

Visualisierung WS2014

Zusammenfassung

Vorlesung 2

Visualization-Pipeline:

Data acquisition → Data enhancement → Visualization mapping → Rendering (3D → 2D)

Wo kann man diese Visualisierungspipeline einbetten?

Computational Sciences = keine Experimente im Labor, sondern virtuelle Experimente. Mit zB virtuellen Mikroskop.

Computational Science = Scientific Computing + Visual Computing
Visual Computing = Scientific visualization + Computer vision + HCI
Scientific Visualization = Volume Visualization + Flow Visualization

Visualization Scenarios

- * Passive Visualization
- * Interactive Visualization
- * Interactive Steering - höhere Komplexität, zusätzliche Vorteile, hier kann auch auf die Datengenerierung zugegriffen werden.

Dateneinteilung:

- * Wo leben die Daten(Dimension), Definitionsbereich auf Wertebereich zB $R^3 \rightarrow R$ (man liest den Skalar), Strömungsdaten zB $R^2 \rightarrow R^2$ 2 Dimensionaler Definitionsbereich und Vektoren als Werte. Verschiedene Attribute sind eine eigene Dimension.
- * Art der Daten (Vektoren, Skalar = numerischer Wert zB Temperatur → man kann Berechnungen durchführen, Non numerical(nominal, ordinal values) , Matrizen), Skalar ist Tensor nullter Ordnung. (weil man null Parameter hat die man variieren kann). Tensor zweiter Ordnung ist eine Matrix(man hat i und j). Tensor erster Ordnung ist ein Vektor. Tensor dritter Ordnung ist eine Menge von Matrizen,
- * Welche Repräsentation macht für gewisse Daten Sinn? inherent spatial domain? which dimension is used what for? 3D auf 1D(CT-data), 2D auf 2D (2D-Flow), 3D -> 3D(Spatial Curve)

Beispiele:

- | | |
|---|--------------------------------------|
| * $N^1 \rightarrow R^1$ value series | bar chart, pie chart, etc. |
| * $R^1 \rightarrow R^1$ function | line graph |
| * $R^2 \rightarrow R^1$ function over R^2 | 2D-heightmap, Konturlinie |
| * $N^2 \rightarrow R^2$ 2D-vector field | Strömungen, LIC |
| * $R^3 \rightarrow R^3$ 3D-flow | Streamlines, Streamsurfaces |
| * $R^3 \rightarrow R^1$ 3D-densities | iso-surfaces in 3D, volume rendering |
| * $(N^1 \rightarrow)R^n$ set of tuples | parallel coordinates, glyphs, icons |

Kartesische Koordinaten gehen auf Rene Descartes zurück.

Gitter:

- * zB. Messpunkte(also die Anzahl mit der ein Objekt abgetastet wird)
- * Wie sind die Daten organisiert?

- * Ist die Nachbarschaft wichtig?
- * Auch Bresenham ist wichtig(Rastarisierung)
- * In 2D Kartesisches Gitter(gleichmäßiger Abstand zwischen Nachbarn, implizite Nachbarschaftsbeziehung, $dx = dy$)
- * Regular Grid ($dx \neq dy$, Abstand zwischen Gitterpunkten)
- * Rectilinear Grid(Abstände sind nicht gleich \rightarrow nicht regulär)
- * Curvilinear Grid(non-orthogonal Grid, Gitterpunkte explizit gegeben, implizite Nachbarschaftsbeziehungen), es ist nicht so einfach wie Punkte angeordnet werden sollen/wie Punkte gewichtet werden.
- * Unstructured Grid - Grid in Form von Tetraeder/Hexahedra, in jedem Tetraeder kann wieder ein Tetraeder sein. Gitterpunkte und Nachbarschaften sind explizit gegeben. Detaillierter in der Nähe des Flugzeuges.

Gestreute Daten(scattered data)

- * Gitter-freie Daten
- * Datenpunkte sind ohne Nachbarschaftsbeziehung gegeben
- * zB Punktwolke, Nachbarschaft wird nicht erhoben \rightarrow Laserscan

Gitter Transformation

- * es kann vorkommen dass man zwischen Gittertypen wechseln muss
- * Fragen: Genauigkeit des re-sampling, design of Algorithms
- * Können auf: physical domain, computational domain und image domain stattfinden

Visualisierung und Farbe

- * Farbe kann Information hervorheben
- * Anzahl der Farben 7 +-2
- * es können ca. 50 - 300 Farbschattierungen unterschieden werden
- * Rainbow color scale ist nicht linear \rightarrow wenn man einen gewissen Bereich weitergeht ist es nicht empfindungsgemäß(nicht überall gleich, es gibt teilweise starke Änderung von grün zu gelb)
- * Farbwahrnehmung hängt stark vom Kontext ab
- * Kanten werden gut erkannt(Mach Band Effekt)

Guidelines für die Verwendung von Farben

- * Kein gesättigtes Blau für details
- * gesättigte Farben sollen nicht vermischt werden
- * Vermeide hohe Farbfrequenzen
- * Redundanz verwenden
- * Farben die etwas vergleichen sollen nah zueinander sein.
- * Farbblindheit

Volume Visualization

- * VolVis - mapping 3D \rightarrow 2D, slicing
- * Volume data - 3D \rightarrow 1D Data, scalar data
- * Voxels vs. cells(1cell hat 8 Voxel)
- * Gradient wird als Normalvektor verwendet
- * Klassifikation - was ist interessant
- * Einfach: Slicing(keine Verdeckung) , Multi Planar Reconstruction(Ansicht von allen Seiten)
- * Direct volume visualization: image order vs object order, Raycasting, Alpha-Compositing
- * zB Computational fluid dynamics
- * Cartesian oder reguläre Gitter vs. Curvi-linear grid oder unstructured Grid(zB Tetraeder)

Vorlesung 3

Rendering Projection - so viel Information und so wenig Pixel

Zwei Arten wie Daten/Gitter interpretiert werden können:

- * Voxel - keine lineare Interpolation, Voxel ist ein Samplepunkt in 3D, man bekommt Unstetigkeiten.
- * Zelle - Werte in Zelle: Interpolation wird verwendet, Ecken sind 8 Voxel, Cell = primitiver Würfel

Interpolation

- * Am einfachsten Nearest Neighbor(wesentlich schlechtere Qualität als zB Trilineare Interpolation, man bekommt starke Artefakte)
- * danach lineare Interpolation(Dichte Wert eines Punktes wird durch die Fläche gegenüber/diagonal des Punktes erfasst). Eine bilineare Interpolation ist keine lineare Interpolation(denn wenn man quer über die Fläche geht entsteht eine gekrümmte Linie/Fläche).
- * Bilineare Interpolation: Vier Punkte A,B,C,D und zwei Parameter u,v mit $0 \leq u \leq 1$ und $0 \leq v \leq 1$. Im wesentlichen macht man zuerst die lineare Interpolation zwischen A und B bzw. zwischen C und D danach interpoliert man die beiden Ergebnisse noch einmal.

