

Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes

(Effiziente Objekterkennung von ungeordneten 3D Szenen mithilfe von Spin Images)

Gliederung:

- 1) Einleitung
- 2) Oberflächenvergleich
- 3) Objekterkennung

1. Einleitung

Ziel:

- gleichzeitige Erkennung von mehreren Objekten in einer 3D Szene, wobei die Objekte in einer Art Bibliothek gespeichert sind



Funktionsweise:

- Erkennung basiert auf erkannten Oberflächen und darauf bekannten Punkten
- ist die Oberfläche erkannt in der Szene, kann eine Beziehung zwischen einem bekannten Objekt und etwas unbekanntem – der Szene – hergestellt werden, um Informationen über die Szene zu erhalten

1. Einleitung

Wozu wird es benötigt:

- Oberflächenerkennung ist eine Technik aus der 3D Welt, die in der Automation und bei Robotern Anwendungen findet

Was benötigt man dazu?

- Sensor, der die Objekte abtastet und die 3D Punkte erzeugt
- Formrepräsentation, um die Informationen über die 3D Punkte effizient zu speichern
- Koordinatensysteme, in dem die Daten beschrieben werden
- 2D Abbildungen

Formrepräsentation:

- Existenz verschiedenster Arten von Formrepräsentationen:
 - verschiedene Achsen
 - Anzahl der Parameter, die zur Beschreibung der Basiselemente genutzt werden (planare Objekte haben viele Basiselemente mit wenigen Parametern
↔ zylindrische Objekte haben wenige Basiselemente mit vielen Parametern)
 - Gauß – Abbildungen für kugelige Objekte
→ Vielzahl an Formrepräsentationen, noch kein Konsens über die beste

1. Einleitung

Koordinatensysteme:

- Sichten–zentriertes Koordinatensystem
 - Beschreibung der Oberflächendaten in Bezug auf das Koordinatensystem, abhängig von der Sicht auf die Oberfläche
 - + einfach zu berechnen
 - bevor verglichen werden kann, ändert sich die Beschreibung der Oberfläche, da die sie eventuell noch transformiert werden muss
 - bei mehreren Sichten muss für jede Sicht eine separate Oberfläche gespeichert werden

- Objekt–zentriertes Koordinatensystem
 - Beschreibung der Oberfläche erfolgt in einem Koordinatensystem, das mit dem Objekt fixiert ist
 - + Sichten–unabhängig
 - + lassen sich direkt vergleichen
 - + kompakter, da alle Sichten in einer Oberflächenbeschreibung beschrieben sind

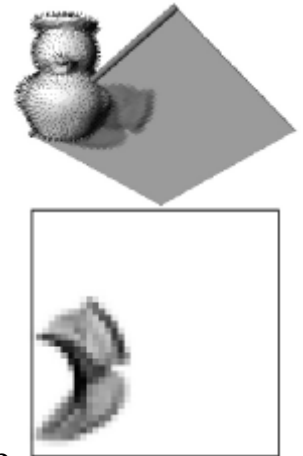
1. Einleitung

2D Abbildungen:

- werden zu einer lokalen Basis eines orientierten Punktes an der Oberfläche eines Objektes erstellt
 - 3D Punkt mit Oberflächennormalen
 - Beschreibung mit nur noch 2 Parametern
 - wird in einem 2D Histogramm erstellt
 - enthält Koordinaten von Punkten der Oberfläche in Bezug auf eine örtliche Basis
- Globale Beschreibung der Form des Objektes und unveränderlich zu starren Transformationen

Was sind Spin Images?

- genaue Beschreibung in Doktorarbeit von Andrew E. Johnson
- Kurzbeschreibung: Darstellung ist ein Array von Variablen
- Name Spin: Bilderzeugung vorstellbar wie ein Blatt, das um die Normale eines Punktes gedreht wird
- + Lokalisierung der Spin Images bei Reduzierung der Erzeugungsparameter ermöglicht Oberflächenerkennung von Durcheinander und Überdeckung
- + große Anzahl der Spin Images sind redundant → statistische Analyse zur Reduzierung und damit Beschleunigung der Erkennung

