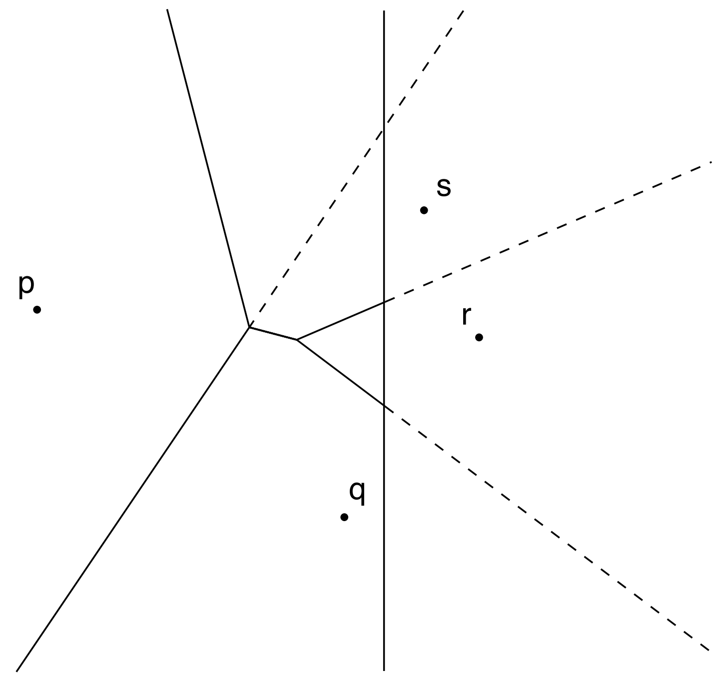
Zusammenfassung Sweep für VD

Elmar Langetepe
University of Bonn

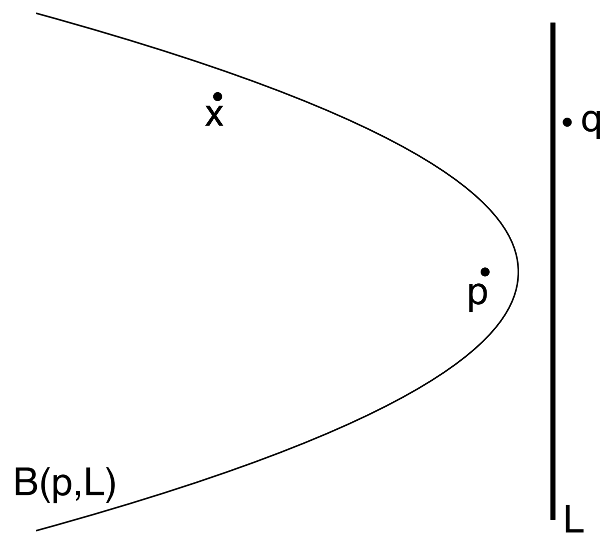
Voronoi Diagramm, Sweep , $O(n \log n)$

- Deterministisch in $O(n \log n)$
- Sweepl. über Punkte, X-sortiert
- Links Sweepline entsteht Voronoi-Diagramm
- Problem: Sweepline kennt nur Punkte links der Linie
- Punkte rechts können weit hineinragen
- Abbhilfe: Sweepline selbst ist auch ein Objekt, simuliert mögliche Punkte weiter rechts



