

Gebäudevisualisierung mit Point clouds

CS561 Verteilte Systeme

Daniel Federau, Lukas Songajlo

5. November 2013

Übersicht

- Point clouds
- Hybrid Images

Was ist eine Point cloud?



- repräsentiert 3-dimensionale, reale Objekte
- jeder Punkt ist Teil der Oberfläche eines Meshs

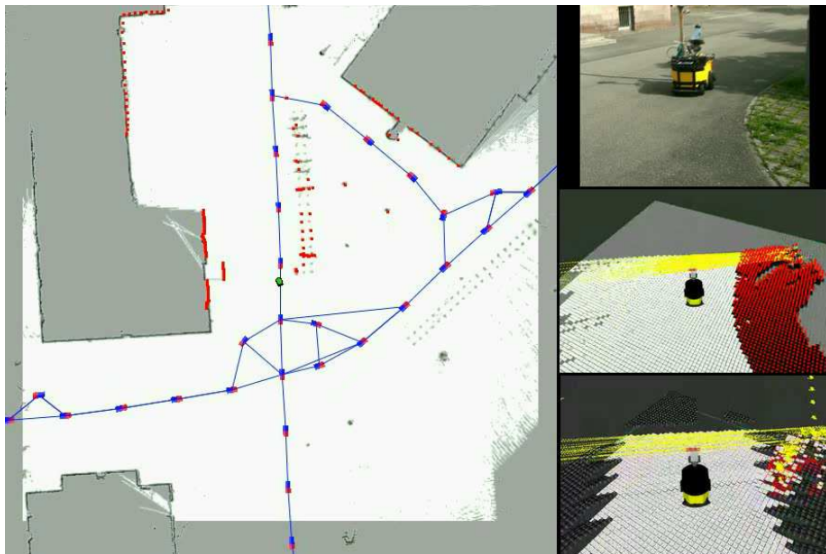
Was ist eine Point cloud? (formal)

- Menge von diskreten Punkten: $P = \{\rho_1, \rho_2, \dots, \rho_i, \dots, \rho_n\}$
- ρ_i ist ein Feature-Vektor:
 $\rho_i = \{x_i, y_i, z_i, r_i, g_i, b_i, temperature_i, \dots\}$

Motivation

- Visualisierung gescannter Objekte am Computer
- Modellierung der Point cloud in 3D Modells
⇒ Oberflächenrekonstruktion
- Anwendung: Unfallvermeidung, Gebäudeplanung einer Stadt

Terrain Detection



“Building Rome in a Day“

- Video Rom
- Projekt der Universität Washington
- Rekonstruktion von Rom aus zahlreichen Bildern
- Download der Bilder von flickr.com
→ Point Cloud durch Feature-Matching
- 21h Berechnungszeit mit ca 500 CPU-Kerne

“Altstadt von Dubrovnik“

- Video Dubrovnik
- 4,619 Bilder + 3,485,717 Punkte
- 5h Feature-Matching
- 17h Visualisierung

Wie rechnet man mit unorganisierten Punkten?

Problem: Redundante Daten

Lösung: jedes ρ_i erhält ein weiteres feature *entropy*

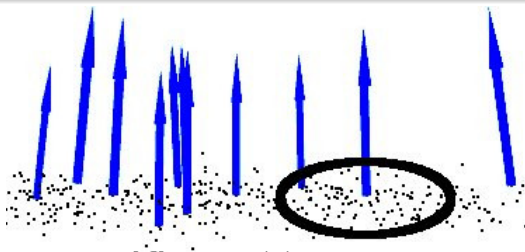
- Punkte mit wenig Informationsgehalt werden gelöscht
- *entropy* abhängig von: Distanzen zueinander, Änderung der Normalen, Farbe

Wie rechnet man mit unorganisierten Punkten?

Probleme: sehr große # Punkte + Messungenauigkeiten

Lösung: downsampling der Punkte mit Gruppierungsmethoden

- Interpolation der Nachbarpunkte



Was ist ein Hybrid Image?

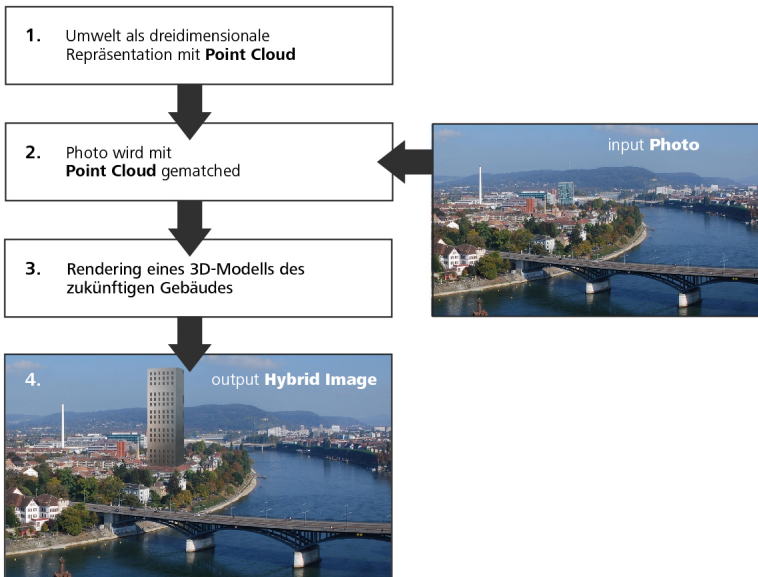


Was ist ein Hybrid Image?

