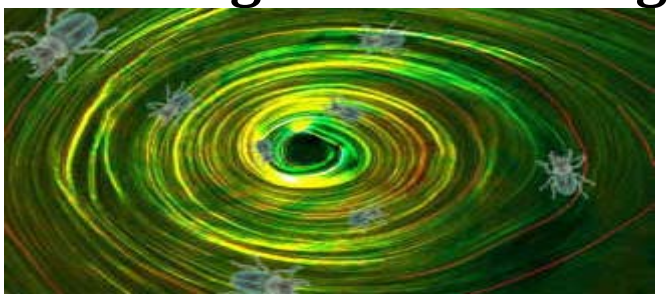
LU Visualisierung - Beispiel 2
Strömungsvisualisierung



Einleitung

Die Datensätze in der Strömungsvisualisierung sind in der Regel größer als in der Volumensvisualisierung, da pro grid-point nicht nur ein einzelner Skalar, sondern mehrere Werte gespeichert werden. Neben einem Vektor, der die Strömungsrichtung angibt, werden auch weitere skalare Werte wie z.B. Druck und Temperatur gespeichert.
Um die Komplexität des zu erstellenden Programms gering zu halten sollen in der Übung nur Algorithmen für zwei-dimensionale Gitter implementiert werden. Die Strömungsdatensätze sollen zumindest mittels Glyphen (z.B. Pfeile), Farbcodierung und einfachen Streamlines dargestellt werden.

Des weiteren soll **eine** der folgenden zwei Aufgaben implementiert werden:

- Evenly Spaced Streamlines:
  Implementieren Sie den Algorithmus von [2] um eine gute Verteilung der streamlines automatisch zu erhalten. Außerdem sollen die Ergebnisse mit den im paper vorgestellten Methoden (tapering, glyph mapping, texture generation) variiert werden können.
- Pong game:
  Die Strömungsdaten sollen verwendet werden um ein Objekt (z.B. einen Ball) über das Strömungsfeld fliegen zu lassen. Weiters sollen vom Benutzer interaktiv bewegte Objekte (z.B. Hindernisse oder ein Schläger) mit dem umherfliegenden Objekt (dem Ball) interagieren können. Das genaue Ziel des Spiels ist frei wählbar und spätestens beim Zwischengespräch mit der Übungsleitung abzusprechen.

Optional können auch noch andere Algorithmen implementiert werden (wie z.B. Line Integral Convolution, Spot Noise, etc.)

Aufgabenstellung

Die Mindestanforderungen bis zum Zwischengespräch

Erstellen Sie einen arrow-plot des Strömungsfeldes, zeichnen Sie vorerst alle Pfeile in einheitlicher Länge. Variieren Sie dann, indem Sie die Stärke des Strömungsfeldes mit entsprechenden Farben darstellen (Legende!) bzw. die Länge der Pfeile entsprechend modifizieren. Es soll des weiteren möglich sein die zusätzlichen Skalarwerte des Datensatzes (Druck, Geschwindigkeit ...) mittels color-coding darzustellen. Versuchen Sie, möglichst aussagekräftige und übersichtliche Bilder zu erzeugen. Weiters sollen streamlines sowohl mit Euler als auch Runge-Kutta Integration zweiter Ordnung erzeugt werden (siehe [1]) und eine Möglichkeit zur Verfügung gestellt werden, diese zu vergleichen.

Die Mindestanforderungen bis zur Abgabe

Entweder

Implementieren Sie den Algorithmus von [2] um eine gute Verteilung der streamlines automatisch zu erhalten. Außerdem sollen die Ergebnisse mit den im paper vorgestellten Methoden (tapering, glyph mapping, texture generation) variiert werden können.
Insgesamt sollen alle vorkommenden Parameter der verwendeten Algorithmen (dsep und dtest, Farben, Dichte, Zoom, ...) interaktiv veränderbar sein.

Oder

Implementieren Sie ein Spiel (ähnlich pong), das die Strömungsdaten zur Steuerung eines Objektes verwendet. Um für den Benutzer das Verhalten des umherfliegenden Objekts vorhersehbarer zu machen sollen geeignete Visualisierungstechniken des Strömungsfeldes verwendet werden. Es soll sich zumindest ein weiteres Objekt auf dem Spielfeld befinden, welches vom Benutzer gesteuert wird und mit dem umherfliegenden Objekt interagieren kann.

Zusätzlich können für die beiden oben genannten Varianten Features wie LIC, spot noise, animated flow, InfoVis-Techniken (wie scatter-plots und brushing), etc. implementiert werden.

Abgabe

Bei der Abgabe werden wie gehabt Fragen gestellt, die das Verständnis der implementierten Techniken überprüfen.
Zur Dokumentation Ihrer Arbeit sollen folgende Dateien und Dokumente erstellt werden:

- Kompilierte Binaries
- Quellcode und Dokumentation (Doxygen)
- Eine Dokumentation des Programms in HTML. Um anderen Leuten die Möglichkeit zu geben das Programm auszuprobieren, soll ein Link auf das Programm zum downloaden enthalten sein.
- Ein Bild mit möglichst hoher Auflösung für den Vis-Contest (s.u.)

Datensätze

Verschiedene Datensätze und Informationen darüber gibt es auf der Homepage zur LU zu finden. Die Datensätze dürfen im Rahmen der VisLU verwendet werden aber nicht weiter gegeben werden!

Vis-Contest

Bei der Präsentation des 2. Beispiels wird das schönste Bild gewählt und der/die Gewinner mit einem Preis belohnt.

Referenzen

[1] "Vector Algorithms", chapter 6.3 in William Schroeder, Ken Martin, Bill Lorensen, "The Visualization Toolkit", pp. 161 - 168, Kitware, Inc., 2002

[2] Bruno Jobard and Wilfrid Lefer, "Creating Evenly Spaced Streamlines of Arbitrary Density", Visualization in Scientific Computing '97, Springer Vienna, 1997, pp. 43 – 54.