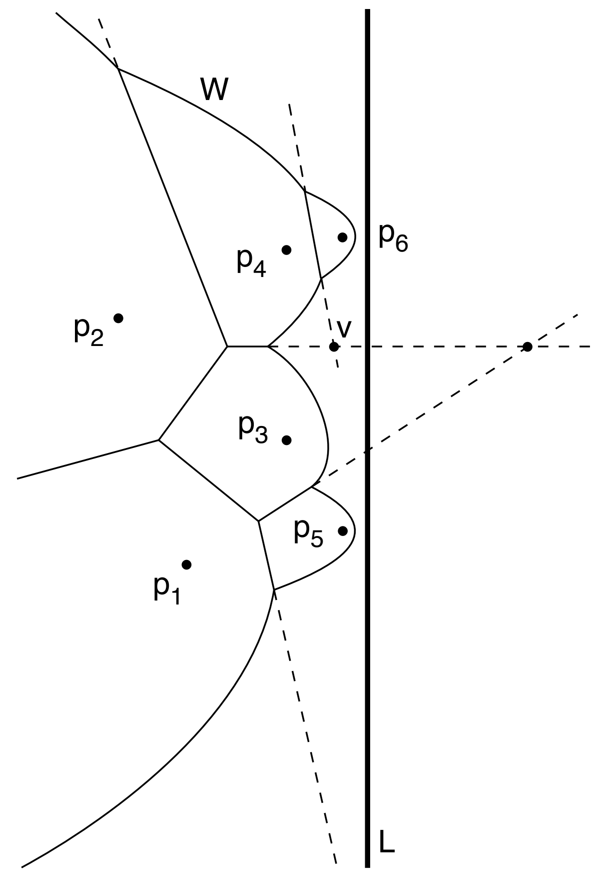
Voronoi Diagramm, Sweepline als Objekt

- Bisektor $B(p, L)$ zwischen L und p: Parabel (Tafel)
- Punkte rechts können die Parabel zwischen p und L nicht mehr betreten
- Parabel ist bereits korrekt bezüglich Zuteilung links



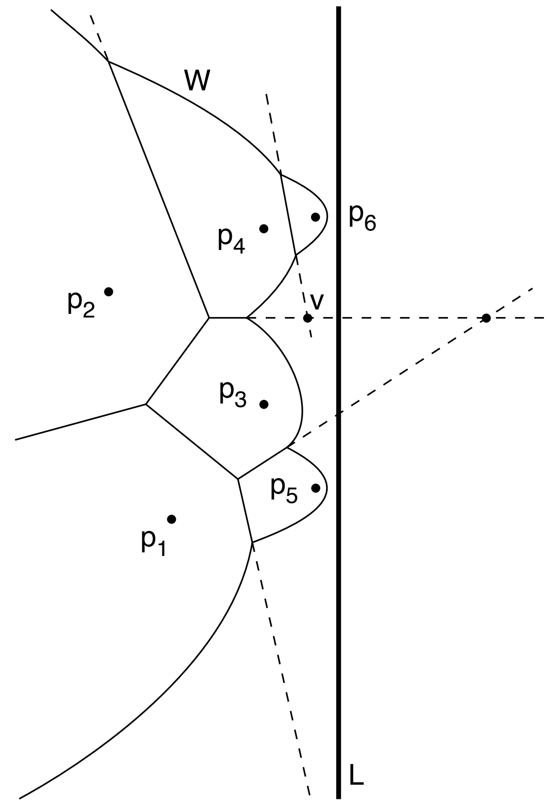
SSS: Wellenfront mit Verlängerungen

- Wellenfront: Mehrere Stücke $B(p_j, L)$ aneinandergereiht
- Links der Wellenfront: $VR(\{p_1, \ldots, p_{i-1})$ fertig
- Gebiet rechts der Wellenfront gehört L
- Zwischen den Parabeln: Spikes, Bisektoren zwischen Punkten
- Im weiteren Verlauf: Voronoi Knoten



Formale Beschreibung SSS Wellenfront

- Parabelstücke geordnet nach Y-Richtung
- Dazwischen Spikes geordnet nach Y-Richtung
- Geordnet nach Punkten: $p_1, p_5, p_3, p_4, p_6, p_4, p_2$
- Abgespeichert in einem Baum
- Schlüssel (Parabeln/Spikes)



Wellenfront ist zusammenhängend!

