

# Hintergrund (1)

Automatisierung von terrestrischen Laserscanning

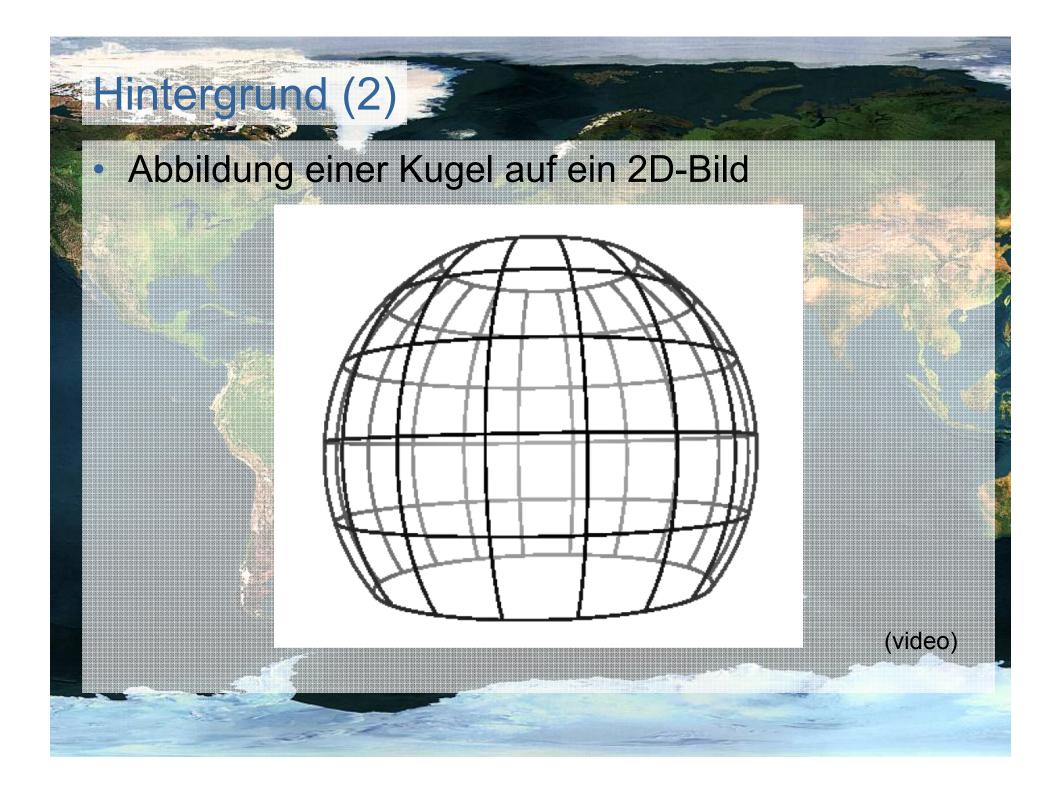


(video)

(animation)

Kombination terrestrisches / kinematisches Laserscanning





### Kugelkoordinaten

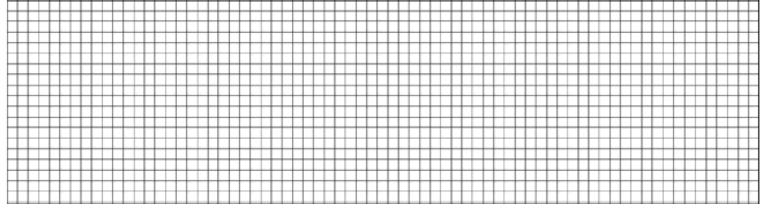
 3D-Laserscanner nutzen das tachymetrischen Messprinzip, bei dem das instrumenteneigene polare Koordinatensystem durch eine vertikale Drehachse, eine horizontale Drehachse



