

Entitäten-Relationen-Modell:

- Entität ist ein abstraktes Objekt, welches durch einen Satz von Attributen bestimmt wird
- Entitäten gleichen Typs weisen den gleichen Satz von Attributen auf
- Entitäten können durch Relationen verbunden werden:
 - 1-1 Relationen: eine Entität von Typ A ist eineindeutig mit einer Entität vom Typ B verbunden
 - 1-n Relation: eine Entität von Typ A ist mit n Entitäten vom Typ B verbunden
 - m-n Relation: m Entitäten von Typ A sind mit n Entitäten vom Typ B verbunden

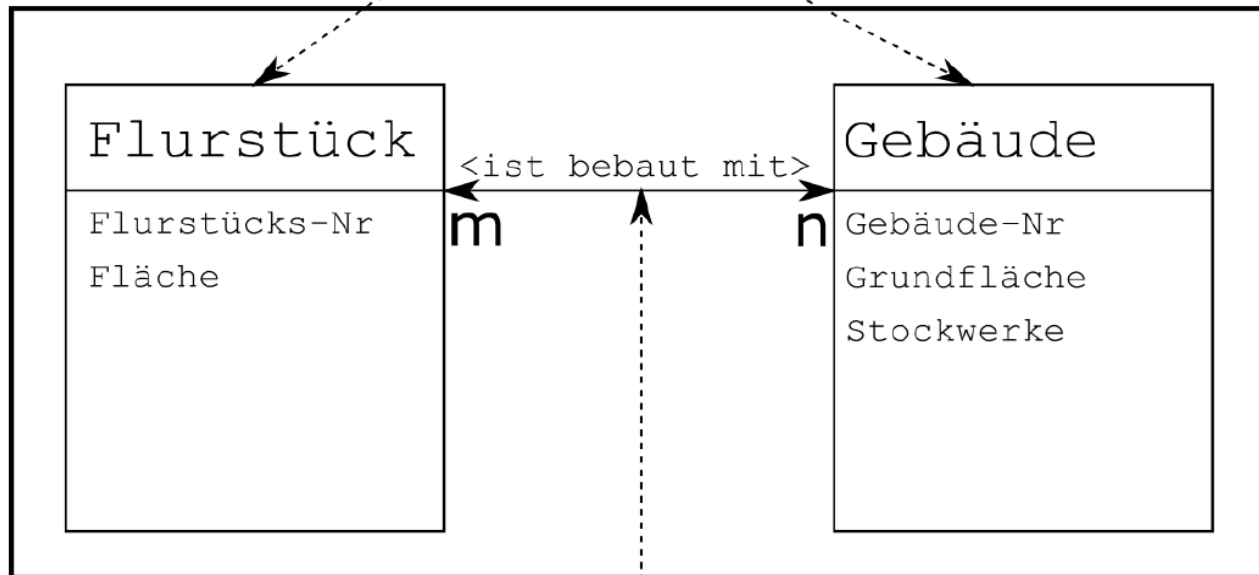
Entitäten-Relationen-Modell:

Entitätsmengen

nt wird

vom Typ B

1
den



Relation

Entitäten-Relationen-Modell:

- Entität ist ein abstraktes Objekt, welches durch einen Satz von Attributen bestimmt wird
- Entitäten gleichen Typs weisen den gleichen Satz von Attributen auf
- Entitäten können durch Relationen verbunden werden:
 - 1-1 Relationen: eine Entität von Typ A ist eineindeutig mit einer Entität vom Typ B verbunden
 - 1-n Relation: eine Entität von Typ A ist mit n Entitäten vom Typ B verbunden
 - m-n Relation: m Entitäten von Typ A sind mit n Entitäten vom Typ B verbunden

Graphen-basierte Modelle:

- Beziehungen zwischen Entitäten werden in gerichteter Form dargestellt
- hierarchische und vernetzte Darstellung möglich

Entitäten-Relationen-Modell:

- Entität ist ein abstraktes Objekt, welches durch einen Satz von Attributen bestimmt wird
- Entitäten Gleichen Typs weisen den Gleichen Satz von Attributen auf
- Entitäten können durch Relationen verbunden werden:
 - 1-1 Relationen: eine Entität von Typ A ist eineindeutig mit einer Entität vom Typ B verbunden
 - 1-n Relation: eine Entität von Typ A ist mit n Entitäten vom Typ B verbunden
 - m-n Relation: m Entitäten von Typ A sind mit n Entitäten vom Typ B verbunden

Graphen-basierte Modelle:

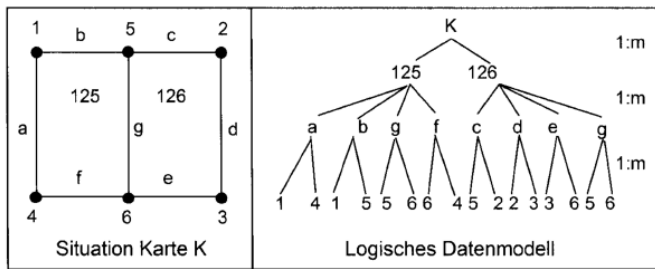


Abbildung 6.43: Hierarchisches Datenmodell.

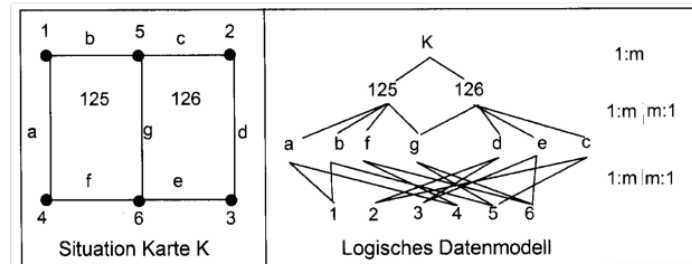
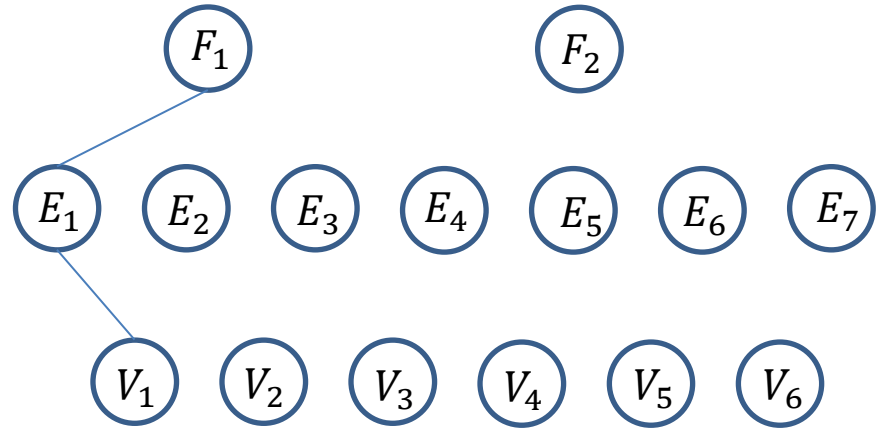
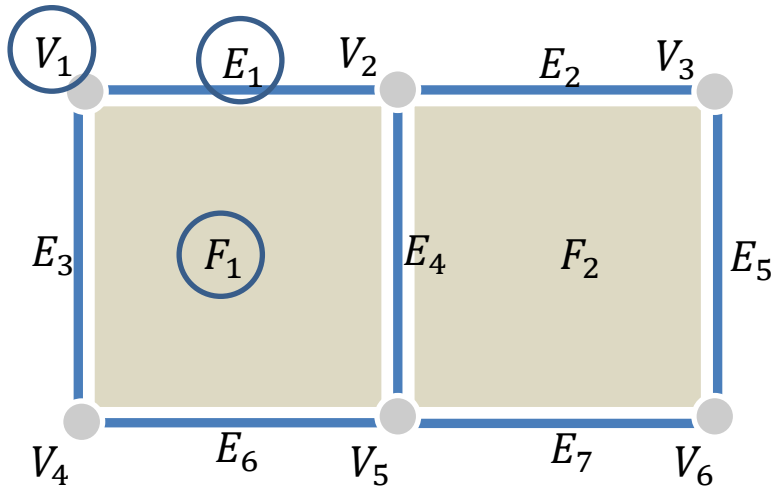


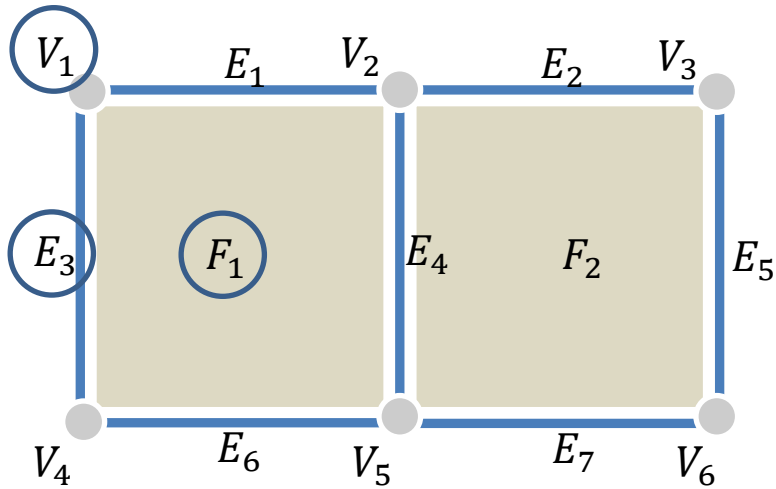
Abbildung 6.44: Netzwerkartiges Datenmodell.

Inzidenzgraph als Repräsentant für ein Graphenbasiertes Datenmodell

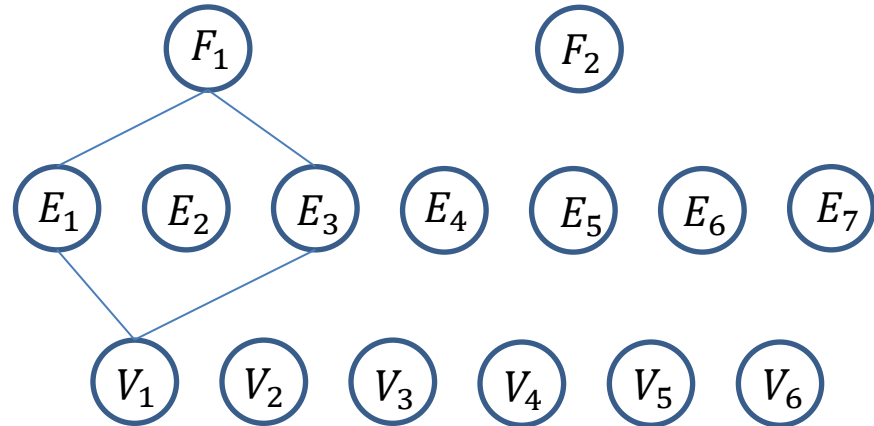


$\{V_1, E_1, F_1\}$

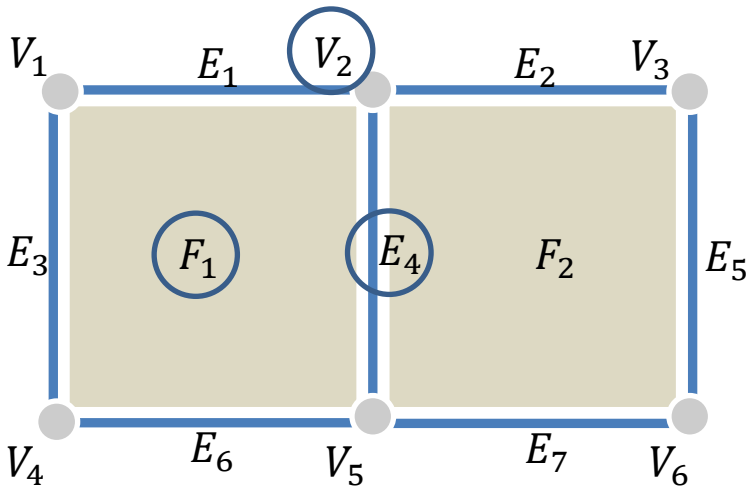
Inzidenzgraph als Repräsentant für ein Graphenbasiertes Datenmodell



