Proseminar Objektposenschätzung

Robert Jeutter

13. November 2021

Damit ein Roboter einen Gegenstand greifen kann, ist es meist notwendig die genaue Lage des Objektes zu kennen. Dies kann sowohl über klassische Verfahren als auch über Deep-Learning-Verfahren erreicht werden. Ziel dieses Seminars ist es den Stand der Technik für die Objektposenschätzung aufzuarbeiten und vorzustellen. Der Fokus sollte dabei auf Verfahren liegen, bei denen zuvor kein Objektmodell benötig wird, sodass auch die Lage von unbekannten Objekten geschätzt werden kann.

1 Motivation

Die Erkennung von Objekten und die Schätzung ihrer Lage in 3D hat eine Vielzahl von Anwendungen in der Robotik. So ist beispielsweise die Erkennung der 3D-Lage und Ausrichtung von Objekten wichtig für die Robotermanipulation. Sie ist auch bei Aufgaben der Mensch-Roboter-Interaktion nützlich, z. B. beim Lernen aus Demonstrationen.
Traditionell wird das Problem der Objektposenschätzung durch den Abgleich von Merkmalspunkten zwischen 3D-Modellen und Bildern angegangen. Diese Methoden setzen jedoch voraus, dass die Objekte reichhaltig texturiert sind, um Merkmalspunkte für den Abgleich zu erkennen. Daher sind sie nicht in der Lage, mit Objekten ohne Textur umzugehen. Die meisten bestehenden Ansätze zur Objektposenschätzung setzen den Zugriff auf das 3D-Modell einer Objektinstanz voraus. Der Zugang zu solchen 3D-Modellen erschwert die Verallgemeinerung auf neue, unbekannte Instanzen. Darüber hinaus erfordern 3D-Modelldatenbanken oft einen nicht unerheblichen manuellen Aufwand und Expertenwissen, um sie zu erstellen, wobei Schritte wie Scannen, Netzverfeinerung oder CAD-Design erforderlich sind. Unordnung und Verdeckungen zwischen den Objekten senken die korrekte Erkennung bei modellbasierten Verfahren zudem deutlich. Bei schablonenbasierten Methoden wird eine starre Schablone konstruiert und verwendet, um verschiedene Stellen im Eingabebild zu scannen. An jeder Stelle wird ein Ähnlichkeitswert berechnet, und die beste Übereinstimmung wird durch den Vergleich dieser Ähnlichkeitswerte ermittelt. Schablonenbasierte Methoden sind nützlich für die Erkennung texturloser Objekte. Sie können jedoch nicht sehr gut mit Verdeckungen zwischen Objekten umgehen, da die Vorlage einen niedrigen Ähnlichkeitswert hat, wenn das Objekt verdeckt ist. Alternativ dazu können Methoden, die eine Regression von Bildpixeln auf 3D-Objektkoordinaten erlernen, um 2D-3D-Korrespondenzen für die 6D-Positionsschätzung herzustellen, nicht mit symmetrischen Objekten umgehen. Außerdem können bei der Verfolgung durch solche dynamische on-the-fly Rekonstruktion von Objekten Fehler entstehen, wenn Beobachtungen mit fehlerhaften Posenschätzungen in das globale Modell einfließen. Diese Fehler wirken sich nachteilig auf die Modellverfolgung in nachfolgenden Bildern aus.

2 Kategorisierung

Motiviert durch die oben genannten Einschränkungen, zielt diese Arbeit auf eine genaue, robuste 6D-Objekterkennung ab, die auf neuartige Objekte ohne 3D-Modelle verallgemeinert werden kann. Die Kategorisierung aller Verfahren erfolgt nach folgendem Schemata

Modell Müssen für das Training oder Nutzung merkmalsbasierte oder Objektmodelle (2D, 3D, CAD) vorhanden sein?

Video-Input Verarbeitet das Verfahren 2D Bilder, 3D Bilder mit Tiefenwahrnehmung?