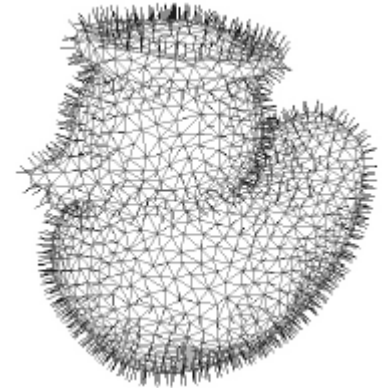


2. Oberflächenerkennung

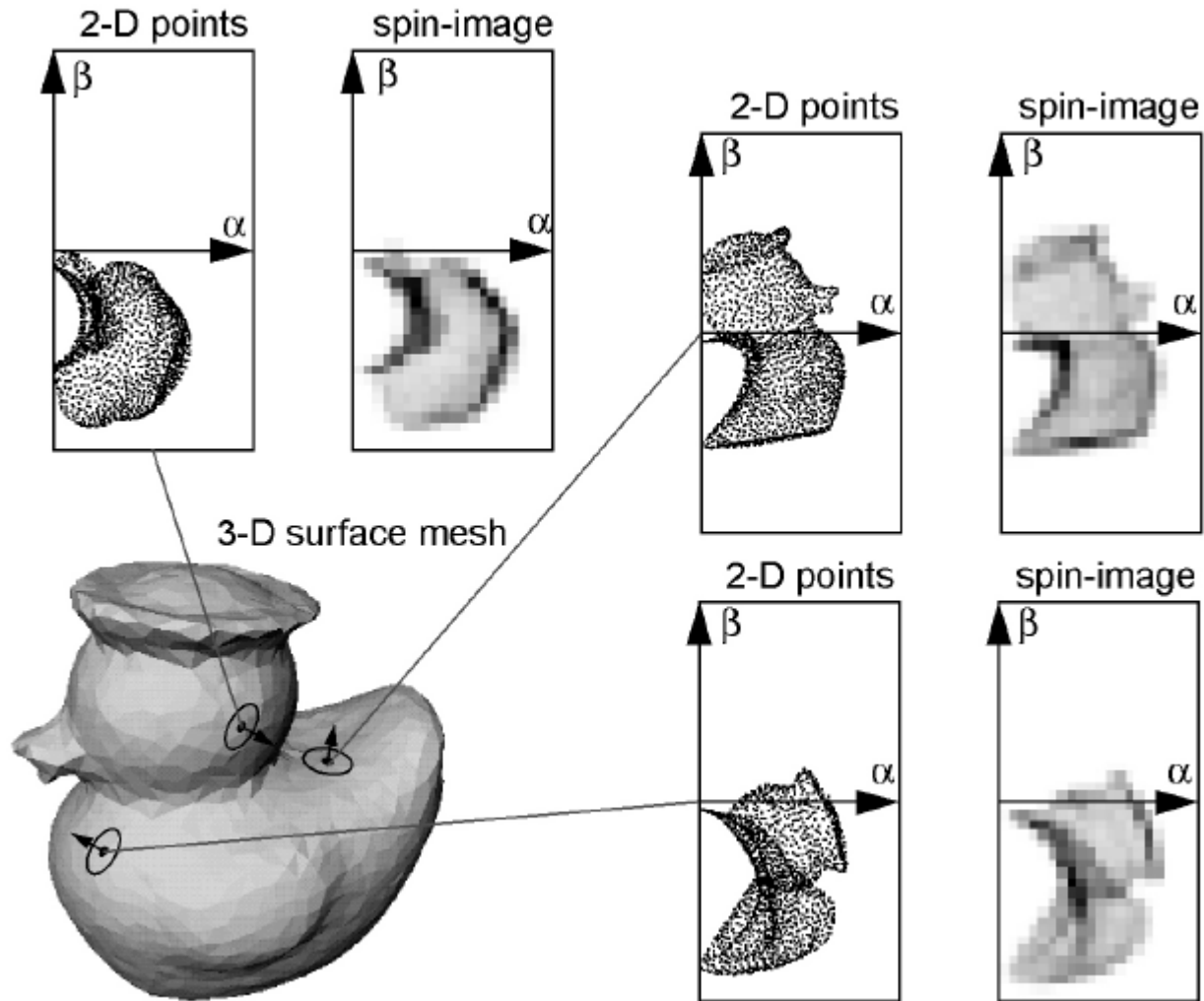
Spin Images Erzeugung

- 3D Punkt mit Richtung – orientierter Punkt
- Definition eines orientierten Punktes an einem Punkt des Oberflächengitternetzes
- Berechnung der Normalen → teilweise Definition eines Objekt-zentrierten Koordinatensystems
- Definition von 2 Koordinaten:
 - Höhenkoordinate a
 - Strahlenkoordinate β , definiert als senkrechtem Abstand durch die Normale
- Erstellung eines 2D Indexes mit a und β
- Koordinaten (a und β) eines Oberflächenpunktes in einem Oberflächenpolygon werden berechnet
- Wiederholung der Prozedur, bis alle Poygoneckpunkte im Spin Image
- kann als Abbildung gedacht werden – dunkle Teile entsprechen Bereiche mit vielen Punkten

