

# GRK-Meeting — 08.02.2010

Dennis Schieferdecker – *dennis.schieferdecker@kit.edu*

Institut für Theoretische Informatik - Algorithmik II



## Vorstellung — Lehrstuhl Prof. Sanders

### Lage

- Am Fasanengarten 5  
76131 Karlsruhe  
Deutschland
- Gebäude 50.34  
2. Stock, Ostflügel

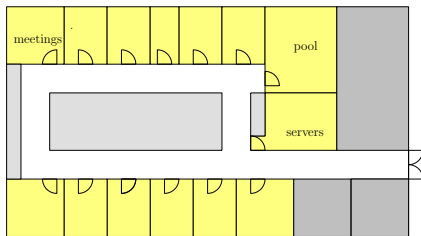


### Forschungsgebiet

- Algorithm Engineering:
  - Entwicklung praktikabler Lösungen mit guten theoretischen Leistungsgarantien,
  - v.a. für diskrete Probleme

### Lage

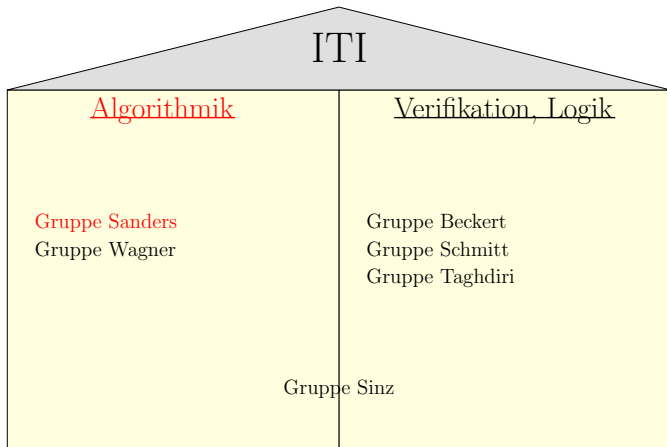
- Am Fasanengarten 5  
76131 Karlsruhe  
Deutschland
- Gebäude 50.34  
2. Stock, Ostflügel

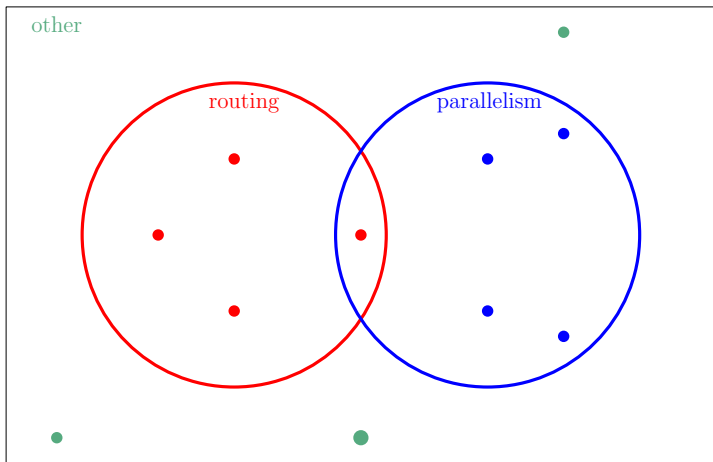


### Forschungsgebiet

- Algorithm Engineering:
  - Entwicklung praktikabler Lösungen mit guten theoretischen Leistungsgarantien,
  - v.a. für diskrete Probleme







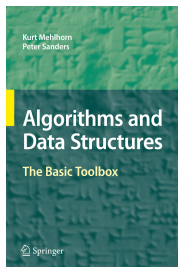
## Professoren

- **Prof. Dr. Peter Sanders**
  - Algorithm Engineering



## Nicht-wissenschaftliche Mitarbeiter

- **Anja Blancani**
  - Sekretärin
- **Norbert Berger**
  - technischer Mitarbeiter



## Professoren

- **Prof. Dr. Peter Sanders**
  - Algorithm Engineering



## Nicht-wissenschaftliche Mitarbeiter

- **Anja Blancani**
  - Sekretärin
- **Norbert Berger**
  - technischer Mitarbeiter



## Professoren

- **Prof. Dr. Peter Sanders**
  - Algorithm Engineering



## Nicht-wissenschaftliche Mitarbeiter

- **Anja Blancani**
  - Sekretärin
- **Norbert Berger**
  - technischer Mitarbeiter



### Wissenschaftliche Mitarbeiter

#### ■ G. Veit Batz

- zeitabhängige Routenplanung

#### ■ Robert Geisberger

- Contraction Hierarchies
- multi-kriterielle Routenplanung

#### ■ Moritz Kobitsch

- energie-effiziente Routenplanung
- MeRegio-Projekt

#### ■ Dennis Luxen

- Verkehrssimulation
- Verteilte Routenplanung



### Wissenschaftliche Mitarbeiter

#### ■ G. Veit Batz

- zeitabhängige Routenplanung

#### ■ Robert Geisberger

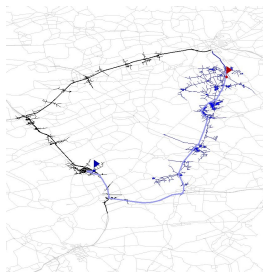
- Contraction Hierarchies
- multi-kriterielle Routenplanung

#### ■ Moritz Kobitsch

- energie-effiziente Routenplanung
- MeRegio-Projekt

#### ■ Dennis Luxen

- Verkehrssimulation
- Verteilte Routenplanung



### Wissenschaftliche Mitarbeiter

#### ■ G. Veit Batz

- zeitabhängige Routenplanung

#### ■ Robert Geisberger

- Contraction Hierarchies
- multi-kriterielle Routenplanung

#### ■ Moritz Kobitsch

- energie-effiziente Routenplanung
- MeRegio-Projekt

#### ■ Dennis Luxen

- Verkehrssimulation
- Verteilte Routenplanung





### Wissenschaftliche Mitarbeiter

#### ■ G. Veit Batz

- zeitabhängige Routenplanung

#### ■ Robert Geisberger

- Contraction Hierarchies
- multi-kriterielle Routenplanung

#### ■ Moritz Kobitsch

- energie-effiziente Routenplanung
- MeRegio-Projekt

#### ■ Dennis Luxen

- Verkehrssimulation
- Verteilte Routenplanung



### Wissenschaftliche Mitarbeiter

#### ■ Jochen Speck

- Scheduling

#### ■ Vitaly Osipov

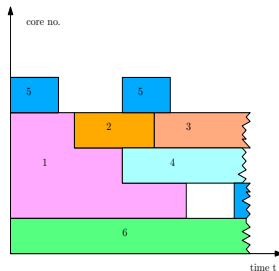
- Algorithmen auf Grafikkarten (CUDA)
- Sortierung, Partitionierung