# Gleichmäßige Rektangularprojektion

$$x = \theta$$

$$x = \theta$$
$$y = \phi$$



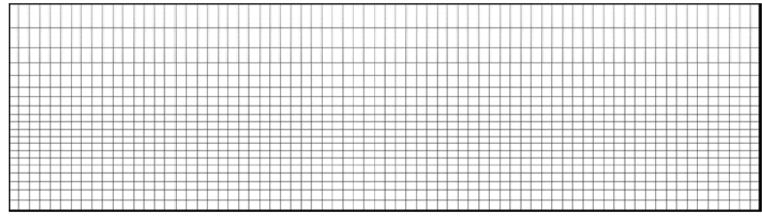




# Zylindrische Projektion

$$x = \theta$$

$$y = \tan \phi$$



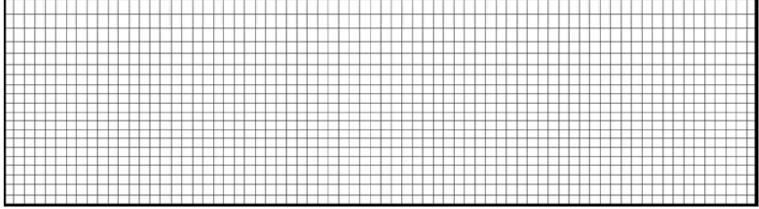


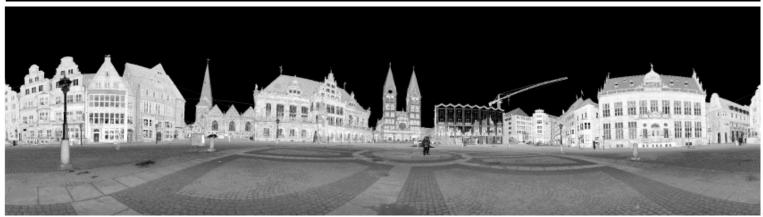


#### **Mercator Projektion**

$$x = \theta$$

$$y = \ln\left(\tan\phi + \frac{1}{\cos\phi}\right)$$



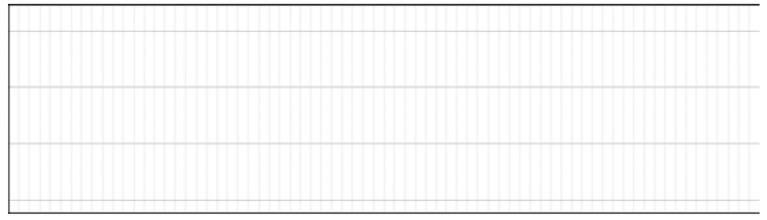


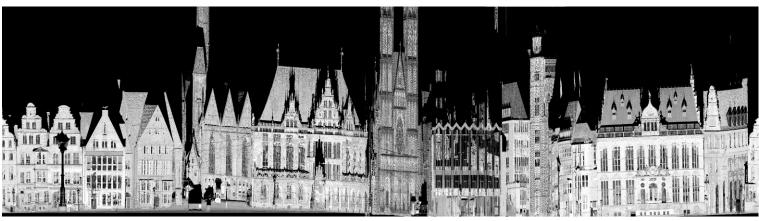


# Hochachsenprojektion

$$x = \theta$$

$$y = z$$

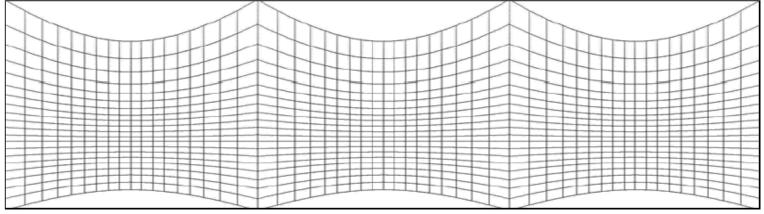






### Rektangularprojektion

$$x = \frac{\cos\phi\sin\left(\theta - \theta_0\right)}{\sin\phi_1\sin\phi + \cos\phi_1\cos\phi\cos\left(\theta - \theta_0\right)}$$
$$y = \frac{\cos\phi_1\sin\phi - \sin\phi_1\cos\phi\cos\left(\theta - \theta_0\right)}{\sin\phi_1\sin\phi + \cos\phi_1\cos\phi\cos\left(\theta - \theta_0\right)}$$

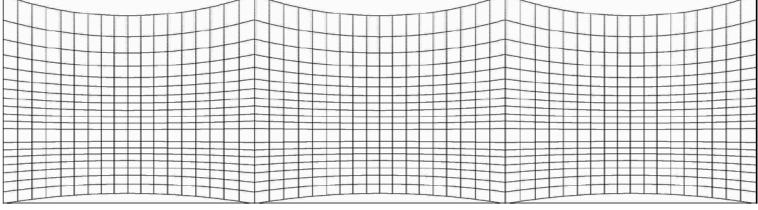






# Pannini-Projektion

$$x = \frac{(d+1)\sin(\theta - \theta_0)}{d + \sin\phi_1 \tan\phi + \cos\phi_1 \cos(\theta - \theta_0)}$$
$$y = \frac{(d+1)\tan\phi \left(\cos\phi_1 - \sin\phi_1 \left(\frac{1}{\tan\phi}\right)\cos(\theta - \theta_0)\right)}{d + \sin\phi_1 \tan\phi + \cos\phi_1 \cos(\theta - \theta_0)}$$