Datensatz mit welchen Datensätzen wurde das Verfahren trainiert oder getestet?

Genauigkeit Wie akkurat ist die Objektposenschätzung im Vergleich?

Ressourcen Wie Ressourcenintensiv ist das Verfahren? Wird spezielle Hardware benötigt?

Laufzeit Mit welcher Geschwindigkeit ist die Verarbeitung von Eingabedaten möglich und stabil?

3 Verschiedene Verfahren

3.1 BundleTrack

BundleTrack[WB21] ist ein Framework für die 6D-Positionsverfolgung neuartiger Objekte, das nicht von 3D-Modellen auf Instanz- oder Kategorieebene abhängt. Es nutzt komplementären Eigenschaften für die Segmentierung und robuste Merkmalsextraktion sowie die speichererweiterte Pose-Graph-Optimierung für die räumlich-zeitliche Konsistenz. Dies ermöglicht eine langfristige, abdriftarme Verfolgung in verschiedenen anspruchsvollen Szenarien, einschließlich erheblicher Verdeckungen und Objektbewegungen.
Im Vergleich zu modernen Methoden, die auf einem CAD-Modell der Objektinstanz basieren, wird eine vergleichbare Leistung erzielt, obwohl die vorgeschlagene Methode weniger Informationen benötigt. Eine effiziente Implementierung in CUDA ermöglicht eine Echtzeitleistung von 10 Hz für das gesamte System. Der Code ist verfügbar unter: github.com/wenbowen123/BundleTrack

Modell ohne Modelle
Video-Input RGB-D
Datensatz NOCS, YCBInEOAT, Davis[Pa17], Youtube-VOS[Xa18]
Genauigkeit kann mit Verdeckung und Objektbewegung gut umgehen. Vergleichbare Leistung mit Methoden mit CAD Modell. Im NOCS-Datensatz:

- $87,4\%$ 5°5cm
- $99,9\%$ IoU25
- $R_{err} = 2,4$
- $T_{err} = 2,1$

Ergebnisse aus AUC Messung

- ADD $87,34\%$
- ADD-S $92,53\%$

Ressourcen effiziente CUDA-Implementierung, ermöglicht es, das rechenintensive Multi-Pair-Feature-Matching sowie die Pose-Graph-Optimierung für die 6D-Objekt-Positionsverfolgung online auszuführen. Alle Experimente wurden auf einem Standard-Desktop mit Intel Xeon(R) E5-1660 v3@3.00GHz Prozessor und einer einzelnen NVIDIA RTX 2080 Ti GPU durchgeführt.
Laufzeit in CUDA Echtzeit mit 10 Hz

3.2 DeepIM

DeepIM[Li+18] basiert auf einem tiefen neuronalen Netzwerk für iterative 6D-Positionsanpassung. Ausgehend von einer anfänglichen 6D-Positionsschätzung eines Objekts in einem Testbild, sagt DeepIM eine relative SE(3)-Transformation voraus, die eine gerenderte Ansicht des Objekts mit dem beobachteten Bild abgleicht. Bei einer anfänglichen Posenschätzung ist das Netzwerk in der Lage, die Pose iterativ zu verfeinern, indem es das gerenderte Bild mit dem beobachteten Bild abgleicht. Durch die iterative Neudarstellung des Objekts auf der Grundlage der verbesserten Posenschätzungen werden die beiden Eingangsbilder des Netzes immer ähnlicher, wodurch das Netz immer genauere Posenschätzungen erzeugen kann. Das Netzwerk wird so trainiert, dass es eine relative Pose-Transformation vorhersagen kann, indem es eine unverzerrte Darstellung der 3D-Position und 3D-Orientierung und einen iterativen Trainingsprozess verwendet.

