Rheinische
Friedrich-Wilhelms-
Universität Bonn

Institut für Informatik
Abteilung VI
Autonome Intelligente Systeme

Rheinische
Friedrich-Wilhelms-Universität Bonn

Bachelorarbeit

Objektposenschätzung mit Keypoints und Part Affinity Fields

Autor:
Moritz Zappel

Erstgutachter:
Prof. Dr. Sven Behnke

Zweitgutachter:
Prof. Dr. Jürgen Gall

Betreuer:
Simon Bultmann, M. Sc.

Datum: 09. März 2021

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Bachelorarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen verwendet habe. Die Stellen der Arbeit sowie evtl. beigefügte Abbildungen, Zeichnungen oder Grafiken, die anderen Werken dem Wortlaut oder Sinn nach entnommen wurden, habe ich unter Angabe der Quelle kenntlich gemacht.

_____ _____

Ort, Datum Unterschrift

Zusammenfassung

Für selbstständig handelnde Roboter ist die 6D-Posenschätzung aus Bildern eine entscheiden Voraussetzung. Das Ziel dieser Arbeit ist es die 6D-Pose unterschiedlicher starrer Objekte aus RGB Bildern zu bestimmen. In dieser Arbeit wird der OpenPose Ansatz (Cao, Hidalgo u. a. 2019) angepasst. Es werden erst charakteristische Punkte auf den Objekten gefunden und diese durch Part Affinity Fields (PAFs) Instanzen zugeordnet. Anschließend wird die 6D Pose mithilfe des Random Sample Consensus (RANSAC)-Algorithmus und des Perspective-n-Point (PnP)-Algorithmus berechnet. Dabei wird auf dem verbreiteten Yale-CMU-Berkeley-Video (YCB-V) Datensatz gearbeitet. Außerdem wird eine einfache Symmetriebehandlung implementiert. Diese Arbeit liefert das drittbeste Ergebnis des Benchmark for 6D Object Pose Estimation (BOP) Wettbewerbs für den YCB-V Datensatz ohne Verwendung von Tiefeninformationen. Des Weiteren schneidet diese Arbeit bei der ADD(-S) Metrik und bei der 2D Projektionsmetrik besser als PoseCNN (Xiang u. a. 2018) aber schlechter als PVNet (Peng u. a. 2019) ab. Überdies wird dargelegt, dass PAFs auch bei Datensätzen mit nur einer Objektinstanz im Bild die Ergebnisse verbessern und Modelle, die nur 1 Objekt erkennen sollen deutlich besser abschneiden als Modelle, die 2 Objekte erkennen sollen. Zudem liefern händisch gewählte charakteristische Punkte bessere Ergebnisse als automatisch gewählte. Die Wahl des RANSAC Grenzwertes ist für die Ergebnisse dieser Arbeit irrelevant. Vor allem quaderförmige Objekte werden gut erkannt, während symmetrische Objekte und Objekte ohne markante Punkte und Textur schwieriger erkannt werden.

Inhaltsverzeichnis

1 Einleitung

Die Schätzung von Objektposen ist eine zentrale Voraussetzung dafür, dass autonome Roboter mit den Objekten in ihrer Umgebung interagieren können, um auf diese Einfluss zu nehmen. Das ist notwendig, damit Roboter selbstständig handeln können. Die Umgebung wird durch Kameras erfasst und die so gewonnenen Bilder werden analysiert, um aus ihnen die 6D-Posen der gesuchten Objekte zu extrahieren. Bei der 6D Objektposenschätzung werden in der Regel entweder die 6D Posen direkt geschätzt oder erst charakteristische Merkmale auf den Objekten bestimmt und dann darauf basierend die 6D Pose geschätzt. In dieser Arbeit wird der zweistufige Ansatz verfolgt. Es werden erst charakteristische Punkte auf den Objekten gefunden. Dann werden diese mithilfe von PAFs Instanzen zugeordnet. Danach wird mittels des RANSAC-Algorithmus und des PnP-Algorithmus die 6D Pose berechnet. Dafür wird OpenPose (Cao, Hidalgo u. a. 2019) adaptiert und mit dem RANSAC-Algorithmus und PnP-Algorithmus ergänzt.