#### ■ Dennis Schieferdecker

- Algorithmen für Sensornetzwerke

#### ■ Johannes Singler

- Parallele Algorithmenbibliotheken  
(z. B. STL, für Externspeicher, für Geometrie)
- Sortieren riesiger Datenmengen



### Wissenschaftliche Mitarbeiter

#### ■ Jochen Speck

- Scheduling

#### ■ Vitaly Osipov

- Algorithmen auf Grafikkarten (CUDA)
- Sortierung, Partitionierung

#### ■ Dennis Schieferdecker

- Algorithmen für Sensornetzwerke

#### ■ Johannes Singler

- Parallele Algorithmenbibliotheken  
(z. B. STL, für Externspeicher, für Geometrie)
- Sortieren riesiger Datenmengen



### Wissenschaftliche Mitarbeiter

#### ■ Jochen Speck

- Scheduling

#### ■ Vitaly Osipov

- Algorithmen auf Grafikkarten (CUDA)
- Sortierung, Partitionierung

#### ■ Dennis Schieferdecker

- Algorithmen für Sensornetzwerke

#### ■ Johannes Singler

- Parallele Algorithmenbibliotheken  
(z. B. STL, für Externspeicher, für Geometrie)
- Sortieren riesiger Datenmengen



### Wissenschaftliche Mitarbeiter

#### ■ Jochen Speck

- Scheduling

#### ■ Vitaly Osipov

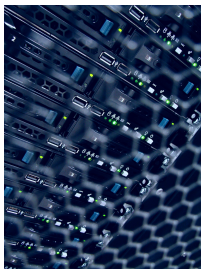
- Algorithmen auf Grafikkarten (CUDA)
- Sortierung, Partitionierung

#### ■ Dennis Schieferdecker

- Algorithmen für Sensornetzwerke

#### ■ Johannes Singler

- Parallele Algorithmenbibliotheken  
(z. B. STL, für Externspeicher, für Geometrie)
- Sortieren riesiger Datenmengen



### externe Mitarbeiter

#### ■ Jan Wassenberg

- Algorithmen zur effizienten Bildauswertung
- Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung (IOSB)



### Mitarbeiter ab März

#### ■ Dr. Johannes Fischer

- Stringalgorithmen
- platzeffiziente Datenstrukturen

#### ■ Christian Schulz

- Partitionierung von Graphen



### externe Mitarbeiter

#### ■ Jan Wassenberg

- Algorithmen zur effizienten Bildauswertung
- Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung (IOSB)



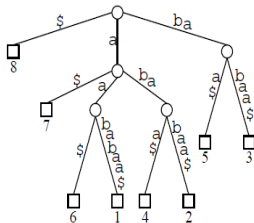
### Mitarbeiter ab März

#### ■ Dr. Johannes Fischer

- Stringalgorithmen
- platzeffiziente Datenstrukturen

#### ■ Christian Schulz

- Partitionierung von Graphen



## externe Mitarbeiter

### ■ Jan Wassenberg

- Algorithmen zur effizienten Bildauswertung
- Fraunhofer Institut für Optronik, Systemtechnik und Bildauswertung (IOSB)



## Mitarbeiter ab März

### ■ Dr. Johannes Fischer

- Stringalgorithmen
- platzeffiziente Datenstrukturen

### ■ Christian Schulz

- Partitionierung von Graphen





## einige Ereignisse der letzten Jahre

- **Google Research Award (2008, 2010)**
  - für Peter Sanders, Dominik Schultes (2008)
  - Bereich: Routenplanung
- **ALGO (2008)**
  - große internationale Algorithmik-Konferenz
  - organisiert von den Lehrstühlen Prof. Sanders und Prof. Wagner
- **Weltrekord im Sortieren großer Datenmengen (2009)**
  - für Tim Kieritz, Mirko Rahn, Peter Sanders, Johannes Singler
  - Sortieren von 100 TB in 3h (auf 195 Knoten)

## einige Ereignisse der letzten Jahre

- **Google Research Award (2008, 2010)**
  - für Peter Sanders, Dominik Schultes (2008)
  - Bereich: Routenplanung
- **ALGO (2008)**
  - große internationale Algorithmik-Konferenz
  - organisiert von den Lehrstühlen Prof. Sanders und Prof. Wagner
- **Weltrekord im Sortieren großer Datenmengen (2009)**
  - für Tim Kieritz, Mirko Rahn, Peter Sanders, Johannes Singler
  - Sortieren von 100 TB in 3h (auf 195 Knoten)

## einige Ereignisse der letzten Jahre

- **Google Research Award (2008, 2010)**
  - für Peter Sanders, Dominik Schultes (2008)
  - Bereich: Routenplanung
- **ALGO (2008)**
  - große internationale Algorithmik-Konferenz
  - organisiert von den Lehrstühlen Prof. Sanders und Prof. Wagner
- **Weltrekord im Sortieren großer Datenmengen (2009)**
  - für Tim Kieritz, Mirko Rahn, Peter Sanders, Johannes Singler
  - Sortieren von 100 TB in 3h (auf 195 Knoten)

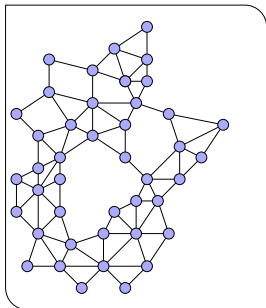
## Boundary Recognition

### Problemstellung

- geg. Sensornetzwerk mit Verbindungen, welche Knoten sitzen am Rand?
- äußerer Rand → Ausdehnung  
innerer Rand → Loch

### Was ist ein Loch / was gehört zum Rand?

- viele mögliche Definitionen:
  - topologisch
  - "weiße Flächen"
  - ...

