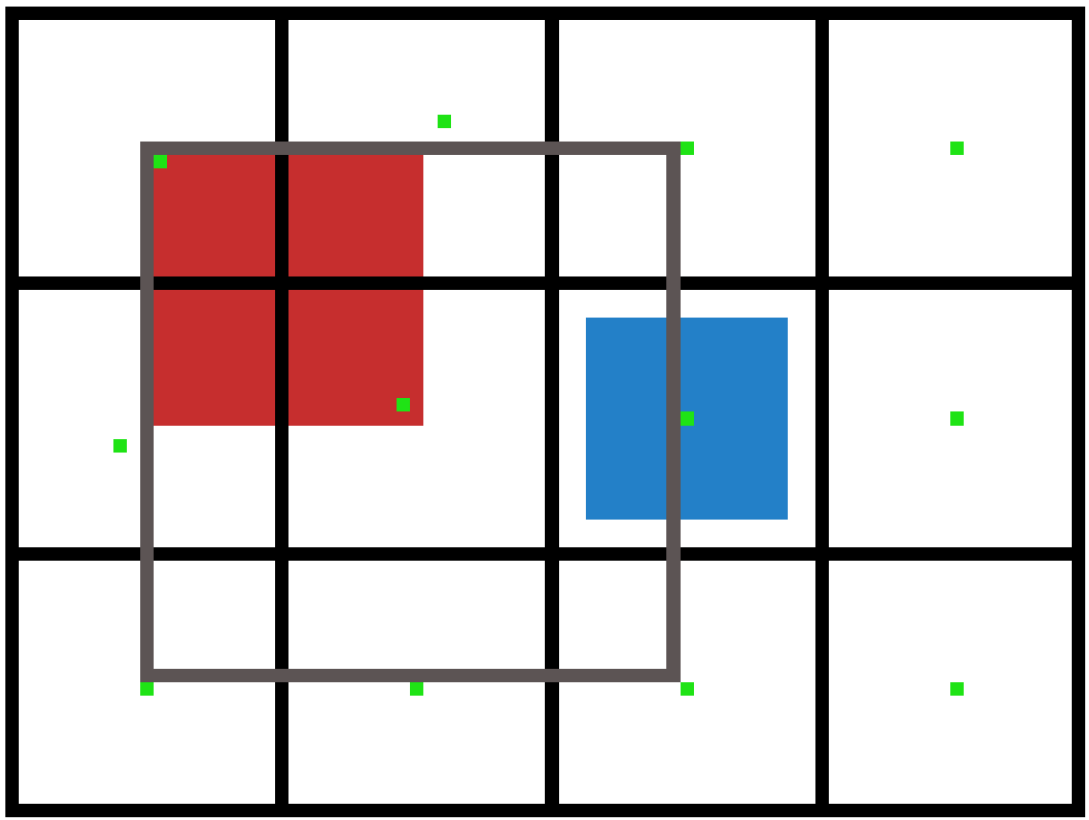


Figure: The area to search for fitting pixels.