Die Autoren von OpenPose haben exemplarisch gezeigt, dass ihr Ansatz auch für die Objektposenschätzung geeignet ist. Diese Idee wird in dieser Arbeit aufgegriffen und überprüft, wobei sich auf die 6D Pose konzentriert wird, statt, wie bei OpenPose, auf die Position im Bild. Das Ziel dieser Arbeit ist es, mithilfe von OpenPose die 6D-Pose unterschiedlicher starrer Objekte aus RGB Bildern zu bestimmen. Dabei wird auf dem verbreiteten YCB-V Datensatz gearbeitet.

Die Arbeit ist wie folgt aufgebaut. Im Kapitel 2 werden Grundlagen erklärt, die für das Verständnis der Bachelorarbeit notwendig sind. Außerdem wird hier die verwendete Notation festgelegt. Dann wird eine Literaturübersicht im Kapitel 3 gegeben. Dabei werden die zwei Herangehensweisen an die 6D-Objektposenschätzung grob skizziert. An dieser Stelle werden sowohl die in dieser Arbeit verwendete Arbeit OpenPose (Cao, Hidalgo u. a. 2019) als auch alternative Ansätze vorgestellt und gegen diese Arbeit abgegrenzt. Ferner wird in diesem Kapitel der in dieser Arbeit verwendete Datensatz vorgestellt. Die genaue methodische Vorgehensweise dieser Arbeit wird im Kapitel 4 erklärt. Es wird zuerst das Netzwerk und sein Aufbau erklärt. Danach werden die Ground Truth Annotationen und die Übertragung zur 6D-Posenschätzung ausgeführt. In Kapitel 5: Evaluation werden verschiedene Metriken erklärt. Es wird zudem genauer auf den BOP Wettbewerb eingegangen, da dieser als state-of-the-art zum Vergleich mit dieser Arbeit verwendet wird. Des

Weiteren werden unterschiedliche systematische Untersuchungen gemacht, um den Einfluss verschiedener Faktoren auf die Ergebnisse zu überprüfen. Zuletzt folgt in Kapitel 6 die Schlussfolgerung dieser Arbeit.

2 Grundlagen

Zunächst werden Grundlagen erläutert, deren Verständnis in den folgenden Kapiteln vorausgesetzt wird. Dazu gehören die Definition der 6D Pose in Abschnitt 2.1, das geometrische Modell der Lochkamera in Abschnitt 2.2, der RANSAC-Algorithmus in Abschnitt 2.3 und der PnP-Algorithmus in Abschnitt 2.4.

2.1 6D-Pose

Eine 6D-Pose besteht aus einer dreidimensionalen Rotation, die als 3x3 Matrix, wie in Formel (2.1), dargestellt wird, und einer dreidimensionalen Transformation, die als 3x1 Vektor, wie in Formel (2.2), dargestellt wird.

$$\boldsymbol{R} = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} \tag{2.1}$$

$$\boldsymbol{t} = \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix} \tag{2.2}$$

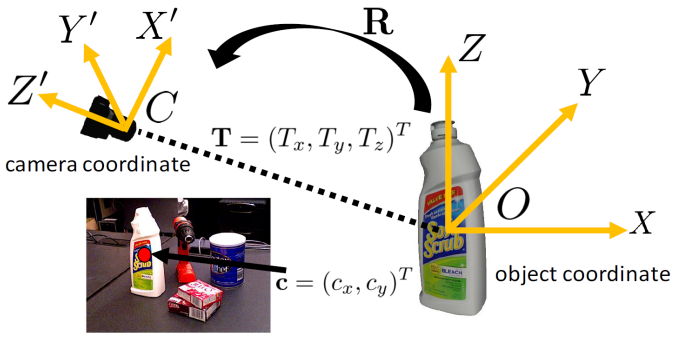


Abbildung 2.1: Transformation von Objekten aus dem Weltkoordinatensystem ins Bildkoordinatensystem (Xiang u. a. 2018)