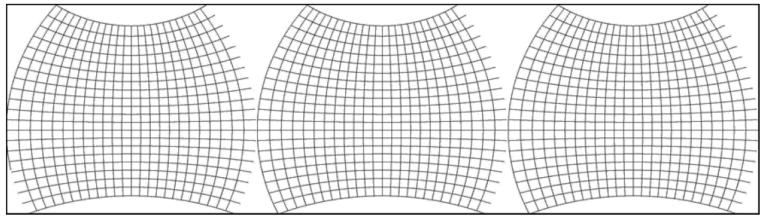






#### Stereographische Projektion

$$x = \frac{2R\cos\phi\sin\left(\theta - \theta_0\right)}{1 + \sin\phi_1\sin\phi + \cos\phi_1\cos\phi\cos\left(\theta - \theta_0\right)}$$
$$y = \frac{2R\left(\cos\phi_1\sin\phi - \sin\phi_1\cos\phi\cos\left(\theta - \theta_0\right)\right)}{1 + \sin\phi_1\sin\phi + \cos\phi_1\cos\phi\cos\left(\theta - \theta_0\right)}$$







# SIFT-Merkmale (1)

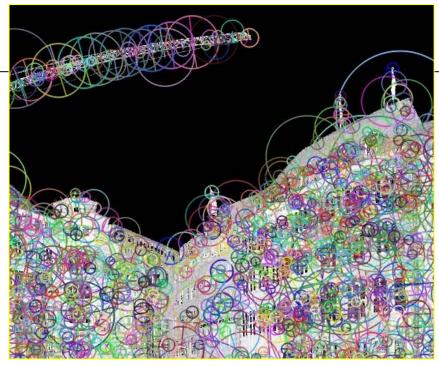
- Erzwinge Skalierungsinvarianz: Berechne die DOG für viele verschiedene Skalen; benutze non-maximum suppression, finde lokale Maxima, als "Keypoint"-Kandidaten
- 2. Lokalisiere Ecken: Für jedes Maximum fitte eine quadratische Funktion. Berechne das Zentrum mit Sub-Pixel Genauigkeit durch Null-Setzen der ersten Ableitung.
- 3. Eliminiere Kanten: Berechne das Verhältnis der Eigenwerte, Wirf diejenigen Keypoints weg, bei denen das Verhälnis größer als ein Schwellwert ist.
- 4. Erzwinge Invarianz bezüglich Rotation: Berechne die Orientierung um Rotationsinvarianz zu erhalten durch das Finden der größten zweiten Ableitung in einem geglätteten Bild. (evtl. gibt es mehrere Orientierungen) Rotiere den Ausschnitt, so dass der Orientierungsvektor nach oben zeigt.
- 5. Berechne eine Merkmalssignatur: Berechne ein "Gradientenhistogram" auf dem Bildausschnitt (4x4 Pixel). Orientiere den Ausschnitt so, dass die größte Gradient nach oben zeigt. Ergebnis: Merkmalvektor mit 128 Werten (15 Einträge, 8 Gradienten).
- 6. Erzwinge Illuminationsinvarianz: Normalisiere Vektor zu Einheitsvekotor. Bearbeite alle Gradienten mit einem Schwellwert, um Invarianz zur Saturierung zu erreichen.

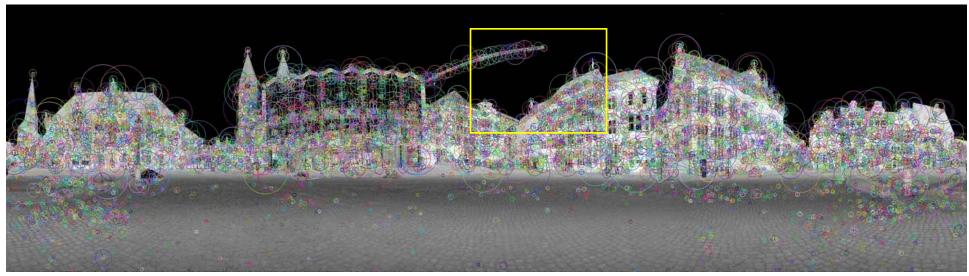


# SIFT-Merkmale (2)

 Typischerweise zirka 30000 Merkmale in einem Bild mit 3.6 Megapixel

Beispiel:



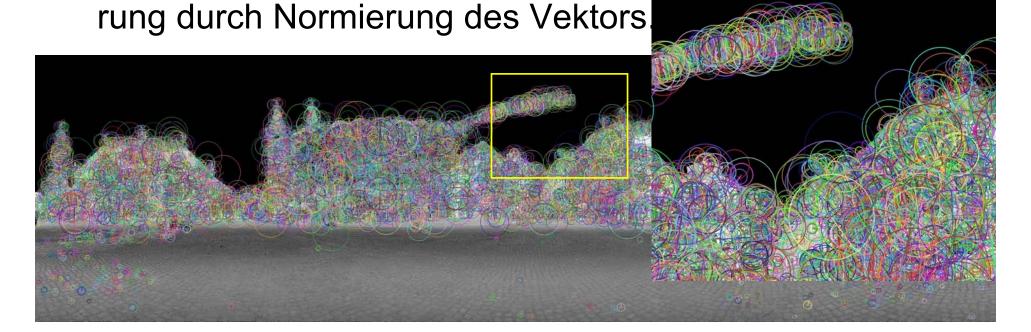




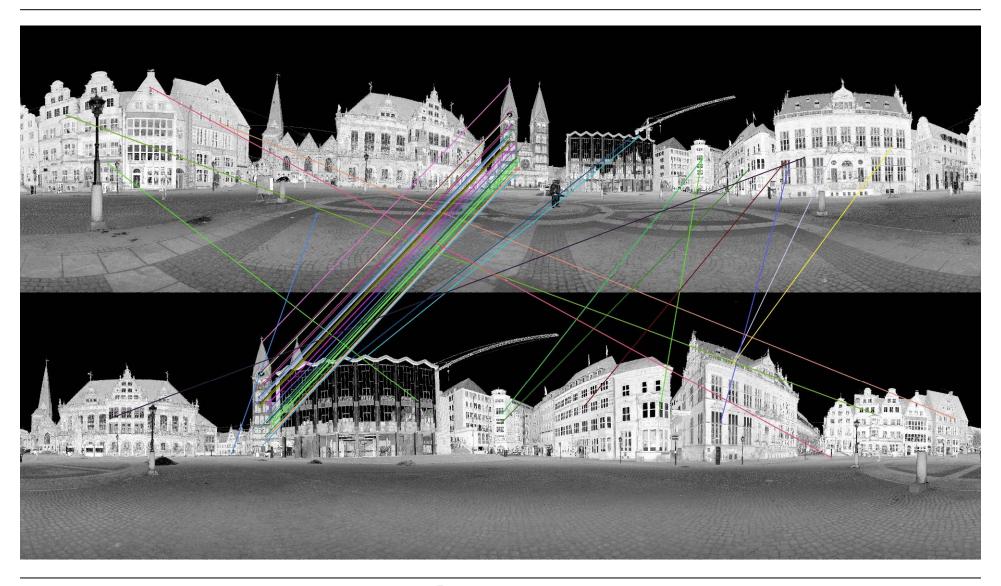
### SURF-Merkmale und Deskriptor

 DOG-Filter (Difference of Gaussian) der SIFT-Merkmale wird approximiert. Dies geschieht durch einen Kasten-Filter (Dob, engl. Difference of Boxes)

 Für jede der 4×4 Regionen wird ein 4-Vektor mit Haar-Merkalen berechnet, konkateniert zu einem Deskriptor der Länge 64. Invarianz gegen Beleuchtung entsteht durch die verwendeten Haar-Merkmale, gegen Skalie-