Modell 3D-CAD-Modell
Video-Input RGB
Datensatz LINEMOD, LM-Occlusion
Genauigkeit LINEMOD

- $85,2\%$ 5°5cm
- $88,6\%$ 6D Pose
- $97,5\&$ 2D Projection

Ressourcen NVIDIA 1080 Ti GPU mit 2 Iterationen während der Tests
Laufzeit 12fps

3.3 MaskFusion

MaskFusion[RBA18] ist ein objektbewusstes, semantisches und dynamisches RGB-D SLAM-System in Echtzeit, das über traditionelle Systeme hinausgeht, die eine rein geometrische Karte einer statischen Szene ausgeben. MaskFusion erkennt, segmentiert und ordnet verschiedenen Objekten in der Szene semantische Klassenlabels zu, während es sie verfolgt und rekonstruiert, selbst wenn sie sich unabhängig von der Kamera bewegen. Während eine RGB-D-Kamera eine unübersichtliche Szene abtastet, erzeugt die bildbasierte semantische Segmentierung auf Instanzebene semantische Objektmasken, die eine Objekterkennung in Echtzeit und die Erstellung einer Darstellung auf Objektebene für die Weltkarte ermöglichen. Im Gegensatz zu früheren, auf Erkennung basierenden SLAM-Systemen benötigt MaskFusion keine bekannten Modelle der Objekte, die es erkennen kann, und kann mit mehreren unabhängigen Bewegungen umgehen. MaskFusion nutzt die Vorteile der semantischen Segmentierung auf Instanzebene,

um semantische Beschriftungen in eine objektbezogene Karte zu integrieren, im Gegensatz zu neueren semantischen SLAM-Systemen, die eine semantische Segmentierung auf Voxel-Ebene durchführen.
MaskFusion ermöglicht dichtes dynamisches RGBD-SLAM in Echtzeit auf Ebene von Objekten. Im Wesentlichen ist MaskFusion ein Multi-Modell-SLAM System, das für jedes Objekt, das es in der Szene erkennt, eine 3D-Darstellung verwaltet das es in der Szene erkennt (zusätzlich zum Hintergrundmodell). Jedes Modell wird unabhängig verfolgt und fusioniert.
Was die Erkennung betrifft, so kann MaskFusion nur Objekte aus Klassen erkennen, auf die die MaskRCNN trainiert wurde (derzeit die 80 Klassen des MS-COCO-Datensatzes) und berücksichtigt keine Fehlklassifizierung von Objektbeschriftungen.

Modell mit Modell
Video-Input RGB-D
Datensatz MS-COCO
Genauigkeit NOCS

- $26,5\%$ 5°5cm
- $64,9\%$ IoU25
- $28,5$ R_err
- $8,3$ T_err

Ergebnisse aus AUC Messung

- ADD $35,07\%$
- ADD-S $41,88\%$

Ressourcen Die Faltungsmaskierung läuft asynchron zum Rest von MaskFusion und erfordert eine spezielle GPU. Sie arbeitet mit 5Hz, und da sie den Grafikprozessor über lange Zeiträume blockiert, wird ein anderen Grafikprozessor für die SLAM-Pipeline verwendet, der mit $>$ 30Hz arbeitet, wenn ein einzelnes Modell verfolgt wird. Bei Vorhandensein mehrerer nicht-statischer Objekten sinkt die Leistung und führt zu einer Bildwiederholrate von 20 Hz für 3 Modelle. Das Testsystem ist mit zwei Nvidia GTX Titan X und einem Intel Core i7, 3.5GHz auszgestattet.
Laufzeit $20 - 30$Hz