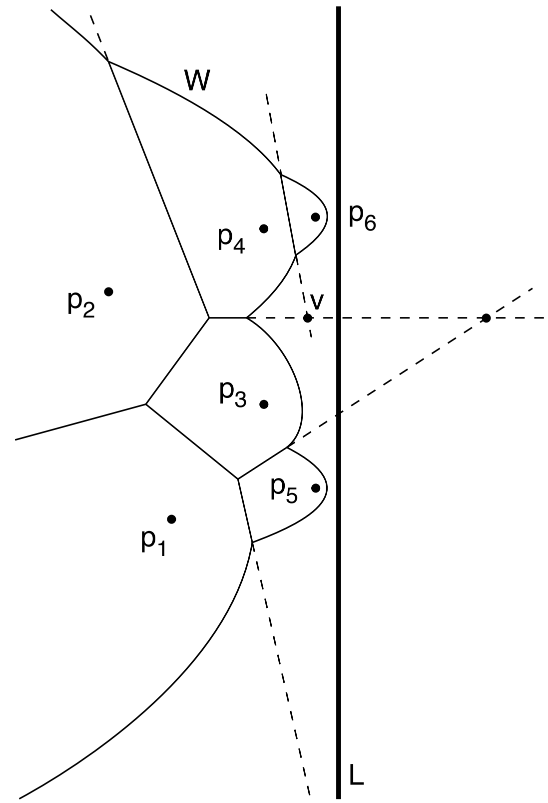
Lemma 6.10 Die Wellenfront ist zusammenhängend und Y-monoton.

- Eigenschaften der Parabeln!
- Definiert über gesamtes Intervall: Je zwei schneiden sich immer!
- Alle Parabeln sind monoton!
- Datenstruktur: Baum mit dynamischen Schlüsseln

Ereignisse des Sweeps

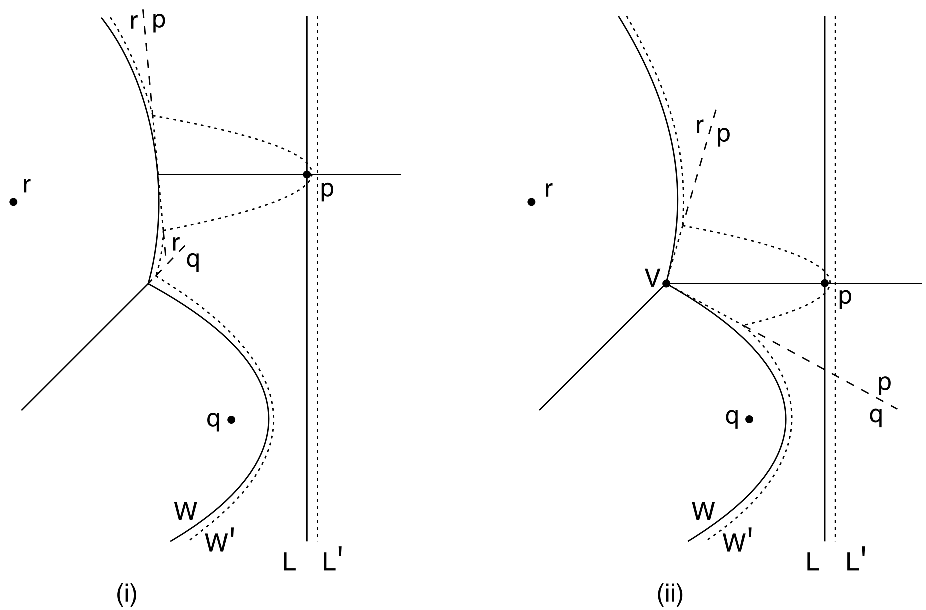
Wann verändert sich die Wellenfront? Nur diskrete Zeitpunkte!