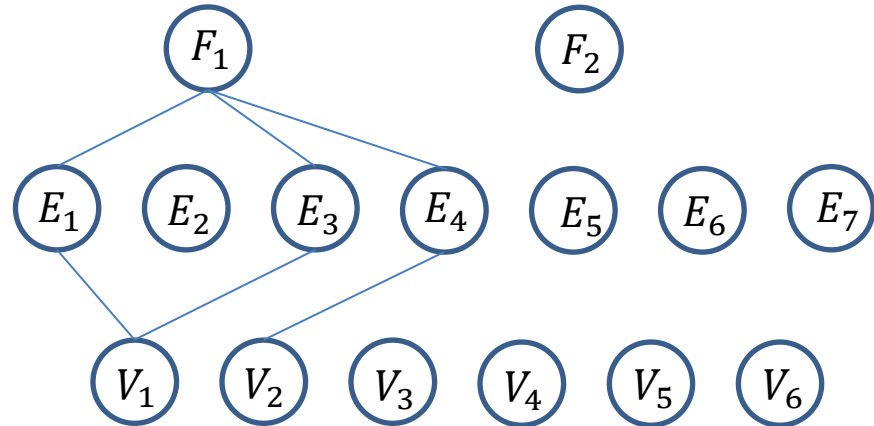
$\{V_1, E_3, F_1\}$



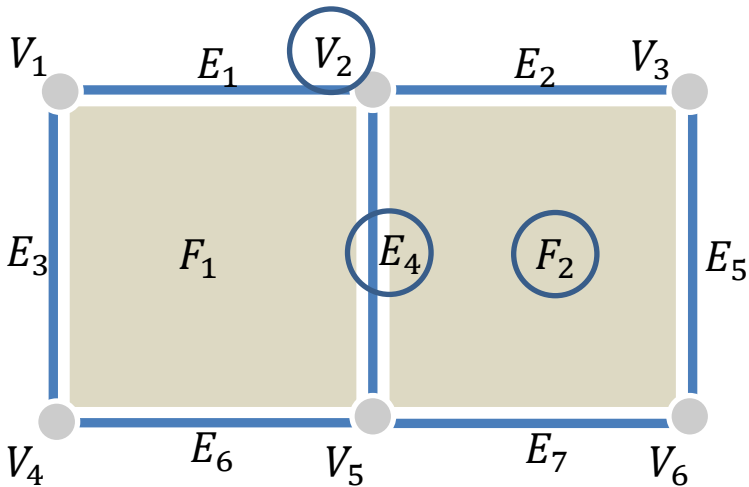
Inzidenzgraph als Repräsentant für ein Graphenbasiertes Datenmodell



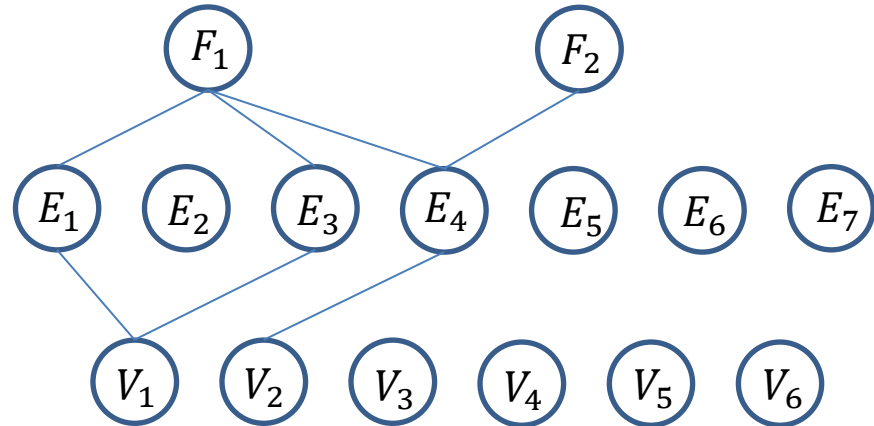
$\{V_2, E_4, F_1\}$



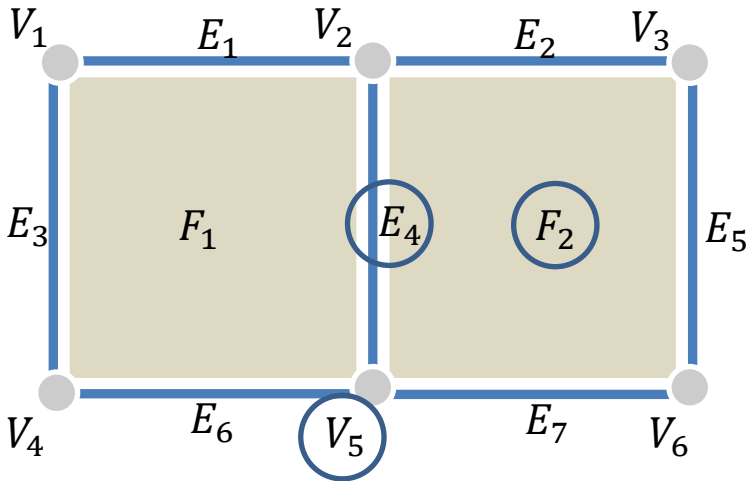
Inzidenzgraph als Repräsentant für ein Graphenbasiertes Datenmodell



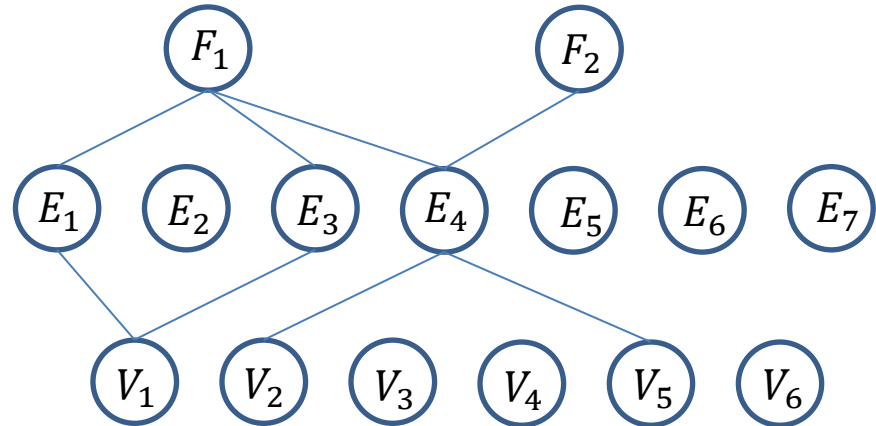
$\{V_2, E_4, F_2\}$



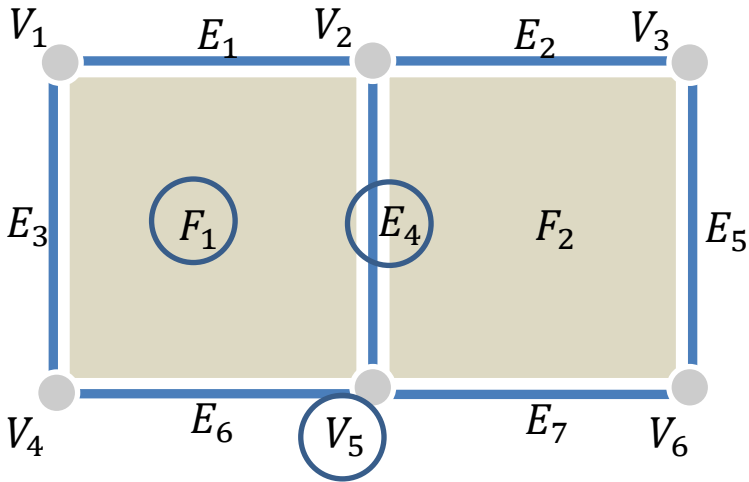
Inzidenzgraph als Repräsentant für ein Graphenbasiertes Datenmodell



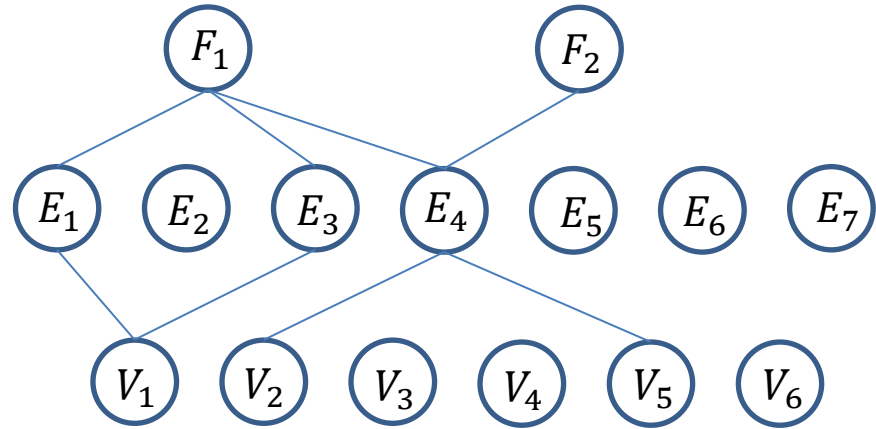
$\{V_5, E_4, F_2\}$



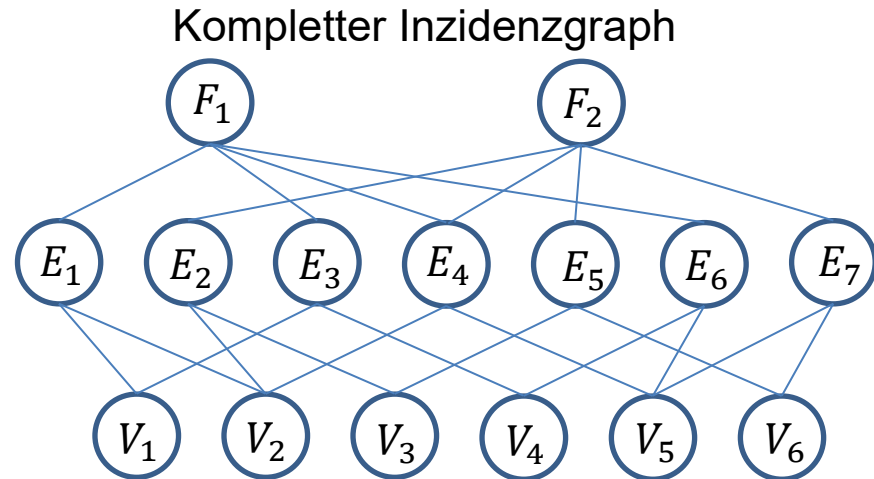
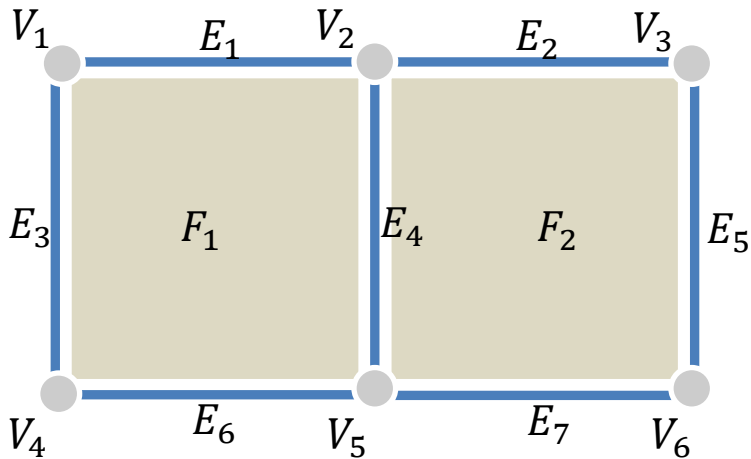
Inzidenzgraph als Repräsentant für ein Graphenbasiertes Datenmodell



$\{V_5, E_4, F_1\}$



Inzidenzgraph als Repräsentant für ein Graphenbasiertes Datenmodell



⇒ alternatives Datenmodell für topologische Beziehungen

Relationales Datenmodell:

- Entitäten als auch deren Beziehungen werden in Tabellen (sogenannten *Relationen*) hinterlegt
- Jede Zeile der Relation entspricht einem *Tupel*, jede Spalte einem *Attribut*
- Attribute, die zur Identifizierung einzelner Tupel dienen, heissen *Schlüssel-Attribute (keys)*; jedes Tupel einer Relation ist eindeutig durch einen *Primärschlüssel* identifiziert
- *Fremdschlüssel* verweisen auf Tupel einer anderen Relation und können mehrfach auftauchen
- Tupel dürfen sich in einer Relation *nicht wiederholen*. Bedeutung einer Relation darf *nicht von der Reihenfolge* der Attribute oder Tupel *abhängen*.

