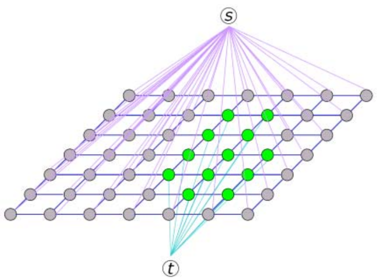
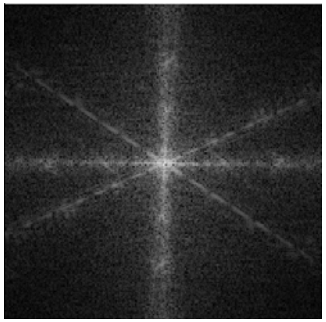


MinCut = MaxFlow

$$y = F \cdot x$$

$$F = \begin{bmatrix} 1 & 1 & 1 & \dots \\ 1 & \omega & \omega^2 & \dots \\ 1 & \omega^2 & \omega^4 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

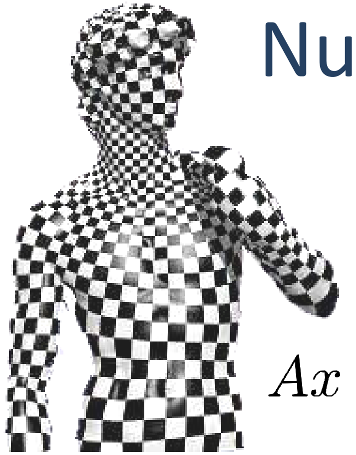
$$x = F^{-1} \cdot y$$



Hauptseminar Computergraphik

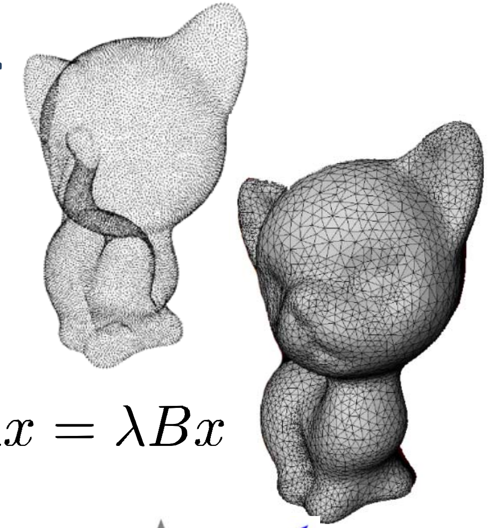
Numerische Algorithmen in der Computergraphik