1. Parabelstück kommt hinzu: Ein neuer Punkt wird getroffen! Punkt-Ereignis, neue Parabel entsteht
2. Parabelstück verschwindet: Die Welle erreicht den Schnittpunkt zweier Spikes, Spike-Ereignis, Voronoi-Knoten entsteht



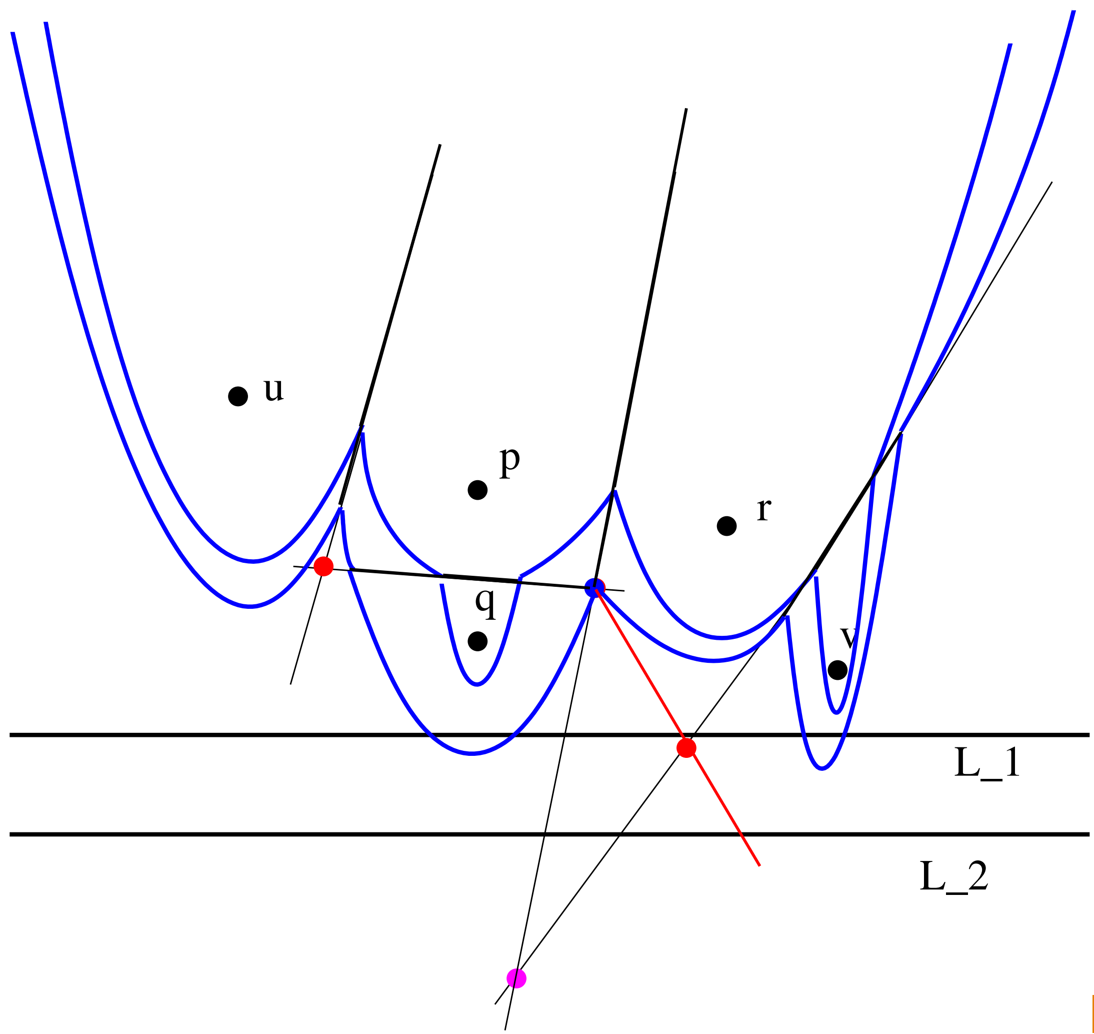
Behandlung: Punkt-Ereignis, SSS akt.

