Histograms of Oriented Gradients for 3D Object Retrival

Philipp Lambracht

***Abstract*— Die 3D Objekterkennung ist ein wichtiges Themengebiet der Mobilen Systeme und der autonomen mobilen Robotik geworden. Ein populärer Ansatz um die Ähnlichkeit zwischen 3D Objekten zu bestimmen, sind globale Deskriptoren. Im Zuge meiner Ausarbeitung für das Proseminar „Mobile Systems Engineering“ habe ich den wissenschaftlichen Artikel „ Histograms of Oriented Gradients for 3d Object Retrieval“ von Maximilian Scherer, Micheal Walter und Tobias Schreck ausgewählt und werde den darin beschriebenen und entwickelten Deskriptor, im folgenden 3DHOG genannt, genauer vorstellen.**

I. EINLEITUNG

Zum Zeitpunkt der Veröffentlichung des von mir ausgewählten Wissenschaftlichen Artikel wurden viele unterschiedliche Methoden zur 3D Objekterkennung vorgestellt. Einzelne Deskriptoren konnten sich bisher nicht als überlegen herausstellen. Es hat sich vielmehr etabliert jeweilige Deskriptoren geschickt zu kombinieren um deren Stärken zu nutzen und somit eine weitaus bessere Performance zu erzielen. Dementsprechend wird der in [3] vorgestellte 3DHOG mit hoch-dimensionalen Merkmal Vektoren verglichen und eine Kombination mit diesem in einem Experiment versucht.
Hauptmotivation für der 3DHOG waren unter anderem bereits erfolgreiche Anpassungen von 2D Bild Analyse Methoden auf 3D Objekterkennung. Es wurde sich für die Anpassung des bereits erfolgreichen HOG aus [1] entschieden.

A. Grundbegriffe

Im folgenden werde ich ein paar wichtige Grundbegriffe für diese Ausarbeitung erläutern.

1) Globaler und partieller Ansatz: Bei der 3D Objekterkennung gibt es zwei verschiedene Ansätze. Der globale Ansatz betrachtet jeweils die komplette Form des 3D Models und es werden nach Ähnlichkeiten gesucht, während der partielle Ansatz nach lokalen Ähnlichkeiten sucht. Hierbei ist zu beachten, dass es bisher keine absolute Lösung des Ähnlichkeitsproblems existiert, weder für den globalen noch für den partiellen Ansatz. Dementsprechend haben entsprechende Lösungsversuche einen heuristische Natur. [3].

2) Histogramm: Histogramme dienen der in der Statistik und Bildverarbeitung dazu Häufigkeiten bestimmter Merkmale visuell darzustellen. Ein einfaches Beispiel aus der Bildverarbeitung wäre ein Histogramm eines Graustufenbildes mit den jeweils darin vorkommenden Grauwerten.

Tabelle II
GRAUWERTBILD ALS MATRIZE

a_{00}	a_{01}	a_{02}	a_{03}
a_{10}	100	50	235
a_{20}	73	42	150
a_{30}	30	125	0

Tabelle I
GRAUWERT HISTOGRAMM

Grauwert	Anzahl
150	30
20	5
...	...
255	10

Eine Detail reichere Einführung im Bezug auf Bildverarbeitung ist in [2] zu finden.

3) Gerichtete Gradienten: Gerichtete Gradienten werden wie z.B. in [1] äußerst erfolgreich zur Merkmaldetektion für 2D Bilder eingesetzt. Die Verwendung dieses Begriffs kann in [3] und dementsprechend in dieser Ausarbeitung vom mathematischen Begriff abweichen.
Um gerichtete Gradienten zu berechnen, benötigt man Gradientenoperatoren. Hiermit sind Lineare Filter aus der Bildverarbeitung gemeint. In der Einführungslektüre [2] versteht man Filter als Funktionen welche auf Bilder, als Matrizen darstellbar, angewendet werden. Gradientenoperatoren sind gemäß der Definition über differenzierbare Funktionen, eine entsprechende Approximation mit denen man z.B. 2D Bilder „ableiten“ kann. Die Filtermaske 1

$$\begin{bmatrix}-1 & 0 & 1\end{bmatrix} \tag{1}$$

bewirkt z.B. die 1. Ableitung. Dieser Filter kann z.B. für ein 2D Bild dementsprechend in die X-Richtung und in die Y-Richtung angewendet werden.