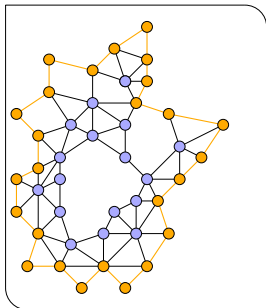


### Problemstellung

- geg. Sensornetzwerk mit Verbindungen, welche Knoten sitzen am Rand?
- äußerer Rand → Ausdehnung
- innerer Rand → Loch

### Was ist ein Loch / was gehört zum Rand?

- viele mögliche Definitionen:
  - topologisch
  - "weiße Flächen"
  - ...

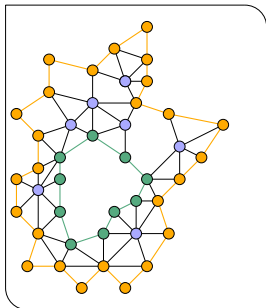


### Problemstellung

- geg. Sensornetzwerk mit Verbindungen, welche Knoten sitzen am Rand?
- äußerer Rand → Ausdehnung  
innerer Rand → Loch

### Was ist ein Loch / was gehört zum Rand?

- viele mögliche Definitionen:
  - topologisch
  - "weiße Flächen"
  - ...

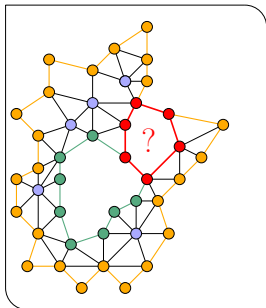


### Problemstellung

- geg. Sensornetzwerk mit Verbindungen, welche Knoten sitzen am Rand?
- äußerer Rand → Ausdehnung  
innerer Rand → Loch

### Was ist ein Loch / was gehört zum Rand?

- viele mögliche Definitionen:
  - topologisch
  - "weiße Flächen"
  - ...



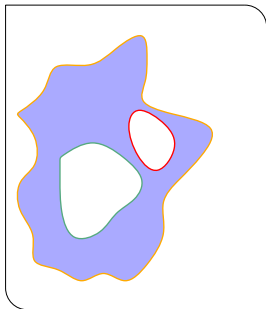


### Problemstellung

- geg. Sensornetzwerk mit Verbindungen, welche Knoten sitzen am Rand?
- äußerer Rand → Ausdehnung  
innerer Rand → Loch

### Was ist ein Loch / was gehört zum Rand?

- viele mögliche Definitionen:
  - topologisch
  - "weiße Flächen"
  - ...

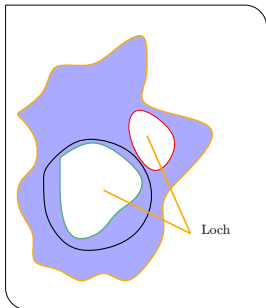


### Problemstellung

- geg. Sensornetzwerk mit Verbindungen, welche Knoten sitzen am Rand?
- äußerer Rand → Ausdehnung
- innerer Rand → Loch

### Was ist ein Loch / was gehört zum Rand?

- viele mögliche Definitionen:
  - topologisch
  - "weiße Flächen"
  - ...

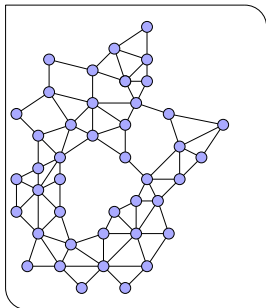


### Problemstellung

- geg. Sensornetzwerk mit Verbindungen, welche Knoten sitzen am Rand?
- äußerer Rand → Ausdehnung  
innerer Rand → Loch

### Was ist ein Loch / was gehört zum Rand?

- viele mögliche Definitionen:
  - topologisch
  - "weiße Flächen"
  - ...

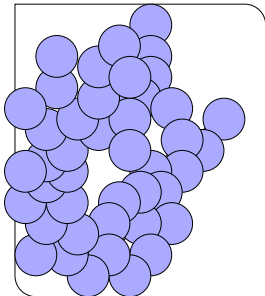


### Problemstellung

- geg. Sensornetzwerk mit Verbindungen, welche Knoten sitzen am Rand?
- äußerer Rand → Ausdehnung  
innerer Rand → Loch

### Was ist ein Loch / was gehört zum Rand?

- viele mögliche Definitionen:
  - topologisch
  - "weiße Flächen"
  - ...

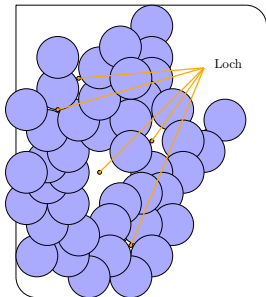


### Problemstellung

- geg. Sensornetzwerk mit Verbindungen, welche Knoten sitzen am Rand?
- äußerer Rand → Ausdehnung  
innerer Rand → Loch

### Was ist ein Loch / was gehört zum Rand?

- viele mögliche Definitionen:
  - topologisch
  - "weiße Flächen"
  - ...

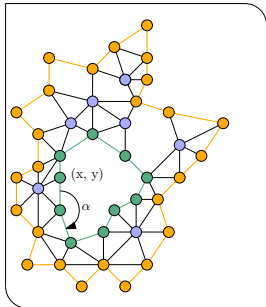


### verschiedene Lösungsansätze

- geometrisch
- topologisch
- statistisch

### unser Ansatz

- keine Ortsinformation, nur Verbindungsinformation
- trotzdem geometrischer Ansatz
- lokaler Algorithmus

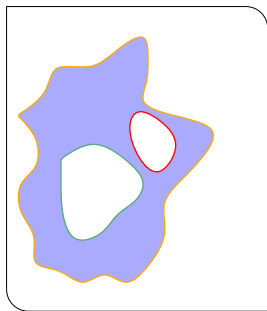


### verschiedene Lösungsansätze

- geometrisch
- topologisch
- statistisch

### unser Ansatz

- keine Ortsinformation, nur Verbindungsinformation
- trotzdem geometrischer Ansatz
- lokaler Algorithmus