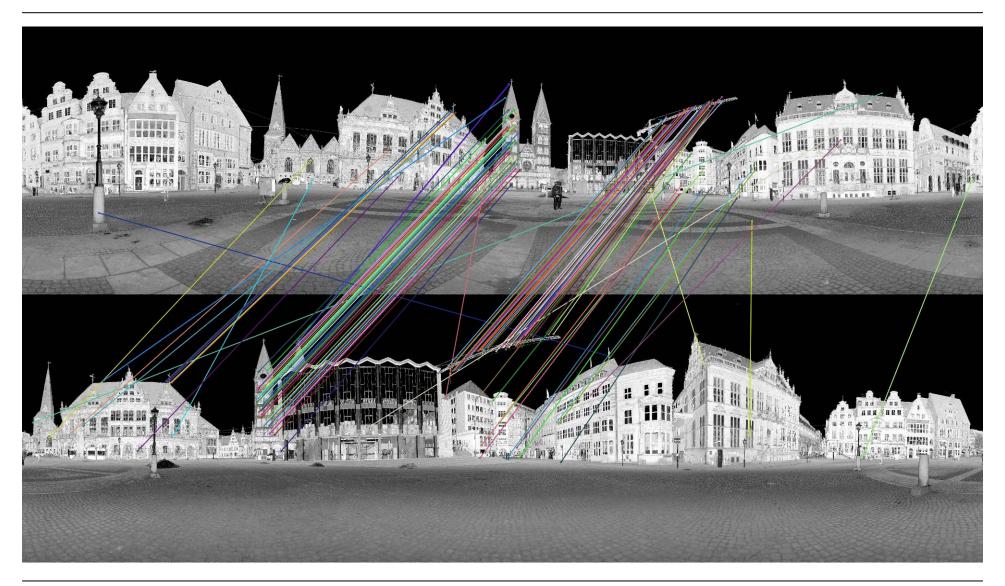


# Merkmalszuordnung (SIFT / SIFT)





# Merkmalszuordnung (SURF / SURF)





#### RANSAC

- Generiere M (eine vordefinierte Anzahl) von Modelhypothesen, jede davon wird mit einem minimaler Teilmenge von Punkten berechnet
- Evaluiere jede Hypothese
- Berechne die übrigen Fehler bei Verwendung aller Daten.
- Punkte mit Fehlern kleiner als ein Schwellwert werden als "Inlier" klassifiziert
- Die Hypothese mit einer maximalen Anzahl von "Inlier" wird ausgewählt. Anschließend wird die Hypothese neu geschätzt nur auf den "Inlier" basierend.



# Ergebnisse (1)

#### Typische Ergebnisse der Merkmalsextraktion

	#Points	Resolution		Features		
Scan id			Projection	Generation	#Features	Extraction
				Time (s)		Time (s)
Scan002	15,246,877	2880x800	Equirectangular	0.18	38,588	38.86
			Cylindrical	0.46	30,342	36.01
			Z-axis	0.17	46,637	41.87
			Mercator	1.08	34,357	37.43
			Rectilinear	1.24	23,724	33.66
			Pannini	2.67	24,291	33.93
			Stereographic	2.74	25.515	34.41

#### Typische Matching und Registrierungsergebnisse

Scan Pair	Resolu- tion	Projection	# Feat.	# Mat- ches	Mat. Time (s)	# In- liers	Min. Er- ror	$\begin{array}{c} \text{Reg.} \\ \text{Time} \\ \text{(s)} \end{array}$
		Equirectangular	74,082	413	430.94	22	50	5.23
		Cylindrical	61,036	384	345.27	27	50	4.9
		Z-axis	90,682	607	530.3	13	70	7.74
$Scan001{\rightarrow}Scan002$	2880x800	Mercator	67,380	366	385.18	23	50	4.67
		Rectilinear	48,654	298	253.35	10	130	2.86
		Pannini	50,263	353	268.5	14	50	4.34
		Stereographic	49,990	316	266.97	15	50	4.07

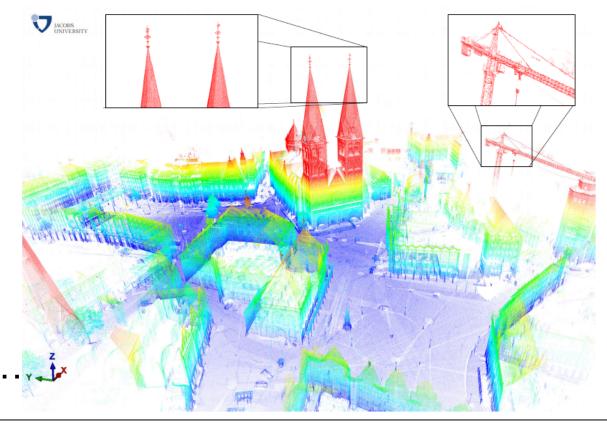
# Ergebnisse (2)

 ...funktioniert alles gut, hängt aber wesentlich vom Überlappungsbereich ab...



# Ergebnisse (3)

- Nur mit der Mercator-Projektion waren wir in der Lage, den kompletten Datensatz (13 3D-Scans) automatisch zu registrieren.
  - Mercator: winkeltreu (formtreu), aber weder flächentreu noch richtungstreu
- Test weiterer
  Merkmale und
  deren Deskripto ren (ORB, FAST,...





### Zusammenfassung & Ausblick

- SIFT / SURF basierte Registrierungen funktionieren sehr gut
- Projektionen von 3D-Scans für merkmalsbasierte Registrierung sollte berücksichtigt werden
- Software: in 3DTK The 3D Toolkit

http://www.threedtk.de/

- Aktuell: Umstellung auf OpenCV
- Zukünftig: Implementierung eines 3D-Operators zur Erkennung natürlicher Landmarken