Higher-Order Reconstruction

- * Wenn sehr hohe Qualität benötigt wird
- * Marschner-Lobb Signal wird zum testen des Signals verwendet
- * Windowed sinc bringt gute Ergebnisse(es gibt aber Rimming Artefakte über/unter schwingen), mit sinc wären ideale Rekonstruktion möglich(ist aber unendlich)

Gradienten in Volumsdaten

- * Man hat bei Volumsdaten keine Flächen sondern nur Punkte ohne Normalvektor \rightarrow deswegen Gradienten.
- * Gradient zeigt in Richtung des höchsten Anstiegs, Isolinien zB bei Berg sind Normalvektoren.
- * Isofläche: wenn man entlang der Tangenten geht ändert sich nichts.
- * Es gibt Forward differencing, backward differencing und Intermediate differencing.

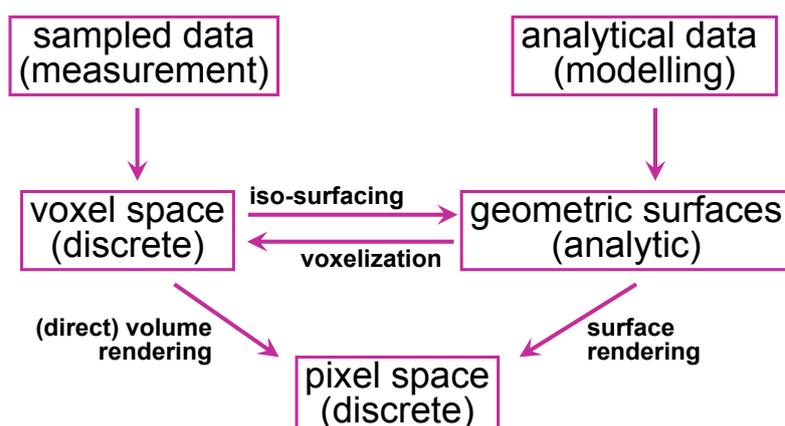
Klassifikation

- * Man hat Daten und an jeder Stelle einen Dichtewert
- * Segmentierung: Teilung des Raums in Bereiche

Transferfunktion

- * Besteht aus zwei Teilen: ein Dichtewert wird eine Farbe und eine Opacity/Durchsichtigkeit(0 = Durchsichtig) zugeordnet.
- * Transferfunktionen haben großen Einfluss auf Ergebnisbild

Konzepte und Begriffe:



- * zB X-Ray Modelling
- * Surfacedefinition
- * Sampling (voxelization) combination
- * Direct Volume Rendering

Slice rendering:

- * Man nimmt Schicht(2D) eines 3D Volumens

Iso-Surface rendering:

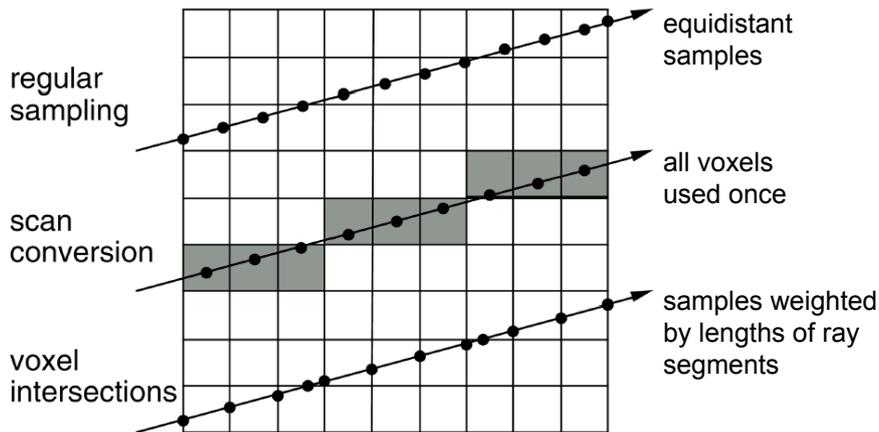
- * Indirekt, man sieht keine inneren Details

Volume Rendering:

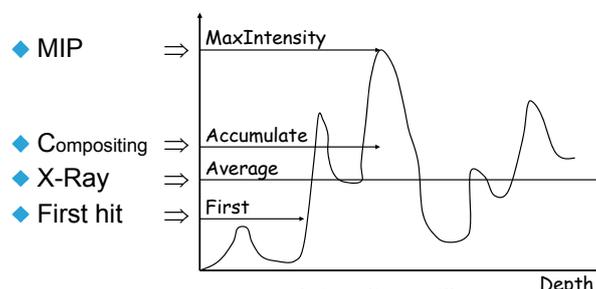
- * man sieht alle Daten
- * es kann zu Verdeckungsproblemen kommen.

Volume Visualization Techniken:

- * Slicing, reguläre Gitter → einfach, Achsen-parallele Schnitte, eventuell kann noch transfer Funktion verwendet werden(Farbe)
- * Splatting - es interessiert nur das Ergebnis Bild, man nimmt eine Zelle betrachtet sie wie ein Schneeball und wirft sie auf das Ergebnisbild. Sie betrachtet die Voxel als kleine Farbkügelchen, die auf die Bildebene geworfen werden und dort zerplatzen. Für normale Gitterstrukturen (Entwicklung Grafikhardware) ist Ray Casting besser.
- * Ray Tracing: Bei Volumen rendering → viewing rays. Dabei werden die Strahlen mit Objekten geschnitten.
- * Ray Casting - Levoy 1988 erster Standard(keine Sekundärstrahlen)
 - Voxel Values (Shading Classification) → Voxel Colors/Voxel Opacities (RayCasting/ Resampling) → Sample Colors/Sample Opacities (Compositing) → Pixel Colors Es werden die Strahlen mit den Dichtewerten in 3D geschnitten.
- * Ray Traversal - Volumen ist auf kartesischem Gitter, Strahl wird auch mit regelmäßigen Abtastpunkten verwendet. Die Punkte haben also einen gleichmäßigen Abstand.



Kombinationstypen



Klassifizierungs Reihenfolge

Vor oder nach dem Shading?