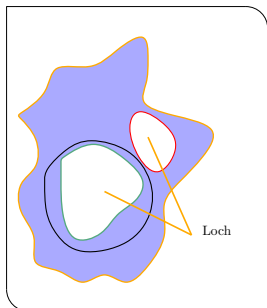


### verschiedene Lösungsansätze

- geometrisch
- topologisch
- statistisch

### unser Ansatz

- keine Ortsinformation,  
nur Verbindungsinformation
- trotzdem geometrischer Ansatz
- lokaler Algorithmus





# Boundary Recognition

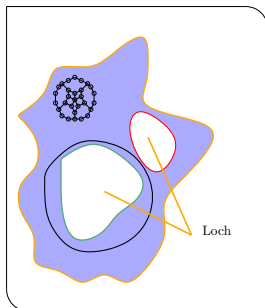
## Einführung

### verschiedene Lösungsansätze

- geometrisch
- topologisch
- statistisch

### unser Ansatz

- keine Ortsinformation, nur Verbindungsinformation
- trotzdem geometrischer Ansatz
- lokaler Algorithmus

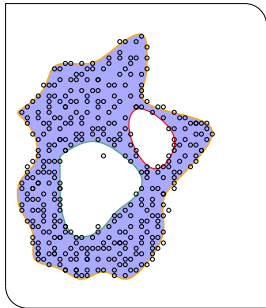


### verschiedene Lösungsansätze

- geometrisch
- topologisch
- statistisch

### unser Ansatz

- keine Ortsinformation,  
nur Verbindungsinformation
- trotzdem geometrischer Ansatz
- lokaler Algorithmus



# Boundary Recognition

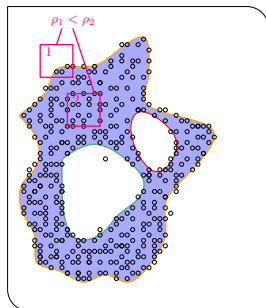
## Einführung

### verschiedene Lösungsansätze

- geometrisch
- topologisch
- statistisch

### unser Ansatz

- keine Ortsinformation, nur Verbindungsinformation
- trotzdem geometrischer Ansatz
- lokaler Algorithmus

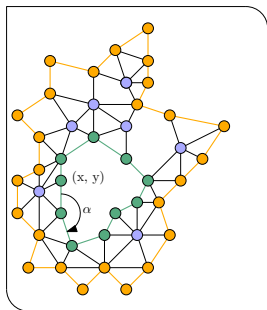


### verschiedene Lösungsansätze

- geometrisch
- topologisch
- statistisch

### unser Ansatz

- keine Ortsinformation, nur Verbindungsinformation
- trotzdem geometrischer Ansatz
- lokaler Algorithmus



# Boundary Recognition

## Multi-Dimensional Scaling (MDS)

### gegeben

- Distanzen  $\delta_{ij}$  zwischen allen Knoten  $i$  und  $j$

### gesucht

- Einbettung  $\{x_1, \dots, x_n\}$  aller Knoten in  $\mathbb{R}^k$  mit  $\sum_{i \neq j} (\|x_i - x_j\| - \delta_{ij})^2$  minimal

### Ablauf

- sei  $\mathbf{X} = [x_1, \dots, x_n]^T$ ,  $\mathbf{B} = (b_{ij})$  mit
$$b_{ij} = -\frac{1}{2} \left( \delta_{ij}^2 - \frac{1}{n} \sum \delta_{rj}^2 - \frac{1}{n} \sum \delta_{is}^2 + \frac{1}{n^2} \sum \sum \delta_{rs}^2 \right) \quad (\text{double-centering})$$
- $\mathbf{B} = \mathbf{X}\mathbf{X}^T$  diagonalisierbar  $\Rightarrow \mathbf{B} = \mathbf{V}\mathbf{D}\mathbf{V}^T$
- Verwende größte  $k$  EW, um  $\mathbf{X} = \mathbf{V}\mathbf{D}^{1/2}$  anzunähern

# Boundary Recognition

## Multi-Dimensional Scaling (MDS)

### gegeben

- Distanzen  $\delta_{ij}$  zwischen allen Knoten  $i$  und  $j$

### gesucht

- Einbettung  $\{x_1, \dots, x_n\}$  aller Knoten in  $\mathbb{R}^k$   
mit  $\sum_{i \neq j} (\|x_i - x_j\| - \delta_{ij})^2$  minimal

### Ablauf

- sei  $\mathbf{X} = [x_1, \dots, x_n]^T$ ,  $\mathbf{B} = (b_{ij})$  mit  

$$b_{ij} = -\frac{1}{2} \left( \delta_{ij}^2 - \frac{1}{n} \sum \delta_{rj}^2 - \frac{1}{n} \sum \delta_{is}^2 + \frac{1}{n^2} \sum \sum \delta_{rs}^2 \right) \quad (\text{double-centering})$$
- $\mathbf{B} = \mathbf{X}\mathbf{X}^T$  diagonalisierbar  $\Rightarrow \mathbf{B} = \mathbf{V}\mathbf{D}\mathbf{V}^T$
- Verwende größte  $k$  EW, um  $\mathbf{X} = \mathbf{V}\mathbf{D}^{1/2}$  anzunähern

# Boundary Recognition

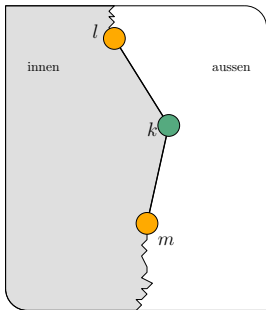
## Locherkennung

### Def.: Randknoten

- Knoten  $k$  ist Randknoten, wenn
  - Winkel  $\sphericalangle(l, k, m) > \alpha$  und
  - f.a. gemeinsamen Nachbarn  $q$  von  $l, m$  gilt  $\sphericalangle(q, k, m) < 0$  bzw.  $> \sphericalangle(l, k, m)$

### Problemfälle

- Schwierigkeiten mit Randdefinition  
→ eine Definition wählen
- Extremfälle  
→ lösen durch Nachoptimierung