Formel 2 zeigt ein Beispiel, wie ein Element aus Tabelle II abgeleitet wird. In diesem Fall in X-Richtung. An den Rändern muss jeweils eine Randbehandlung vorgenommen werden. Werte können z.B. gespiegelt werden.

$$a'_{22}x = -75 + 0 + 150 = 75 \tag{2}$$

Mit den Gradientenoperatoren lässt sich jeweils die Gradientenlänge bzw. -betrag und die Gradientenrichtung berechen. Die Forlmen (3) und (4), entnommen aus [2] zeigen jeweils die Berechnung für 2D Bilder. I_x bzw. I_y steht jeweils für die Ableitung in X- bzw. Y-Richtung. Mit dem Parameter p ist den entsprechende Pixel gemeint.

$$G_l(p) = \sqrt{I_x^2(p) + I^2y(p)} \tag{3}$$

$$G_r(p) = arctan_2(-I_y(p), I_x(p)) \tag{4}$$

4) 3D Mesh: 3D Mechs werden dazu verwendet um 3D Objekte digital zu speichern. Es werden Informationen über die Vertices (Punkte), Kanten, Flächen, Polygone sowie falls nötig Informationen über die Oberfläche (z.B. Farbe) gespeichert. In dem von mir Ausgewählten Artikel [3] werden Meshs aus schon bestehenden Performance-Tests genommen um die Leistungsfähigkeit des 3DHOG zu messen.

II. HAUPTTEIL

In diesem Abschnitt werde ich zunächst den 2DHOG aus [1] kurz vorstellen, mit dem Hauptthema 3DHOG fortfahren und zuletzt das in [3] durchgeführte Experiment aufgreifen.

A. 2DHOG

Im Bereich der 2D Objekterkennung aus Bildern existieren bereits erfolgreiche Methoden. Der Skale-Invariant-Feature-Transform Algorithmus (SIFT),genauere Beschreibung z.B. in [2] zu finden, arbeitet mit aggregierten Gradienten. Beim 2DHOG werden hingegen die Gradienten entsprechend ihrer Richtung in Histogrammen eingeordnet.
Die Idee hinter dem 2DHOG ist, dass sich Form und Aussehen von Objekten mit Gradienten beschreiben lassen. Dies ist selbst möglich, ohne die genaue Position der Gradienten zu kennen. Der 2DHOG läuft grob nach folgenden Schema ab. Zuerst werden die Farbwerte des Bilds,auf dem die Detektion durchgeführt wird, normalisiert. Danach wird das Bild in gleich große, rechteckige Zellen aufgeteilt. Dabei können einzelne Zellen überlappen. Für jede dieser Zellen werden Histogramme für die jeweils berechneten Gradienten angelegt. Die Einteilung erfolgt entsprechend ihrer Richtung. Die Ergebnisse müssen normalisiert werden. Die HOGs werden mittels Detektionsfenster extrahiert und an eine Support Vector Machine (SVM) weitergeben. Danach kann entschieden werden, ob das entsprechende Objekt gefunden wurde. Im Fall von [1] Menschen. In dem eben genannten Wissenschaftlichen Artikel hat sich durch Experimente herausgestellt, dass die einfache Ableitungsmaske 1 zur Berechnung der Gradienten die besten Ergebnisse liefert. Es wurden andere Ableitungsfilter, wie z.B. der Sobel-Operator (Formel 5, entnommen aus [2]), jedoch waren die Ergebnisse eher enttäuschend.

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} S_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{5}$$

Auch wurde, zwecks Optimierung, mit Gaußfiltern experimentiert. Eine Performenceverbesserung wurde ebenfalls nicht erzielt. Detailliertere Informationen über den 2DHOG sind in [1] zu finden.