„There‘s beautiful math inside…“

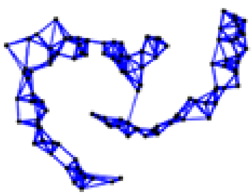


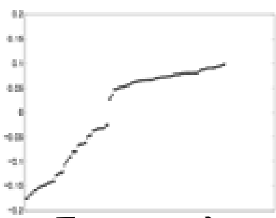
$$Ax = b$$

Dipl. Medieninf. Sören König
Di 3.DS, INF E 009



$$Ax = \lambda Bx$$

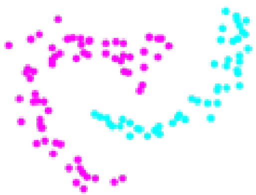




$$\lambda_2$$

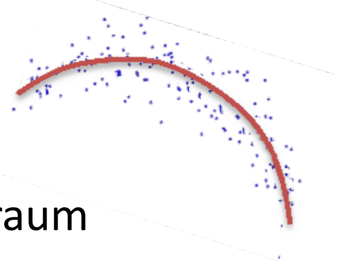
$$L = D - A$$

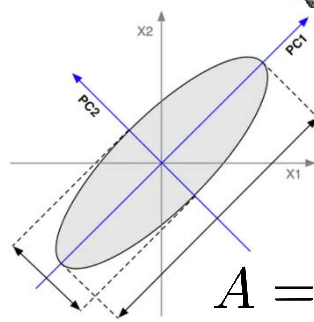
$$Lx = \lambda x$$



K-means im Eigenraum

$$A^T A x = A^T b$$





$$A = USV^T$$

Gliederung

- Scheinerwerb
- Inhaltliches zum Vortrag/ zur Ausarbeitung
- Themenvorstellung

Scheinerwerb

- Vortrag Dauer (ca. 40 min) mit Diskussion (5-10 min)
  - Vortrag muss eine Woche vor dem Termin fertig sein und mit dem Betreuer durchgesprochen worden sein!
  - Folienabgabe als PDF oder PPT für unsere Website
- Schriftlichen Ausarbeitung Umfang 10-20 Seiten
  - Abgabe bis Semesterende als PDF
  - Erstellung mit Latex wird empfohlen
  - Qualitätssicherung durch jeweiligen Betreuer
- Regelmäßige Teilnahme am Seminar

Inhaltliches zum Vortrag/Ausarbeitung

- Grundlagenteil:
  - Problematik durch ein oder zwei selbstgewählte Beispiele in der Computergraphik/Computervision motivieren
  - Graphische Problemvisualisierung :

    Kernidee, Grundproblem oder Algorithmus soll möglichst intuitiv und kompakt auf ein oder zwei Folien anhand eines **eigenen** Bildes/**eigener** Grafik/Folienanimation **visualisiert** werden
  - formale Problemspezifikation (z.B. Formeln Zusammenhang herstellen)
  - Vorstellung aktueller Algorithmen/Verfahren sowie Bibliotheken

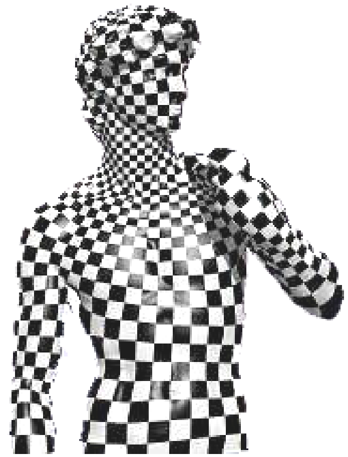
- Angewandter Teil:
  - Vorstellung eines aktuellen CG-Papers, in dem die vorgestellte Technik eine wichtige Rolle spielt (Auswahl des Papers vorher zusammen mit zugeteiltem Betreuer abstimmen)

Themenvorstellung

1. Lineare Gleichungssysteme, Spezielle Matrixklassen, Matrixfaktorisierung und direkte Löser

$$Ax = b$$

Lineares Gleichungssystem (LGS)



$$Ax = B$$

Lineares Gleichungssystem mit mehreren rechten Seiten

Matrixfaktorisierungen sind die „Building Blocks" der Linearen Algebra

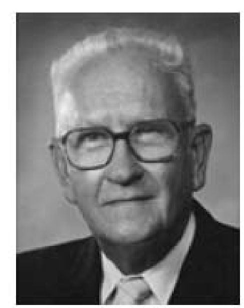
$$A = (P)LU = QR = LL^T = U\Sigma V^T = Q\Lambda Q^{-1} = \ldots$$

