

## Übungsblatt 3 - Polygontriangulierung

**Ausgabe:** 27. April 2011

**Abgabe:** 03. Mai 2011

### 1 Korrektheitsbeweis

In der Vorlesung wurde der Algorithmus **MakeMonotone** vorgestellt. Im Verlauf der Ausführung des Algorithmus wird die Funktion **handleMergeVertex** aufgerufen. Beweise die Korrektheit der Funktion.

*Hinweis:* Für Details über **MakeMonotone** und die verwendeten Datenstrukturen schaue dir die Vorlesungsfolien an.

---

**Algorithmus 1:** handleMergeVertex

---

```
 $e \leftarrow$  rechte Kante  
if  $isMergeVertex(helper(e))$  then  
   $\mathcal{D} \leftarrow$  füge  $helper(e)v$  ein  
lösche  $e$  aus  $\mathcal{T}$   
 $e' \leftarrow$  Kante links von  $v$  in  $\mathcal{T}$   
if  $isMergeVertex(helper(e'))$  then  
   $\mathcal{D} \leftarrow$  füge  $helper(e')v$  ein  
 $helper(e') \leftarrow v$ 
```

---

### 2 Lineare Laufzeit?

Der in der Vorlesung vorgestellte Algorithmus zur Partitionierung eines Polygons in  $y$ -monotone Teilpolygone hat eine worst-case Laufzeit von  $\mathcal{O}(n \log n)$ . Modifiziere den Algorithmus so, dass die worst-case Laufzeit auf  $\mathcal{O}(n)$  sinkt, unter der Annahme, dass die Anzahl der Wendeknoten in den Eingabe-Polygonen in  $\mathcal{O}(1)$  ist.

*Bitte umblättern*

### 3 Randabdeckung genug?

Beweise, oder widerlege folgende Aussage:

Sei  $\mathcal{P}$  ein einfaches Polygon. Positioniert man Kameras so, dass alle Kameras zusammengenommen den vollständigen Rand des Polygons abdecken, dann decken sie auch automatisch das gesamte Innere des Polygons ab.

### 4 Polygon-Splitting

Gebe einen  $\mathcal{O}(n \log n)$  Algorithmus an, der ein einfaches Polygon, bestehend aus  $n$  Knoten, in zwei einfache Polygone aufteilt wobei jedes aus höchstens  $\lfloor 2n/3 \rfloor + 2$  Knoten besteht.

*Hinweis:* Trianguliere das Eingabe Polygon und verwende dann den Dualgraphen dieser Triangulierung.