- Spin Images ein und des selben Objektes werden ähnlich sein, aber nie gleich, da nur eine zufällige Auswahl an Punkten genutzt wird



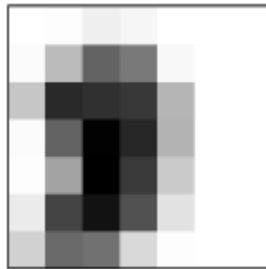
2. Oberflächenerkennung



2. Oberflächenerkennung

Parameter zur Spin Images Erzeugung

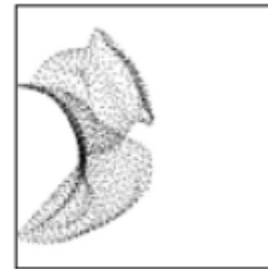
- geometrische Breite – die Auflösung der Spin Images
 - Beeinflussung der Größe bei Speicherung
 - Effekt auf die Beschreibbarkeit
 - wird in einem Vielfachen der Auflösung der Oberflächennetzpolygone angegeben
 - möglich, da Polygonauflösung mit der Größe der Größe der Formmerkmalen und der Dichte der Punkte und den Polygonen in Relation steht
 - 4 mal so groß – beschreibt das Modell nicht wirklich (a)
 - $\frac{1}{4}$ mal so groß – zu genaue Beschreibung; Platzverschwendung (c)
 - Genau so groß – idealer Mittelweg (b)



(a)



(b)

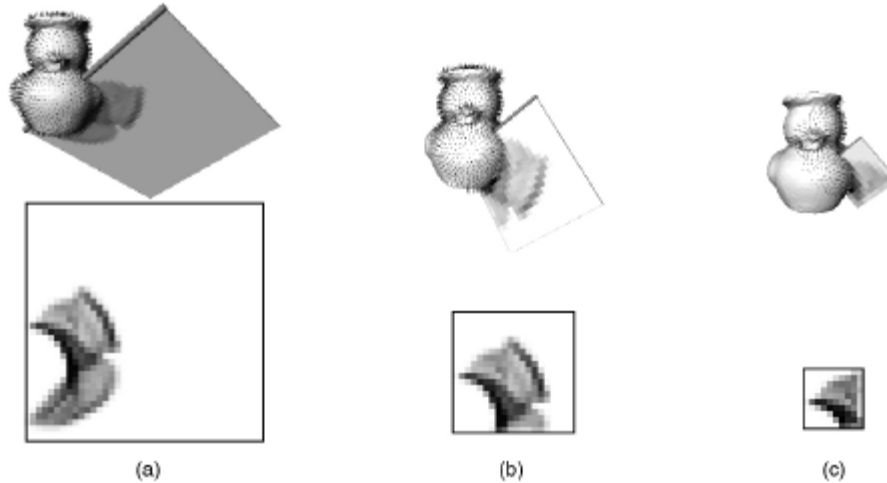


(c)

2. Oberflächenerkennung

Parameter zur Spin Images Erzeugung

- Bildbreite
 - entspricht der Fensterbreite bei 2D Bildern
 - Nach Tests: in etwa so groß setzen wie die breite des Modells