Direktes Lösen mittels Divide & Conquer …

$$Ax = b \xrightarrow[A = LU]{} LUx = b$$

$$\downarrow$$

$$Ly = b \longrightarrow Ux = y$$



Householder

2. Singulärwertzerlegung und Eigenwertprobleme

$$A = U\Sigma V^T$$

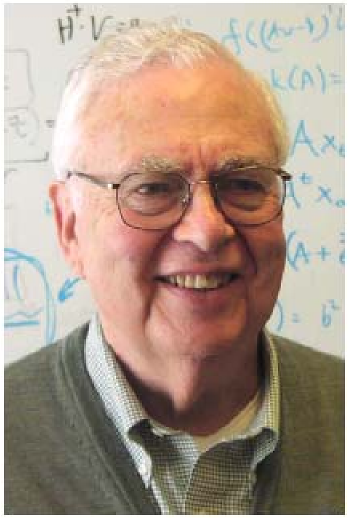
Singulärwerte in Diagonale

Orthonormale Matrizen

$$A = Q\Lambda Q^{-1}$$

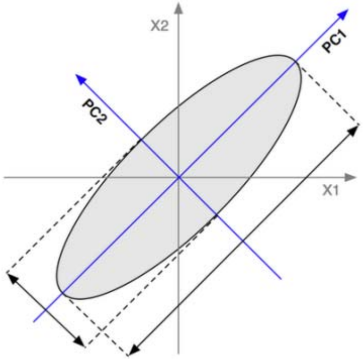
Eigenwerte in Diagonale

Eigenvekoren in Spalten



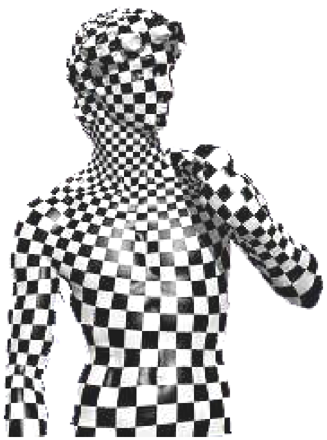
Prof. Gene Golub

- Bestimmung des Rangs einer Matrix
- Lösungen von über- und unterbestimmten LGS
- Kleinste Quadrate Lösungen (Modell fitten)
- Kompression / Matrixapproximation
- Invertieren
- Hauptachsentransformation
- …



… und sein Nummernschild

3. Krylov-Unterraummethoden - Iterative Löser für Gleichungssysteme und Eigenwertprobleme

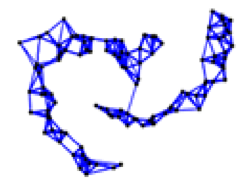


Parametrisierung

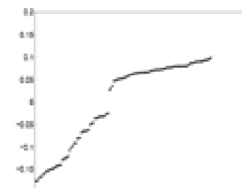
führt zu 2 LGSen der Größe nxn mit
n Anzahl Knoten im Netz!

Spectral Clustering:

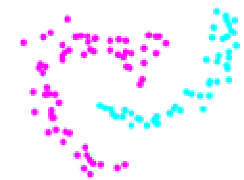
λ_2



$L = D - A$



$Lx = \lambda x$



K-means im Eigenraum

Matrix ist extrem groß und dünnbesetzt!
Direktes Lösen/Faktorisieren zu aufwendig!

Iteratives Lösen
linearer Gleichungssysteme

CG, GMRES-Verfahren

Iterative Berechnung einiger weniger,
großer Eigenwerte/-vektoren

Arnoldi-, Lanczos-Verfahren

4. Fast Fourier Transform (FFT)

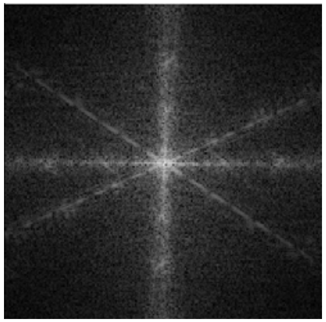
Diskrete Fourier Transformation

$$y_k = \sum_{j=0}^{N-1} e^{-2\pi i jk/N} \cdot x_j = \sum_{j=0}^{N-1} \omega^{jk} \cdot x_j$$

$$F = \begin{bmatrix} 1 & 1 & 1 & \dots \\ 1 & \omega & \omega^2 & \dots \\ 1 & \omega^2 & \omega^4 & \dots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