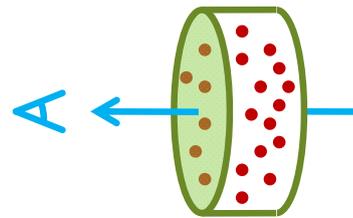
- * Pre-Interpolative
- * Post-Interpolative - liefert genauere Resultate, ist aber aufwändiger

Alpha - Compositing

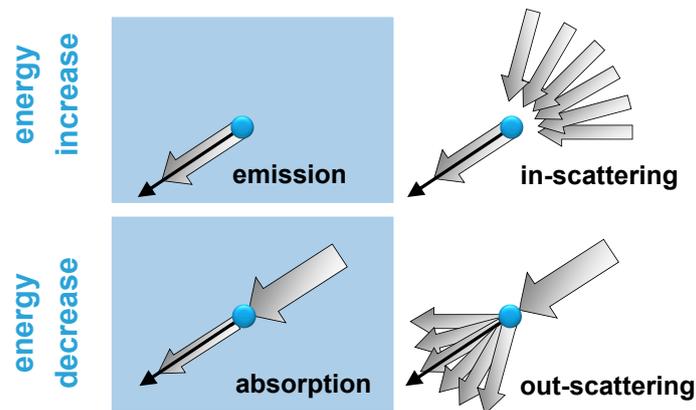
- * Alpha Blending wird in Computerspielen und anderen Grafikanwendungen benutzt, um die Transparenzeigenschaften von Objekten zu modellieren.

Globale Beleuchtung - Luft ist nicht Durchsichtig

- * Emission (light particles)
- * Absorption (dark fog, Verdeckung)
- * Emission und Absorption (Wolken)
- * Streuung
- * Man sieht sich die Partikel um einen Zylinder an und bestimmt dadurch die Werte

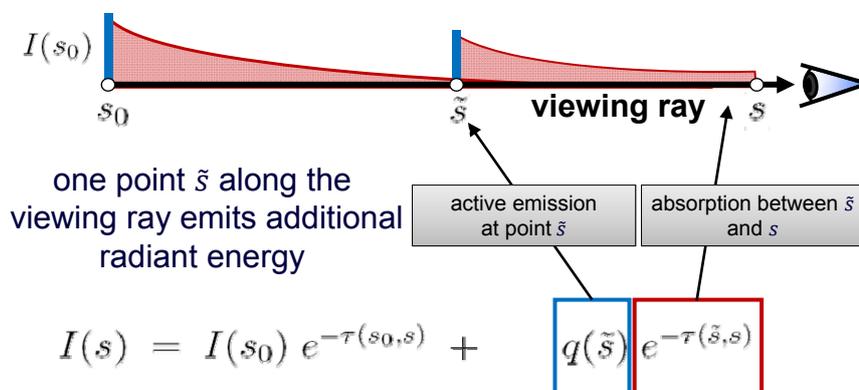


Physical Model of Radiative Transfer



Analytic Model

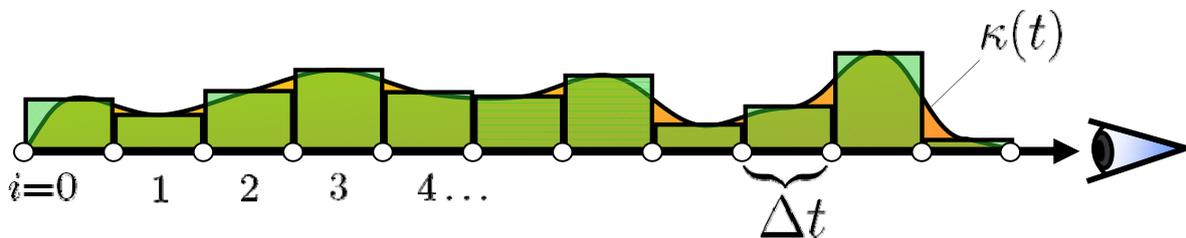
- * Licht schwächt sich über die Distanz ab $e^{-\tau}$
- * je höher die Anzahl der Partikel desto höher ist die Abschwächung
- * Hinterer Teil wird zusätzlich addiert(Integral)



Vorlesung 4

Absinth + Wasser = Streueffekte(Physikalischer Effekt)/Louche-Effekt

Numerische Approximation



- * Dichtewert entlang des Strahls wird mittels Riemannsummen approximiert. → Integral wird zur Summe
- * Messpunkte werden gleichmäßig verteilt
- * es wird Farbe und Durchsichtigkeit approximiert
- * Back to Front Compositing: Die Voxel werden vom Hintergrund zum Betrachter durchlaufen. Der für die vorherigen Voxel berechnete Farbwert muss jeweils mit der Transparenz und der Farbe des aktuellen Voxels verrechnet werden(akkumuliert werden). Funktioniert wie ein Paintersalgorithmus → man sieht den Aufbau.
- * Front to Back Compositing: Die Voxel werden vom Betrachter(Augpunkt) aus zum Hintergrund hin durchlaufen. Die Farbe weiter hinten liegender Voxel geht entsprechend kaum noch in die Gesamtfarbe ein, so dass der Algorithmus bei Unterschreitung eines bestimmten Schwellenwertes abgebrochen werden kann. Man muss sich merken welche Durchsichtigkeit und welche Farbwerte es gibt, es muss sich also mehr gemerkt werden. Ergebnisse sind bei beiden Verfahren gleich, beide können als Alphablending bezeichnet werden.
- * Numerische Lösungen: Die Farbe kann mit dem Alpha-Wert vor multipliziert werden. → Vereinfachung(Problem: Transparentes Rot == Transparentes Schwarz) → Verlust von Präzession

Ray Casting - Beispiele

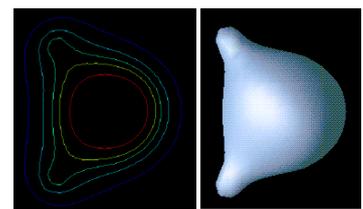
- * Tornado Visualisierung
- * CT-Scans
- * Molekulare Daten

Hardware-Volumen Visualisierung

- * 3D-Texturen, die Grafikkarte ist auf Computerspiele optimiert, dabei sind Texturen sehr wichtig. Im Texturspeicher der Grafikkarte können Volumendaten gespeichert werden. Die Volumendaten werden dem Viewport angepasst.
- * 2D-Texturen - Man kann nur entlang einer Achse Volumendaten speichern.