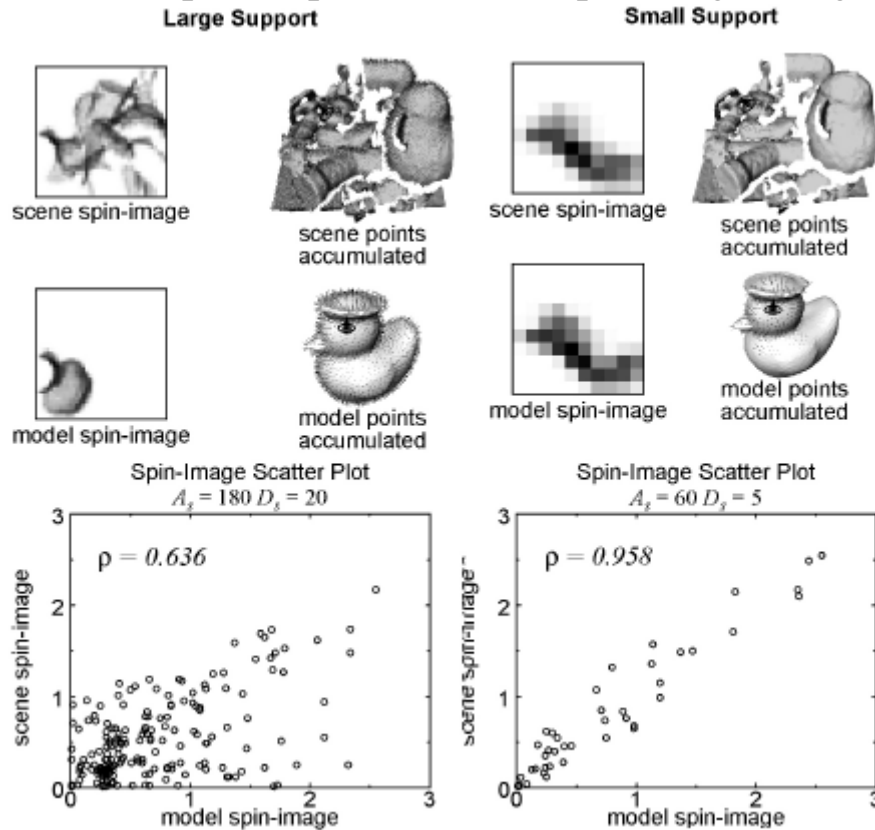


- (a) 40 Pixel (b) 20 Pixel und (c) 10 Pixel breit

2. Oberflächenerkennung

Parameter zur Spin Images Erzeugung

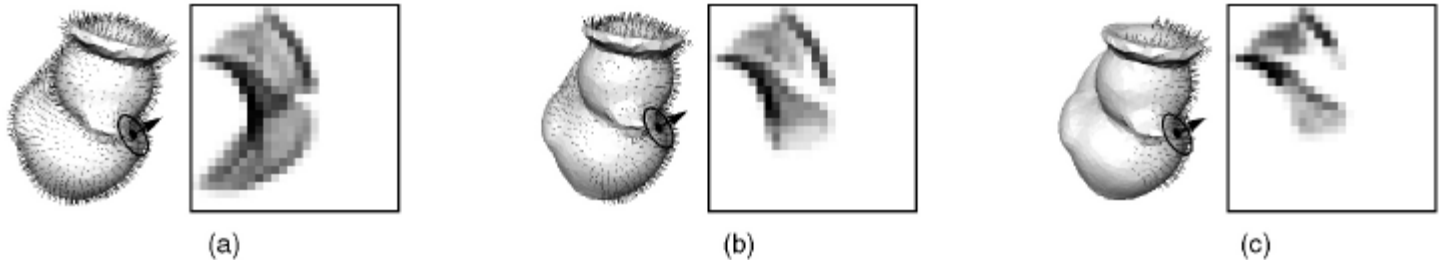
- Support Distance (DS)
 - Bildbreite * geometrische Breite
 - bestimmt den Speicherplatz, den das Spin Image belegt



2. Oberflächenerkennung

Parameter zur Spin Images Erzeugung

- Support Angle (AS)
 - Maximalwinkel zwischen der Richtung der orientierten Punktbasis und der Oberflächennormalwertes der Punkte
 - Bsp: Punkte A und B mit der Position und Normalen p_A und p_B sowie n_A und n_B
B wird ins Spin Image aufgenommen, wenn $a \cos(n_A * n_B) < AS$ ist
 - wird genutzt, um die Auswirkungen des Durcheinander einzugrenzen