Für eine Transformation einer 6D-Pose werden die Rotation und die Transformation, wie in Formel (2.3) zusammengefügt.

$$\boldsymbol{T} = \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{2.3}$$

Die Koordinaten, die für eine 6D-Transformation verwendet werden sind homogen und haben die Form, die in Formel (2.4) zu sehen ist, damit Rotation und Transformation in einem Schritt berechnet werden können.

$$\boldsymbol{p} = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{2.4}$$

Die Formel (2.5) beschreibt, wie eine Transformation T auf einen Punkt $\boldsymbol{p}$ angewandt wird, um einen transformierten Punkt $\boldsymbol{p}^*$ zu erhalten. Die Form von $\boldsymbol{p}$ bleibt bei $\boldsymbol{p}^*$ durch die Transformation erhalten.

$$\boldsymbol{p}^* = \boldsymbol{T} \cdot \boldsymbol{p} \tag{2.5}$$

Bei dieser Arbeit werden starre Objekte betrachtet, die durch eine Rotation und Translation vom Weltkoordinatensystem in das Kamerakoordinatensystem transformiert werden, wie in Abbildung 2.1 dargestellt. Danach werden die Objekte in die Bildebene projiziert. Dieser Schritt ist ebenfalls in der Abbildung 2.1, am Beispiel des Objektzentrums, zu sehen.

2.2 Lochkamera

Eine Lochkamera ist ein einfaches Kameramodell. Das Licht fällt durch ein Loch auf ein Objekt in einem dunklen Raum. Von diesem Objekt wird es zurückgeworfen und trifft auf eine Projektionsfläche, an der dann ein Abbild des Objekts zu sehen ist. Dieser Aufbau ist im Bild 2.2 zu erkennen. Es findet keine weitere Bündelung des Lichts durch Objektive statt.

Die Form dreidimensionaler Punkte ist in Gleichung (2.4) beschrieben. Analog

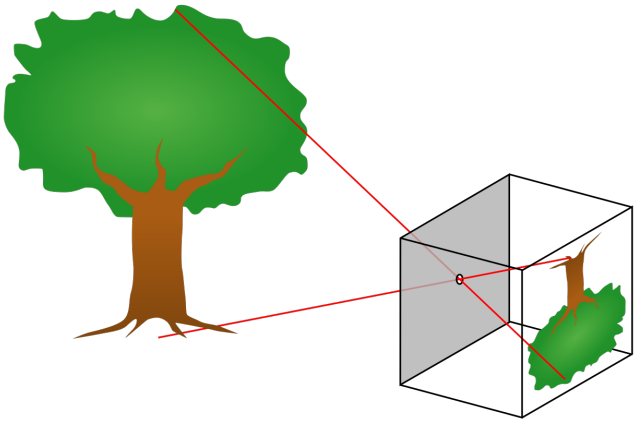


Abbildung 2.2: Darstellung einer Lochkamera

haben zweidimensionale Punkte in der Bildebene die Form, wie sie in Gleichung (2.6) dargestellt ist. Bei beiden Punkten erhält man die Koordinaten, indem alle Koordinaten durch die letzte geteilt werden.

$$\boldsymbol{u} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} \tag{2.6}$$

Ein Punkt $\boldsymbol{p}$ im Weltkoordinatensystem wird durch die Gleichung (2.7) als Punkt $\boldsymbol{u}$ in die Bildebene projiziert (Zhang 2000). Der Ursprung der Bildebene liegt in der oberen linken Ecke. Die x-Achse wird auf die u-Achse, die von links nach rechts verläuft, abgebildet und die y-Achse wird auf die v-Achse, die von oben nach unten verläuft, abgebildet.