B. 3DHOG

1) Erweiterung des 2DHOG auf 3DHOG: Der erste Schritt, die Berechnung der Gradienten, erweist sich bei der Erweiterung auf 3DHOG ein wenig komplizierter. Zunächst Benötigt man eine Notation für Nachbarschaft und Intensität für die Polygon Meshs. In [3] wird dafür ein dreidimensionales euklidisches Distanzfeld berechnet. Dieses Feld ist als eine reellwertige Funktion aufzufassen, welche auf einem diskreten, regulären 3D Gitter definiert ist. Das Gitter umfasst dabei das komplette Volumen des Meshs [3]. Die jeweiligen Gitterzellen können auch als Voxel bezeichnet werden. Jeder Voxel enthält dabei die Information über den Abstand seines Zentrums zur Oberfläche des Meshs.

$$f : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \mapsto \mathbb{R}$$

$$f(x, y, z) = \min_{x \in \Sigma} \|x - center(x, y, z)\|_2 \tag{6}$$

Eine Definition der Funktion ist bei Formel (6) zu sehen, entnommen aus [3]. Σ ist hierbei die Menge aller Punkte auf der Oberfläche des Meshs und $center : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \mapsto p$ liefert die Koordinate des Zentrums des Voxels zurück.
Auf das berechnete Distanzfeld kann man z.B. die Filtermaske $\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$ aus [1] anwenden für die Gradientenberechnung.
Das Distanzfeld wird sehr stark von Position und Größe des Objekts beeinflusst. Dementsprechend muss das Mesh vor der Distanzfeldberechnung normalisiert werden. In [3] greift man deshalb auf Translationsinvarianz (das Zentrum der Masse des Meshs wird in den Ursprung verlegt), Skalierungsinvarianz (Skallierung des Meshs in den Einheitswürfel), sowie eine Normalisierung für Rotation mittels gewichteter PCA analyse.
Der zweite Schritt ist um weiten simpler. Die dreidimensionalen Gradienten werden jeweils entsprechend ihrer Richtung in Histogramme für die einzelnen Zellen eingeordnet. Hierfür werden die Gradienten in sphärische Koordinaten entsprechend Formel (7), entnommen aus [3], umgerechnet.

$$\begin{pmatrix} \theta \\ \phi \\ r \end{pmatrix} = \begin{pmatrix} \arccos \dfrac{z}{\sqrt{x^2 + y^2 + z^2}} \\ atan2(x, y) \\ \sqrt{x^2 + y^2 + z^2} \end{pmatrix} \tag{7}$$

Die Einordnung erfolgt entsprechend ihrer Richtung ($Zenit\theta \in [0, \pi)$ und $Azimut\phi \in [0, 2\pi)$)

2) 3DHOG Extraktionsalgorithmus : Der schematische Ablauf des Extraktionsalgorithmus ist in Abbildung 1 zu sehen. Genaueres über die Implementation ist in dem in [3] mitgelieferten Sourcecode [1] zu entnehmen

C. Experiment

Im folgenden werde ich das in [3] durchgeführte Experiment und dessen Ergebnisse vorstellen. Um die Effizienz der Deskriptoren zu vergleichen werden Precision-and-recall-Diagramme verwendet

[1] www.gris.informatik.tu-darmstadt.de/projects/vsa/3dhog/3dhog.zip

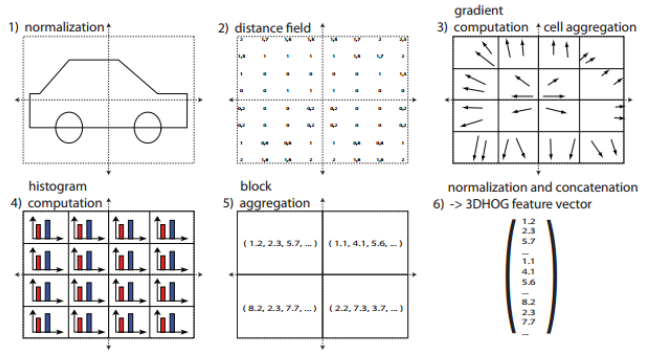

Abbildung 1. Vektor Extraktionspipeline aus [3]