Relationales Datenmodell:

- Entitäten als auch deren Beziehungen werden in Tabellen (sogenannten *Relationen*) hinterlegt
- Jede Zeile der Relation entspricht einem *Tupel*, jede Spalte einem *Attribut*
- Attribute die zur Identifizierung einzelner Tupel dienen heissen *Schlüssel-Attribute (keys)*;

Polygone				
Polygon #	Formation	Lithologie	Alter	Zeitspanne
1	Shelly Fm.	Kalkstein	Oberkarbon	323.2 – 298.9 MJ
2	Grit Fm.	Sandstein	Oberkarbon	323.2 – 298.9 MJ
3	Slab Fm.	Schiefer	Oberkarbon	323.2 – 298.9 MJ
4	Mount Fm.	Granit	Kreide	145 – 66 MJ
5	Mount Fm.	Granit	Kreide	145 – 66 MJ
6	Volcano Fm.	Tuff	Trias	251.9 – 201.3 MJ
7	Mount Fm.	Granit	Kreide	145 – 66 MJ
8	Shelly Fm.	Kalkstein	Oberkarbon	323.2 – 298.9 MJ
9	Slab Fm.	Schiefer	Oberkarbon	323.2 – 298.9 MJ
10	Shelly Fm.	Kalkstein	Oberkarbon	323.2 – 298.9 MJ

hrfach

f nicht

Relationales Datenmodell:

- Es sollen nach Möglichkeit Redundanzen in den Relationen vermieden werden, u.a. um spätere Erweiterungen oder Änderungen zu vereinfachen. Dazu können Relationen *normiert* werden:
- Numerische Codes für textbasierte Attribute einführen

Polygon #	Formation #	Formation	Polygone		Alter #	Alter	Zeitspanne
			Lithologie #	Lithologie			
1	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 – 298.9 MJ
2	3	Grit Fm.	6	Sandstein	5	Oberkarbon	323.2 – 298.9 MJ
3	4	Slab Fm.	5	Schiefer	5	Oberkarbon	323.2 – 298.9 MJ
4	1	Mount Fm.	2	Granit	8	Kreide	145 – 66 MJ
5	1	Mount Fm.	2	Granit	8	Kreide	145 – 66 MJ
6	5	Volcano Fm.	3	Tuff	7	Trias	251.9 – 201.3 MJ
7	1	Mount Fm.	2	Granit	8	Kreide	145 – 66 MJ
8	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 – 298.9 MJ
9	4	Slab Fm.	5	Schiefer	5	Oberkarbon	323.2 – 298.9 MJ
8	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 – 298.9 MJ

Relationales Datenmodell:

- Es sollen nach Möglichkeit Redundanzen in den Relationen vermieden werden, u.a. um spätere Erweiterungen oder Änderungen zu vereinfachen. Dazu können Relationen *normiert* werden:
- Numerische Codes für textbasierte Attribute einführen
- **1. Normalform:**
 - alle Attributwerte müssen *atomar* sein, d.h. sie bestehen nur aus einem einzigen Wert
 - die Relation muss *frei von Wiederholungsgruppen* sein, d.h. verschiedene Attribute dürfen keine gleiche oder gleichartige Bedeutung haben (numerische Codierung ausgenommen)

Polygone								
Polygon #	Formation #	Formation	Lithologie #	Lithologie	Alter #	Alter	Beginn Zeitspanne	Ende Zeitspanne
1	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ
2	3	Grit Fm.	6	Sandstein	5	Oberkarbon	323.2 MJ	298.9 MJ
3	4	Slab Fm.	5	Schiefer	5	Oberkarbon	323.2 MJ	298.9 MJ
4	1	Mount Fm.	2	Granit	8	Kreide	145 MJ	66 MJ
5	1	Mount Fm.	2	Granit	8	Kreide	145 MJ	66 MJ
6	5	Volcano Fm.	3	Tuff	7	Trias	251.9 MJ	201.3 MJ
7	1	Mount Fm.	2	Granit	8	Kreide	145 MJ	66 MJ
8	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ
9	4	Slab Fm.	5	Schiefer	5	Oberkarbon	323.2 MJ	298.9 MJ
8	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ

Relationales Datenmodell:

- Es sollen nach Möglichkeit Redundanzen in den Relationen vermieden werden, u.a. um spätere Erweiterungen oder Änderungen zu vereinfachen. Dazu können Relationen *normiert* werden:
- Numerische Codes für textbasierte Attribute einführen
- **1. Normalform:**
 - alle Attributwerte müssen *atomar* sein, d.h. sie bestehen nur aus einem einzigen Wert
 - die Relation muss *frei von Wiederholungsgruppen* sein, d.h. verschiedene Attribute dürfen keine gleiche oder gleichartige Bedeutung haben (numerische Codierung ausgenommen)
- **2. Normalform:**
 - die Relation liegt bereits in 1. Normalform vor
 - jedes Nicht-Schlüsselattribut muss von *allen* Primärschlüsseln abhängen (voll funktional abhängig)

Weitere Datenmodelle

Polygone								
Polygon #	Formation #	Formation	Lithologie #	Lithologie	Alter #	Alter	Beginn Zeitspanne	Ende Zeitspanne
1	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ
1-2	3	Grit	6	Sandstein	5	Oberkarbon	323.2 MJ	298.9 MJ
3	4	Slab	5	Schiefer	5	Oberkarbon	323.2 MJ	298.9 MJ
4	1	Mount Fm.	2	Kalkstein	8	Kreide	119.0 MJ	66 MJ
5	1	Mount Fm.	2	Kalkstein	8	Kreide	119.0 MJ	66 MJ
6	5	Volcano Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ
7	1	Mount Fm.	2	Kalkstein	8	Kreide	119.0 MJ	66 MJ
8	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ
9	4	Slab Fm.	5	Schiefer	5	Oberkarbon	323.2 MJ	298.9 MJ
8	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ

Jetzt sind Polygon *und* Formation Primärschlüssel, aber nicht alle Attribute sind voll funktional abhängig von den Primärschlüsseln!

Es müsste eine neue Relation angelegt werden, die nur Formation# als Primärschlüssel hat.

- die Relation muss *hier* von Wiederholungsgruppen sein, d.h. verschiedene Attribute dürfen keine gleiche oder gleichartige Bedeutung haben (numerische Codierung ausgenommen)
- **2. Normalform:**
 - die Relation liegt bereits in 1. Normalform vor
 - jedes Nicht-Schlüsselattribut muss von *allen* Primärschlüsseln abhängen (voll funktional abhängig)

Relationales Datenmodell:

- Es sollen nach Möglichkeit Redundanzen in den Relationen vermieden werden, u.a. um spätere Erweiterungen oder Änderungen zu vereinfachen. Dazu können Relationen *normiert* werden:
- **3. Normalform:**
 - die Relation liegt bereits in 2. Normalform vor
 - Alle Nicht-Schlüsselattribut sind untereinander unabhängig (d.h. sie müssen *nicht-transitiv* bzgl. dem Primärschlüssel sein)



Polygone								
Polygon #	Formation #	Formation	Lithologie #	Lithologie	Alter #	Alter	Beginn Zeitspanne	Ende Zeitspanne
1	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ
2	3	Grit Fm.	6	Sandstein	5	Oberkarbon	323.2 MJ	298.9 MJ
3	4	Slab Fm.	5	Schiefer	5	Oberkarbon	323.2 MJ	298.9 MJ
4	1	Mount Fm.	2	Granit	8	Kreide	145 MJ	66 MJ
5	1	Mount Fm.	2	Granit	8	Kreide	145 MJ	66 MJ
6	5	Volcano Fm.	3	Tuff	7	Trias	251.9 MJ	201.3 MJ
7	1	Mount Fm.	2	Granit	8	Kreide	145 MJ	66 MJ
8	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ
9	4	Slab Fm.	5	Schiefer	5	Oberkarbon	323.2 MJ	298.9 MJ
8	2	Shelly Fm.	7	Kalkstein	5	Oberkarbon	323.2 MJ	298.9 MJ

Formation ist nicht-transitiv von *Polygon* abhängig, aber *Lithologie* nicht \Rightarrow Relation aufteilen

Weitere Datenmodelle

Polygone	
Polygon #	Formation #
1	2
2	3
3	4
4	1
5	1
6	5
7	1
8	2
9	4
8	2

Formation			
Formation #	Formation	Lithologie #	Alter #
1	Mount Fm.	2	8
2	Shelly Fm.	7	5
3	Grit Fm.	6	5
4	Slab Fm.	5	5
5	Volcano Fm.	3	7

Lithologie	
Lithologie #	Lithologie
2	Granit
3	Tuff
5	Schiefer
6	Sandstein
7	Kalkstein

Alter			
Alter #	Alter	Beginn Zeitspanne	Ende Zeitspanne
5	Oberkarbon	323.2 MJ	298.9 MJ
7	Trias	251.9 MJ	201.3 MJ
8	Kreide	145 MJ	66 MJ

Relationales Datenmodell:

- Es sollen nach Möglichkeit Redundanzen in den Relationen vermieden werden, u.a. um spätere Erweiterungen oder Änderungen zu vereinfachen. Dazu können Relationen *normiert* werden:
- **3. Normalform:**
 - die Relation liegt bereits in 2. Normalform vor
 - Alle Nicht-Schlüsselattribut sind untereinander unabhängig (d.h. sie müssen *nicht-transitiv* bzgl. dem Primärschlüssel sein)
- Weiter Normalformen können hinzugefügt werden, aber die 3. Normalform reicht aus, um sinnvoll Operationen der relationalen Algebra wie **Vereinigungen**, **Differenzen**, **Durchschnitte**, **karthesische Produkte**, **Projektionen**, **Selektion** von Relationen zu definieren.
- Änderungen und Ergänzungen können ebenfalls einfach vorgenommen werden

Weitere Datenmodelle

Table 2-6. Editing the relation shown in Table 2-4 B before normalization. It has been discovered that the Mount Fm. is a gabbro not a granite (!). Even in this very trivial example it can be seen that the change must be repeated exactly the same in three places in the unnormalized table. Change lithology # and lithology each time the Mount Fm. occurs, see items in bold italics.