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

SLIC: Simple Linear Iterative Clustering

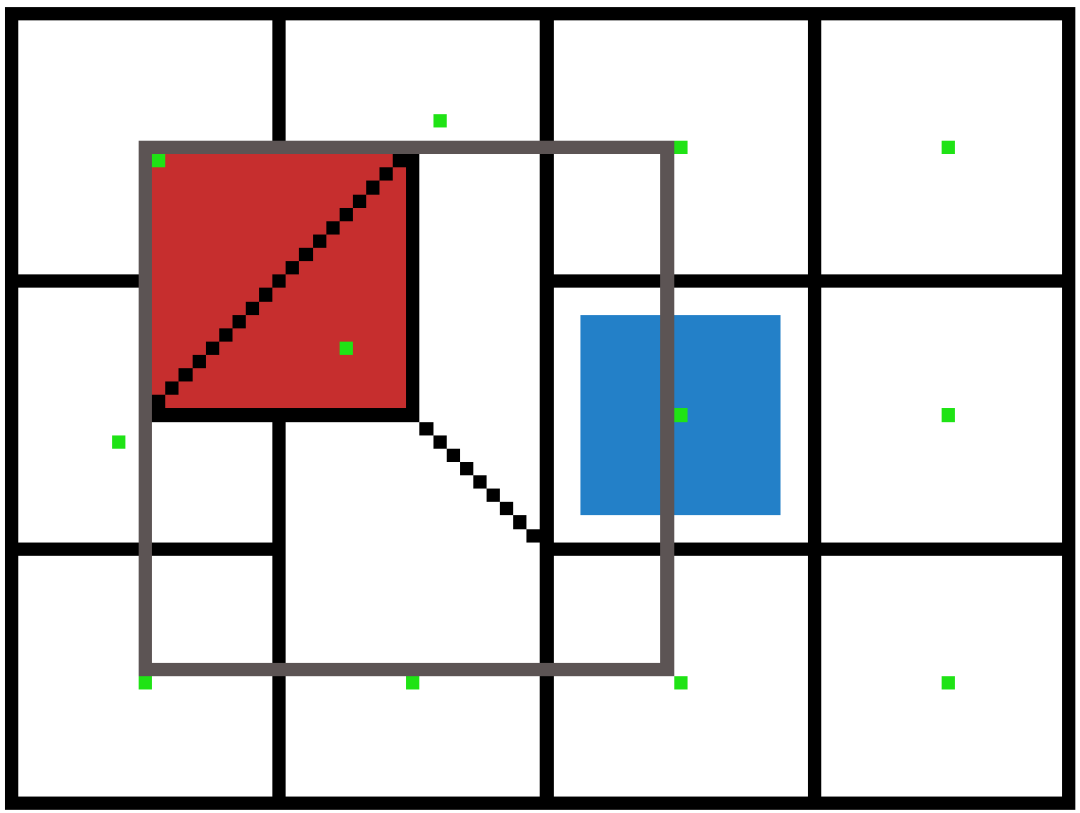


Figure: A few iterations of adding pixels and recentering.

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

SLIC: Simple Linear Iterative Clustering

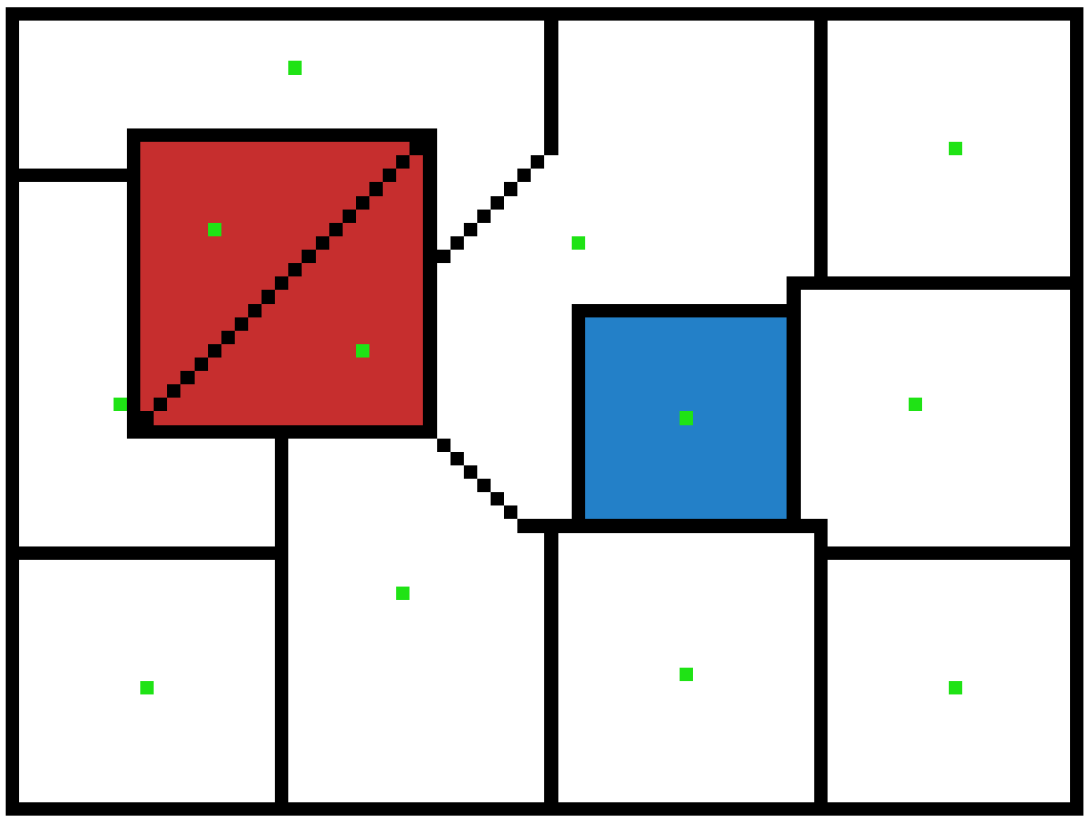


Figure: The superpixels after recentering and adding similar pixels.

3D Data Processing

Example: Segment Anything Model (SAM)