# Boundary Recognition

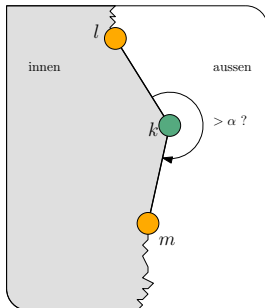
## Locherkennung

### Def.: Randknoten

- Knoten  $k$  ist Randknoten, wenn
  - Winkel  $\sphericalangle(l, k, m) > \alpha$  und
  - f.a. gemeinsamen Nachbarn  $q$  von  $l, m$  gilt  $\sphericalangle(q, k, m) < 0$  bzw.  $> \sphericalangle(l, k, m)$

### Problemfälle

- Schwierigkeiten mit Randdefinition  
→ eine Definition wählen
- Extremfälle  
→ lösen durch Nachoptimierung





# Boundary Recognition

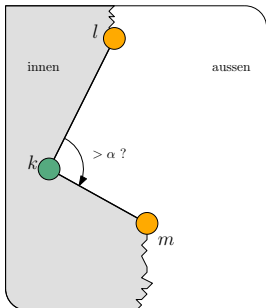
## Locherkennung

### Def.: Randknoten

- Knoten  $k$  ist Randknoten, wenn
  - Winkel  $\sphericalangle(l, k, m) > \alpha$  und
  - f.a. gemeinsamen Nachbarn  $q$  von  $l, m$  gilt  $\sphericalangle(q, k, m) < 0$  bzw.  $> \sphericalangle(l, k, m)$

### Problemfälle

- Schwierigkeiten mit Randdefinition  
→ eine Definition wählen
- Extremfälle  
→ lösen durch Nachoptimierung



# Boundary Recognition

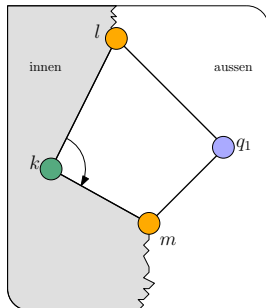
## Locherkennung

### Def.: Randknoten

- Knoten  $k$  ist Randknoten, wenn
  - Winkel  $\sphericalangle(l, k, m) > \alpha$  und
  - f.a. gemeinsamen Nachbarn  $q$  von  $l, m$  gilt  $\sphericalangle(q, k, m) < 0$  bzw.  $> \sphericalangle(l, k, m)$

### Problemfälle

- Schwierigkeiten mit Randdefinition  
→ eine Definition wählen
- Extremfälle  
→ lösen durch Nachoptimierung



# Boundary Recognition

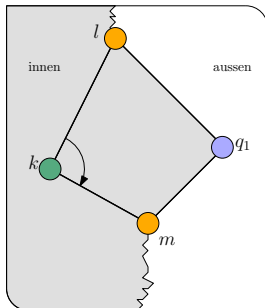
## Locherkennung

### Def.: Randknoten

- Knoten  $k$  ist Randknoten, wenn
  - Winkel  $\sphericalangle(l, k, m) > \alpha$  und
  - f.a. gemeinsamen Nachbarn  $q$  von  $l, m$  gilt  $\sphericalangle(q, k, m) < 0$  bzw.  $> \sphericalangle(l, k, m)$

### Problemfälle

- Schwierigkeiten mit Randdefinition  
→ eine Definition wählen
- Extremfälle  
→ lösen durch Nachoptimierung



# Boundary Recognition

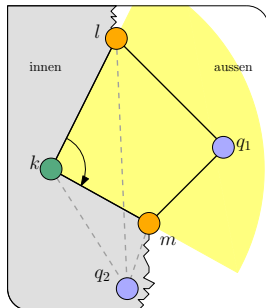
## Locherkennung

### Def.: Randknoten

- Knoten  $k$  ist Randknoten, wenn
  - Winkel  $\sphericalangle(l, k, m) > \alpha$  und
  - f.a. gemeinsamen Nachbarn  $q$  von  $l, m$  gilt  $\sphericalangle(q, k, m) < 0$  bzw.  $> \sphericalangle(l, k, m)$

### Problemfälle

- Schwierigkeiten mit Randdefinition  
→ eine Definition wählen
- Extremfälle  
→ lösen durch Nachoptimierung



# Boundary Recognition

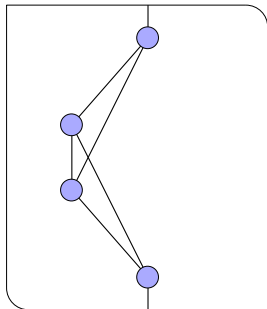
## Locherkennung

### Def.: Randknoten

- Knoten  $k$  ist Randknoten, wenn
  - Winkel  $\sphericalangle(l, k, m) > \alpha$  und
  - f.a. gemeinsamen Nachbarn  $q$  von  $l, m$  gilt  $\sphericalangle(q, k, m) < 0$  bzw.  $> \sphericalangle(l, k, m)$

### Problemfälle

- Schwierigkeiten mit Randdefinition
  - eine Definition wählen
- Extremfälle
  - lösen durch Nachoptimierung



# Boundary Recognition

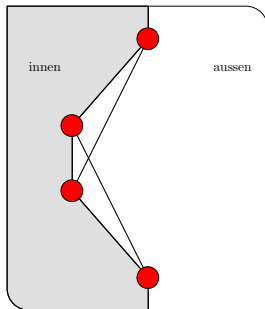
## Locherkennung

### Def.: Randknoten

- Knoten  $k$  ist Randknoten, wenn
  - Winkel  $\sphericalangle(l, k, m) > \alpha$  und
  - f.a. gemeinsamen Nachbarn  $q$  von  $l, m$  gilt  $\sphericalangle(q, k, m) < 0$  bzw.  $> \sphericalangle(l, k, m)$

### Problemfälle

- Schwierigkeiten mit Randdefinition  
→ eine Definition wählen
- Extremfälle  
→ lösen durch Nachoptimierung



# Boundary Recognition

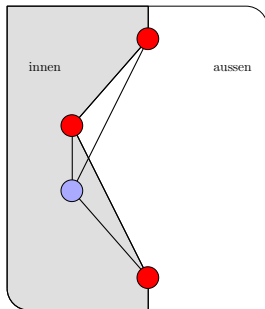
## Locherkennung

### Def.: Randknoten

- Knoten  $k$  ist Randknoten, wenn
  - Winkel  $\sphericalangle(l, k, m) > \alpha$  und
  - f.a. gemeinsamen Nachbarn  $q$  von  $l, m$  gilt  $\sphericalangle(q, k, m) < 0$  bzw.  $> \sphericalangle(l, k, m)$