Polygon #	Formation #	Formation Name	Lithology #	Lithology	Age #	Age
1	2	Shelly Fm.	7	Limestone	5	Pennsylvanian
2	3	Grit Fm.	6	Sandstone	5	Pennsylvanian
3	4	Slab Fm.	5	Shale	5	Pennsylvanian
4	<u>1</u>	Mount Fm.	<u>4</u>	<u>Gabbro</u>	8	Cretaceous
5	<u>1</u>	Mount Fm.	<u>4</u>	<u>Gabbro</u>	8	Cretaceous
6	5	Volcano Fm.	3	Tuff	7	Triassic
7	<u>1</u>	Mount Fm.	<u>4</u>	<u>Gabbro</u>	8	Cretaceous
8	<u>2</u>	Shelly Fm.	7	Limestone	5	Pennsylvanian
9	4	Slab Fm.	5	Shale	1	Pennsylvanian
10	2	Shelly Fm.	7	Limestone	5	Pennsylvanian

Table 2-7. After normalization, the POLYGON and AGE relations remain unchanged, and two small changes are needed in the FORMATION and LITHOLOGY relations. Suppose this change is made in a real situation for a geological map containing several thousand polygons. The editing changes would still be the same as shown here, because the POLYGON relation requires no modification.

FORMATION relation (1 change shown in italics)

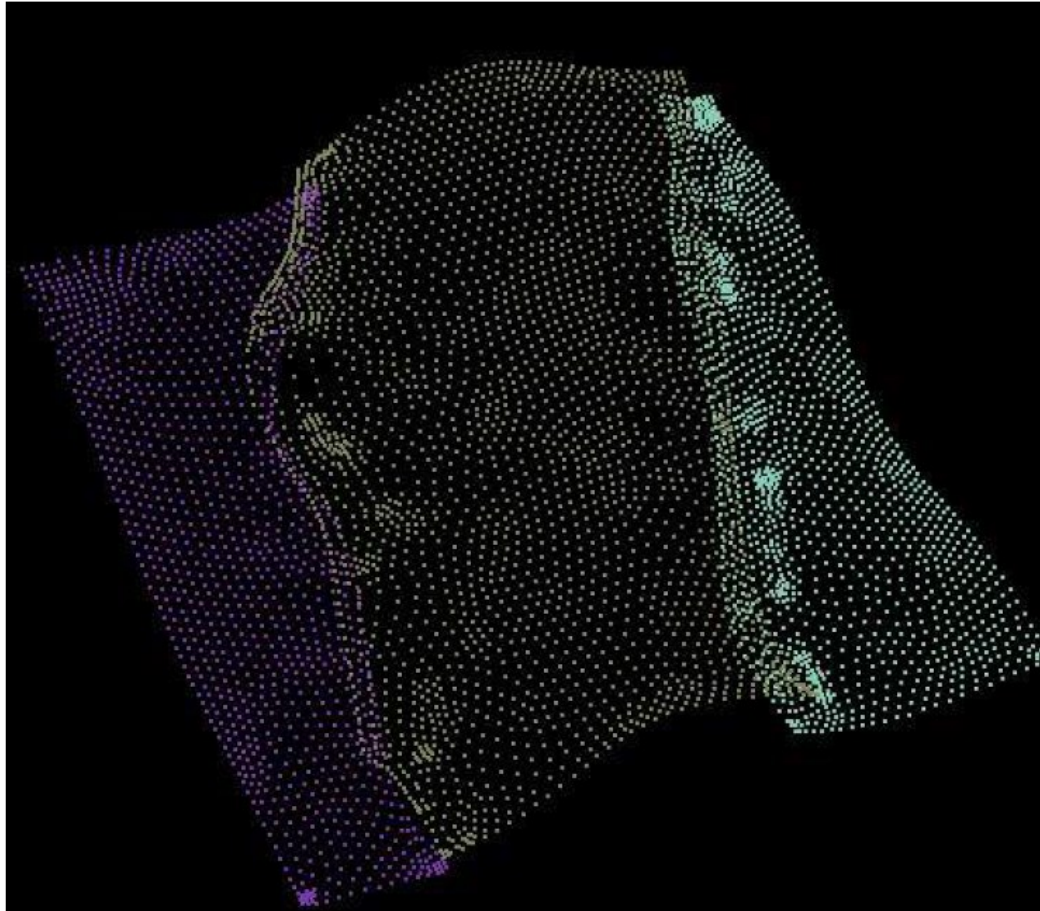
Formation #	Formation Name	Lithology #	Age #
1	Mount Fm.	<u>4</u>	8
2	Shelly Fm.	7	5
3	Grit Fm.	6	5
4	Slab Fm.	5	5
5	Volcano Fm.	3	7

LITHOLOGY relation
(1 insertion shown in italics)

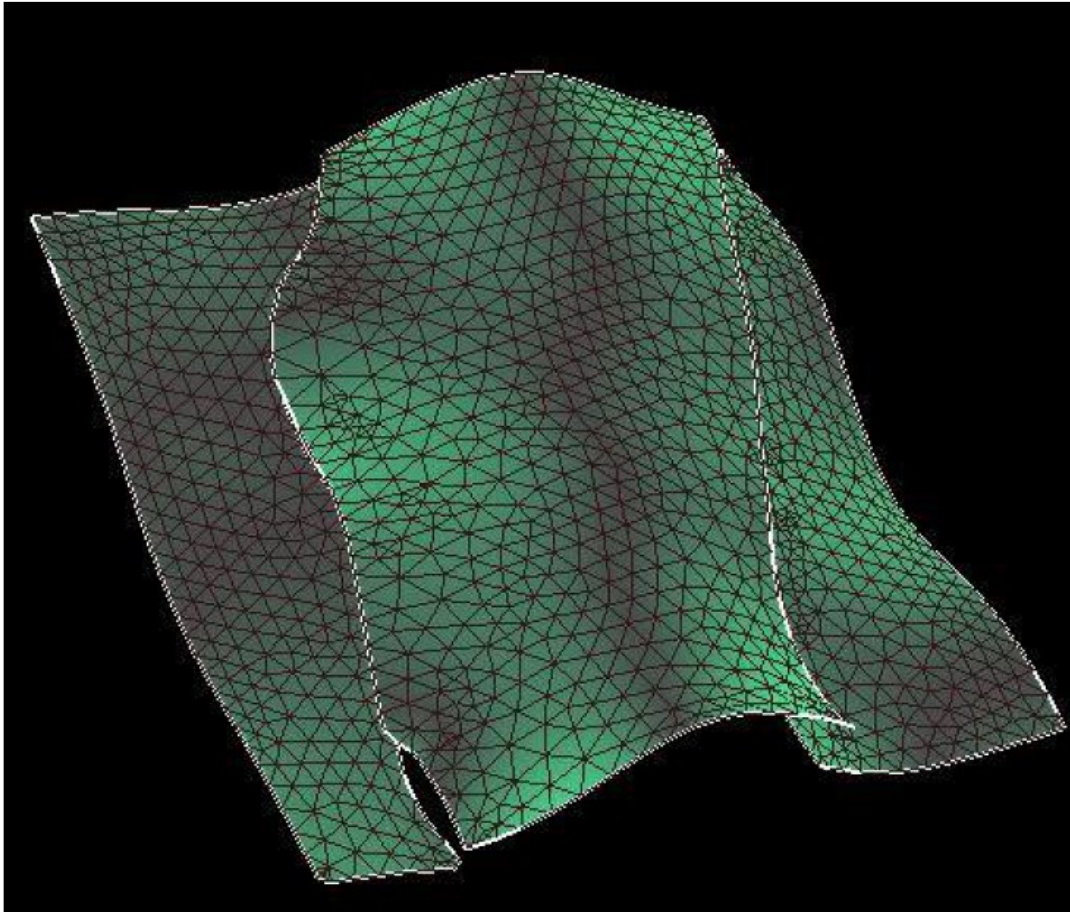
Lithology #	Lithology
2	Granite
3	Tuff
<u>4</u>	<u>Gabbro</u>
5	Shale
6	Sandstone
7	Limestone

1. Definitionen, Funktionen, Anwendungen (Vorlesung 1)
2. Koordinatensysteme und -transformationen (Vorlesung 2+3)
3. Räumliche Datenmodellierung (Vorlesung 4 - 6)
- 4. Vermaschungen**
5. Räumliche Interpolation
6. Transformationen, Filtermethoden, Sonstiges

Zerlegungen und Triangulierung (Partition and Triangulation)

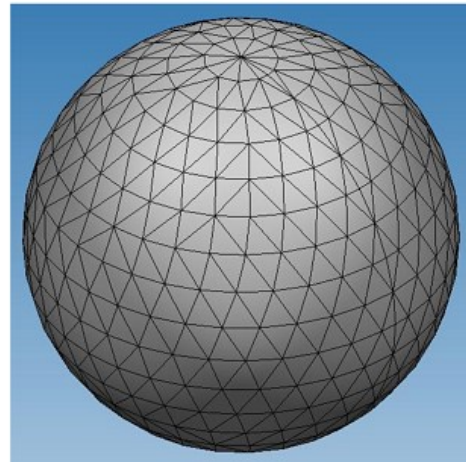
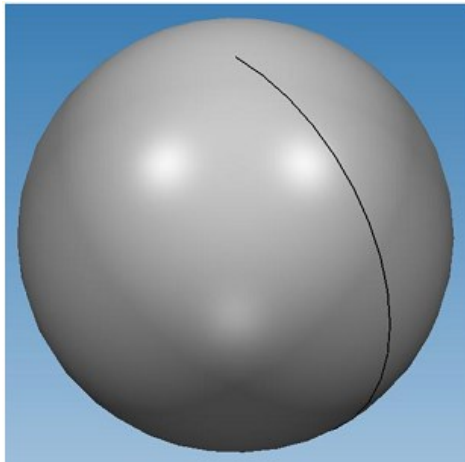


Zerlegungen und Triangulierung (Partition and Triangulation)



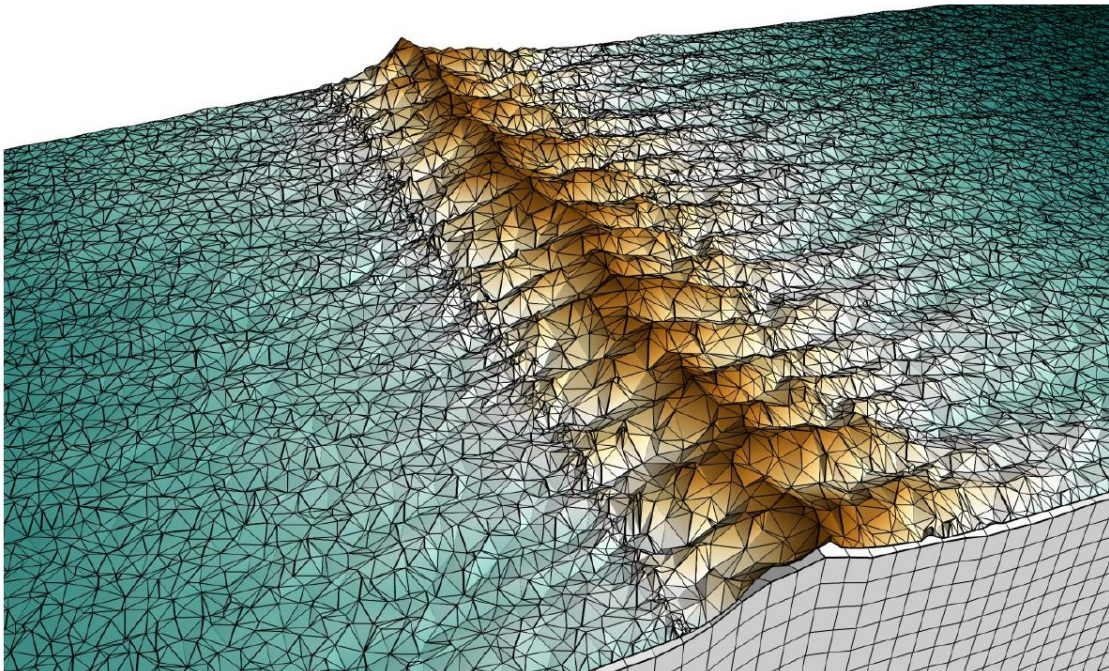
Zerlegungen und Triangulierung (Partition and Triangulation)

Im Bereich “computer aided geometric design” oder “computational geometry” bezieht sich der Terminus “Zerlegung” auf jedes beliebige Verfahren, um ein beliebiges n -dimensionales Objekt in eine Menge benachbarter polytopaler* n -dimensionaler Zellen zu zerlegen.