- Zwei Fälle: Innerh. Parabelst., Schnittpkt von Parabelstücken
- Zwei neue Spikes, ggf. neuer Voronoi Knoten
- Parabelstücke sortiert im Baum: Schlüssel $B(L, p_j)$ Parabel
- Zugriff $O(\log m)$, m Anzahl der Parabeln, Akt.: $O(\log m)$



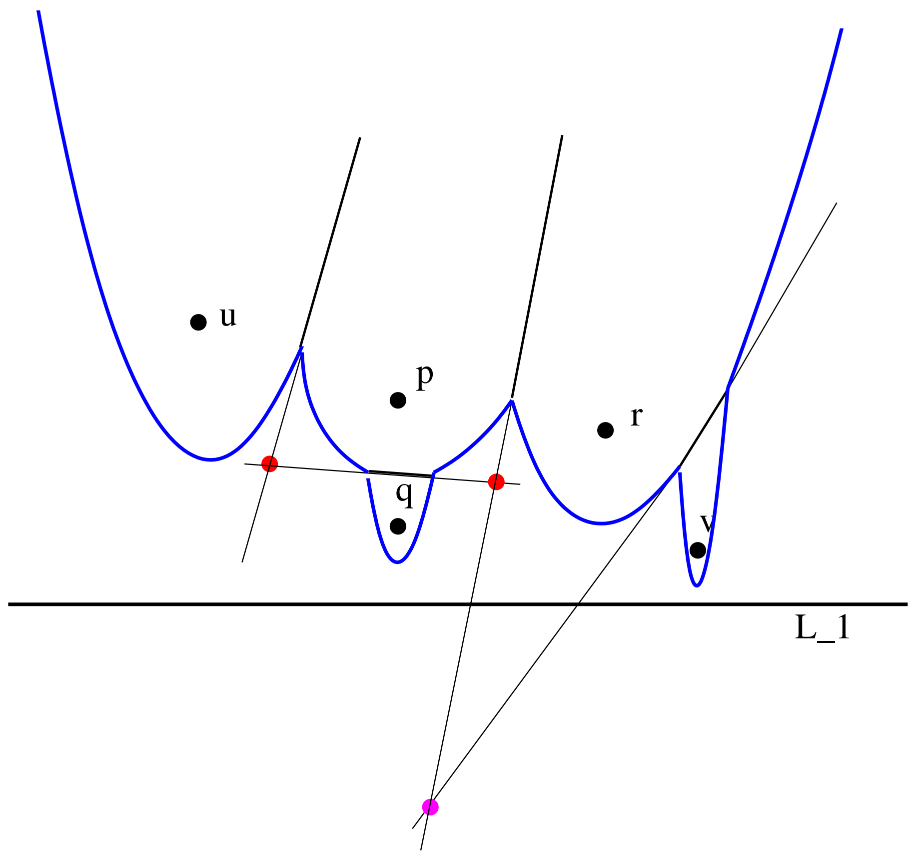
Behandlung: Spike-Ereignis, SSS-akt

- Parabelst. verschwindet, Spikes schneiden sich, Voronoi-Knoten $O(n)$ viele
- Zeitpunkt des Ereignisses (Position L_2) weiter zurück
- Parabel $B(L, p_j)$ entfernen: $O(\log m)$
- Ein neuer Spike zwischen neuen Nachbarn: $O(1)$
- Vorderster Schnittpunkt, nur Nachbarn



Ereignisstruktur akualisieren: m Parabeln

- Sei m Kompl. Wellenfront
- Priority Queue,
  $O(n)$ Punktereignisse,
  sortieren: $O(n \log n)$
- Spike-Ereign. (Voro-Knoten $O(n)$):
  Schnittpunkte der Spikes,
  jeweils die vordersten!
- Wie Segment-Schnitt-Sweep:
  Schnitte nur mit Nachbarn,
  nachgelagerte Events
- Stets nur ersten Schnitt
  einfügen, andere entfernen,
  Anzahl: $O(m)$,
  Einf./Entf. $O(\log m)$