3.4 Neural Analysis-by-Synthesis

Der neuronale Analyse-durch-Synthese-Ansatz [Che+20] ist zur Schätzung der Objektposition auf Kategorieebene. Durch den Einsatz eines gelernten Bildsynthesemoduls ist dieser Ansatz in der Lage, die 3D-Pose eines Objekts aus einem einzigen RGB- oder RGB-D-Bild zu ermitteln, ohne dass ein Zugriff auf instanzspezifische 3D-CAD-Modelle erforderlich ist. Der Kerngedanke der Analyse durch Synthese besteht darin, ein Vorwärtsmodell (z.B. eine Grafikpipeline) zu nutzen, um verschiedene Bilder zu erzeugen, die möglichen geometrischen und semantischen Zuständen der Umgebung entsprechen. Anschließend wird der Kandidat ausgewählt, der am besten mit der gemessenen visuellen Evidenz übereinstimmt.
Zunächst wird ein Pose-Aware-Bildgenerator mit Multi-View-Bildern von synthetischen Objekten aus dem ShapeNet-Datensatz trainiert, der in der Lage ist, Objektbilder zu erzeugen, die die Pose und Erscheinung des Eingabeobjekts genau wiedergeben. Zum Zeitpunkt der Inferenz mit einem segmentierten Bild als Eingabe schätzt die Methode die Objektpose durch iterative Optimierung der Objektpose und -form, um die Diskrepanz zwischen dem Eingabebild und dem synthetisierten Bild zu minimieren.
Der Schwerpunkt liegt auf starren Objekten.

Modell ohne
Video-Input RBG und RGB-D
Datensatz ohne
Genauigkeit NOCS

- 95% average translation precision
- 90% average orientation precision

Ressourcen
Laufzeit

3.5 6-PACK

6-PACK[Wan+19a] ist ein auf Bildverarbeitung basierender 6D-Posen Anker-basierter Kategorie-Level Keypoint Tracker. Dieser verfolgt einen kleinen Satz von Keypoints in RGB-D-Videos und schätzt die Objektpose durch Akkumulation der relativen Poseänderungen über die Zeit. Diese Methode erfordert kein bekanntes 3D-Modell. Stattdessen umgeht es die Notwendigkeit der Definition und Schätzung der absoluten 6D-Pose durch einen neuartigen Ankermechanismus, der der Vorschlagsmethodik für die 2D-Objekterkennung entspricht. Diese Anker bieten eine Grundlage für die Erzeugung von 3D-Keypoints. Im Gegensatz zu früheren Methoden, die manuelle Keypoint-Anmerkungen erfordern, wird ein unüberwachten Lernansatz eingesetzt, der die optimale Menge an 3D-Keypoints für die Verfolgung ermittelt. Diese Keypoints dienen als kompakte Repräsentation des Objekts, aus der der Pose-Unterschied zwischen zwei benachbarten Frames effizient geschätzt werden kann. Diese Keypoint-basierte Darstellung führt zu einer robusten und Echtzeit-6D-Positionsverfolgung. Darüber hinaus wurde 6-PACK auf einer HSR-Roboterplattform eingesetzt und gezeigt, dass die Methode Echtzeit-Tracking und Roboter-Interaktion ermöglicht.

Modell Kategorie-bezogene 3D Modellen
Video-Input RGB-D
Datensatz NOCS, ShapeNetCore
Genauigkeit NOCS

- 33, 3% 5°5cm
- 94, 2% IoU25
- 16, 0 R_err
- 3, 5 T_err

Ressourcen getestet mit NVIDIA GTX1070 GPU und Intel Core i7-6700K CPU, verfolgt Posen mit 10 Hz mit weniger als 30% des GPU-Speicherplatzes (etwa 2 GB)
Laufzeit > 10fps real-time interaction

3.6 PoseCNN

Ein neues CNN für die 6D-Objektposenschätzung. PoseCNN[Xia+17] entkoppelt die Schätzung von 3D-Rotation und 3D-Translation. Es schätzt die 3D-Verschiebung eines Objekts, indem es sein Zentrum im Bild lokalisiert und seinen Abstand zur Kamera vorhersagt. Durch Regression jedes Pixels auf einen Einheitsvektor in Richtung des Objektzentrums kann das Zentrum unabhängig vom Maßstab robust geschätzt werden. Noch wichtiger ist, dass die Pixel das Objektzentrum auch dann wählen, wenn es von anderen Objekten verdeckt wird. Die 3D-Rotation des Objekts wird durch Regression auf eine Quaternion-Darstellung-Darstellung geschätzt. Es werden zwei neue Verlustfunktionen für die Rotationsschätzung eingeführt, wobei der ShapeMatch-Verlust für symmetrische Objekte entwickelt wurde. Dadurch ist PoseCNN in der Lage, Okklusion und symmetrische Objekte in unübersichtlichen Szenen zu verarbeiten. Dies eröffnet den Weg zur Verwendung von Kameras mit einer Auflösung und einem Sichtfeld, die weit über die derzeit verwendeten Tiefenkamerasysteme hinausgehen. Manchmal führt SLOSS zu lokalen Minimums im Pose-Raum führt, ähnlich wie ICP[QK18].
Die Methode erreicht Ende-zu-Ende 6D Posenschätzung und ist sehr robust gegenüber Verdeckungen zwischen Objekten.