Indirekte Volumen Visualisierung - Iso Surface Display

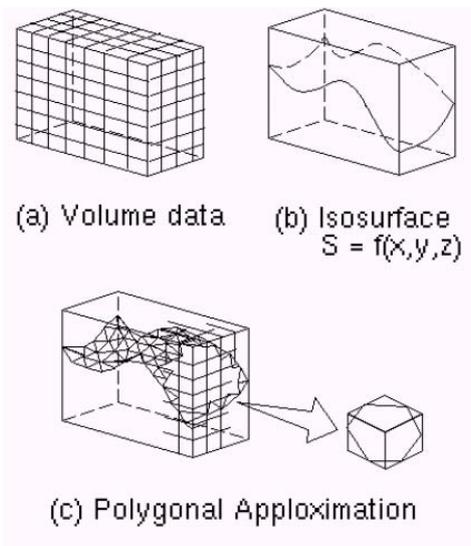
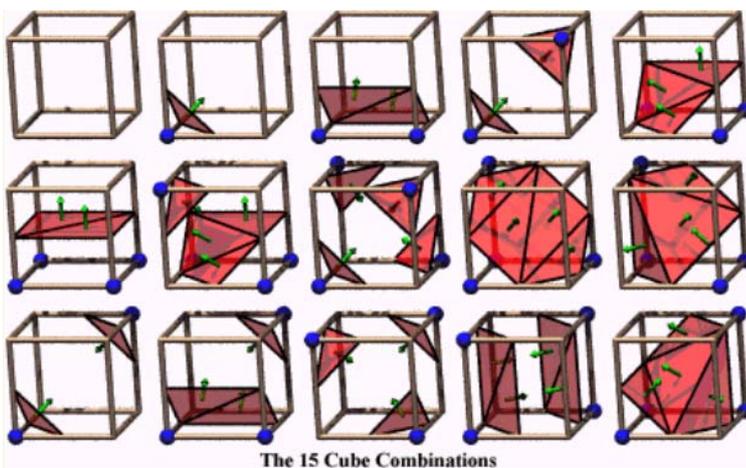
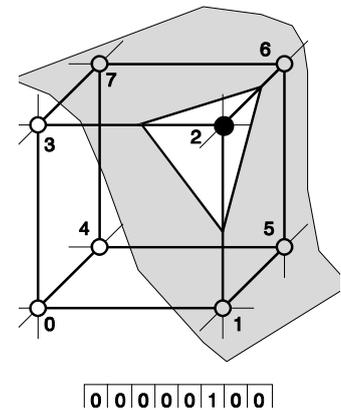
- * Isoflächen sind Flächen, die im Raum benachbarte Punkte gleicher Merkmale oder Werte einer bestimmten Größe wie zum Beispiel Temperatur oder Dichte miteinander verbinden.
- * Indirekte Verfahren visualisieren das Volumen mit Hilfe einer polygonalen Zwischenrepräsentation. Es wird also eine Zwischenstruktur benötigt → indirekt
- * Marching Cubes zählt zu diesen indirekten Verfahren, beruht darauf dass sich die Isofläche die man benötigt lokal berechnet wird.



- * Die Isofläche(Iso-Surfaces) wird erzeugt indem in einem Querschnitt alle Punkte mit dem gleichen Dichtewert verbunden werden.
- * Dazu braucht man starke Übergänge(aussagekräftige Isowerte) —> sind in der Medizin vorhanden(zB Knochen zu Weichteile), bei Strömungen nicht so leicht möglich.

Marching Cubes

- * Gestalt der Zelle nur aufgrund der Eckpunkte
- * ist ein Algorithmus zur Darstellung von Isoflächen in der 3D-Computergrafik. Er nähert eine Isofläche durch eine Polygongrafik an.
- * Die Idee von Marching Cubes ist es, das gegebene Voxelmodell eines Objekts zunächst in kleine Würfel (cubes) zu zerlegen und anschließend von einem Würfel zum nächsten zu „marschieren“ und zu bestimmen, wie die Oberfläche des Objekts den jeweiligen Würfel durchschneidet.
- * Man vergleicht die Eckpunkte mit den Isowerten, darüber oder darunter.
- * Algorithmus:
 1. Erstelle einen Würfel
 2. Klassifiziere jeden Voxel, überprüfe ob Fläche innerhalb des Würfels ist: Wert > Iso Wert -> Punkt liegt ausserhalb
 3. Erstelle einen Index - verwende binäres Labeling innerhalb = 1, außerhalb = 0, es gibt 256 verschiedene Möglichkeiten wie Würfel durchlaufen werden kann. —> das kann auf 15 vermindert werden.(durch Ausnützung von Symmetrie)
 4. Lookup Edge List 256 auf 15 reduzieren, man kann mit 0 bis 4 Dreiecken auskommen.
 5. Oberflächenschnittpunkt für jede Dreiecks-Kante interpolieren. Man bekommt innerhalb der Zelle den Ort an dem der Schnittpunkt liegt.
 6. Berechne die Normale für jede Ecke(Vertex) des Würfels
 1. Problemfälle: zweideutige Fälle: es können Löcher entstehen, Lösung: Entscheidung für einen Fall. Man bekommt konsistente Lösung(keine Löcher) aber es können Fehler gemacht werden. Oder man löst es exakt —> Asymptotische Entscheidung(Hyperbeln werden zur Entscheidung verwendet, Signum), Es gibt insgesamt 6 Problemfälle.



Vorlesung 5

Strömungsvisualisierung - man möchte sich zum Beispiel den Strömungskanal ersparen, es gibt aber auch Experimental flow visualization -> es werden etwa Partikel eingespritzt und deren Geschwindigkeit gemessen. Wichtige Frage ist auch wie Verdeckungen/Tiefeneindruck der Strömungslinie visualisiert wird. Kartoffelstempel ist wie Spotnoise.

Woher kommen die Strömungsdaten?

- * Strömungssimulation
 - * Flugzeug- / Auto - Entwicklung
 - * Wettersimulationen
 - * Medizin
- * Strömungsmessung
 - * Windtunnel, Flüssigkeitstunnel, Schockwelle von Ultraschall
 - * Schlieren-, Schattentechniken
- * Strömungsmodelle
 - * Gewöhnliche Differenzialgleichungen

2D vs. Oberfläche vs. 3D

- * 2D-Strömung - 2D x 2D auf einer Schicht
- * Strömung auf einer Oberfläche, zB auf Oberfläche eines Flugzeugs, abseits davon ist die Strömung uninteressant.
- * 3D-Strömung

Stetige(zeitlich unabhängige) vs. Zeit-Abhängige Strömung

- * Stetig - Strömung bleibt gleich über die Zeit - Weniger Speicher wird benötigt, man hat an jeder Stelle des Gitters einen Vektor
- * Zeitabhängig (unstetig) - Wesentlich komplexere Wechselbeziehung

Direkte vs. Indirekter Strömungsvisualisierung

- * Direkt: die Strömung wird direkt dargestellt zum Beispiel mit Pfeilen
- * Indirekt: Objekte wie Strömungslinien werden berechnet und dargestellt

Experimentelle Strömungsvisualisierung

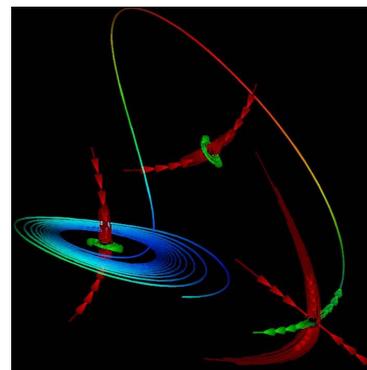
- * Injektion von Farbe, Rauch oder Partikel
- * Optische Methoden: Schlieren, Schatten

Particle Image Velocimetry

- * Die Geschwindigkeit von einem Partikel wird zu zwei aufeinanderfolgenden Zeiten gemessen.
- * Partikel werden durch ein Sieb möglichst gut verteilt. Durch ein Lightshield(Stroboskop) werden die Partikel erfasst.