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

State-of-the-Art Image Segmentation using Transformer Embeddings

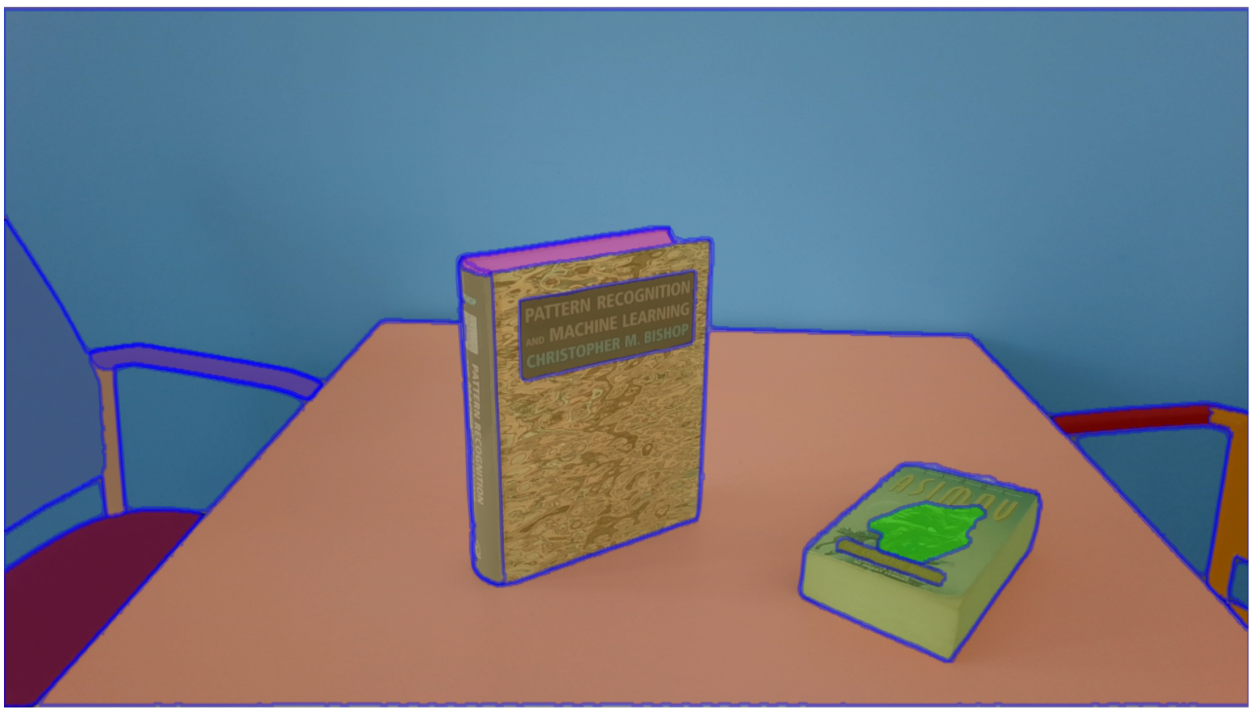
- Zero-Shot Learning, trained on 1 Billion masks and 11 Million images

thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Example: Segment Anything Model (SAM)

State-of-the-Art Image Segmentation using Transformer Embeddings

- Zero-Shot Learning, trained on 1 Billion masks and 11 Million images

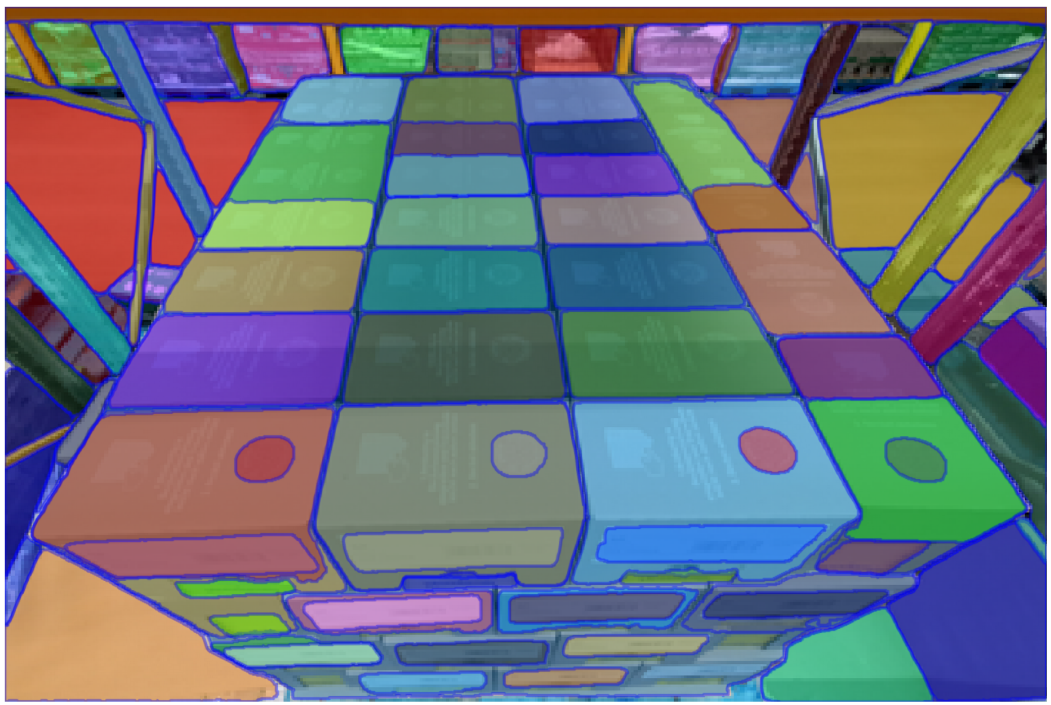


thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Example: Segment Anything Model (SAM)

State-of-the-Art Image Segmentation using Transformer Embeddings

- Zero-Shot Learning, trained on 1 Billion masks and 11 Million images



thws
Technical University
of Applied Sciences
Würzburg-Schweinfurt

Example: Segment Anything Model (SAM)

State-of-the-Art Image Segmentation using Transformer Embeddings

- Zero-Shot Learning, trained on 1 Billion masks and 11 Million images