### Problemfälle

- Schwierigkeiten mit Randdefinition
  - eine Definition wählen
- Extremfälle
  - lösen durch Nachoptimierung



# Boundary Recognition

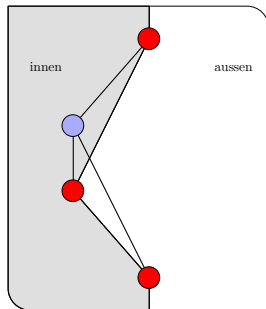
## Locherkennung

### Def.: Randknoten

- Knoten  $k$  ist Randknoten, wenn
  - Winkel  $\sphericalangle(l, k, m) > \alpha$  und
  - f.a. gemeinsamen Nachbarn  $q$  von  $l, m$  gilt  $\sphericalangle(q, k, m) < 0$  bzw.  $> \sphericalangle(l, k, m)$

### Problemfälle

- Schwierigkeiten mit Randdefinition  
→ eine Definition wählen
- Extremfälle  
→ lösen durch Nachoptimierung





# Boundary Recognition

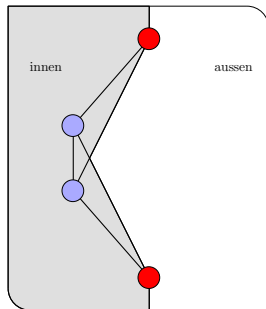
## Locherkennung

### Def.: Randknoten

- Knoten  $k$  ist Randknoten, wenn
  - Winkel  $\sphericalangle(l, k, m) > \alpha$  und
  - f.a. gemeinsamen Nachbarn  $q$  von  $l, m$  gilt  $\sphericalangle(q, k, m) < 0$  bzw.  $> \sphericalangle(l, k, m)$

### Problemfälle

- Schwierigkeiten mit Randdefinition
  - eine Definition wählen
- Extremfälle
  - lösen durch Nachoptimierung



# Boundary Recognition

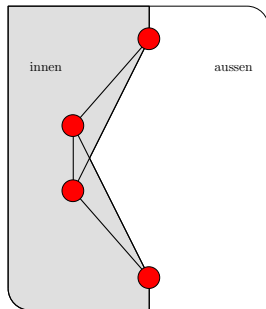
## Locherkennung

### Def.: Randknoten

- Knoten  $k$  ist Randknoten, wenn
  - Winkel  $\sphericalangle(l, k, m) > \alpha$  und
  - f.a. gemeinsamen Nachbarn  $q$  von  $l, m$  gilt  $\sphericalangle(q, k, m) < 0$  bzw.  $> \sphericalangle(l, k, m)$

### Problemfälle

- Schwierigkeiten mit Randdefinition  
→ eine Definition wählen
- Extremfälle  
→ lösen durch Nachoptimierung



# Boundary Recognition

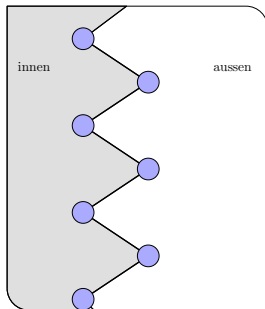
## Locherkennung

### Def.: Randknoten

- Knoten  $k$  ist Randknoten, wenn
  - Winkel  $\sphericalangle(l, k, m) > \alpha$  und
  - f.a. gemeinsamen Nachbarn  $q$  von  $l, m$  gilt  $\sphericalangle(q, k, m) < 0$  bzw.  $> \sphericalangle(l, k, m)$

### Problemfälle

- Schwierigkeiten mit Randdefinition  
→ eine Definition wählen
- **Extremfälle**  
→ lösen durch Nachoptimierung



# Boundary Recognition

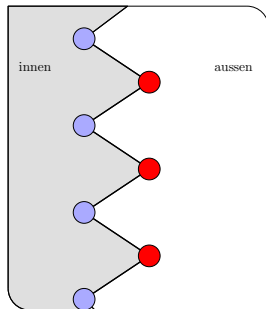
## Locherkennung

### Def.: Randknoten

- Knoten  $k$  ist Randknoten, wenn
  - Winkel  $\sphericalangle(l, k, m) > \alpha$  und
  - f.a. gemeinsamen Nachbarn  $q$  von  $l, m$  gilt  $\sphericalangle(q, k, m) < 0$  bzw.  $> \sphericalangle(l, k, m)$

### Problemfälle

- Schwierigkeiten mit Randdefinition  
→ eine Definition wählen
- **Extremfälle**  
→ lösen durch Nachoptimierung



# Boundary Recognition

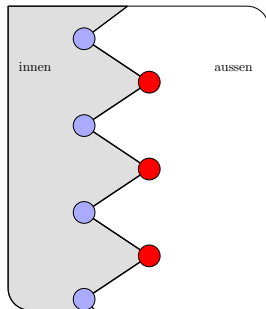
## Locherkennung

### Def.: Randknoten

- Knoten  $k$  ist Randknoten, wenn
  - Winkel  $\sphericalangle(l, k, m) > \alpha$  und
  - f.a. gemeinsamen Nachbarn  $q$  von  $l, m$  gilt  $\sphericalangle(q, k, m) < 0$  bzw.  $> \sphericalangle(l, k, m)$

### Problemfälle

- Schwierigkeiten mit Randdefinition
  - eine Definition wählen
- Extremfälle
  - lösen durch Nachoptimierung