Zerlegungen und Triangulierung (Partition and Triangulation)

Im Kontext von GIS werden Zerlegungen verwendet, wenn ein Geoobjekt über seine Bestandteile beschrieben werden soll. Diesen Bestandteilen können dadurch eigene Attribute zugewiesen werden.



Zerlegungen und Triangulierung (Partition and Triangulation)

Im Kontext von GIS werden Zerlegungen verwendet, wenn ein beliebiges Geobjekt über seine Bestandteile beschrieben werden soll. Diesen Bestandteilen können dadurch eigene Attribute zugewiesen werden.

Auf diesen Bestandteilen lassen sich Berechnungen zumeist sehr viel einfacher durchführen, als auf dem Objekten selbst. Eine digitale visuelle Darstellung eines beliebigen Objektes ist ohne eine Vermaschung des Objektes in einfache diskrete Zellen oft nicht möglich.

Zerlegungen sind eine spezielle Form des Vektormodells.

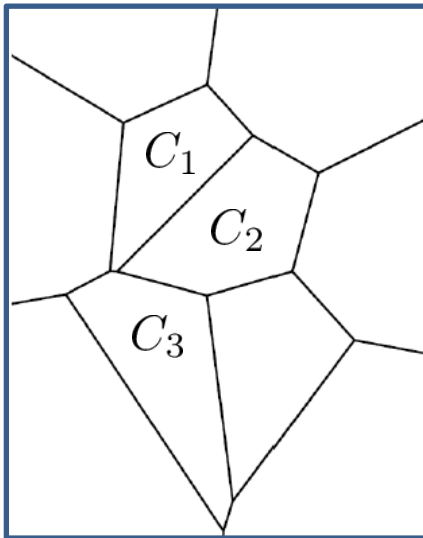
Zerlegung / Vermaschung

Eine **Zerlegung** (*partition*) einer Region $\Omega \subset \mathbb{R}^n$ ist eine Menge offener Zellen $C_i \subset \mathbb{R}^n, i = 1, \dots, N$ mit $C_i \cap C_j = \emptyset, i \neq j$, so dass gilt

$$\Omega = \bigcup_{i=1, \dots, N} C_i$$

Eine solche Zerlegung wird auch als Vermaschung (*tesselation*) bezeichnet.

Die „*Offenheit*“ der Zellen ist für eine exakte mathematische Beschreibung notwendig ...



Voronoi Vermaschung

Gegeben sei eine Punktmenge $P = \{p_i \in \mathbb{R}^n : i = 1, \dots, N\}$. Eine **Voronoi-Zelle** V_i beschreibt die Menge

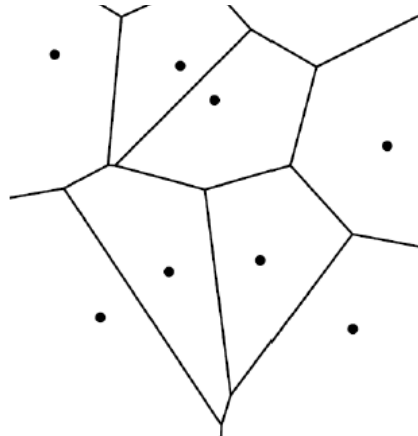
$$V_i = \{x \in \mathbb{R}^n : |x - p_i| < |x - p_j|, i \neq j\}.$$

Eine solche Zelle ist eine spezielle Form eines Polytops (z.B. ebener Polyeder in 3D od. ebenes Polygon in 2D). Dabei bilden die Zellen $C_i = V_i$ „annähernd“* eine Zerlegung des \mathbb{R}^n .

*Warum nur „annähernd“?

Formal müssten die M Grenzkanten aller Zellen mit als zusätzliche Zellen $C_i, i = N + 1, \dots, N + M$ und alle K Knoten, an denen sich Kanten treffen, als zusätzliche Zellen $C_i, i = N + M + 1, \dots, N + M + K$ mit einbezogen werden.

Diese Zellen werden durch die „Offenheits“-Bedingung nicht automatisch erfasst.



Voronoi Vermaschung

Gegeben sei eine Punktmenge $P = \{p_i \in \mathbb{R}^n: i = 1, \dots, N\}$. Eine **Voronoi-Zelle** V_i beschreibt die Menge

$$V_i = \{x \in \mathbb{R}^n: |x - p_i| < |x - p_j|, i \neq j\}.$$

Eine solche Zelle ist eine spezielle Form eines Polytops (z.B. ebener Polyeder in 3D od. ebenes Polygon in 2D). Dabei bilden die Zellen $C_i = V_i$ „annähernd“* eine Zerlegung des \mathbb{R}^n .

Der **Bisector** zwischen zwei Punkten p_i, p_j ist gegeben durch

$$B_{i,j} = \{x \in \mathbb{R}^n: |x - p_i| = |x - p_j|, i \neq j\}$$

Und teilt den \mathbb{R}^n in zwei Halbräume. $B_{i,j}$ repräsentiert die Grenze des Halbraums

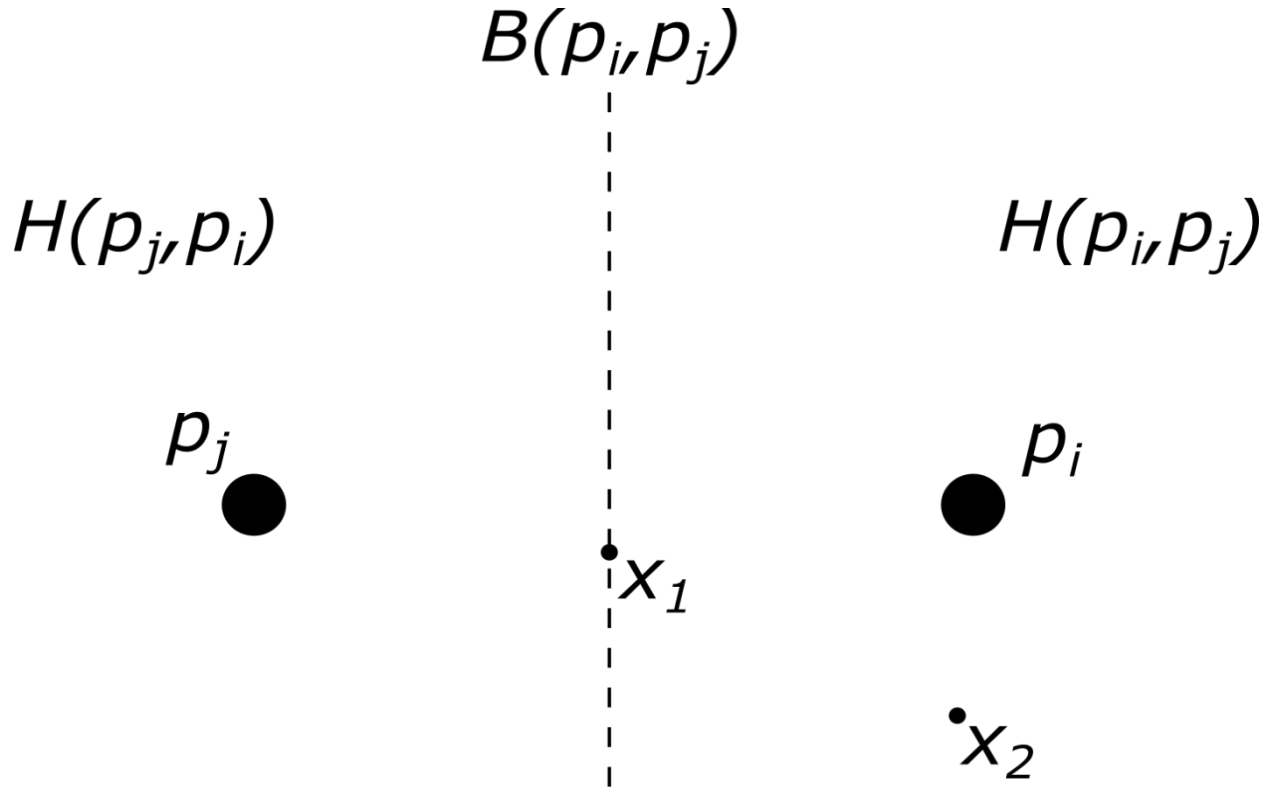
$$H_{i,j} = \{x \in \mathbb{R}^n: |x - p_i| < |x - p_j|, i \neq j\}.$$

Jede Voronoi-Zelle kann über die Schnittmenge

$$V_i = \bigcup_{j=1, \dots, N; i \neq j} H_{i,j}$$

repräsentiert werden.