Dynamisches System Visualisierung

- * zB Lorenz-system, dabei tritt komplexes Verhalten auf, zB gibt es anziehende und abstoßende Punkte(Pendel), sollte anschaulich machen, dass im atmosphärischen Strömungsbild kleine Ursachen große Wirkung zeigen können(Schmetterlingseffekt).
- * Man mach Linearisierung —> man nimmt die partiellen Ableitung und bekommt eine Matrix, diese Matrix beschreibt wie die Strömung in einem gewissen Bereich ist.
- * Sattelpunkte sind interessant aber schwierig in Griff zu bekommen.
- * Strömungsverhalten kann auch handgezeichnet visualisiert werden

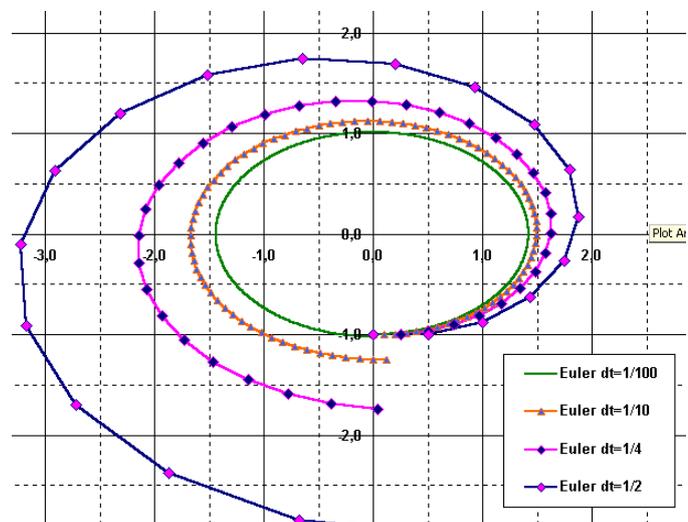


Strömungsvisualisierung mit Pfeilen

- * einfachste Variante
- * Direkte Art Strömung zu Visualisierung
- * Wie lang werden Pfeile gewählt? Normalisierte Pfeile vs. Pfeile die über die Geschwindigkeit ihre Größe verändern
- * Sinnvoll im 2D Bereich
- * problematisch in 3D(durch Perspektive nicht eindeutig), Verbesserung: es können 3D Pfeile verwendet werden(Zylinder mit Kegel) oder die man verwendet Schichten. Eine Möglichkeit ist es Pfeile in 3D Modell nur auf Erdoberfläche anzuzeigen.
- * wird oft verwendet
- * Pfeile können zusätzlich mit Farbe codiert werden

Strömungslinie

- * Definiert als Kurve, wobei an jeder Stelle die Tangentialrichtung dem zugrundeliegenden Strömungsfeld entspricht
- * Man möchte nicht einen Pfeil haben sondern eine Linie
- * Man beobachtet einen bewegten Partikel wie er sich von einem Ort zum nächsten bewegt → das ist die Strömungslinie
- * Strömungslinien in 2D sind gut um sich Übersicht zu verschaffen, es ist auch möglich an der Spitze die Linie dicker zu zeichnen.
- * Man macht eine Linearisierung und nimmt an dass sich in einem ganz kleinen Zeitraum der Partikel nicht bewegt. → damit verkommt das Integral zu einer Summe(Euler Integral), Je kleiner die Schrittweite gewählt wird desto kleiner ist der Fehler(bsp. Ellipse). Problem: Fehler wird immer in eine Richtung gemacht → Extrembeispiel ist die Ellipse, bei Kurven sind die Ergebnisse besser.
- * Bei einer Schrittweite dt von $1/100$ ist der letzte Punkt wieder fast im Ursprung.(2% entfernt), man benötigt aber 889 Schritte.
- * Verbesserung von Euler ist Runge-Kutta Ansatz, Fehler wirkt sich im Quadrat aus(Runge-Kutta Verfahren zweiter Ordnung), aber Aufwändiger, man verwendet einen sogenannten Preview Vektor. Der Vektor wird in der Mitte unterteilt.
 1. Man macht halben Eulerschritt
 2. Evaluere den Flussvektor
 3. Verwenden ihn im Ursprung
- * Runge Kutta ist mit 9 Schritten($dt = 1$) besser als Euler mit 72 Schritten($dt = 1/9$)



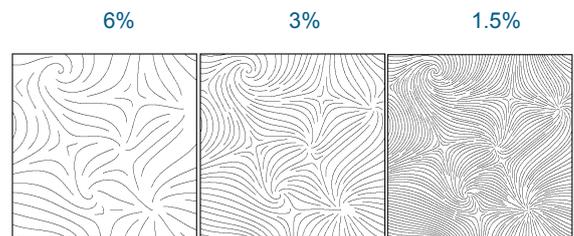
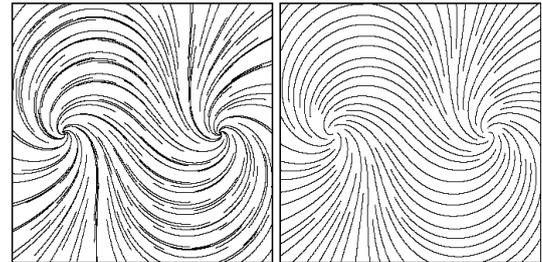
Vorlesung 6

Visualisierung mit Partikeln

- * Partikelpfad = Strömungslinie
- * Es gibt zwei Variante(für zeitabhängige Daten):
 - * Pfadlinien: Langzeit Pfad eines Partikels
 - * Schlierenlinien(streak): Man lässt mehrere Partikel aus, man verfolgt die Gruppe von Partikeln.
- * Für stetige Strömungen sind die Ergebnisse für beide Varianten gleich
- * Strömungslinien in 3D können auch als Sweeps dargestellt werden
- * Beleuchtung von Strömungslinien, im 3D Raum werden nur wichtige Linien beleuchtet.

Platzierung von Strömungslinien

- * Problem: wenn ein gleichmäßiges Gitter verwendet wird kommt es zu ungleichmäßigen Resultaten
- * Deswegen benötigt man einen Füllalgorithmus(nicht mit Farbe sondern mit Strömungslinien) → Strömungslinien sollen nicht zu nah beieinander liegen.
- * Ansatz Algorithmus: Man legt eine Entfernung/Distanz fest, dann überlegt man sich von wo man noch eine Strömungslinie loslassen kann.(Das soll kein Bereich sein indem schon viele Strömungslinien gezeichnet wurden.) Man nimmt einen Startpunkt der mindestens eine gewisse Distanz hat(d_{sep}), dann rechnet man sich für diesen Startpunkt eine Strömungslinie aus(aus Symmetriegründen Vorwärts und Rückwärts Integration, also in und gegen die Pfeilrichtung). Diese Strömungslinie wird so lange verfolgt/gezeichnet bis sie zu nahe(d_{test}) an eine andere Strömungslinie kommt.
- * Abbruch der Strömungslinie:
 - * wenn zu Nah an Nachbarn
 - * wenn außerhalb des Rechteckigen Bereichs
 - * wenn Strömung in Fixpunkt kommt(Strömung = 0)
 - * Wenn sich die Strömungslinie selbst zu nahe kommt
 - * Nach einer maximalen Anzahl von Schritten
- * Strichstärke kann auch in Abhängigkeit der Abstände verändert werden
- * Um die Orientierung besser zu erkennen können auch Glyphen verwendet werden.



