Proseminar Objektposenschätzung

Robert Jeutter

12. 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[**BundleTrack**] 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[**Davis**], Youtube-VOS[**Youtube-vos**]
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[**Deepim**] 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, LINEMOD Occlusion
Genauigkeit LINEMOD -Datensatz:

- $85,2\%$ 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**]

Modell ohne Modell
Video-Input
Datensatz
Genauigkeit NOCS Datensatz

- $26,5\%$ 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
Laufzeit

3.4 Neural Analysis-by-Synthesis

[**CategoryLevelObject**]

Modell
Video-Input
Datensatz
Genauigkeit
Ressourcen
Laufzeit

3.5 6-PACK

6-PACK[**6pack**] 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 Datensatz:

- $33,3\%$ 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

[**PoseCNN**]

Modell
Video-Input
Datensatz
Genauigkeit
Ressourcen
Laufzeit

Ein neues CNN für die 6D-Objektposenschätzung. PoseCNN schätzt die 3D-Verschiebung eines Objekts, indem es sein Zentrum im Bild lokalisiert und seinen Abstand zur Kamera vorhersagt. Die 3D-Rotation des Objekts wird durch Regression auf eine Quaternion-Darstellung geschätzt. Dabei führt man eine neue Verlustfunktion ein, die es PoseCNN ermöglicht, symmetrische Objekte zu behandeln. Erreicht Ende-zu-Ende 6D Posenschätzung und ist sehr robust gegenüber Verdeckungen zwischen Objekten.
PoseCNN entkoppelt die Schätzung von 3D-Rotation und 3D-Translation. Es schätzt die 3D-Verschiebung durch Lokalisierung des Objektzentrums und Vorhersage des Zentrumsabstands. 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-Drehung wird durch Regression auf eine Quaternion-Darstellung vorhergesagt. 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. Wir stellen fest, dass SLOSS manchmal zu lokalen Minimums im Pose-Raum führt, ähnlich wie ICP. Es wäre interessant, in Zukunft einen effizienteren Umgang mit symmetrischen Objekten in der 6D-Positionsschätzung zu erforschen.

3.7 Robust Gaussian Filter[GaussianFilter]

Modell
Video-Input
Datensatz
Genauigkeit
Ressourcen
Laufzeit

4 Vergleich verschiedener Verfahren

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

GNN-basierte

	?	Farbbild	Tiefenbild	3D Point-cloud
3D Modell	RGF[**GaussianFilter**]	ICP Matching		
	dbotPF[**dbotPF**]	DeepIM[**Deepim**]		
	se-TrackNet[**se-TrackNet**]			
Kategorie Modell	NOCS[**NormalizedObjectCoordiante**]	Iterative Matching		
	KeypointNet[**KeypointNet**]		6-PACK[**6pack**]	
ohne Modell	ICP[**ICP**], TEASER++[**Teaser++**]	Iterative Point		
	MaskFusion[**MaskFusion**]		PoseCNN[**PoseCNN**] BundleTrack[**BundleTrack**]	

5 Fazit

Glossar

2D Projection Die 2D-Projektionsmetrik berechnet den durchschnittlichen Abstand der 3D-Modellpunkte die auf das Bild projiziert werden, unter Verwendung der geschätzten Pose und der Grundwahrheits-Pose. Eine geschätzte Pose ist korrekt, wenn der durchschnittliche Abstand kleiner als 5 Pixel ist.. 2

5°5cm Prozentsatz der Schätzungen mit einem Orientierungsfehler $< 5°$und einem Translationsfehler $< 5cm$ - je höher, desto besser. 2, 3

6D Pose Die 6D-Positionsmetrik berechnet den durchschnittlichen Abstand zwischen den 3D-Modellpunkten, die mit Hilfe der der geschätzten Pose und der Grundwahrheits-Pose. Für symmetrische Objekte verwenden wir den Abstand der geschlossenen Punkte für die Berechnung des durchschnittlichen Abstands. Eine geschätzte Pose ist korrekt, wenn der durchschnittliche Abstand innerhalb von 10% des 3D-Modelldurchmessers liegt.. 2

AUC Area Under Curve: Die Ergebnisse werden anhand der Genauigkeitsschwelle AUC berechnet, die von ADD gemessen wird, das einen exakten Modellabgleich durchführt, und ADD-S, das für die Bewertung symmetrischer Objekte konzipiert ist.. 2

CNN Convolutional Neural Network: Besitzt pro Convolutional Layer mehrere Filterkerne, sodass Schichten an Feature Maps entstehen, die jeweils die gleiche Eingabe bekommen, jedoch aufgrund unterschiedlicher Gewichtsmatrizen unterschiedliche Features extrahieren.. 3

IoU25 (Intersection over Union) Prozentualer Anteil der Fälle, in denen die Überschneidung von Vorhersage und 3D Bounding Box größer ist als 25% ihrer Vereinigung - je höher, desto besser besser. 2, 3

LINEMOD LINEMOD ist ein RGB+D-Datensatz, der sich zu einem De-facto-Standard-Benchmark für 6D-Positionsschätzungen entwickelt hat. Der Datensatz enthält schlecht texturierte Objekte in einer unübersichtlichen Szene. 15 texturlose Haushaltsgegenstände mit Farbe, Form und Größe. Jedem Objekt ist ein Testbild zugeordnet, das eine kommentierte Objektinstanz mit erheblicher Unordnung, aber nur leichter Verdeckung zeigt.. 2

LINEMOD Occlusion Bietet zusätzliche Ground-Truth-Annotationen für alle modellierten Objekte in einer der Testgruppen von LIMEMOD. Dies führt anspruchsvolle Testfälle mit verschiedenen Verdeckungsgraden ein. Die Trainingsbilder sind die gleichen wie die für LIMEMOD.. 2

NOCS Der Datensatz enthält 6 Objektkategorien: Flasche, Schüssel, Kamera, Dose, Laptop und Becher. Drei von diesen sind Kategorien mit symmetrischen Achsen.. 2, 3

Quaternion-Darstellung Darstellung V einer Gruppe G, die einen G-invarianten Homomorphismus $J : V \to V$ besitzt, der antilinear ist und $J^2 = -Id$ erfüllt.. 3

R_err mittlerer Orientierungsfehler in Grad - je geringer desto besser. 2, 3

ShapeNetCore ShapeNetCore ist eine Teilmenge des vollständigen ShapeNet-Datensatzes mit einzelnen sauberen 3D-Modellen und manuell verifizierten Kategorie- und Ausrichtungsannotationen. Er umfasst 55 gängige Objektkategorien mit etwa 51.300 einzigartigen 3D-Modellen. Die 12 Objektkategorien von PASCAL 3D+, einem beliebten 3D-Benchmark-Datensatz für Computer Vision, werden alle von ShapeNetCore abgedeckt.. 3

T_err mittlerer Übersetzungsfehler in Zentimetern - je niedriger, desto besser. 2, 3

YCBInEOAT Dieser Datensatz hilft, die Effektivität der 6D-Positionsverfolgung während der Robotermanipulation zu überprüfen. Er enthält 9 Videosequenzen, die von einer statischen RGB-D-Kamera aufgenommen wurden, während die Objekte dynamisch manipuliert werden. Die Videos beinhalten 5 YCB Objekte: Glas, Dose, Zuckerbox, Bleichereiniger und Keksbox.. 2