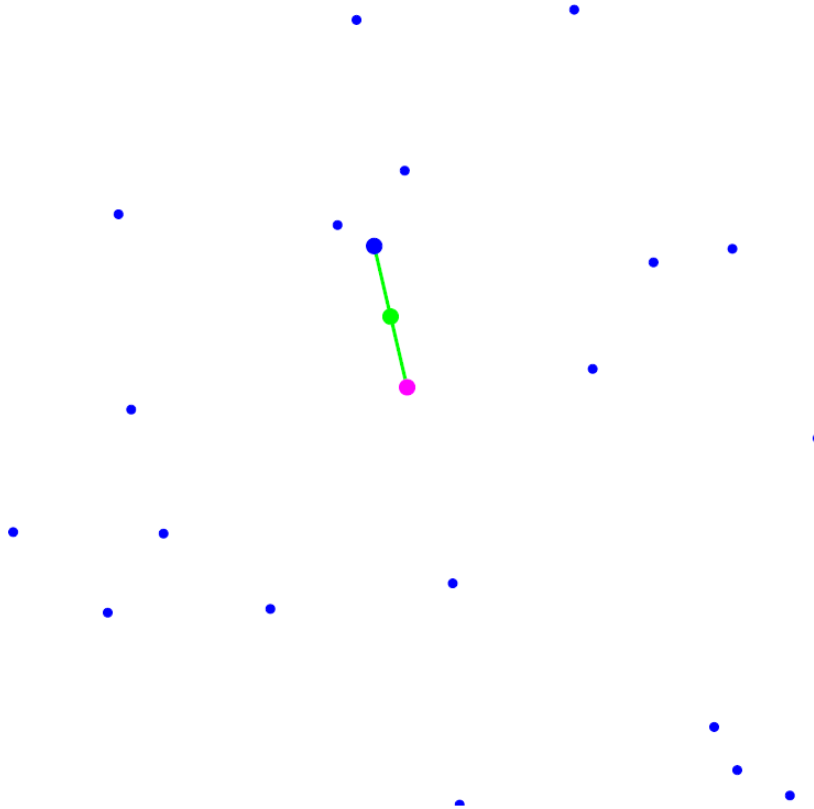
Voronoi Vermaschung



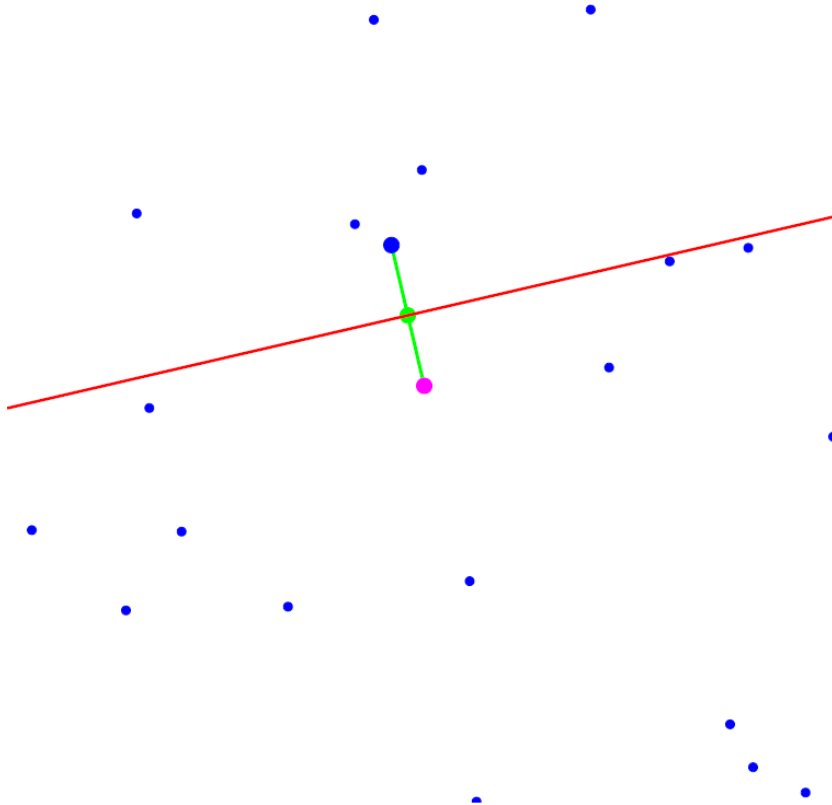
$$|x_1 - p_i| = |x_1 - p_j| \rightarrow x_1 \in B(p_i, p_j)$$