$$s\boldsymbol{u} = \boldsymbol{A}[\boldsymbol{R}\ \boldsymbol{t}]\boldsymbol{p} \tag{2.7}$$

Dabei ist s ein beliebiger Faktor zum Skalieren. $[\boldsymbol{R}\ \boldsymbol{t}]$ ist die Matrix, die die extrinsischen Parameter enthält. Das sind die Rotation $\boldsymbol{R}$ und Translation $\boldsymbol{t}$, die das Weltkoordinatensystem in das Kamerakoordinatensystem transformieren. $\boldsymbol{R}$ und $\boldsymbol{t}$ sind wie in Gleichung (2.3) angeordnet. $\boldsymbol{A}$ ist die Matrix, die die intrinsischen Kameraparameter enthält. $\boldsymbol{A}$ beinhaltet die Projektion vom Kamerakoordinatensystem in die Bildebene. Der Aufbau von $\boldsymbol{A}$ ist in Gleichung (2.8) genauer beschrieben.

$$\boldsymbol{A} = \begin{pmatrix} f_u & \gamma & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.8}$$

(u_0, v_0) sind die Koordinaten des optischen Zentrums, dem Punkt, der genau senkrecht hinter dem Loch der Lochkamera liegt. Das optische Zentrum liegt in der Regel in der Mitte des Bildes. Die Brennweiten f_u und f_v sind die Skalierungsfaktoren für die Achsen u und v in der Bildebene. Dabei transformieren f_u und f_v das Objekt aus dem Maßstab des Kamerakoordinatensystems in den entsprechenden Maßstab in Pixeln. γ ist der Parameter, der die Schräge des Achsen u und v beschreibt. γ kann wie in Gleichung (2.9) berechnet werden.

$$\gamma = f_u \cdot \tan(\alpha) \tag{2.9}$$

Dabei ist α der Winkel zwischen einer Achse, die rechtwinklig zur v-Achse steht, und der u-Achse. In dem in dieser Arbeit verwendeten Datensatz gilt immer $\gamma = 0$.

2.3 Random Sample Consensus (RANSAC)-Algorithmus

Der RANSAC-Algorithmus (Fischler und Bolles 1981) teilt eine Datenmenge in Ausreißer und valide Daten ein. Die Vorgehensweise wird anhand des Beispiels der 6D-Posenschätzung erklärt. Bei der 6D-Posenschätzung wird von dem Modell der 6D-Pose ausgegangen, das auch in Abbildung 2.1 veranschaulicht wird.

Der RANSAC-Algorithmus funktioniert folgendermaßen.

1. Eine zufällige Teilmenge der erkannten charakteristischen Punkte wird ausgewählt.

2. Eine Schätzung für die Pose wird basierend auf dieser Teilmenge erstellt.

3. Alle Punkte, die in diese geschätzte Pose passen werden der Teilmenge hinzugefügt.

4. Falls die Teilmenge genug Punkte enthält geschieht folgendes.

    a) Die Schätzung wird mithilfe der neuen Teilmenge verbessert.

b) Falls die Teilmenge bisher die meisten Punkte enthält, wird sie gespeichert.

Eine feste Anzahl von Wiederholungen werden von diesem Prozess durchgeführt. Die Teilmenge, die nachher die meisten Punkte enthält wird als valide Daten ausgegeben. Alle Punkte, die nicht zu dieser Teilmenge gehören werden als Ausreißer klassifiziert.