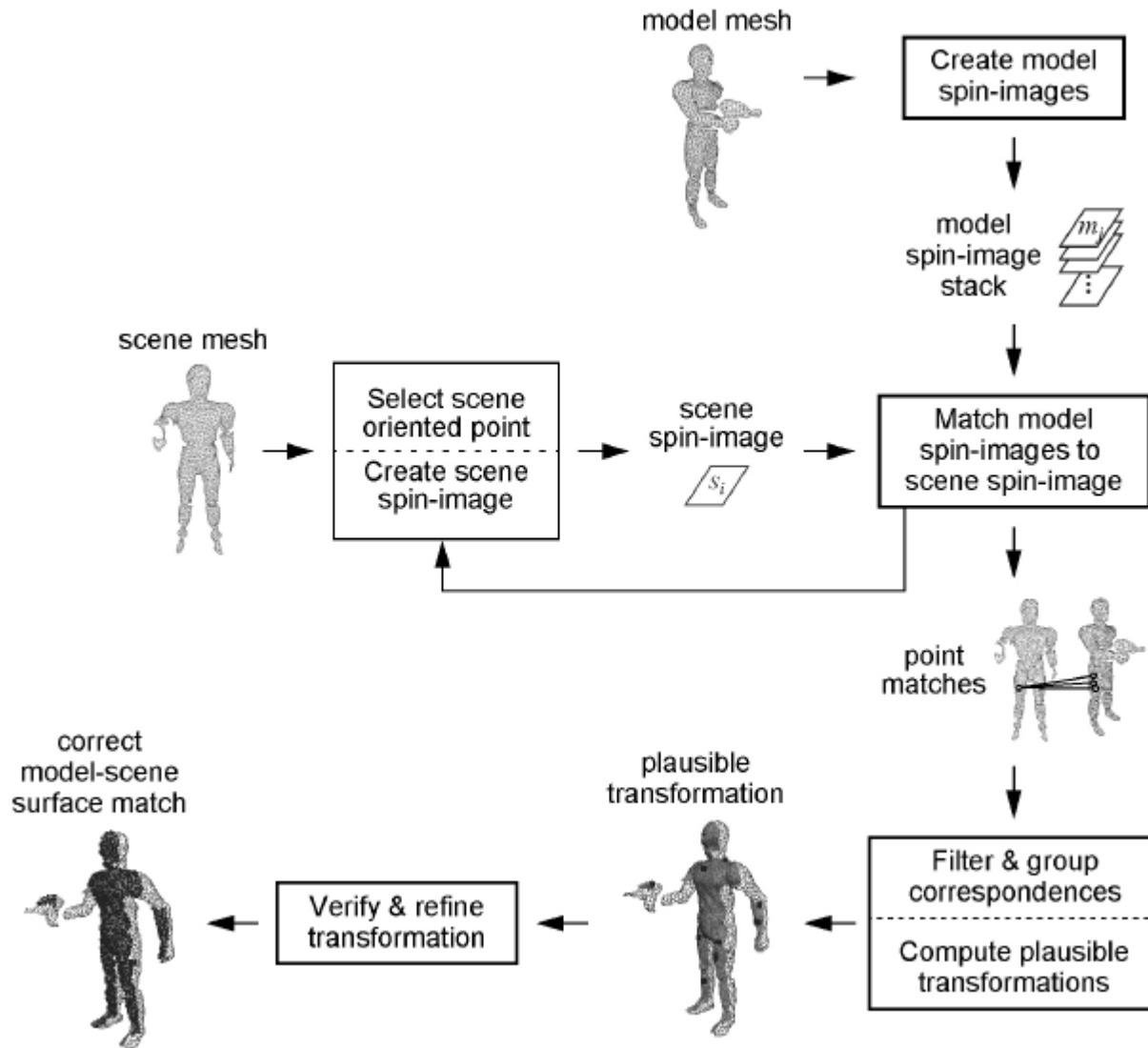
- (a) 180°, (b) 90° und (c) 60° AS
- DS und AS haben Auswirkungen auf die Anzahl der Punkte, die in der Szene genutzt werden; je größer diese ist, desto ungenauer wird die Erkennung, da mehr Punkte genutzt werden, die gar nicht zum Objekt gehören

2. Oberflächenerkennung

Oberflächenvergleichsverfahren

- Berechnung des Korrelationskoeffizienten mit Spin Images von Punkten der anderen Oberfläche
- bevor verglichen wird, werden alle Spin Images des Modells erstellt
- zufällige Auswahl von Punkten des Polygonnetzes und Berechnung der Spin Images
- Erstellung einer Punktkorrespondenz zwischen dem Punkt und dem am besten passenden Punkt der anderen Oberfläche
- Wiederholung bis eine große Anzahl an Punkten untersucht wurden
- Punkte werden gruppiert und Ausreißer entfernt
- Vergleich mithilfe eines modifizierten iterativen closed – point Algorithmus
- Vergleich mit der größten Überdeckung wird ermittelt