$$|x_2 - p_i| < |x_2 - p_j| \rightarrow x_2 \in H(p_i, p_j)$$

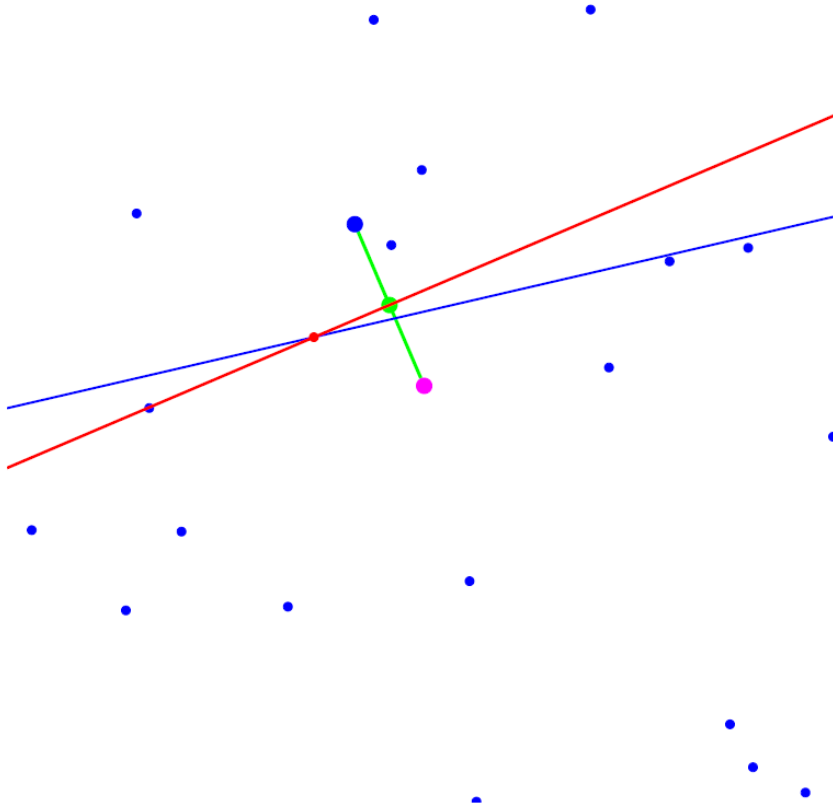
Beispiel: Voronoi Vermaschung in 2-D



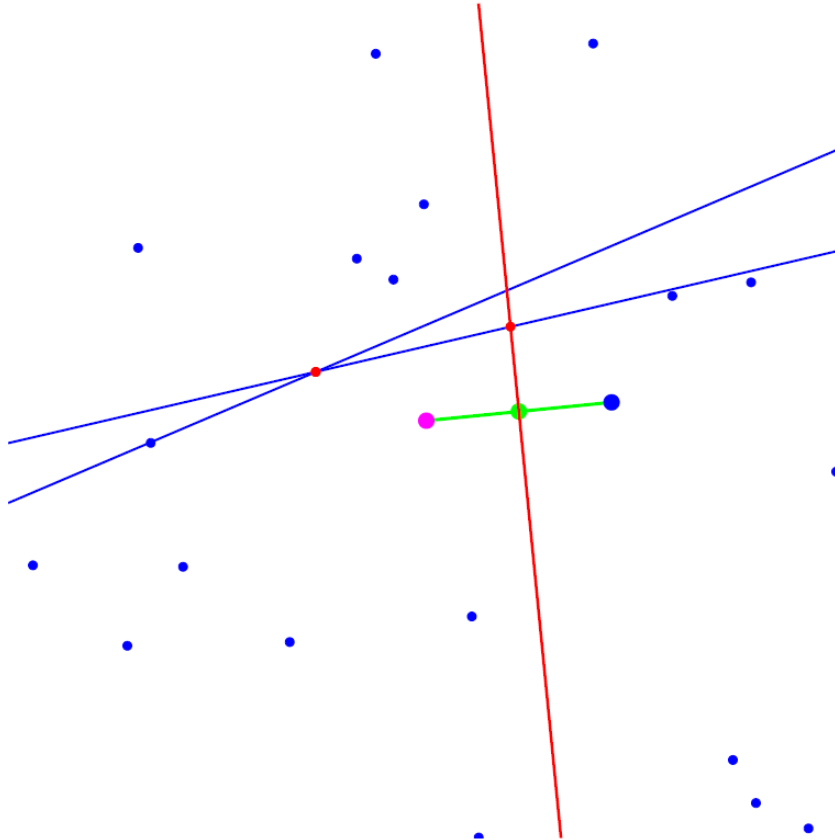
Beispiel: Voronoi Vermaschung in 2-D



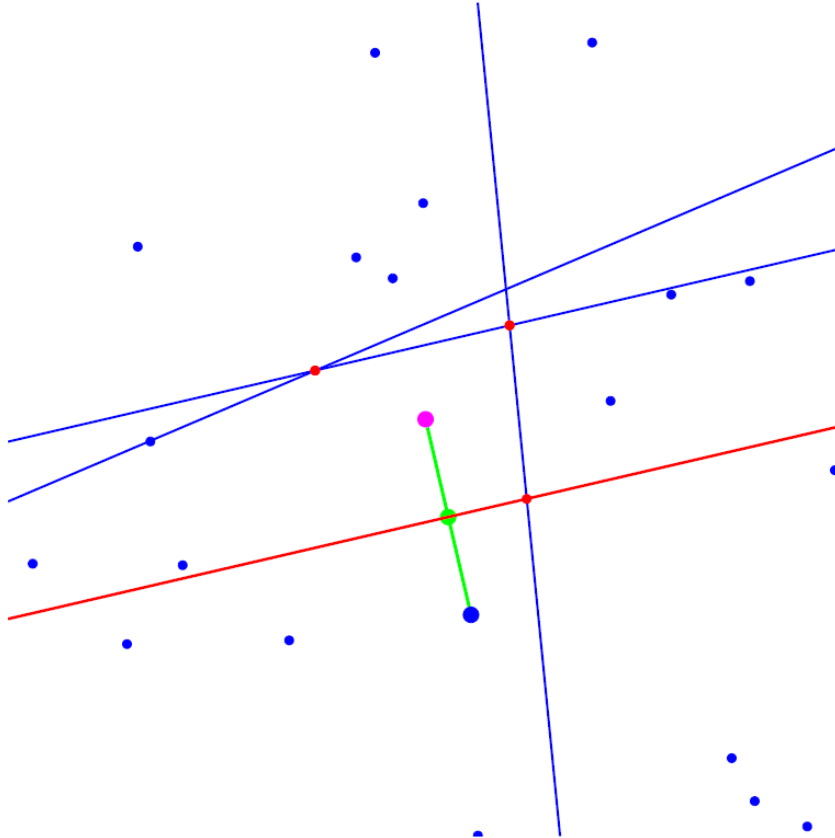
Beispiel: Voronoi Vermaschung in 2-D



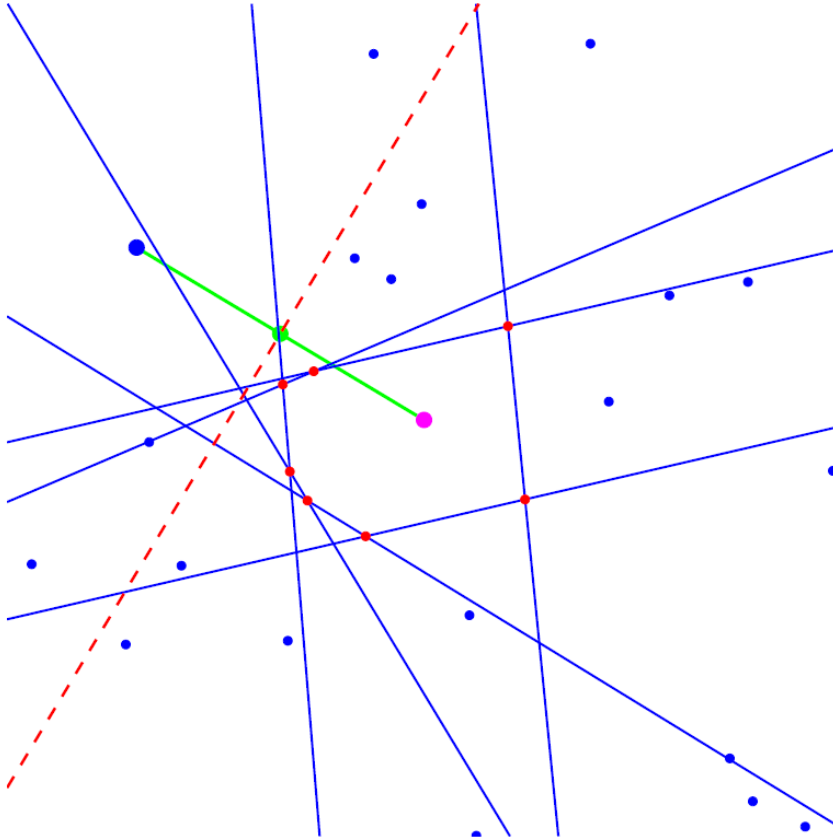
Beispiel: Voronoi Vermaschung in 2-D



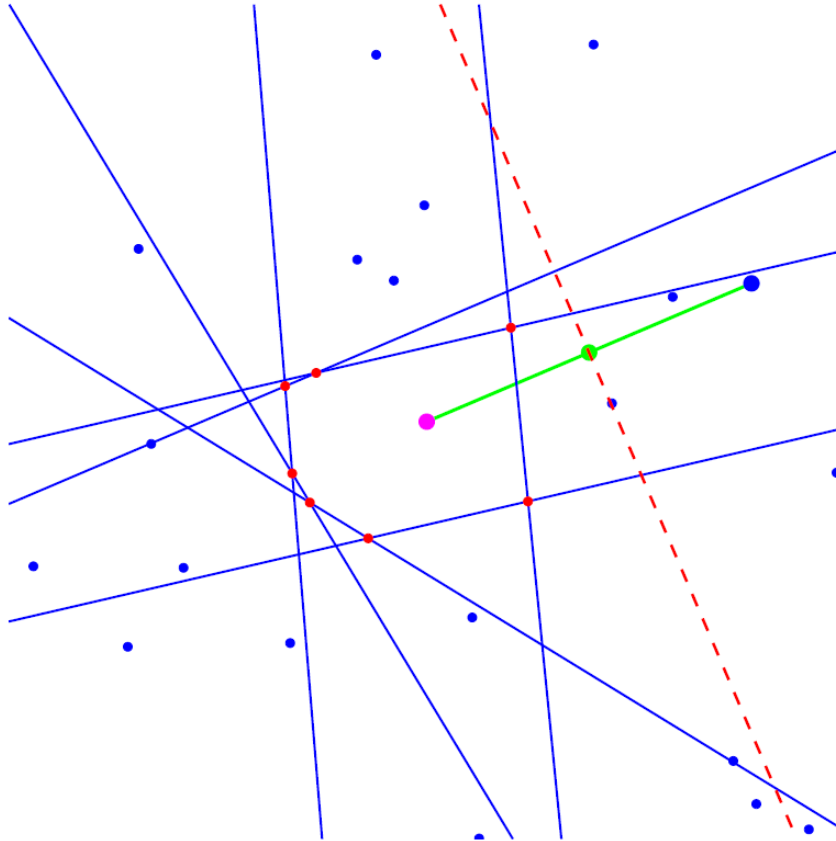
Beispiel: Voronoi Vermaschung in 2-D



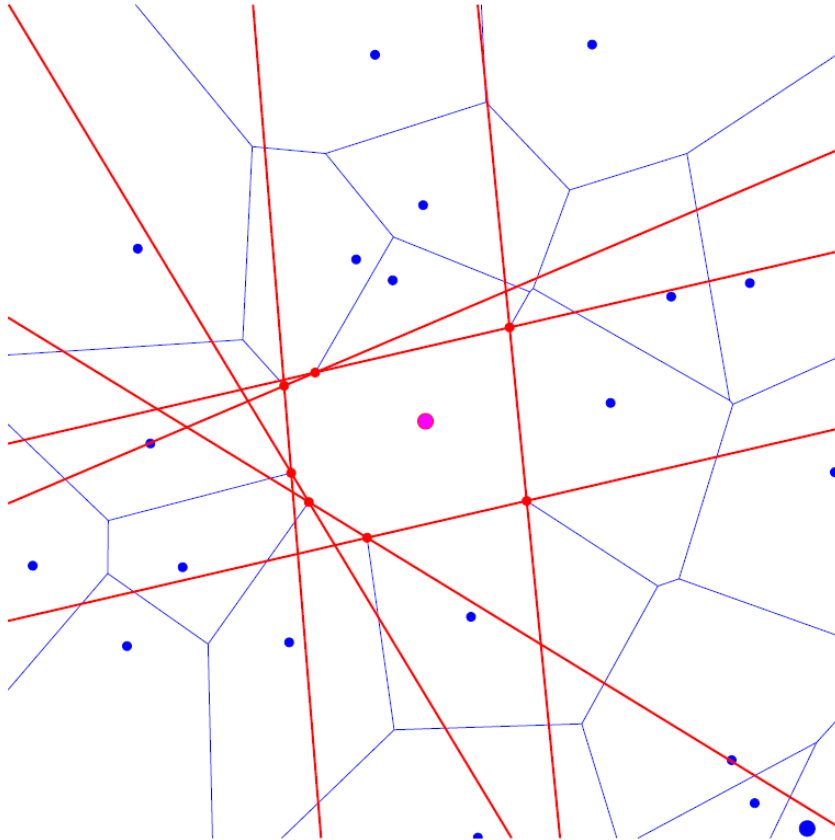
Beispiel: Voronoi Vermaschung in 2-D



Beispiel: Voronoi Vermaschung in 2-D



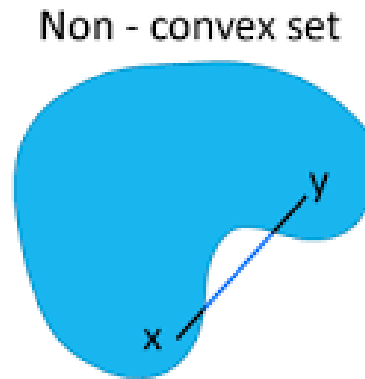
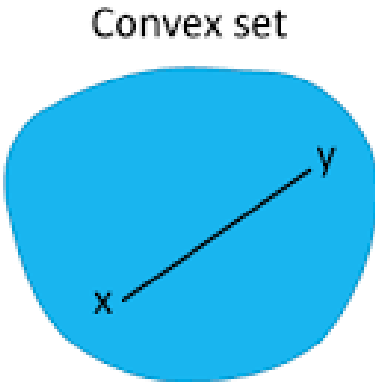
Beispiel: Voronoi Vermaschung in 2-D



Voronoi Vermaschung

Lemma (Eigenschaften). Sei $\mathcal{V} = \{V_1, \dots, V_N\}$ ist die Voronoi-Vermaschung der Punktmenge $P = \{P_1, \dots, P_N\} \subset \mathbb{R}^{n*}$.
Dann gelten folgende Eigenschaften:

- a) Jede Zelle V_i ist konvex (d.h. wenn V_i zwei Punkte x, y enthält, muss auch die komplette Strecken zwischen x und y enthalten sein).



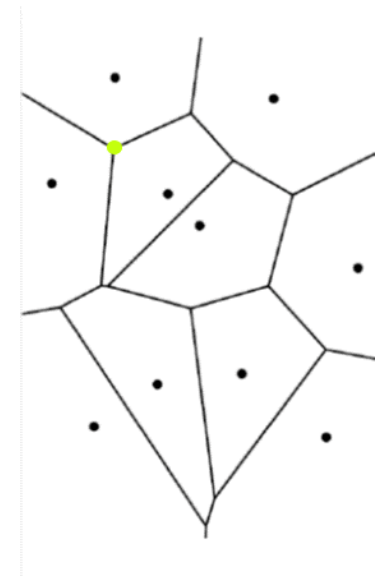
**“in general position“:*
es dürfen keine $n + 1$ Punkte
auf einer $n - 1$ -Kugel liegen

Voronoi Vermaschung

Lemma (Eigenschaften). Sei $\mathcal{V} = \{V_1, \dots, V_N\}$ ist die Voronoi-Vermaschung der Punktmenge $P = \{P_1, \dots, P_N\} \subset \mathbb{R}^n$.

Dann gelten folgende Eigenschaften:

- a) Jede Zelle V_i ist konvex (d.h. wenn V_i zwei Punkte x, y enthält, muss auch die komplette Strecken zwischen x und y enthalten sein).
- b) Jeder Vertex einer Voronoi-Vermaschung liegt auf der Grenze von exakt $n + 1$ verschiedenen Zellen $V_i \in \mathcal{V}$

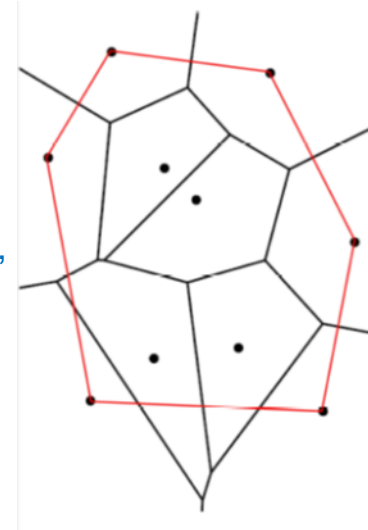


Voronoi Vermaschung

Lemma (Eigenschaften). Sei $\mathcal{V} = \{V_1, \dots, V_N\}$ ist die Voronoi-Vermaschung der Punktmenge $P = \{P_1, \dots, P_N\} \subset \mathbb{R}^n$.

Dann gelten folgende Eigenschaften:

- Jede Zelle V_i ist konvex (d.h. wenn V_i zwei Punkte x, y enthält, muss auch die komplette Strecken zwischen x und y enthalten sein).
- Jeder Vertex einer Voronoi-Vermaschung liegt auf der Grenze von exakt $n + 1$ verschiedenen Zellen $V_i \in \mathcal{V}$.
- Eine Zelle V_i ist *unbegrenzt* (unendlich ausgedehnt), wenn p_i auf der konvexen Hülle $\mathcal{C}(P)$ von P liegt. Diese ist das kleinste konvexe n -Polytop, welches P umfasst.

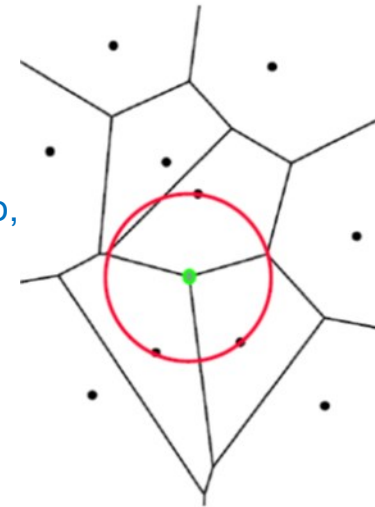


Voronoi Vermaschung

Lemma (Eigenschaften). Sei $\mathcal{V} = \{V_1, \dots, V_N\}$ ist die Voronoi-Vermaschung der Punktmenge $P = \{p_1, \dots, p_N\} \subset \mathbb{R}^n$.

Dann gelten folgende Eigenschaften:

- Jede Zelle V_i ist konvex (d.h. wenn V_i zwei Punkte x, y enthält, muss auch die komplette Strecken zwischen x und y enthalten sein).
- Jeder Vertex einer Voronoi-Vermaschung liegt auf der Grenze von exakt $n + 1$ verschiedenen Zellen $V_i \in \mathcal{V}$.
- Eine Zelle V_i ist *unbegrenzt* (unendlich ausgedehnt), wenn p_i auf der konvexen Hülle $\mathcal{C}(P)$ von P liegt. Diese ist das kleinste konvexe n -Polytop, welches P umfasst.
- Sei X ein Vertex der zu den Zellen $V_{i_1}, \dots, V_{i_{n+1}}$ gehört. Dann ist X der Mittelpunkt einer n -Kugel durch die Punkte $p_{i_1}, \dots, p_{i_{n+1}}$. Es liegt kein anderer Punkt von P innerhalb dieser n -Kugel.



Voronoi Zerlegung – Tabellarische Darstellung 2D

Es muss in irgendeiner Form beschrieben werden, wie die Voronoizellen geometrisch definiert sind:

Tabelle 17: nodes

nodes		
nodeID	x	y
1	x_1	y_1
\vdots	\vdots	\vdots
max. Anzahl der Nodes n	x_n	y_n

Tabelle 18: voronoiCells

voronoiCells	
cellID	nodeList
1	nodeID_1, nodeID_2, nodeID_3, ...
\vdots	\vdots
max. Anzahl der Zellen m	nodeID_i, nodeID_j, nodeID_k, ...