1) Verwendete Benchmarks: Für das Experiment wurden drei etablierte Benchmarks genommen, welche 3D Mesh Modelle enthalten. Der Princton Shape Benchmark (PSB), 2009 SHREC Generic Shape Retrieval Contest dataset (SHREC) und Konstanz 3D shape database (KN-DB). Die einzelnen 3D Modelle sind in verschiedene Klassen eingeteilt (z.B. Menschen, Tiere, Autos, ..), damit verschiedene 3D Deskriptoren besser miteinander verglichen werden können. Je nachdem um welche Objektart es sich handelt, liefern Deskriptoren unterschiedlich gute Ergebnisse.

Tabelle III

Benchmark	anz. Modelle	anz. Klassen	durchschn. anz. M. pro K.
PSB	1814	92	10
KN-DB	473	55	9
SHREC	720	40	18

2) Verwendete Vergleichsdeskriptoren: Im folgenden werde ich die Vergleichsdeskriptoren kurz anreißen. Details sind in [4] zu finden.

a) 438-dimensional Depth-Buffer Descriptor (DBD438): Dieser Deskripter nutzt das aus der Computergrafik bekannte Tiefenpufferverfahren. Nach [3] gilt dieser Deskriptor als einer der effektivsten.

b) 300-dimensional Silhoutte-based Descriptor(SIL300): Der SIL300 arbeitet mit der Zerlegung des 3D Models in 2D Silhouetten mit den jeweiligen Achsen (y,z) (z,y) und (x,y).

c) 136-dimensional Descriptor based on Radial Extent function (RSH136) : Bei diesem Deskriptor wird mit der Ausdehnung von 3D Objekten gearbeitet. Die Objekte werden entlang gegebenen Richtungen(entlang von vorher definierten Strahlen) gemessen.

d) 472-dimensonal Hybrid Descriptor (DSR472) : Hierbei handelt es sich um einen Hybrid aus den vorigen Deskriptoren. Da hier geschickt die einzelnen Stärken kombiniert werden, ist sogar dem DBD438 überlegen.

3) Ergebnisse des Experiment: Zunächst wurde überprüft, ob sich zur Gradientenberechnung ggf. Gradienten der 2. Ableitung (Formel 8) bessere Ergebnisse liefern.

$$\begin{bmatrix} 1 & 0 & -2 & 0 & 1 \end{bmatrix} \tag{8}$$

Da dies der Fall war, wurden alle weiteren Benchmarks mit der 2. Ableitung durchgeführt. Im Vergleich mit den einzelnen Deskriptoren schlägt sich der 3DHOG Relativ gut, schneidet sogar beim SHREC Benchmark besser als die anderen Deskriptoren ab (Abbildung 2).

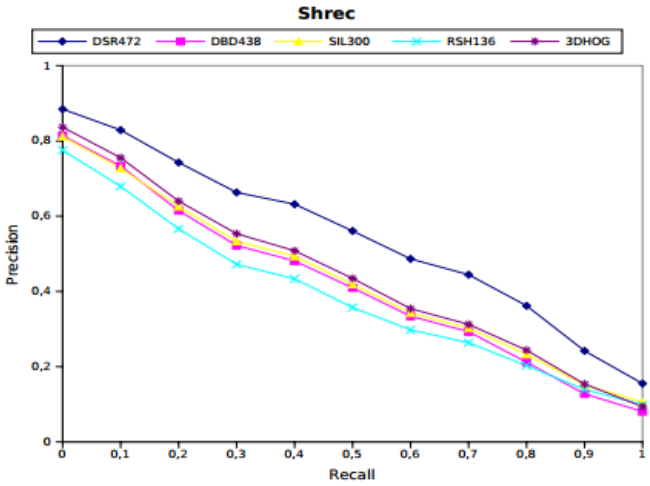

Abbildung 2. SHREC Precision-Recall-Diagramm aus [3]