Modell ohne
Video-Input RGB, RGB-D
Datensatz YCB-Video, LINEMOD, LM-Occlusion
Genauigkeit Ergebnisse aus AUC Messung bei RGB

- ADD 53, 7%
- ADD-S 75, 9%

Ergebnisse aus AUC Messung bei RGB-D mit ICP

- ADD 79, 3%
- ADD-S 93, 0%

Ergebnisse aus LM-Occlusion

- PoseCNN Color 24, 9%
- PoseCNN+ICP 78, 0%

Ressourcen
Laufzeit

3.7 Robust Gaussian Filter

Modellbasierte 3D-Verfolgung von Objekten bei dichten Tiefenbildern als Eingabe mithilfe Robuster Gauss Filter[Iss+16]. Zwei Schwierigkeiten schließen die Anwendung eines Standard-Gauß-Filters auf dieses Problem aus. Tiefensensoren sind von Messrauschen gekennzeichnet und werden durch eine Robustifizierungsmethode für Gaußfilter behoben. Dadurch wird eine heuristische Ausreißer-Erkennungsmethode verwendet, die einfache Messungen ablehnt, wenn sie nicht mit dem Modell übereinstimmen. Daneben sind die Rechenkosten des Standard-Gauß-Filters aufgrund der hochdimensionalen Messung unerschwinglich. Um dieses Problem zu lösen wird eine Annäherung verwendet um die Rechenkomplexität des Filters zu reduzieren. In quantitativen Experimenten mit realen Daten wurde gezeigt, dass diese Methode besser abschneidet als der Standard-Gauß-Filter. Außerdem mit einer auf Partikelfiltern basierenden Verfolgungsmethode bei vergleichbarer Recheneffizienz eine verbesserte Genauigkeit und Glattheit der Schätzungen erzielt. Die Methode zur Verfolgung der 6-Grad Freiheitsgrades und der Geschwindigkeit eines Objekts mit Tiefenmessungen nutzt eine Standard-Tiefenkamera. Der vorgeschlagene Algorithmus läuft mit der Bildrate der Kamera von 30 Hz auf nur einem CPU-Kern.

Modell CAD-Modell
Video-Input RGB-D
Datensatz Bayesian Object Tracking
Genauigkeit

- Verschiebungsfehler Mittelwert 0, 03008[mm]
- Verschiebungsfehler Median 0, 00489[mm]
- Winkelfehler Mittelwert 0, 39644[Grad]
- Winkelfehler Median 0, 06076[Grad]

Ressourcen 640×480 Pixel auf einem CPU-Kern
Laufzeit 30Hz

4 Fazit

benötigen	?	Farbbild	Tiefenbild	3D Pointcl
3D Modell	RGF[Iss+16] dbotPF[Wüt+13] se-TrackNet[Wa20]	Contour Matching DeepIM[Li+18]		Robust Ga
Kategorie Modell	NOCS[Wan+19b] KeypointNet[Sa18]	Feature Matching	6-PACK[Wan+19a]	
ohne Modell	ICP[QK18] TEASER++[HC20]	Iterative Closest Point Analyse-durch-Synthese [Che+20] PoseCNN[Xia+17]	MaskFusion[RBA18] Analyse-durch-Synthese [Che+20] BundleTrack[WB21] PoseCNN[Xia+17]+ICP[QK18]	

Tabelle 1: Übersicht unterschiedlicher Verfahren

5 Vergleich verschiedener Verfahren

Vergleich der unterschiedlichen Methoden unterscheidbar nach Klassische, RNN-basierte, CNN-basierte, GNN-basier

Literatur

[Che+20] Xu Chen u. a. *Category Level Object Pose Estimation via Neural Analysis-by-Synthesis*. Aufgerufen 27.10.2021. 2020. arXiv: 2008.08145. URL: arxiv.org/abs/2008.08145.

