

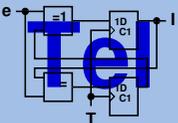
# Zwischenpräsentation

## Belegarbeit

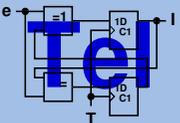
Kerstin Vanselow

s9040180@mail.inf.tu-dresden.de

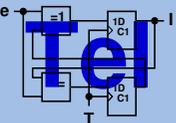
TU Dresden, Institut für Technische Informatik



Untersuchung zur Visualisierung von Partikelströmungen und  
Implementierung der Ergebnisse für den  
Mikrofluidikdesignsimulator SVFS.



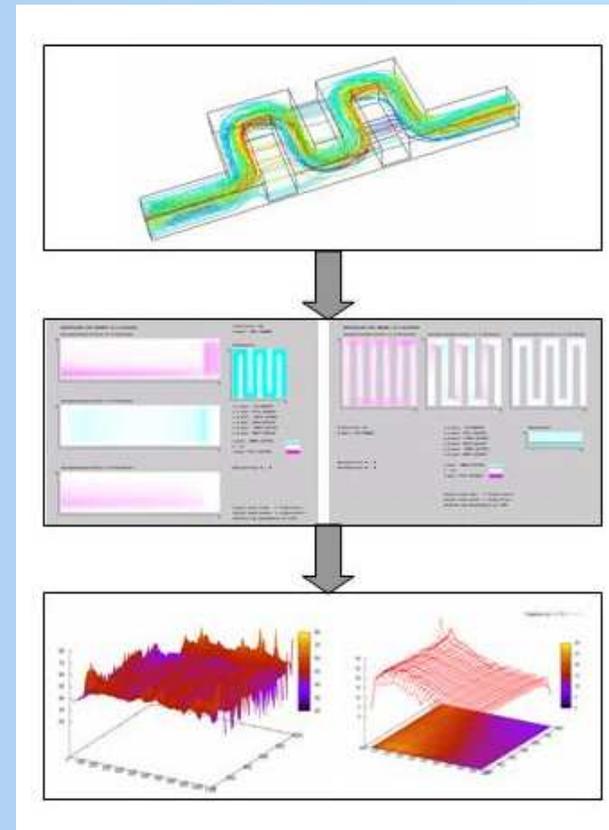
- ◆ Partikel- Simulator SVFS 1.0
- ◆ Konzeption
- ◆ Realisierung
- ◆ Ausblick
- ◆ Quellen
- ◆ Glossar



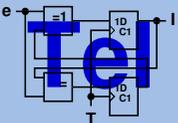
# Anwendungsgrundlage

Ziel ist der Nachweis von verschiedensten Proteinen geringster Konzentration, welche in einem großen Probenvolumen enthalten sind.

- ◆ CFD Software FLUENT
- ◆ Konvertierungs- und Visualisierungssoftware
- ◆ SVFS Simulationssystem

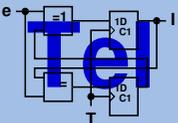


Der Partikel- Simulator SVFS 1.0 ist eine Simulationssoftware zur Ermittlung der Zahl der gebundenen Proteine und der für die Messung benötigten Zeit, ausgehend von der Nachweisgrenze des Sensorsystems, der Konzentration und dem Volumen der Probe [2].



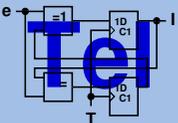
# Simulatorparameter

- ◆ Simulationsergebnisse: optimale Prozessparameter für die größtmögliche Ausbeute an gebundenen Liganden bei schnellstmöglichem Probandurchsatz.
- ◆ Eingabe: unterschiedliche Mikrofluidikdesigns und Prozessparameter.
- ◆ Ausgabe: Anzahl und Position der auf der Sensoroberfläche anbindenden Liganden.

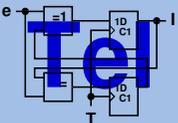


# Simulatorproblematik

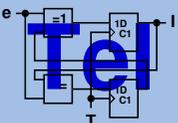
- ◆ Idealfall: Kanalhöhe über Sensorflächen minimal und Strömungsgeschwindigkeit im Bereich der Diffusionsgeschwindigkeit.
- ◆ Einschränkung: technische Realisierbarkeit und kurze Messzeit.



Ziel der Belegarbeit ist die Konzeption und Realisierung von Softwarekomponenten für die Visualisierung, Optimierung und das Überprüfen von Mikrofluidikdesigns.