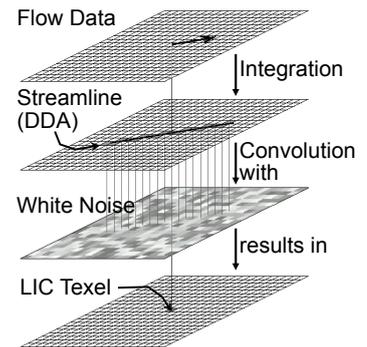
Objekte in 3D

- * Streamribbons: Es kann ein Strömungsband(Streamribbon) verwendet werden, dadurch könne auch Verwirbelungen dargestellt werden
- * Streamsurface: Man lässt eine Linie los und schaut wie sie sich durch Oberfläche bewegt. Nicht nur Partikel wie bei Streamlines.
- * Stream Arrows: Pfeile(können auch mit Textur belegt werden)
- * Flow Volumes(Sammlung von Farbpigmenten): Man geht nicht von einem einzigen Startpunkt aus sondern von einer kleinen Kugel und zu jedem inneren Punkt der Kugel berechnet man sich eine Strömungslinie, dadurch bekommt man eine Vielzahl von Linien, wenn zu viele → Dichteberechnung(wie viele Strömungslinien gehen durch Voxel durch?)
- * Streamtubes ähnlich zu Streamribbons, es wird aber kein Band losgelassen sondern ein Sweep(zB. 5-Eck/3-Eck)



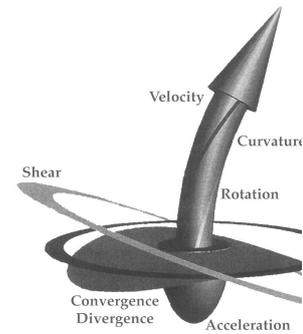
Line Integral Convolution - Strömungsvisualisierung in 2D oder auf Oberflächen

- * In 3D nicht wirklich gut anwendbar, wegen Überdeckung
- * Strömung wird als Textur visualisiert
- * Feldrichtungen werden überall angezeigt
- * Funktioniert wie Tintentröpfchen(White Noise) die auf Blatt Papier aufgetragen werden. Oder Eisenspäne mit Magnet
- * Um eine LIC-Darstellung für solch ein Strömungsfeld zu erhalten, wird ein zweidimensionales weißes Rauschen entlang der Linien des Strömungsfeldes mit einem Kern mit lokalem Träger gefaltet (im einfachsten Fall: lokal gemittelt). Außerdem wird das Bild mit einem Weichzeichner(Gauß Filter) gefaltet.
- * Die Pixelwerte entlang der Feldlinien sind dadurch stark korreliert, orthogonal dazu hingegen - durch das Rauschen - fast unkorreliert. Dadurch heben sich die Feldlinien optisch vom Hintergrund ab und werden sichtbar.
- * Kann auch in 3D gemacht werden, ein Problem ist aber die Verzerrung
- * Alternativen: Spot-Noise(Ellipsen werden entlang der Strömung gelegt), Vector Kernel, Line Bundles/Splats(man hat Textursplats)
- * Nachteil: Aufwändig, Richtung wird nicht dargestellt.



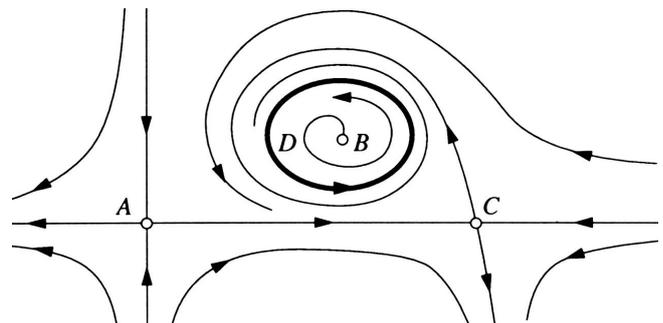
Glyphs resp. Icons

- * Man hat bei einer Strömung verschiedenste Parameter zB. Druck, Rotationsverhalten, Beschleunigungsverhalten
- * zB. durch Ringe oder Scheiben, Länger oder Orientierung
- * Man versucht auch Verwirbelungen zu bestimmen



Flow Topology

- * Eine karge Visualisierung
- * abstrakte Struktur einer Strömung
- * Möglichkeit eines Zyklus, man startet an Ort und kommt wieder zurück.
- * Sattelpunkte sind gefährlich.
- * Es gib Separatritzen zB Wasserscheiden, damit kann das Strömungsfeld in Bereiche unterteilt werden in denen das Verhalten gleich ist
- * Lorenz System: In 3D kommen noch seltsame Attraktoren(ein Fraktal, das nicht in geschlossener Form geometrisch beschrieben werden kann, Abstoßend/Anziehend) hinzu.
- * Timesurface: Veranschaulicht eine Menge von Punkten wo nicht die räumliche sondern die zeitliche Entwicklung im Vordergrund steht. Dabei sieht man Scherung.
- * Auf Oberflächen gibt es folgende Topologien:
 - * Feste Punkte
 - * Separatritzen



Vorlesung 7

Informationsvisualisierung(Hans Rosling Videos)

„The use of computer-supported, interactive, visual representations of abstract data to amplify cognition“

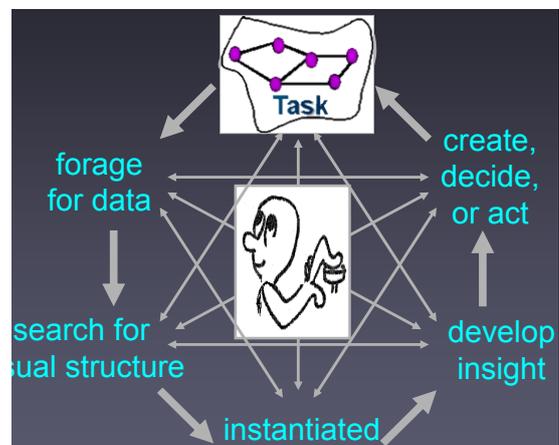
Mensch ist stark im erkennen von Mustern (Preattentive Erkennung). Deswegen können zB Ziffern die andere Farbe haben viel schneller erkannt werden → schneller gezählt werden.