[HC20] J. Shi H. Yang und L. Carlone. *TEASER: Fast and Certifiable Point Cloud Registration*. Website. 2020. arXiv: 2001.07715. URL: arxiv.org/abs/2001.07715.

[Iss+16] Jan Issac u. a. „Depth-based object tracking using a Robust Gaussian Filter“. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)* (März 2016). Aufgerufen 27.10.2021. DOI: 10.1109/icra.2016.7487184. URL: dx.doi.org/10.1109/ICRA.2016.7487184.

[Li+18] Yi Li u. a. „DeepIM: Deep Iterative Matching for 6D Pose Estimation“. In: *International Journal of Computer Vision* 128.3 (Nov. 2018). Aufgerufen 16.10.2021, S. 657–678. ISSN: 1573-1405. DOI: 10.1007/s11263-019-01250-9. URL: arxiv.org/abs/1804.00175.

[Pa17] J. Pont-Tuset und et al. *The 2017 davis challenge on video object segmentation*. Website. 2017. arXiv: 1704.00675. URL: arxiv.org/abs/1704.00675.

[QK18] J. Park Q.-Y. Zhou und V. Koltun. *Open3d: A modern library for 3d data processing*. Website. 2018. arXiv: 1801.09847. URL: arxiv.org/abs/1801.09847.

[RBA18] Martin Rünz, Maud Buffier und Lourdes Agapito. *MaskFusion: Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects*. Aufgerufen 27.10.2021. 2018. arXiv: 1804.09194. URL: arxiv.org/abs/1804.09194.

[Sa18] S. Suwajanakorn und et al. *Discovery of latent 3d keypoints via end-to-end geometric reasoning*. Website. 2018. arXiv: 1807.03146. URL: https://arxiv.org/abs/1807.03146.

[Wa20] B. Wen und et al. *se (3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains*. Website. 2020. arXiv: 2007.13866. URL: arxiv.org/abs/2007.13866.

[Wan+19a] Chen Wang u. a. *6-PACK: Category-level 6D Pose Tracker with Anchor-Based Keypoints*. Aufgerufen 27.10.2021. 2019. arXiv: 1910.10750. URL: arxiv.org/abs/1910.10750.

[Wan+19b] He Wang u. a. *Normalized Object Coordinate Space for Category-Level 6D Object Pose and Size Estimation*. Aufgerufen 27.10.2021. 2019. arXiv: 1901.02970. URL: arxiv.org/abs/1901.02970.

[WB21] Bowen Wen und Kostas Bekris. *BundleTrack: 6D Pose Tracking for Novel Objects without Instance or Category-Level 3D Models*. Website. Aufgerufen 23.10.2021. 2021. arXiv: 2108.00516. URL: arxiv.org/abs/2108.00516.

[Wüt+13] M. Wüthrich u. a. *Probabilistic object tracking using a range camera*. Website. 2013. DOI: 10.1109/iros.2013.6696810. arXiv: 1505.00241. URL: arxiv.org/abs/1505.00241.

[Xa18] N. Xu und et al. *Youtube-vos: A large-scale video object segmentation benchmark*. Website. 2018. arXiv: 1809.03327. URL: arxiv.org/abs/1809.03327.

[Xia+17] Yu Xiang u. a. *PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes*. Website. Aufgerufen 16.10.2021. 2017. arXiv: 1711.00199. URL: arxiv.org/abs/1711.00199.