- ◆ *Korrekte* Darstellung zur Nutzung der Visualisierung für Auswertungszwecke
  - Anhand der Darstellung kann ein vorzeitiger Abbruch einer Simulation bei fehlerbehaftetem Verlauf initiiert werden
- ◆ *Optimierung*, vor allem bezüglich Geschwindigkeit, da eine Echtzeitdarstellung angestrebt wird
  - Lokalisierung und Umgehung beziehungsweise Ausweitung von Bottlenecks



*Physikalisches Problem:*

Kanaldimensionen  $1.100\mu\text{m} \times 1.000\mu\text{m} \times 50\mu\text{m}$

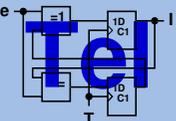
*Mathematisches Problem: (worst case)*

Kanal 1.100.000 Zellen

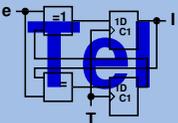
zusätzlich 25.000 Partikel

*Darstellungsproblem:*

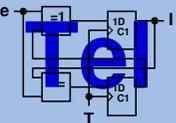
Darstellung von 1.100.000 GL\_QUADS sowie 25.000  
GL\_POINTS



- ◆ Z-Buffer Algorithmen
  - Darstellung nur der sichtbaren Partikel oder Dreiecke des Kanals
- ◆ DisplayList und VertexArray
- ◆ Segmentierung (Clustering)
  - Algorithmen zur Vereinfachung von Meshes

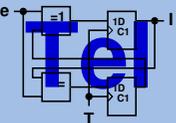


- ◆ Partikelsystem
  - Schnelle Darstellung vieler einzelner Partikel über deren Position
  - OpenGL Umsetzung mit GL\_TRIANGLES oder GL\_POINTS
  
- ◆ typedef struct {  
float xpos, ypos, zpos;  
float r, g, b;  
bool bind;  
}Particle;



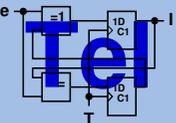
## Gtk+ 2.0

- ◆ bekanntes Widget- Toolkit für X-Window-Systeme
- ◆ Vorteile:
  - programmiert in C
  - vielfältige Sammlung an Widgets
  - gute Dokumentation
- ◆ Alternativen: Qt, Motif, FLTK

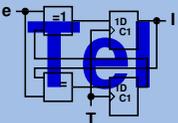


## GtkGLExt 1.2.0

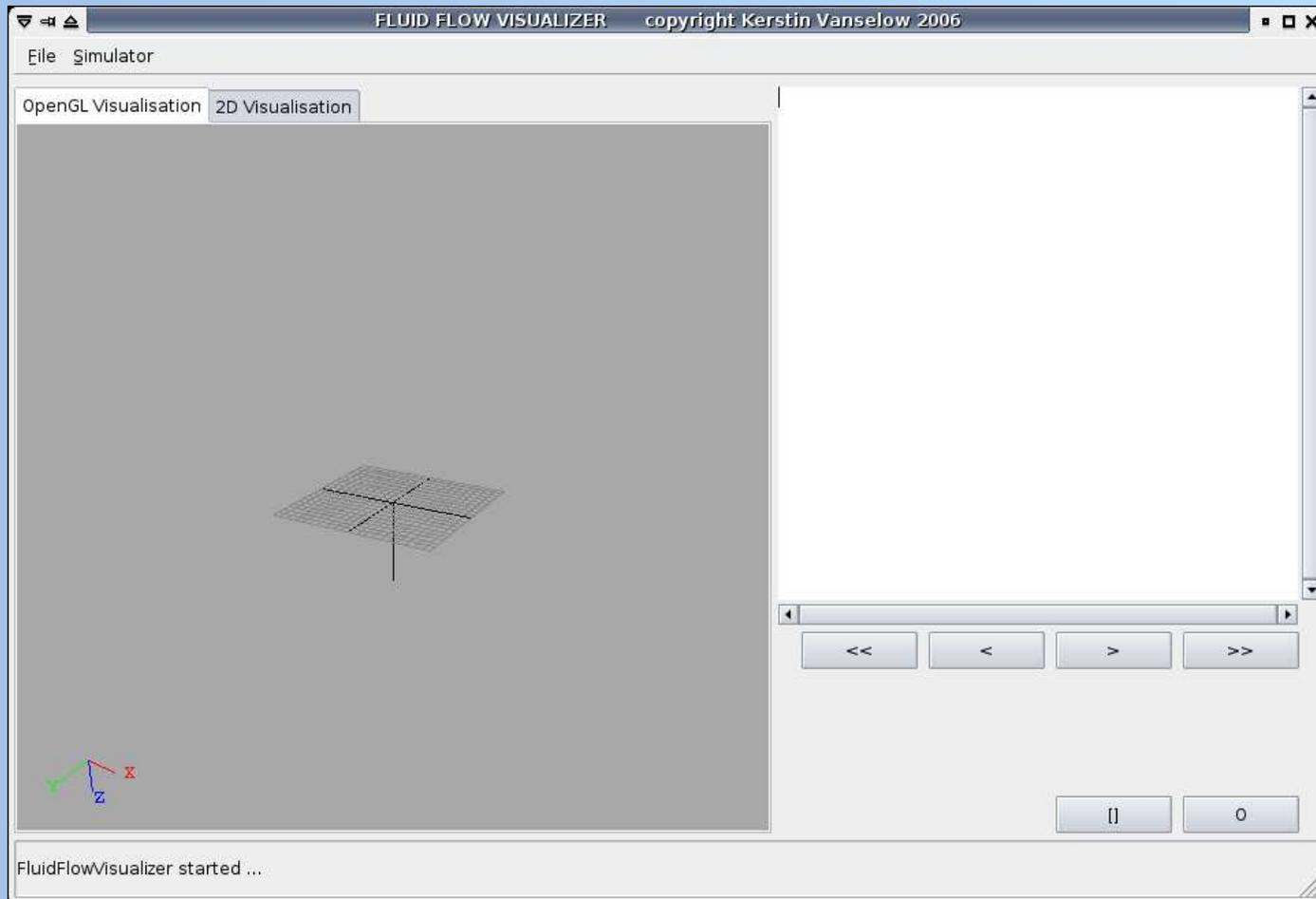
- ◆ OpenGL- Erweiterung für GTK+ 2.0
- ◆ notwendig, da Standard- GTK- Widgets nicht zur Darstellung von OpenGL befähigt sind  
*gtk\_widget\_set\_gl\_capability(...);*