Tabelle 19: cellCenters

cellCenters		
cellID	x_center	y_center
1	x_1	y_1
\vdots	\vdots	\vdots
max. Anzahl der zellen m	x_m	y_m

Interpolation mittels Voronoi Zerlegung: Nearest Neighbor Interpolation

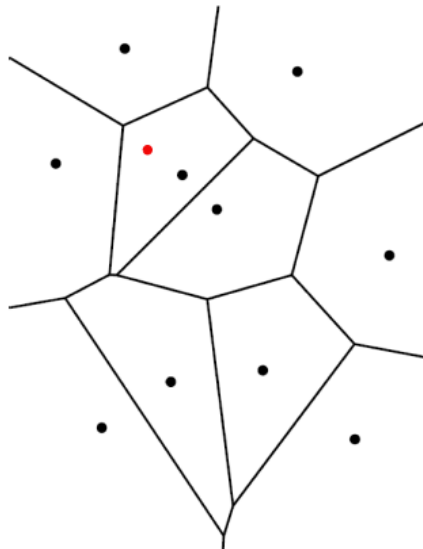
Gegen sein eine Punktmenge $P = \{p_1, \dots, p_N\} \subset \mathbb{R}^n$ und ein Messwert/Datum h_i wird jedem Punkt p_i zugewiesen.

Nearest-Neighbor-Interpolation: Gegen sein ein beliebiger Punkt $x \in \mathbb{R}^n$, an dem kein Messwert $h(x)$ bekannt ist.

Dieser kann mittels der zugehörigen Voronoi-Vermaischung $\mathcal{V}(P)$ vorhergesagt werden:

- x kann nur in einer Voronoi-Zelle liegen $x \in V_i$
- $h(x) = h_i$

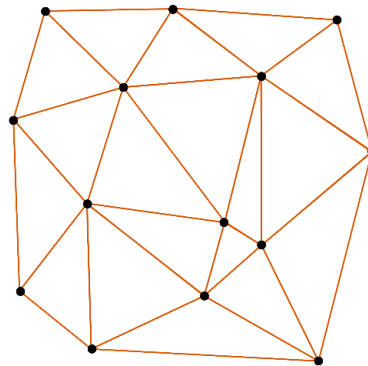
Wie gut wäre eine solche Vorhersage?



Vermaschungen als Graphen

Ein Graph $\mathcal{G} = (V, E)$ ist ein Tupel von abstrakten Knoten/Vertices $V = \{v_1, \dots, v_N\}$ und Kanten $E = \{e_1, \dots, e_N\}$, von denen jede zwei Knoten verbindet. Ein Graph wird als „einfach“ (*simple*) bezeichnet, wenn Folgendes gilt:

1. Zwei Knoten werden durch maximal eine Kante verbunden,
2. Kanten haben keine Orientierung
3. Es existieren keine Schleifen (*loops*, Kanten mit identischem Start- und Endpunkt)

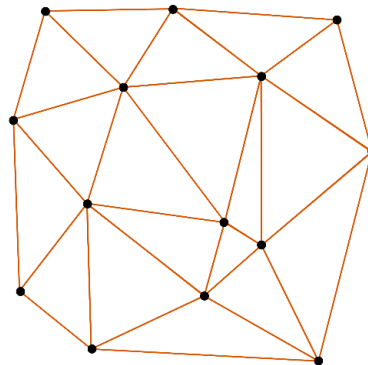


Vermaschungen als Graphen

Ein Graph $\mathcal{G} = (V, E)$ ist ein Tupel von abstrakten Knoten/Vertices $V = \{v_1, \dots, v_N\}$ und Kanten $E = \{e_1, \dots, e_N\}$, von denen jede zwei Knoten verbindet. Ein Graph wird als „einfach“ (*simple*) bezeichnet, wenn Folgendes gilt:

1. Zwei Knoten werden durch maximal eine Kante verbunden,
2. Kanten haben keine Orientierung
3. Es existieren keine Schleifen (*loops*, Kanten mit identischem Start- und Endpunkt)

Knoten entsprechen 0-Zellen, Kanten 1-Zellen und Flächen, begrenzt durch einen Kantenzug, 2-Zellen. Kanten bestimmen die „Nachbarschaft“ von Flächen und Knoten die Nachbarschaft von Kanten.



Vermaschungen als Graphen

Ein Graph kann mittels seiner **Adjazenz-Matrix** $A_V \in \mathbb{R}^{N \times N}$:

$$(A_V)_{i,j} = \begin{cases} 1, & \text{für } i \neq j \text{ und } v_i, v_j \text{ sind über eine Kante verbunden} \\ 0, & \text{andernfalls} \end{cases}.$$

Eine äquivalente Charakterisierung kann über die Inzidenz-Matrix $I_{VE} \in \mathbb{R}^{N \times N}$ erfolgen:

$$(I_{VE})_{i,j} \begin{cases} 1, & \text{falls Vertex } v_i \text{ ist Endpunkt von Kante } e_j \\ 0, & \text{andernfalls} \end{cases}.$$

Jede Spalte in I_{VE} eines einfachen Graphen enthält den Wert 1 genau zwei mal.

Vermaschungen als Graphen

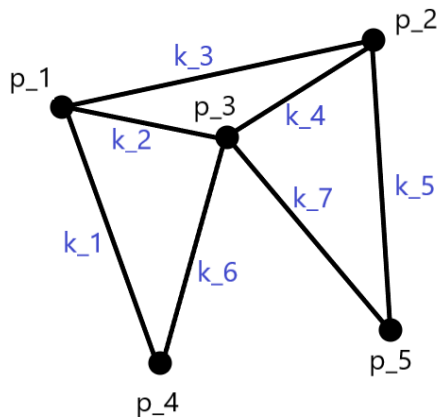
Ein Graph kann mittels seiner **Adjazenz-Matrix** $A_V \in \mathbb{R}^{N \times N}$:

$$(A_V)_{i,j} = \begin{cases} 1, & \text{für } i \neq j \text{ und } v_i, v_j \text{ sind über eine Kante verbunden} \\ 0, & \text{andernfalls} \end{cases}$$

Eine äquivalente Charakterisierung kann über die Inzidenz-Matrix $I_{VE} \in \mathbb{R}^{N \times N}$ erfolgen:

$$(I_{VE})_{i,j} \begin{cases} 1, & \text{falls Vertex } v_i \text{ ist Endpunkt von Kante } e_j \\ 0, & \text{andernfalls} \end{cases}$$

Jede Spalte in I_{VE} eines einfachen Graphen enthält den Wert 1 genau zwei mal.



Adjazenz-Matrix dieses Beispiel-Graphen:

$$A_V = \begin{pmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Vermaschungen als Graphen

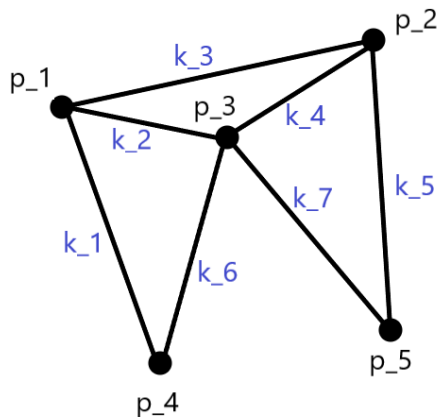
Ein Graph kann mittels seiner **Adjazenz-Matrix** $A_V \in \mathbb{R}^{N \times N}$:

$$(A_V)_{i,j} = \begin{cases} 1, & \text{für } i \neq j \text{ und } v_i, v_j \text{ sind über eine Kante verbunden} \\ 0, & \text{andernfalls} \end{cases}$$

Eine äquivalente Charakterisierung kann über die Inzidenz-Matrix $I_{VE} \in \mathbb{R}^{N \times N}$ erfolgen:

$$(I_{VE})_{i,j} \begin{cases} 1, & \text{falls Vertex } v_i \text{ ist Endpunkt von Kante } e_j \\ 0, & \text{andernfalls} \end{cases}$$

Jede Spalte in I_{VE} eines einfachen Graphen enthält den Wert 1 genau zwei mal.



Inzidenz-Matrix dieses Beispiel-Graphen:

$$I_{VE} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

Vermaschungen als Graphen

Ein Graph kann mittels seiner **Adjazenz-Matrix** $A_V \in \mathbb{R}^{N \times N}$:

$$(A_V)_{i,j} = \begin{cases} 1, & \text{für } i \neq j \text{ und } v_i, v_j \text{ sind über eine Kante verbunden} \\ 0, & \text{andernfalls} \end{cases}.$$

Eine äquivalente Charakterisierung kann über die Inzidenz-Matrix $I_{VE} \in \mathbb{R}^{N \times N}$ erfolgen:

$$(I_{VE})_{i,j} \begin{cases} 1, & \text{falls Vertex } v_i \text{ ist Endpunkt von Kante } e_j \\ 0, & \text{andernfalls} \end{cases}.$$

Jede Spalte in I_{VE} eines einfachen Graphen enthält den Wert 1 genau zwei mal.

Es gilt:

$$(A_V)_{i,j} = \begin{cases} (I_{VE} I_{VE}^T)_{i,j} & \text{für } i \neq j \\ 0, & \text{sonst} \end{cases}$$

Topologie einer Voronoi Vermaschung

Neben Kanten und Vertices, beinhaltet einen 2-D Voronoi-Vermaschung auch die Flächen F_1, \dots, F_K . Die zugehörigen Inzidenz-Matrizen können wie folgt repräsentiert werden:

$$(I_{VE})_{i,j} = \begin{cases} 1, & \text{falls } v_i \text{ ist Endpunkt von } e_j \\ 0, & \text{sonst} \end{cases}$$

$$(I_{EF})_{i,j} = \begin{cases} 1, & \text{falls } e_i \text{ ist Grenzkante von } F_j \\ 0, & \text{sonst} \end{cases}$$

$$(I_{VF})_{i,j} = (I_{VE}I_{EF})_{i,j} = \begin{cases} 1, & \text{falls } v_i \text{ ist Grenzknoten von } F_j \\ 0, & \text{sonst} \end{cases}$$

Topologie einer Voronoi Vermaschung

Neben Kanten und Vertices, beinhaltet eine 2-D Voronoi-Vermaschung auch die Flächen F_1, \dots, F_K . Die zugehörigen Inzidenz-Matrizen können wie folgt repräsentiert werden:

$$(I_{VE})_{i,j} = \begin{cases} 1, & \text{falls } v_i \text{ ist Endpunkt von } e_j \\ 0, & \text{sonst} \end{cases}$$

$$(I_{EF})_{i,j} = \begin{cases} 1, & \text{falls } e_i \text{ ist Grenzkante von } F_j \\ 0, & \text{sonst} \end{cases}$$

$$(I_{VF})_{i,j} = (I_{VE}I_{EF})_{i,j} = \begin{cases} 1, & \text{falls } v_i \text{ ist Grenzknoten von } F_j \\ 0, & \text{sonst} \end{cases}$$

Diese Matrizen beschreiben die Nachbarschaftsbeziehungen zwischen Vertices, Kanten und Flächen. So lässt sich die Nachbarschaft zweier Voronoi-Zellen (Flächen) über folgende Matrix beschreiben:

$$(A_F)_{i,j} = \begin{cases} (I_{EF}^T I_{EF})_{i,j}, & \text{falls } i \neq j \\ 0, & \text{sonst} \end{cases}$$