DFT: $y = F \cdot x$

Inverse DFT: $x = F^{-1} \cdot y$



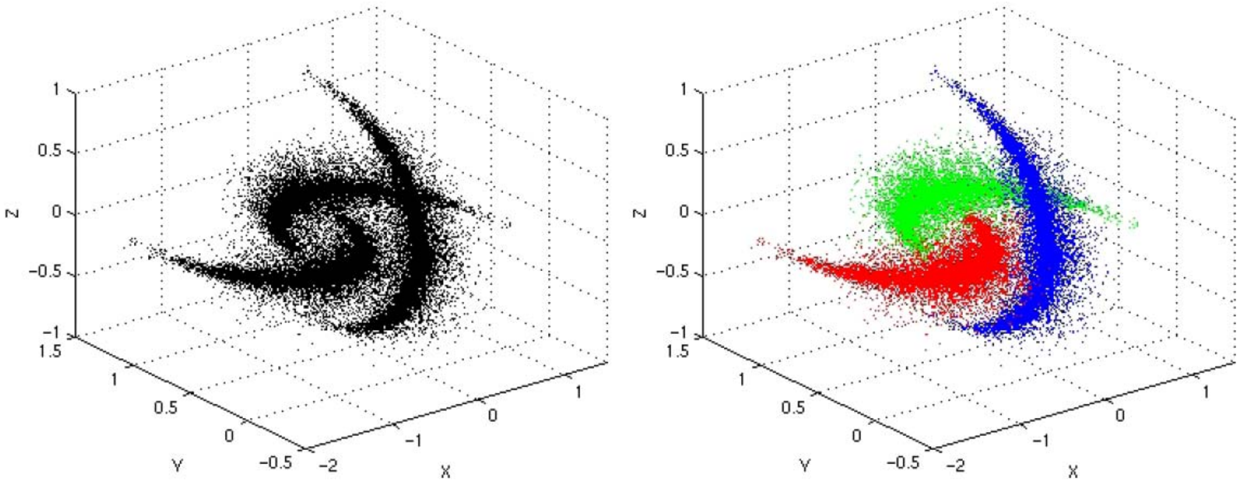
2D DFT

FFT ist ein Trick das Produkt von F mit x zu beschleunigen auf N log N!

Faltung im Ortsraum (x) entspricht Multiplikation im Frequenzraum (y)

Frequenzanalyse, Schnelle Multiplikation von Polynomen,…

5. Fast Multipole Method (FMM), Fast Gaussian Transform (FGT), ...



Mean-Shift Clustering

Approximation der Dichte-Funktion mittels Summe aus Gaussverteilungen

Z.B.: In 1D: $$S(t_i) = \sum_{j=1}^{N} f_j e^{-\frac{(s_j - t_i)^2}{\sigma^2}}, i = 1, \ldots, m$$

Dichtefunktion an m Stellen auswerten für n Sites hat Aufwand: mn

FGT löst Problem in m+n   FMM abstrahiert das Problem auf weitere Kernelfunktionen!

6. Minimaler Schnitt und Maximaler Fluss in Graphen (MinCut-MaxFlow)

Finde ein optimales binäres Labeling L für eine Kostenfunktion F!

$$\hat{L} = \underset{L \in \{0,1\}^n}{\operatorname{argmin}} F(L).$$

Kostenfunktion mit folgender Form :

$$F(L) = \sum_{i=1}^{n} E^i(L_i) + \sum_{\substack{i<j \\ i,j \in \{1,\dots,n\}}} E^{i,j}(L_i, L_j).$$

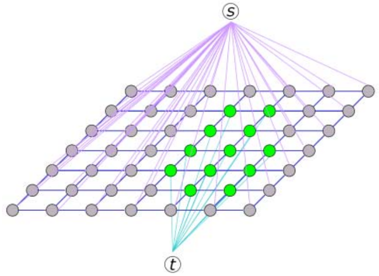
Kosten für Gesamtlabeling

Wie schlecht passt Pixellabel zum Segment?

Wie schlecht passt Pixellabel zu den benachbarten Pixellabeln?

Übertrage Problem auf die Suche nach einen minimalen Schnitt/ Maximalen Fluß in einem speziellen Graphen:

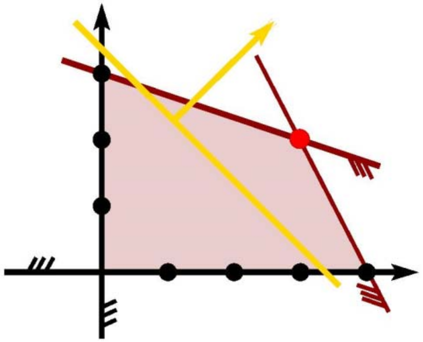
MinCut = MaxFlow



7. Linear Programming, Quadratic Programming

Maximiere eine lineare Funktion:

LP: $$f(x_1, x_2, \ldots, x_n) = c_1 x_1 + c_2 x_2 + \ldots + c_n x_n$$



unter den Nebenbedingungen:

$$A\mathbf{x} \leq \mathbf{b}$$

GJK-Algorithmus: Abstand zweier polygonaler Netze

QP: Minimiere eine quadratische Funktion:

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \mathbf{c}^T \mathbf{x}$$

unter den Nebenbedingungen:

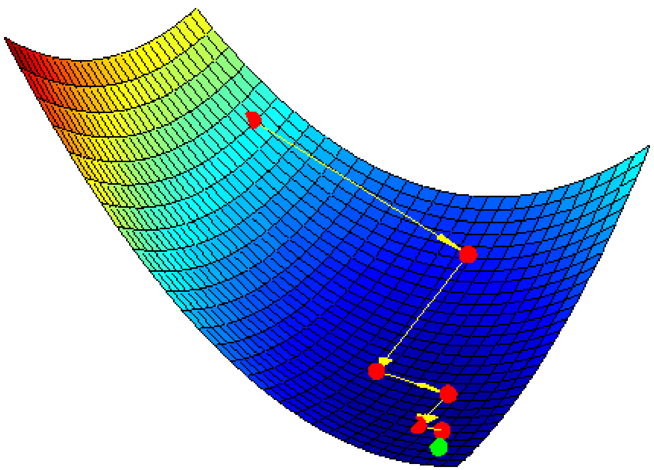
Support Vector Machines

$$A\mathbf{x} \leq \mathbf{b} \qquad E\mathbf{x} = \mathbf{d}$$

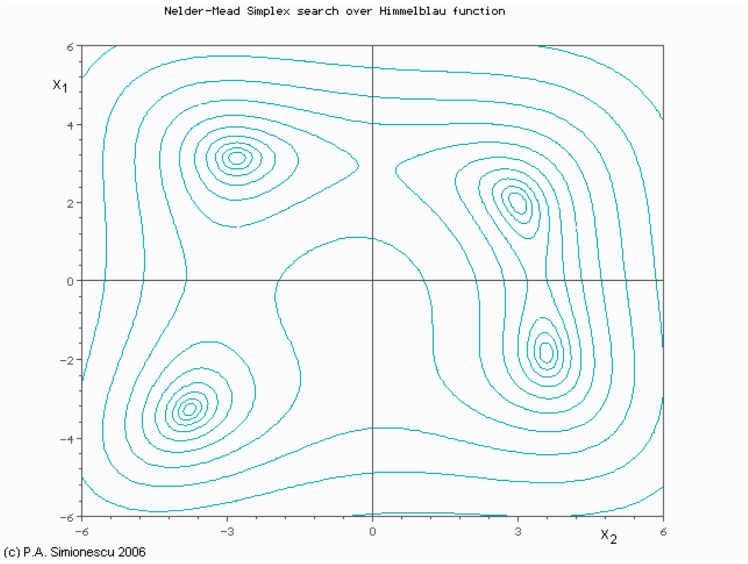
Interior Point Method, Active Set Method, CG Method, …