- Verbindung von realen und virtuellen Elementen in einem Bild
- statischer Einsatz von Augmented Reality

Zielsetzung

- Visualisierung von zukünftigen Bauprojekten in einer realen Umgebung
- betrachten des Gebäude aus jedem Blickwinkel
- Präsentation für potentielle Kunden, Anwohner und Projektbeteiligte



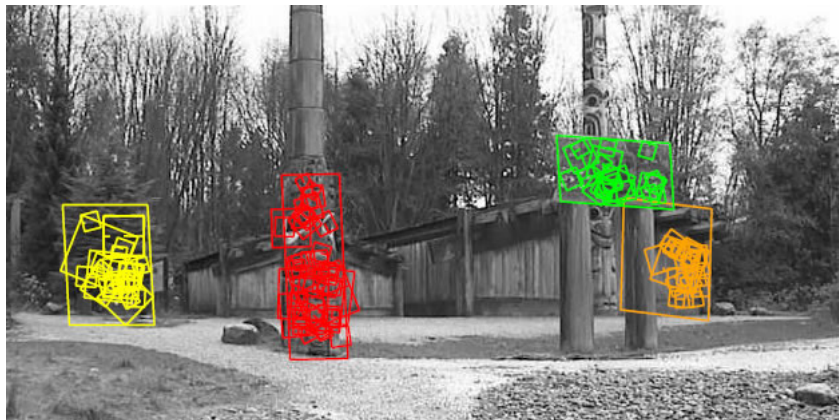
1. Arbeitsschritt: Point cloud erstellen

- Benötigt eine Reihe von Bildern der gleichen Umgebung aus unterschiedlichen Perspektiven
- Für jedes Bild B_i wird ein Feature -Set F_i mit f_1, \dots, f_n feature Vektoren erstellt
→ SIFT-Algorithmus
- Features aus verschiedenen Bildern werden verglichen
- Suchen von Punkten, die in mehreren Bildern vorkommen
→ Repräsentation durch die Point cloud

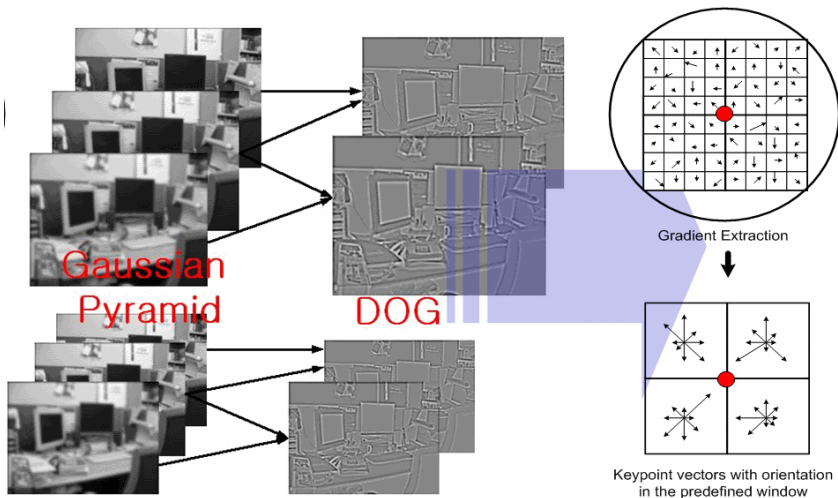
Scale-invariant feature transform (SIFT)

- wurde 1999 von David Lowe entwickelt
- Features sind invariant gegen Skalierung, Rotation, Beleuchtung, Blickwinkel
- es werden nur lokale Informationen betrachtet
- ein Feature wird durch einen Vektor mit 128 Einträgen dargestellt





1. Arbeitsschritt: Feature Extraktion



1. Arbeitsschritt: Matching von Feature Sets in verschiedenen Bildern

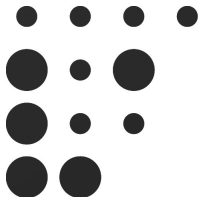
- “ k -nearest neighbour search”
- Es werden der nächste und der zweitnächste Nachbar gesucht ($k = 2$):
 - ist der Abstand der beiden Nachbarn zu klein wird es nicht als *match* verzeichnet
 - überschreitet der Abstand einen gewissen Schwellwert, werden die zwei Features als *match* gewertet

2. Arbeitsschritt: Matching mit vorhandener Point cloud

- Erstellen eines SIFT-Feature-Set des Input Bildes
- Matching der Features mit der Point cloud
- Durch die Transformation der Point cloud erhält man Perspektive und Aufnahmeort des Bildes

3. Arbeitsschritt: Rendering des 3D-Modells

- Zeitpunkt der Aufnahme des Bildes bestimmt den Ort der Lichtquelle (Stand der Sonne)
- Objekte miteinbeziehen, die Sicht blockieren
- Point cloud liefert Informationen über Perspektive
- richtige Ausrichtung des 3D-Modells



hKDM

**Hochschule für Kunst, Design
und Populäre Musik**

**University of Arts and Applied
Sciences**

Quellen

- Approximierung von point clouds in 3D Meshes:
http://geom.ibds.kit.edu/papers/point_clouds.pdf
- alternative Einsatzmöglichkeiten + allg. Informationen zum Thema Point clouds:
<http://ais.informatik.uni-freiburg.de/teaching/ws10/robotics2/pdfs/rob2-12-ros-pcl.pdf>
- SIFT:
<http://www.aishack.in/2010/05/sift-scale-invariant-feature-transform/>
- SIFT-Algorithmus (Bild):
<http://ispl.korea.ac.kr/Research/datafusion/navigation.html>
- Building Rome in one day:
<http://grail.cs.washington.edu/rome/index.html>
- Workflow Gebäudevisualisierung:
Remembering in the Metaverse, Florian Müller

The End

Danke für die Aufmerksamkeit.
offene Fragen?