3. Objekterkennung



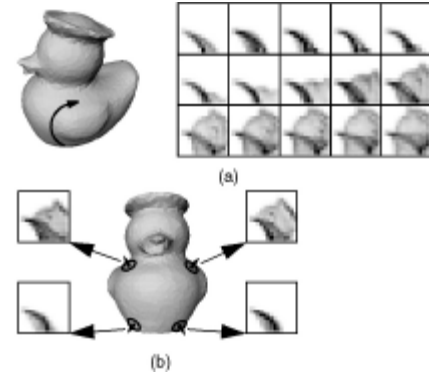
3. Objekterkennung

- jedes Modell ist in der Bibliothek als Polygonmodell gespeichert
 - vor der Erkennung werden Spin Images für alle Ansichten erzeugt und gespeichert
 - während der Erkennung wird ein Punkt der Szene ausgewählt und dessen Spin Image berechnet
 - errechnetes Spin Image wird mit allen Spin Images aller Modell verglichen
 - das am besten passende Spin Image zeigt das am besten passende Modell und die am besten passende Polygonecke
 - nach Vergleich vieler Spin Images werden korrespondierende Punkte an den Oberflächenvergleichsalgorithmus übergeben
- ineffiziente Erkennung:
- jeder Vergleich benötigt eine Beziehung von 2 Spin Images – eine Operation der Größenordnung der relativ der Größe der Arrays der Spin Images ist
 - unakzeptable Verarbeitungsgeschwindigkeit zur Erkennung einer großen Anzahl an Modellen
- Spin Images können komprimiert werden – höhere Geschwindigkeit

3. Objekterkennung

Spin Images Kompression

- Spin Images von Punkten die nahe beieinander liegen, ähneln sich sehr
- bei Oberflächensymmetrie: Punkte die auf gegenüberliegenden Seiten des Objektes liegen in Beziehungen gebracht werden
- Beziehung zwischen Punkten kann genutzt werden, um Spin Images mittels Bildkompression komprimieren, um den Vergleich zu beschleunigen
- Spin Images werden als Vektoren angesehen in einem D – dimensionalen Vektorraum, wo D die Anzahl der Pixel im Spin Image ist
- Beziehungen zwischen Spin Images werden mit einem niederdimensionalen Unterraum dargestellt
- allgemeine Methode der Bildkompression in der Objekterkennung ist ‚principal component analysis‘ (PCA) auch Karhunen – Loeve Erweiterung genannt
 - Berechnung von Richtungen größter Varianz einer Menge von Vektoren
 - Berechnung der Eigenvektoren der Kovarianzmatrix der Menge von Vektoren kann eine orthogonale Matrix – Eigenraum – abgebildet werden, in denen die Vektoren beschrieben sind
 - Entfernung l_2 im Spin Image – Raum zwischen 2 Spin Images ist genau so groß wie die l_2 Entfernung von 2 Spin Images im Eigenraum



3. Objekterkennung

Spin Images Kompression

- Vektoren werden in einen Unterraum abgebildet, der durch die größten Eigenwerte definiert ist
- Annahme: die Modellbibliothek hat N Modelle x_i der Größe D_i
- Mittel aller Spin Images: $\bar{x} = \sum_{i=1}^N x_i$
- Subtrahierung des Mittel von jedem einzelnen Spin Image – effektivere Berechnung der Hauptrichtung durch PCA für die Beschreibung des Unterschiedes der Spin Images
- $\hat{x}_i = x_i - \bar{x}$ Menge der Spin Images, die als $D \times N$ – Matrix mit jeder Spalte die Differenz aus Mittel und Spin Image: $s^m = [x_i \ \hat{x}_2 \ \dots \ \hat{x}_N]$
- Kovarianz der Spin Images ist in $D \times D$ - Matrix gegeben durch $c^m = s^m (s^m)^t$
- Eigenvektoren von C sind dann berechnet durch das Lösen des Eigenvektorproblems: $I^m e_i^m = c^m e_i^m$
- ist die Dimension der Spin Images nicht zu groß (~ 200), kann der Standart Jacobi Algorithmus genutzt werden zur Berechnung der Eigenvektoren und Eigenwerte
- Model – Projektions – Dimension wird über eine Rekonstruktionsmatrix bestimmt, die abhängig ist von der Genauigkeit der Rekonstruktion und der Varianz unter den Bilder

