

# Planare Einbettungen mit maximaler Außenfläche

Carsten Gutwenger



Vorlesung  
**Automatisches Zeichnen von Graphen**  
 WS 07/08  
 18. Dezember 2007



TU Dortmund, Fakultät für Informatik, Ls11

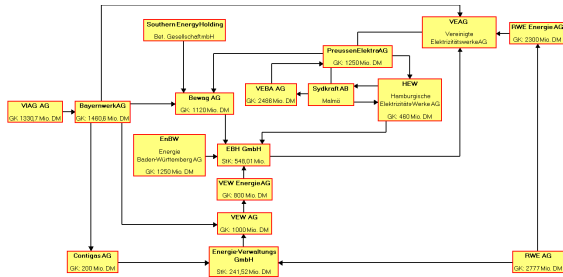
## Überblick

- Warum Einbettungen mit großer Außenfläche?
- Der Algorithmus
  - 2-zusammenhängende Graphen
  - Zusammenhängende Graphen
- Implementierungen im OGDf

→ **Nikolausvortrag**  
 13:19 Uhr Audimax



## Motivation (1)



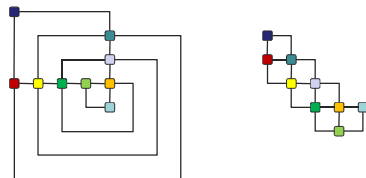
## Motivation (2)

### Der Topology-Shape-Metrics Approach:

1. Bestimme Topologie → **planare Einbettung**
2. Bestimme orthogonale Form der Zeichnung  
 → **Minimiere Anzahl der Knicke**
3. Bestimme endgültige Koordinaten (Kompaktierung)  
 → **Minimiere Kantenlängen (oder Fläche)**

## Motivation (3)

Ein 2-zusammenhängender Graph:



Knicke: 13 → **7**  
 Fläche: 7 × 7 → **4 × 4**

**Außenfläche: 3 → 9**

## Motivation (4)

Ein nicht 2-zusammenhängender Graph:



Knicke: 11 → **5**  
 Fläche: 9 × 6 → **6 × 4**

**Außenfläche: 3 → 15**

## Literatur

### Verwandte Arbeiten:

D. Bienstock, C. L. Monma, *On the complexity of embedding graphs to minimize certain distance measures*, Algorithmica 5(1), 1990, pp. 93-109.

- Polynomialzeit-Algorithmen zur Minimierung verschiedener Distanzmaße (*radius, width, outerplanarity, face depth*).

M. Pizzonia, R. Tamasia, *Minimum depth graph embedding*, Proc. ESA 2000, pp. 356-367.

- Linearzeit-Algorithmen zur Minimierung der Blockschachtelungstiefe für 2-zshgd. Graphen mit **fester** Einbettung.

## Literatur

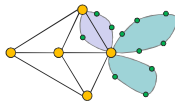
C. Gutwenger, P. Mutzel, *Graph embedding with minimum depth and maximum external face*, Proc. Graph Drawing 2003, pp. 259-272.

- Maximierung der Außenfläche über alle planare Einbettungen.
- Minimierung der Blockschachtelungstiefe über alle planaren Einbettungen.
- Kombination von beiden (Maximale Außenfläche über alle planaren Einbettungen minimaler Blockschachtelungstiefe.)

## 2-zusammenhängende Graphen

Betrachte einen Skeletongraphen:

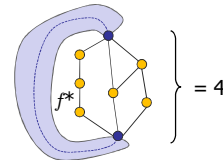
- jede Kante hat eine Länge  $\geq 1$
- jeder Knoten hat eine Länge  $\geq 0$
- Größe einer Fläche :=  
Summe Kantenlängen + Summe Knotenlängen



## Komponentenlänge (1)

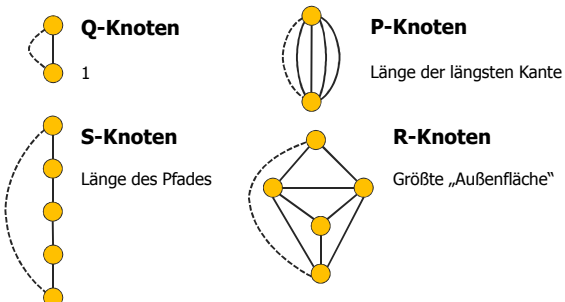
Komponentenlänge einer Skeletonkante:

- $\text{expansion}^+(e)$  := Expansionsgraph von  $e$  + Kante  $e$
- Suche Einbettung von  $\text{expansion}^+(e)$  mit einer Fläche  $f^*$ , die  $e$  enthält und maximale Größe hat.
- Komponentenlänge von  $e$  := Größe von  $f^*$  - Länge von  $e$



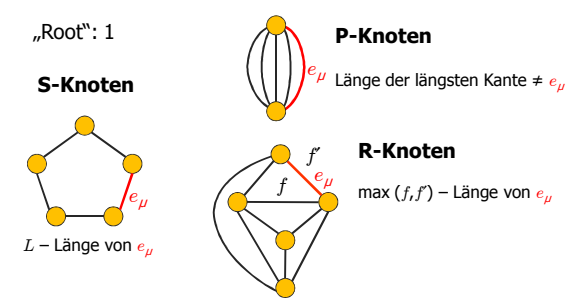
## Komponentenlänge (2)

„Bottom-up“ Berechnung:



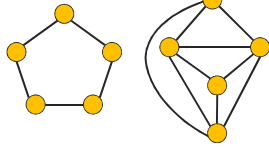
## Komponentenlänge (3)

„Top-down“ Berechnung für Referenzkanten:



## Bestimme größte Fläche

Betrachte jedes Skeleton:



### S- und R-Knoten

Wähle größte Fläche.

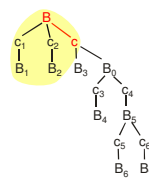


### P-Knoten

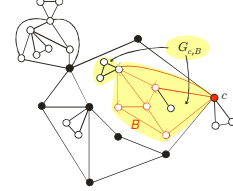
Wähle Fläche aus den beiden längsten Kanten

## Zusammenhängende Graphen (1)

Untergraph  $G_{c,B}$ :



BC-Baum



Graph  $G$

## Zusammenhängende Graphen (2)

Wir wollen berechnen:

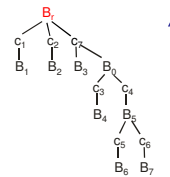
- $\text{cstrLength}(B,c) :=$   
Größe einer maximalen Fläche von  $G_{c,B}$  die  $c$  enthält.
- $\text{length}_B(c) :=$  Länge von Knoten  $c$  in Block  $B$

**Beobachtung:** Es gilt

$$\text{length}_B(c) = \sum_{\substack{(c,B) \in \text{BC-Tree} \\ B' \neq B}} \text{cstrLength}(B',c)$$

## Zusammenhängende Graphen (3)

1. Phase: „Bottom-up“

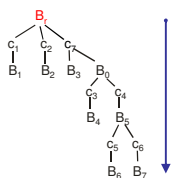


Berechne

- $\text{length}_B(v)$  für alle  $v \in B$ , die nicht „parent“ von  $B$  sind
- $\text{cstrLength}(B,c)$  für alle  $c \rightarrow B$

## Zusammenhängende Graphen (4)

2. Phase: „Top-down“



Berechne

- Maximale Fläche in  $B$
- $\text{cstrLength}(B,c)$  für alle  $B \rightarrow c$  (\*)
- $\text{length}_B(c)$  für jeden „parent“  $c$  von  $B$  (\*\*)

(\*) nach Vorberechnung in  $O(\sum \text{deg}(c'))$  möglich,  $c'$  Repräsentanten von  $c$  im SPQR-Baum

(\*\*) Vorberechnung von  $L := \sum \text{cstrLength}(B',c)$

## Finally...

### Theorem:

Sei  $G=(V,E)$  ein planarer, zusammenhängender Graph. Dann kann (mit Hilfe von BC- und SPQR-Bäumen) eine planare Einbettung von  $G$  mit maximaler Außenfläche in Zeit  $O(|V|+|E|)$  berechnet werden.

Implementierung im OGDF:

- `EmbedderMaxFace`

Weitere Einbettungsvarianten:

- `EmbedderMaxFaceLayers`
- `EmbedderMinDepth`, `EmbedderMinDepthMaxFace`,...