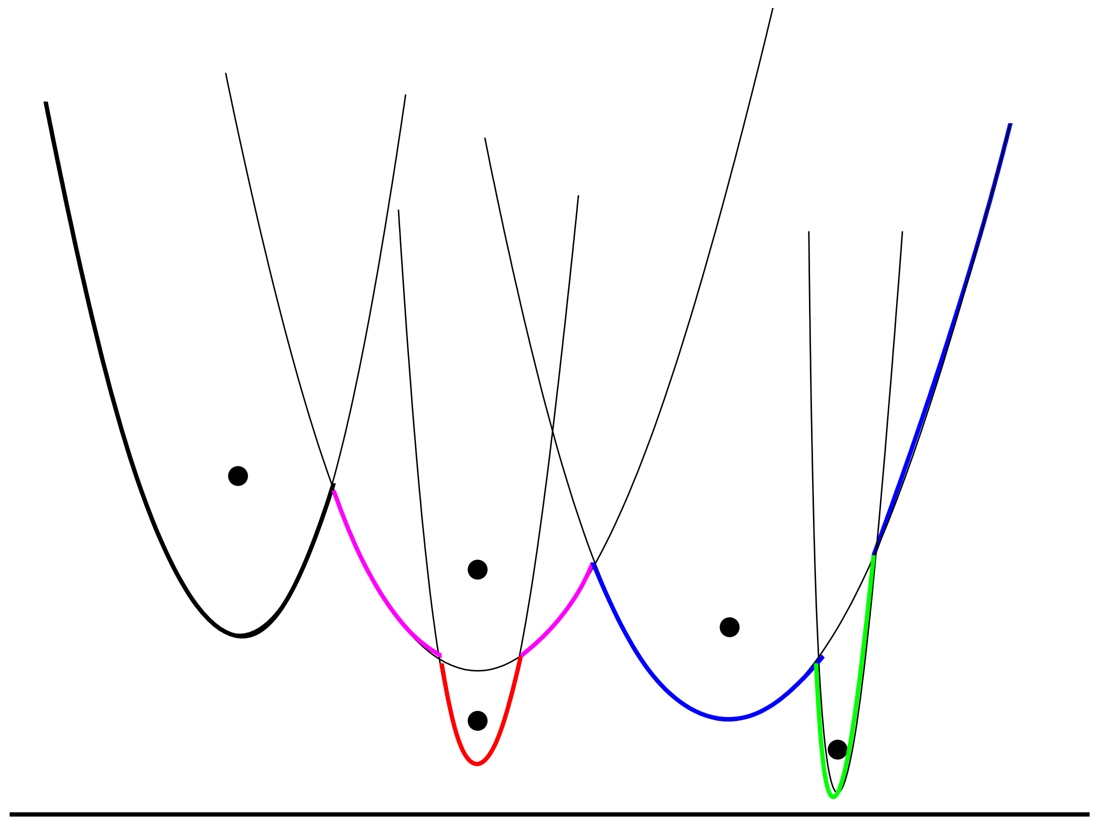


Gesamtlaufzeit/Gesamtkomplexität

- m Parabeln in SSS
- Max. $O(n+m)$ Ereignisse in ES
- Maximal $O(n)$ Ereignisse finden statt: Punktereign./Voronoi-Knoten
- Einfügen/Entfernen Events, Einfügen/Entfernen Parabelen
- n-mal $O(\log(n+m))$ Laufzeit!
- Komplexität der Wellenfront: $m \in O(n)$
- Begründung!

Komplexität der Wellenfront

- n Parabeln, von denen sich zwei maximal zweimal schneiden können
- Definiert auf dem gesamten Intervall
- DSS von n Buchstaben der Ordnung 2: $\lambda_2(n) \in O(n)$



Ergebnis

Theorem 6.11 Das Voronoi Diagramm von n Punkten läßt sich mit dem Sweep Algorithmus in Zeit $O(n \log n)$ und mit Platz $O(n)$ berechnen, das ist optimal.

Beweis: $O(n)$ Events in je $O(\log n)$ Zeit

Korrektheit: Algorithmus schreibt Voronoi-Diagramm links der Wellenfront *in den Sand*

Kapitel Buch

Kapitel 6 Seite 281 mitte – 294 mitte