3. Objekterkennung

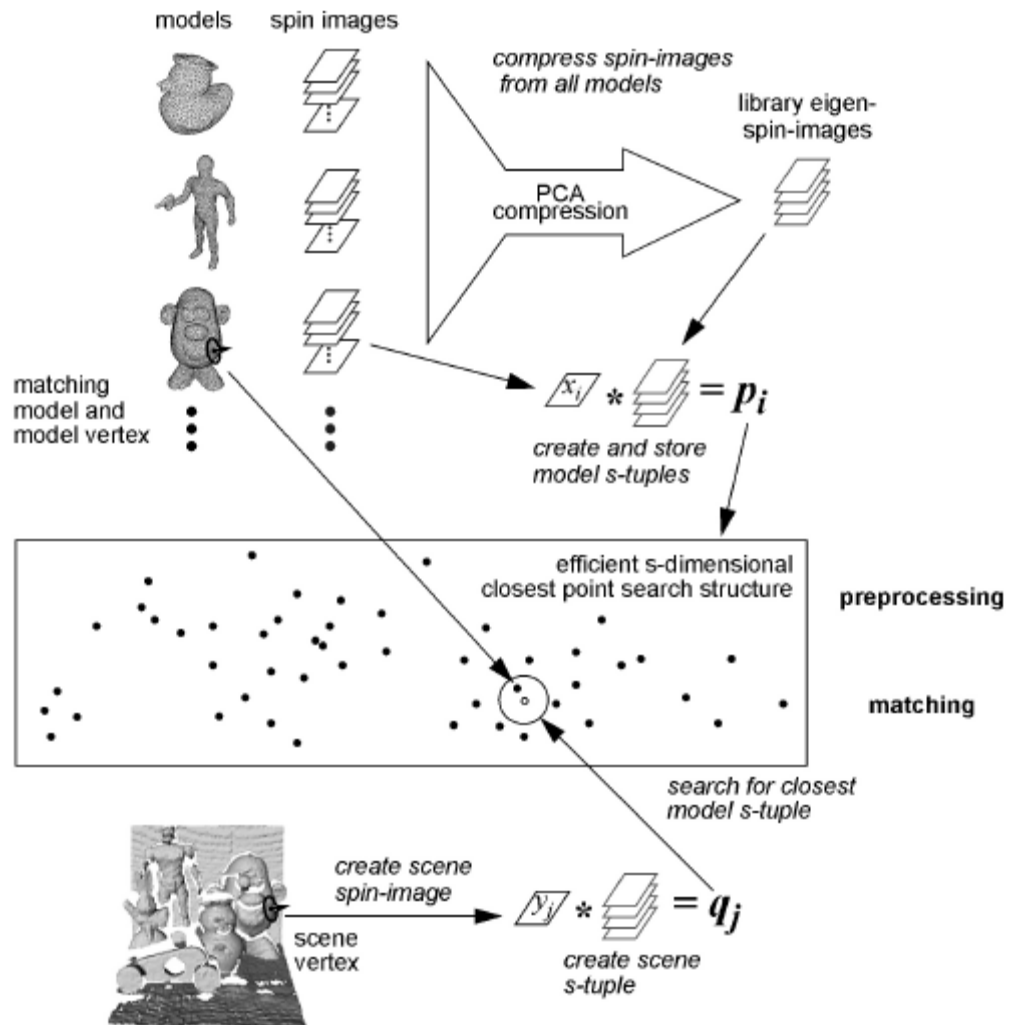
Spin Images Kompression

- jedes Spin Image eines Modells ist in s – dimensionalen Unterraum abgebildet, der den Eigenvektor der größten Eigenwerte umfasst, das s – Tupel des Projektionskoeffizienten p_j
- komprimierte Repräsentation des Spin Images: $p_j = (x_j e_i^m, x_j e_2^m, \dots, x_j e_s^m)$
- Maß der Kompression: $\frac{s}{D}$

Vergleich komprimierter Spin Images

- möglich, Spin Images in einer Zeit zu vergleichen, die unterhalb der Anzahl der Modell – Spin Images bei Nutzung effizienter closed – point Suchstrukturen liegt
- nötig, ein Szene s – Tupel zu erzeugen mit den selben Parametern, mit denen die Modell s – Tupel erzeugt wurden
- zur Ermittlung des am besten passenden Modell Spin Images zum Szenen Spin Images wird die l_2 Entfernung genutzt
- das gefundene naheste s – Tupel wird gegen das Spin Image ersetzt, das mit dem Szenen Spin Image in Beziehung gesetzt wurde (l_2 Entfernung ist nicht die selbe wie beim Korrelationskoeffizientenvergleich beim Spin Image Vergleich)
- Suchstruktur von Nene und Nayar genutzt, um den nahesten Punkt zu finden
- Effizienz beruht auf Annahme, dass ein Punkt nur interessant ist, wenn er weniger als einen bestimmten Abstand vom untersuchten Punkt hat