Entsprechend wenig überraschend ist er aber dem Hybrid Deskriptor unterlegen. Dennoch schneidet der 3DHOG beim SHREC Benchmark bei einzelnen Klassen besser ab, als der Hybrid. Zu sehen in Abbildung 3

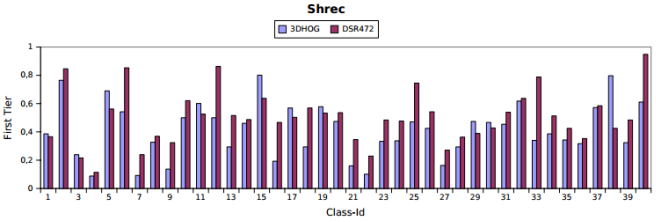

Abbildung 3. 1. Tier Präzision der verschieden Klassen des SHREC Benchmark aus [3]

Dies legt nahe, dass der 3DHOG wichtige 3D Merkmale erkennt, die den anderen Deskriptoren entgehen [3].
Entsprechende Versuche, in denen der 3DHOG mit dem DSR472 kombiniert wurde führten zu einer Performanceverbesserung, wie in Abbildung 4 zu sehen.

Der 3DHOG stellt damit einen wertvollen Beitrag zur Verbesserung von 3D Objekterkennungssystemen da [3].

III. DISKUSSION

Dieser Abschnitt befasst sich mit der optimalen Parameter des 3DHOG, welche sich aus dem Experiment aus [3] ergeben hat. Außerdem gehe ich ebenfalls auf ein dort eingetreten Problem bei der Gradientendefinition ein.

A. Parameterwahl des 3DHOG

Das Experiment von Scherer, Walter und Schreck in [3] hat gezeigt, dass die Parameterwahl für den 3DHOG die

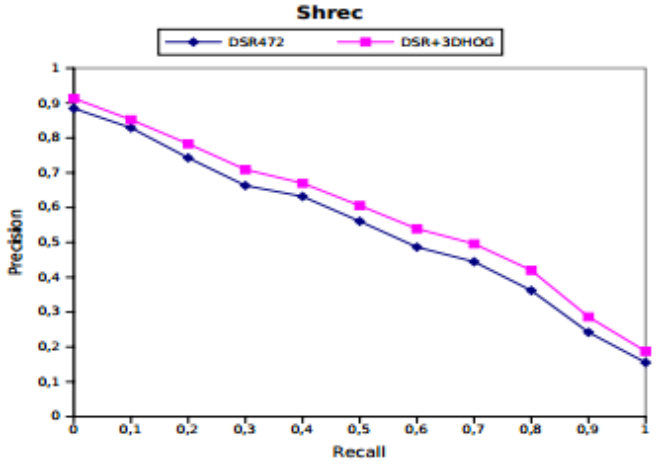

Abbildung 4. SHREC Precision-Recall-Diagramm aus [3]

Performance stark beeinflussen kann. Eine sich als optimal erwiesene Einstellung ist in Abbildung IV zu finden.

Tabelle IV
OPTIMALE PARAMETERWAHL FÜR 3DHOG, ENTNOMMEN AUS [3]

Parameter	Wert
$r_{x,y,z}$	$\frac{2}{52}$
$c_{x,y,z}$	12 vxl
bins(θ)	9
bins(ϕ)	9
$b_{x,y,z}$	2 Zellen
S $o_{x,y,z}$	0 Zellen
Dimensionalität	5184