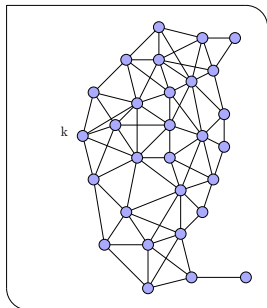


# Boundary Recognition

## Algorithmus MDS-BR

### Algorithmus MDS-BR

- `test_boundary(k)`, f.a. Knoten  $k$
- `refine_results()`



# Boundary Recognition

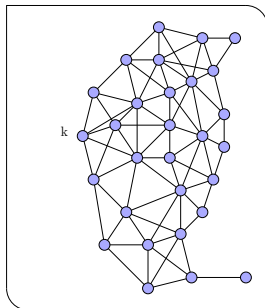
## Algorithmus MDS-BR

### Algorithmus MDS-BR

- `test_boundary(k)`, f.a. Knoten  $k$
- `refine_results()`

### Algorithmus: `test_boundary(k)`

- extrahiere 2-hop Nachbarschaft von  $k$
- bestimme Einbettung mit MDS (hop-count als Distanzmaß)
- teste auf Randknoten in Einbettung



# Boundary Recognition

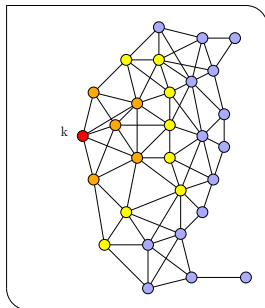
## Algorithmus MDS-BR

### Algorithmus MDS-BR

- `test_boundary(k)`, f.a. Knoten  $k$
- `refine_results()`

### Algorithmus: `test_boundary(k)`

- extrahiere 2-hop Nachbarschaft von  $k$
- bestimme Einbettung mit MDS (hop-count als Distanzmaß)
- teste auf Randknoten in Einbettung





# Boundary Recognition

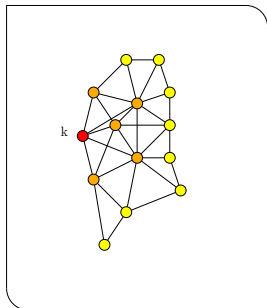
## Algorithmus MDS-BR

### Algorithmus MDS-BR

- `test_boundary(k)`, f.a. Knoten  $k$
- `refine_results()`

### Algorithmus: `test_boundary(k)`

- extrahiere 2-hop Nachbarschaft von  $k$
- bestimme Einbettung mit MDS (hop-count als Distanzmaß)
- teste auf Randknoten in Einbettung



# Boundary Recognition

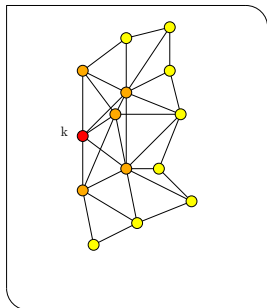
## Algorithmus MDS-BR

### Algorithmus MDS-BR

- `test_boundary(k)`, f.a. Knoten  $k$
- `refine_results()`

### Algorithmus: `test_boundary(k)`

- extrahiere 2-hop Nachbarschaft von  $k$
- bestimme Einbettung mit MDS (hop-count als Distanzmaß)
- teste auf Randknoten in Einbettung



# Boundary Recognition

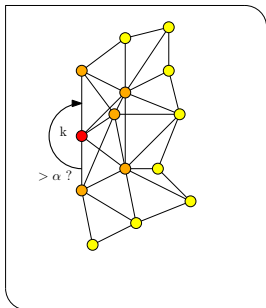
## Algorithmus MDS-BR

### Algorithmus MDS-BR

- `test_boundary(k)`, f.a. Knoten  $k$
- `refine_results()`

### Algorithmus: `test_boundary(k)`

- extrahiere 2-hop Nachbarschaft von  $k$
- bestimme Einbettung mit MDS (hop-count als Distanzmaß)
- teste auf Randknoten in Einbettung



# Boundary Recognition

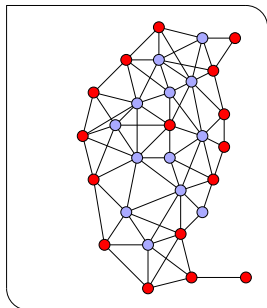
## Algorithmus MDS-BR

### Algorithmus MDS-BR

- `test_boundary(k)`, f.a. Knoten  $k$
- `refine_results()`

### Algorithmus: `test_boundary(k)`

- extrahiere 2-hop Nachbarschaft von  $k$
- bestimme Einbettung mit MDS (hop-count als Distanzmaß)
- teste auf Randknoten in Einbettung



# Boundary Recognition

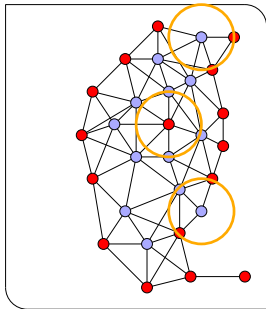
## Algorithmus MDS-BR

### Algorithmus MDS-BR

- `test_boundary(k)`, f.a. Knoten  $k$
- `refine_results()`

### Algorithmus: `refine_results()`

- entferne Knoten aus Randmenge, die  $< \beta$  Nachbarn auf dem Rand haben
- addiere Knoten zur Randmenge, die  $< \gamma$  Nachbarn nicht auf dem Rand haben



# Boundary Recognition

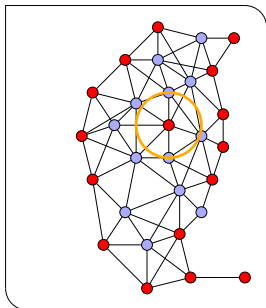
## Algorithmus MDS-BR

### Algorithmus MDS-BR

- `test_boundary(k)`, f.a. Knoten  $k$
- `refine_results()`

### Algorithmus: `refine_results()`

- entferne Knoten aus Randmenge, die  $< \beta$  Nachbarn auf dem Rand haben
- addiere Knoten zur Randmenge, die  $< \gamma$  Nachbarn nicht auf dem Rand haben



# Boundary Recognition

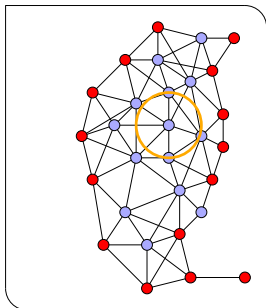
## Algorithmus MDS-BR

### Algorithmus MDS-BR

- `test_boundary(k)`, f.a. Knoten  $k$
- `refine_results()`

### Algorithmus: `refine_results()`

- entferne Knoten aus Randmenge, die  $< \beta$  Nachbarn auf dem Rand haben
- addiere Knoten zur Randmenge, die  $< \gamma$  Nachbarn nicht auf dem Rand haben