- * Nomograph: ist ein Diagramm, an dem Werte einer mathematische Funktion näherungsweise abgelesen werden können. Die Genauigkeit, mit der die Funktionswerte abgelesen werden, hängt von der Genauigkeit ab, mit der die Markierungen ablesbar sind.
- * Diagramme: Grafische Darstellung die Zusammenhänge Visualisieren.

Externe Hilfsmittel(zB Kerben in Holz ritzen) → Informationsdesign → Visualisierung(Computer basierte) → SciVis(physikalische Daten, 2D/3D) oder InfoVis(abstrakte Daten, zB Datenbanken)

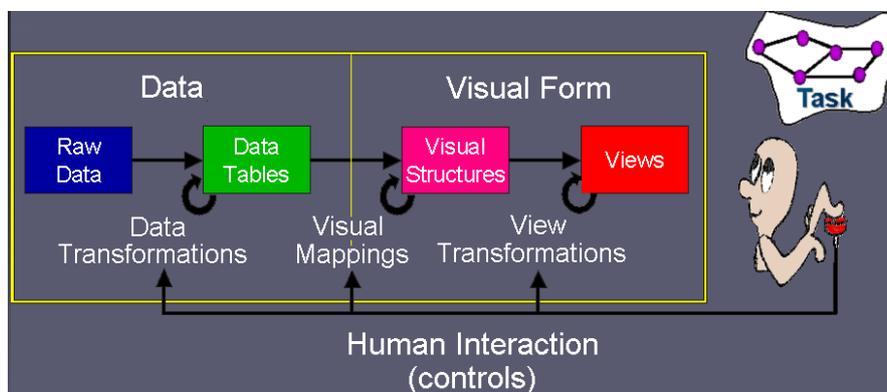
Knowledge Crystallization

- * Wenn Optimierungsfunktion nicht eindeutig ist dann braucht man zur Entscheidung einen Menschen, zB Wohnungssuche
- * Übersicht, Zoom, Filter, Details, Browse, Search Query
- * Record, Cluster, Detect Pattern
- * Create, Delete, Manipulate, Read pattern, Read compare
- * Extract, Compose
- * zB. Dynamic HomeFinder, man kann die guten Wohngegenden mit Slidern schnell finden
- * Table Lens Tool: Man bekommt schnell einen Überblick und kann dann hineinzoomen/ausklappen, dadurch muss man das Fenster nicht immer hin und her schieben. → Aggregation (Zusammenfassung) der Daten.
- * Cost Structure: Wie schnell können Aufgaben gelöst werden, welche Fehler treten auf.



InfoVis Referenzmodell

- * sieht ähnlich aus wie die Visualisierungspipeline
- * man geht von den gewonnenen Daten aus

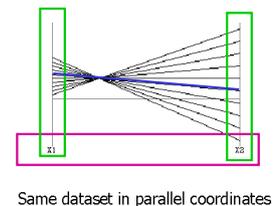
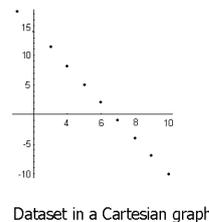


- * RawData → Data Tables: zB Twitter, Plagiate. Es kann zB in standardisierte Form gebracht werden zB Word Vectors. Wenn zwei Dokumente ein ähnliches vorkommen von Wörtern haben dann werden sie als ähnlich bezeichnet. Es kann auch nach Sätzen, Absätzen, Kapiteln, Buchstaben, Bildern kategorisiert werden.

- * Raw Data Probleme: Inkonsistente Darstellung von Daten, Fehlende Daten(Lasse ich sie aus oder kennzeichne ich sie als fehlend), verschiedene Tabellenstruktur
- * Data Transformation: Daten können Nominal(Name), Ordinal(zB Sozialversicherungsnummer) oder Quantitativ(Berechnungen können durchgeführt werden, sqrt(PLZ) macht wenig Sinn) sein.
- * Visual Mappings:
 - * Expressiveness: Durch die Abbildung auf ein visuelles Element werden gewisse Charakteristika der Daten zum Ausdruck gebracht, es sollen aber nur die Inhalte der Daten zum Ausdruck gebracht werden.
 - * Effectiveness: Hat mit Kosten zu tun, die visuelle Representation soll die Lösung eines Problems rasch ermöglichen.
 - * Man kann Marks(Punkte, Linien, Bereiche) verwenden
 - * Verbindungen können verwendet werden
 - * Die Anordnung der Achsen ist wichtig: Composition(zB Scatterplots), Overloading(man verwendet den Platz für zwei verschiedenen Darstellungen), Folding, Recursion(zB 2x2)
 - * Koordinatenachsen müssen nicht orthogonal zueinander sein zB. Starplot.
- * Visuelle Strukturen: 1D,2D,3D oder Multivariable(>3D, Chernoff Faces, 3D Streudiagramm, Charles Minard: Napoleon Karte, Parallele 2D Achsen-> man brauch geschulte Nutzer), Bäume, Netzwerke
- * 2D Visual Structures: Scatterplots zB Spotfire. Vorteil: Mensch erkennt die Muster

Multivariable Data

- * Parallele Koordinaten: wichtig ist dass man Interaktionen damit machen kann, zB Achsen vertauschen oder Auswahl von einem Wert = Brushing(0 oder 1), Skalierung der Koordinatenachse. Smooth brushing = teilweise Auswahl(Farbintensität zwischen 0 und 1)
- * Bild rechts = keine Korrelation.



Bäume

- * repräsentieren hierarchische Strukturen
- * Planarität = keine Überschneidung von Kanten
- * Klarheit = die Kanten sollen möglichst kurz sein, möglichst wenig Kurven machen oder nicht gekrümmt sein ...
- * Saubere Darstellung
- * Hierarchische Darstellung in eine Richtung
- * 2D Baum/3D Baum(Cone Tree), Balloon View(Wurzel in der Mitte, Nachfolger kreisförmig herum), Radial Layout
- * Tree Maps: Venn Diagramm(rund —> schlechte Raumausnutzung), Tree Diagramm, Nested Treemap, Treemap(Hierarchie ist nicht mehr rekonstruierbar, aber Raum ist gut ausgenützt, zB. es zeigt an welche Dateien auf der Festplatte viel Platz beanspruchen)

Netzwerke

- * Werden für Netzwerke, U-Bahn, Telefonsysteme und das Internet verwendet
- * Beispiel U-Bahn in Knotenpunkten wird mehr Platz beansprucht, deswegen werden die Daten verzerrt. Außerdem sind die Richtungen verändert. Wichtig ist aber das die Topologie vorhanden bleibt.
- * Metabolische Netzwerke: Netzwerke die zeigen wie Stoffe vom Körper verarbeitet werden.(sehr komplex, man kann zB Kanten weglassen)

View Transformationen

- * Wen Daten nicht nur statisch untersucht werden sollen
- * Overview + Detail
- * Zooming
- * Fokus + Context(ausgewählte Daten werden genauer bezeichnet)

Vorlesung 8

Informationsvisualisierung:

- * Sammlung von Abstrakten Daten aus zB sozialen Medien, Dokumente.
- * Raum ist bei Informationsvisualisierung im Gegensatz zur wissenschaftlichen Visualisierung nicht fix vorgegeben, außerdem muss man sich eine Metrik überlegen.