Der Parameter $r_{x,y,z}$ legt die Anzahl der Voxel im Distanz Feld fest. $r_{x,y,z}$ steht jeweils für die Kantenlänge jedes Voxels. Je kleiner die Kantenlänge gewählt wird desto weniger Informationen gehen verloren, jedoch erhöht sich die Rechenzeit. Die Zellengröße $c_{x,y,z}$ legt fest wie viele Gradienten in ein Histogramm aufgenommen werden. Damit lässt sich der Grad der Lokalität des Deskriptors festlegen [3]. Mit den Parametern bins(θ) und bins(ϕ) wird die Feinheit der Einteilungen des Histogramms festgelegt. Hier hat man die Wahl zwischen Genauigkeit und Stabilität. Die nächsten beiden Parameter legen jeweils die Größe der Blöcke ($b_{x,y,z}$) und ihre Überlappung ($o_{x,y,z}$) fest. Damit wird festgelegt, wie viele Benachbarte Zellen miteinander normalisiert werden. Diese beiden Parameter wurden zunächst vielversprechenden Parametern aus [1] gewählt. Die Überlappung hatte jedoch nicht den erwarteten positiven Effekt. Deshalb wurd der Wert 0 gewählt.

B. Alternative Gradientendefinition

Durch Experimentieren hat sich herausgestellt, das sich die 2. Ableitung für die Gradientenberechnung des 3DHOG als effektiver erwiesen hat. Scherer, Walter und Schreck [3] begründen es damit, dass nicht nur Informationen über Lokale Extrema in der nähe Oberfläche des Meshs, sondern auch Informationen innerhalb des Meshs nützlich sein können. Zudem liefert die 1. Ableitung nicht genau

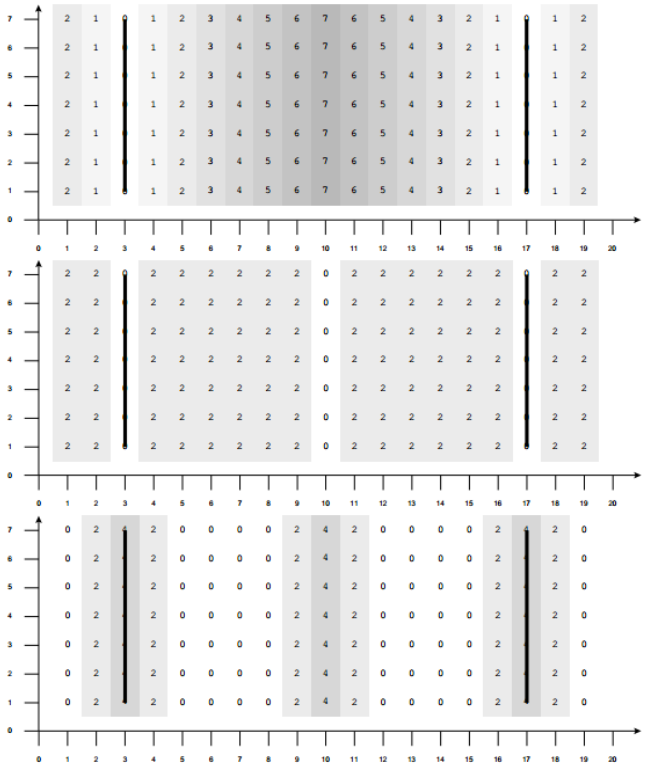

Abbildung 5. 2D Darstellung eines Distanz Feldes, entnommen aus [3]. Das oberste Bild zeigt das Eigentliche Distanz Feld, das mittlere unter Verwendung der 1. Ableitung, das untere mit der 2. Ableitung

das, was man unter Gradienten in der Bildverarbeitung versteht. Einen Pixel kann man als einen Punkt in der Welt verstehen, in dem Informationen über das reflektierte Licht gespeichert werden. Der entsprechende Gradient würde z.B. an Ecken von Wänden an stärke zunehmen. Abbildung 5 zeigt deutlich, dass dies bei der 1. Ableitung nicht der Fall ist. Die 2. Ableitung hingegen erfüllt die Grundauffassung von Gradienten in der Bildverarbeitung.

REFERENCES

[1] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.
[2] Lutz Priese. *Computer Vision*. Springer Vieweg, 2015.
[3] Maximilian Scherer, Michael Walter, and Tobias Schreck. Histograms of oriented gradients for 3d object retrieval. 2010.
[4] Dejan V. Vranić. *3D Model Retrieval*. PhD thesis, University of Leipzig, 2004.