# Boundary Recognition

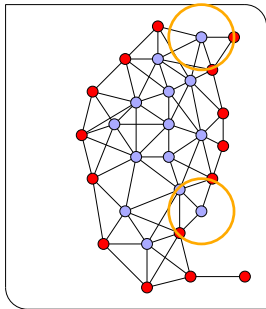
## Algorithmus MDS-BR

### Algorithmus MDS-BR

- `test_boundary(k)`, f.a. Knoten  $k$
- `refine_results()`

### Algorithmus: `refine_results()`

- entferne Knoten aus Randmenge, die  $< \beta$  Nachbarn auf dem Rand haben
- addiere Knoten zur Randmenge, die  $< \gamma$  Nachbarn nicht auf dem Rand haben





# Boundary Recognition

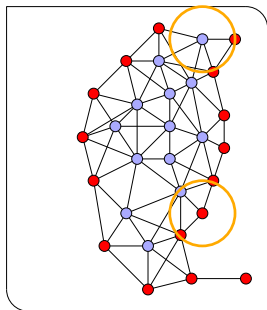
## Algorithmus MDS-BR

### Algorithmus MDS-BR

- `test_boundary(k)`, f.a. Knoten  $k$
- `refine_results()`

### Algorithmus: `refine_results()`

- entferne Knoten aus Randmenge, die  $< \beta$  Nachbarn auf dem Rand haben
- addiere Knoten zur Randmenge, die  $< \gamma$  Nachbarn nicht auf dem Rand haben



# Boundary Recognition

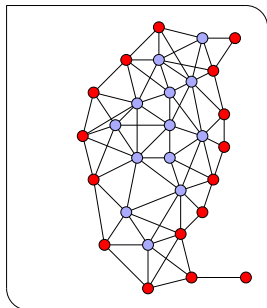
## Algorithmus MDS-BR

### Algorithmus MDS-BR

- `test_boundary(k)`, f.a. Knoten  $k$
- `refine_results()`

### Algorithmus: `refine_results()`

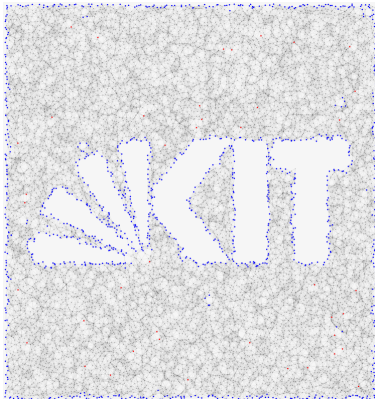
- entferne Knoten aus Randmenge, die  $< \beta$  Nachbarn auf dem Rand haben
- addiere Knoten zur Randmenge, die  $< \gamma$  Nachbarn nicht auf dem Rand haben



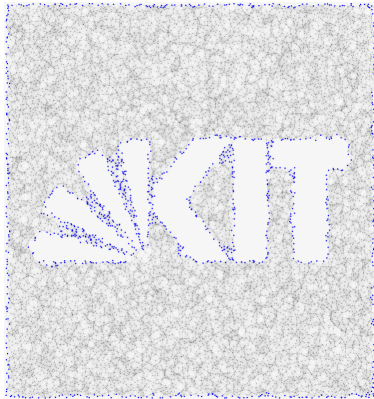
# Boundary Recognition

## Ergebnisse

MDS-BR



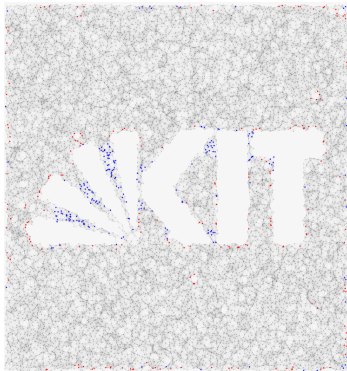
ECBR



# Boundary Recognition

## Ergebnisse

Vergleich

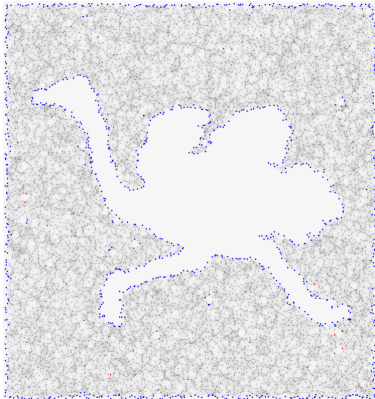


rot: bei MDS-BR und nicht bei ECBR  
blau: bei ECBR und nicht bei MDS-BR

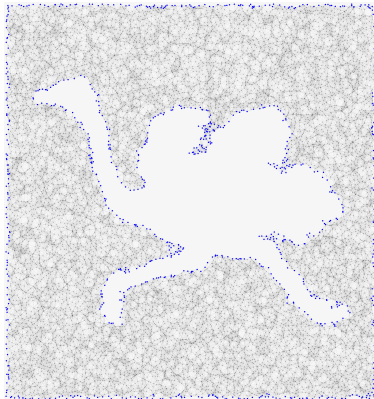
# Boundary Recognition

## Ergebnisse

MDS-BR



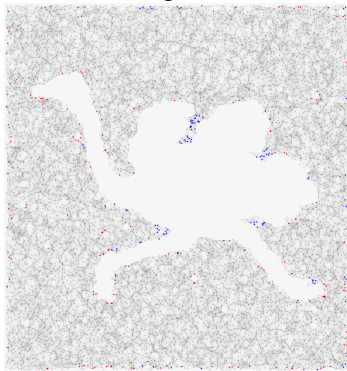
ECBR



# Boundary Recognition

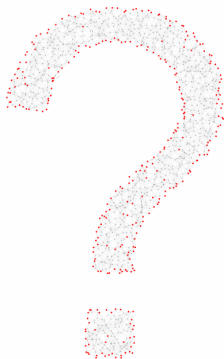
## Ergebnisse

Vergleich



**rot:** bei MDS-BR und nicht bei ECBR  
**blau:** bei ECBR und nicht bei MDS-BR

# Danke für Ihre Aufmerksamkeit!



## Zeit für Fragen