3. Objekterkennung



3. Objekterkennung

Vergleich komprimierter Spin Images

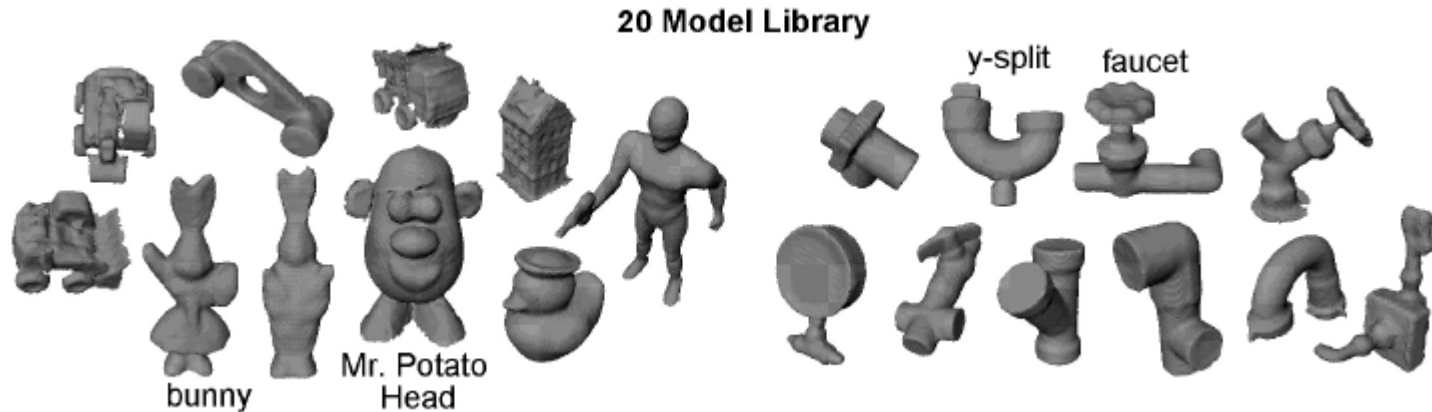
- vor Erkennung werden alle Modelloberflächen auf die selbe Größe skaliert
- Spin Images für jedes Modell sind in Bibliothek gespeichert
- Projektionsauflösung s bestimmbar
- Modell s – Tupel werden für jedes Modell erzeugt und in einer closed – Point Suchstruktur gespeichert

- zur Erkennung wird nur ein Bruchteil der Punkte der Szene ausgewählt
- Berechnung der Spin Images mithilfe der Szenedaten
- Erzeugung der Szene s – Tupel durch Abbildung der Szene Spin Images auf die Modell Eigenvektoren
- Szenen s – Tupel wird genutzt, um einen Anfangspunkt in der aktuellen Modell – Suchstruktur zu erhalten
- Rückgabe einer Liste mit Modell s – Tupeln, die passen könnten
- Übergabe der s – Tupelliste an einen Oberflächenvergleichsalgorithmus

3. Objekterkennung

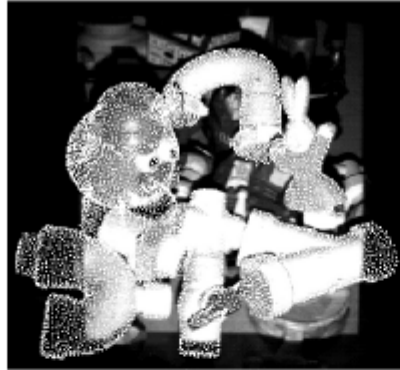
Resultate der Erkennung

- Erstellung einer Bibliothek mit (hier) 20 Objekten
- Mehrere Ansichten der Modelle wurden berücksichtigt
- 3 Sichten der Szene werden erzeugt: 3D Front, 3D Oben und Intensitätsansicht

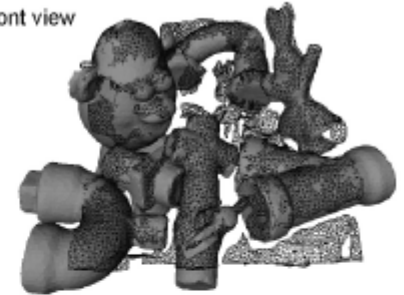
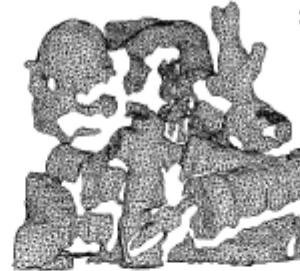


3. Objekterkennung

intensity image



3-D front view



3-D top view