Overview + Detail

- * Man kann zwischen Detail und Globaler Ansicht wechseln
 - * Detail nur in einem Fenster
 - * Zoomen und ersetzen (Zooming soll nicht zu stark sein, da sonst die Korrelation zwischen Übersicht und Zoomfenster nicht mehr ersichtlich ist)
 - * Bifocal Vergrößerung (wie Lupe)
 - * Fischaugen Ansicht (Verzerrung, Topologie bleibt gleich, Winkel und Längen verändern sich)
- * Fokus + Context: man hat einen kontinuierlichen Übergang zwischen dem Fokus und dem Kontextbereich. Es wird das gemacht was beim Auge auch passiert: das Gehirn wird nicht mit übermäßigen Informationen überfordert, man sieht also nur die Dinge im Fokus genau. Man kann zB den Fokusbereich in gesättigten Farben darstellen und den Bereich (Context) herum nur in Grauwerten. Beispiele:
 - * Hyperbolic Tree: im Zentrum des Baumes ist mehr Platz.
 - * Perspective Wall: zB Kalender, Gegenwart ist wichtiger (größer/im Zentrum)
 - * Document Lens

Visual Transfer Function

- * Farbe, Sättigung, nur Kontur, ...
- * DOI - Degree of Interest Function
- * man möchte keine realistische Darstellung sondern die Verbindung zum Gehirn optimal nutzen.

Interaktion

1. Analyse: Man kann verdacht bestätigen
 2. Präsentation: von Daten
 3. Exploration: Man weiß vorher nicht welche Zusammenhänge gefunden werden.
- * Details on Demand
 - * Dynamic Queries: zB FilmFinder, in Echtzeit können Slider die Ergebnisse verändern
 - * Brushing und Linking: Wenn Daten die Bedingungen erfüllen werden sie hervorgehoben

Visual Analytics:

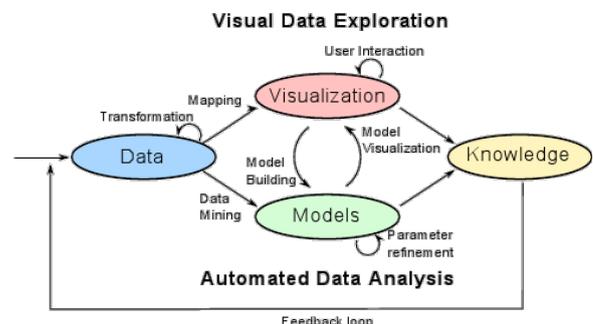
USA - Homeland Security, sammelt Daten um Terroristen zu finden.

Hurrikan Katrina - wo sollen Barrieren errichtet werden.

Problem: Große Menge an dynamischer Daten
Visual Analytics = Automated meth.+ Visualization

Ziele: Visual Analytics

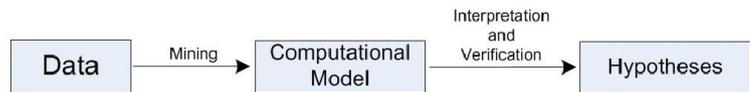
- * Dynamische und Unvollständige Daten zu behandeln
- * Auch das womit man nicht gerechnet wurde herauszufinden
- * Man braucht System mit denen man rechtzeitig reagieren kann (zB Medizin)
- * Entscheidungen müssen kommuniziert werden
- * Man möchte Schlussfolgerungen ziehen können, wobei man interaktive Schnittstellen hat.
- * Entscheidungen müssen vom Betrachter getroffen werden
- * Sehr große und komplexe Daten



- * Automatisierte Methoden(Vorbehandlung) → Visualisierung(Nutzer)
- * Interdisziplinärer Fachbereich: Data mining, Data management, Visualisierung, Human perception and Cognition(wie werden Verschieden Methoden wahrgenommen), Infrastruktur, Evaluierung
- * Herausforderungen:
 - * Data: Große Menge an Daten
 - * Users: Was braucht der Nutzer
 - * Design: Unterstütze Designer
 - * Technology: Was ist die nötige Infrastruktur um die Daten abzugreifen

Data Mining

- * Automatische Erfassung von Informationen
- * Knowledge Discovery: Semi oder Vollautomatisiert
- * Es gibt die Ability Matrix in der dargestellt wird welche Aufgaben besser vom Computer und welche besser vom Menschen erfüllt werden. (zB. Datenspeicherung vom Computer)



Grafiken

- * Der Gesichtssinn ist die Breitbandverbindung zum Gehirn
- * Muster können vom Menschen besser erkannt werden
- * Statistik zB. in Tabelle hat ihre Grenzen → Punkte zeichnen, aber auch Visualisierung kann Irreführend sein.
- * „Above all else, show the data“ - [Tuftte]
- * Data(ground truth) → Information(Phänomene) → Knowledge(Ursachen) → Wisdom(Mögliche Interaktion)
- * Mantra: overview first, zoom/filter, details on demand - [Shneidermann, 1996]
- * Extended Mantra für Visual Analytics: Analyse first, show the important, zoom/filter, analyse further, details on demand
- * The purpose of computers is insight not numbers.

Ungenauigkeiten

- * In der Realität muss man Unsicherheit berücksichtigen
- * Man kann aber trotz Unsicherheiten/Ungenauigkeiten Rückschlüsse ziehen(siehe Statistik)
- * What is not surrounded by uncertainty cannot be the truth - [Richard Feynman]
- * Doubt is not a pleasant condition, but certainty is absurd - [Voltaire]
- * In manchen Bereichen möchte man Unsicherheiten komplett ausblenden, zB Arzt.
- * Ungenauigkeiten wegen: Messgeräten, nicht Messbarkeit können schon in Daten eliminiert werden oder erst später

Data Management Challenges

- * Semantisches Management
- * Big Data
 - * Meteorologie
 - * Gentechnik(zB. wie wer wo von Afrika ausgewandert ist)
 - * Graph in Gehirn(Große Menge an Nervenzellen messen)
 - * Komplexe physische Simulationen(Kombination verschiedener Modelle)
 - * Business Intelligence

Visual Steering to Support Decision Making in Visdom

- * Bei Überschwemmung soll eine Entscheidung getroffen werden, was geflutet werden soll.
- * Erst wenn Strömung gewisse Geschwindigkeit unterschritten hat → Sandsäcke aufbauen.