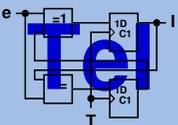
- ◆ Das Pflichtenheft "ist die vertraglich bindende, detaillierte Beschreibung einer zu erfüllenden Leistung, z.B. (...) der Erstellung eines Computerprogrammes" [1].
- ◆ Nach Helmut Balzert wurde ein Pflichtenheft erstellt, um die Anforderung der entstehenden Software genau zu spezifizieren.



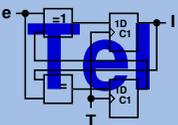
# Realisierung Screenshot



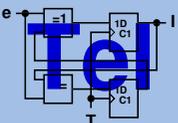
- ◆ Einlesen der Daten für den Partikelstrom, die vom Simulator generiert werden
- ◆ gegebenenfalls Verifikation und Speicherung der Kanaldaten
- ◆ Darstellung des Teilchenstroms im OpenGL- Fenster
- ◆ Auswahl eines Teilchens und Ausgabe seiner Eigenschaften
- ◆ Navigation (räumlich und zeitlich) durch die Simulationsvisualisierung
- ◆ Export der Renderingsequenz in ein Präsentationsformat, wahrscheinlich \*.avi
- ◆ 2D Darstellung zur einfachen Navigation



- ◆ [en.wikipedia.org](http://en.wikipedia.org)
- ◆ [de.wikipedia.org/wiki/Pflichtenheft](http://de.wikipedia.org/wiki/Pflichtenheft) [1]
- ◆ [en.wikipedia.org/wiki/Binary\\_Space\\_Partitioning](http://en.wikipedia.org/wiki/Binary_Space_Partitioning)
- ◆ [en.wikipedia.org/wiki/Particle\\_system](http://en.wikipedia.org/wiki/Particle_system)
- ◆ [en.wikipedia.org/wiki/Z-buffering](http://en.wikipedia.org/wiki/Z-buffering)
- ◆ [www.opengl.org](http://www.opengl.org)
- ◆ [www.gtk.org](http://www.gtk.org)
- ◆ [www.k-3d.org/gtkglext/Main\\_Page](http://www.k-3d.org/gtkglext/Main_Page)

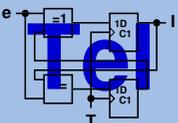


- ◆ *Modellierung, Dimensionierung und Optimierung des Mikrofluidikdesigns für einen Proteinchip auf SPR- Basis zum Nachweis geringster Konzentration in großen Probevolumina.* F.Sonntag und S.Vogelsang. 2006. [2]
- ◆ *Simulation und Charakterisierung von Mikrofluidiksystemen zur mäanderförmigen Strömungsführung in Lab-on-a-Chip Systemen.* F.Sonntag und S.Vogelsang. 2006.
- ◆ *Progressive Meshes.* Hugues Hoppe. Siggraph 1996.
- ◆ *OpenGL superbible.* Richard S.Wright,Jr. and Benjamin Lipchak. Sams Publishing 2005.



Vielen Dank für Ihre Aufmerksamkeit.  
Für Fragen stehe ich gern zur Verfügung.

Kerstin Vanselow



*Diffusion*: Nettostofftransport von Teilchen durch den Übergang von einem Nichtgleichgewicht in ein Gleichgewicht

*Ligand*: ein ungeladenes Molekül

*Mesh*: Netz aus zusammenhängenden Teilflächen

*Vertex*: Eckpunkt eines Polygons

*Widget*: Komponente einer Benutzeroberfläche