8. Nichtlineare Optimierung (Gradient Descent, Newton's method, Levmar, downhill simplex)

Numerische Berechnung von Minimas und Maximas einer nichtlinearen Funktion



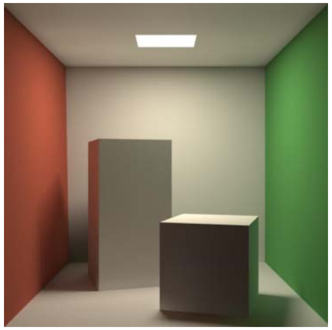
Iterativer Gradientenabstieg



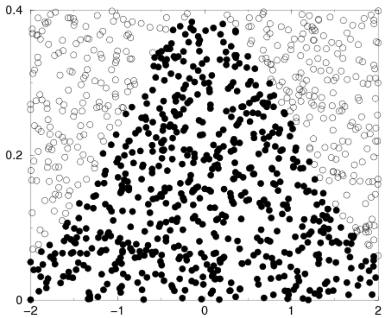
Downhill Simplex Verfahren

9. Sampling Methoden: Random Sampling, Stratified Sampling und die Monte Carlo Methode

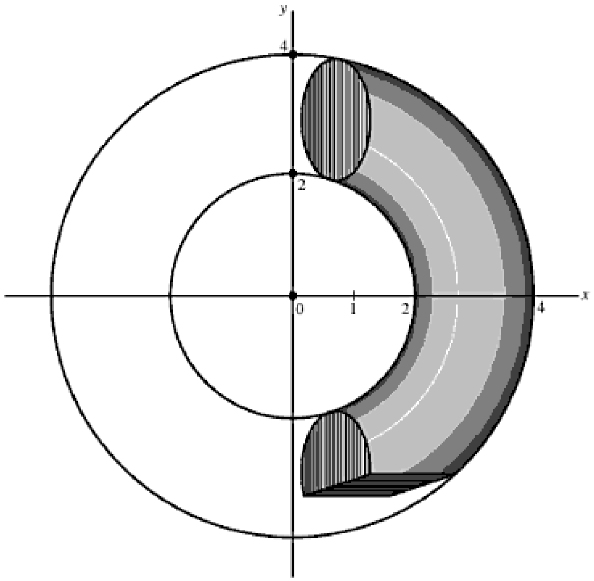
Wie wertet man komplizierte Integrale aus um Flächen oder Volumen zu berechnen?



Renderinggleichung lösen



Random Sampling einer Gaussfunktion



Torus geschnitten mit zwei Ebenen

Wie generiert man am besten die Samples?

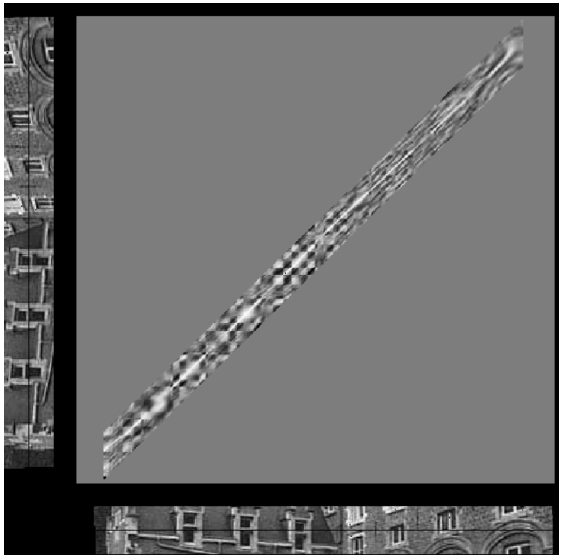
Wie viele Samples braucht man?

10. Dynamic Programming

- Algorithmisches Designmuster
  - Wiederverwendung von Zwischenergebnissen
  - Vermeidung von Redundanz innerhalb von Berechnungen

Fib(0) = 0, Fib(1)=1;
Fib(n) = Fib(n-1) + Fib(n-2)

Fib(20) = Fib(19) + Fib(18)
Fib(20) = (Fib(18) + Fib(17)) + Fib(18)



- Sequence/Substring Matching
- Bspline auswerten (DeBoor)
- Magnetische Lasso in Photoshop
- …

11. Optimierung mit Methoden der Variationsrechnung

Am Beispiel des optischen Flusses:

- Geg.: Bildfunktion I(x,y,t)
- Ges.: Verschiebefeld u (Verschiebung in x),v (Verschiebung in y)

$$I_x u + I_y v + I_t = 0$$

Grauwert ist konstant

$$|\nabla u|^2 + |\nabla v|^2 \rightarrow min$$

Änderungen in Verschiebefeld ist möglichst klein (glatt)

$$E(u,v) = \int_\Omega (I_x u + I_y v + I_t)^2 + \alpha(|\nabla u|^2 + |\nabla v|^2) dxdy$$

Energiefunktional fordert beide Annahmen für den gesamten Bildbereich!

Gesucht sind Funktionen u,v die Energiefunktional E minimieren!

Bedingung für Optimum liefert der Euler-Lagrange Formalismus.

Euler Lagrange Gleichungen aufstellen :

$$(I_x u + I_y v + I_t) I_x - \alpha \Delta u = 0$$

$$(I_x u + I_y v + I_t) I_y - \alpha \Delta v = 0$$

Diskretisieren und Lösen…

Und mit noch ein paar kleinen Verbesserungen…