2.4 Perspective-n-Point (PnP)-Algorithmus

Mithilfe des PnP-Algorithmus (Lepetit, Moreno-Noguer und Fua 2008) wird für einenge von gegebenen zueinander gehörigen 2D und 3D Punkten eines Objekts die 6D Pose des Objekts berechnet. Bei dem PnP-Algorithmus wird auch von dem in Abbildung 2.1 veranschaulichten Modell der 6D-Pose ausgegangen. Dementsprechend wird versucht die Rotation R und die Translation $\boldsymbol{t}$ zu berechnen. Das wird erreicht, indem die n Punkte im Weltkoordinatensystem

$$\boldsymbol{p}_i, i = 1, ..., n \tag{2.10}$$

als gewichtete Summe von vier Kontrollpunkten

$$\boldsymbol{c}_j, j = 1, ..., 4 \tag{2.11}$$

dargestellt werden, wie es in der Formel (2.12) zu sehen ist. Im Folgenden sind die Punkte im Kamerakoordinatensystem mit ^{c} und die Punkte im Weltkoordinatensystem mit ^{w} markiert.

$$\boldsymbol{p}_i^w = \sum_{j=1}^{4} a_{ij} \boldsymbol{c}_j^w, \ mit \ \sum_{j=1}^{4} a_{ij} = 1 \tag{2.12}$$

Wie in Gleichung (2.13) dargestellt, gilt das gleiche für das Kamerakoordinatensystem.

$$\boldsymbol{p}_i^c = \sum_{j=1}^{4} a_{ij} \boldsymbol{c}_j^c \tag{2.13}$$

Diese vier virtuellen Kontrollpunkte könnten beliebig gewählt werden. Aber um

die Stabilität des Algorithmus zu verbessern, wird stattdessen das Zentrum der Punkte als der erste Kontrollpunkt gewählt und die drei Hauptachsen der stärksten Varianz zur Wahl der restlichen drei Punkte verwendet. Die Projektion in die Bildebene wird durch die Gleichung (2.14) beschrieben.

$$\forall i, w_i \begin{bmatrix} \boldsymbol{u}_i \\ 1 \end{bmatrix} = \boldsymbol{A}\boldsymbol{p}_i^c = \boldsymbol{A} \sum_{j=1}^{4} a_{ij} \boldsymbol{c}_j^c \tag{2.14}$$

Die Matrix $\boldsymbol{A}$ ist die interne Kameramatrix und $\{\boldsymbol{u}_i\}_{i=1,\ldots,n}$ sind die Projektionen der Punkte $\{\boldsymbol{p}_i\}_{i=1,\ldots,n}$ aus dem Kamerakoordinatensystem in die Bildebene. w_i ist ein Skalar. Wenn die Gleichung (2.14) erweitert wird, indem die Punkte $\boldsymbol{c}_j^c$ durch $[x_j^c, y_j^c, z_j^c]^T$ und die Punkte $\boldsymbol{u}_i$ durch $[u_i, v_i]^T$ ersetzt werden und die interne Kameramatrix ausgeschrieben wird, wird die Gleichung (2.15) erhalten.

$$\forall i, w_i \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} = \begin{bmatrix} f_u & 0 & u_c \\ 0 & f_v & v_c \\ 0 & 0 & 1 \end{bmatrix} \sum_{j=1}^{4} a_{ij} \begin{bmatrix} x_j^c \\ y_j^c \\ z_j^c \end{bmatrix} \tag{2.15}$$

Aus der unteren Zeile der Gleichung (2.15) folgt, die Gleichung (2.16).

$$w_i = \sum_{j=1}^{4} a_{ij} z_j^c \tag{2.16}$$

Wenn die Gleichung (2.16) in die Gleichung (2.15) eingesetzt wird, erhält man die Gleichungen (2.17) und (2.18).

$$\sum_{j=1}^{4} a_{ij} f_u x_j^c + a_{ij}(u_c - u_i) z_j^c = 0 \tag{2.17}$$

$$\sum_{j=1}^{4} a_{ij} f_v y_j^c + a_{ij}(v_c - v_i) z_j^c = 0 \tag{2.18}$$

In den Gleichungen (2.17) und (2.18) kommt w_i nicht mehr vor. Dadurch sind die unbekannten Werte der Gleichung nur noch die 12 Koordinaten der Kontrollpunkte. Die Gleichung (2.15) kann in die Gleichung (2.19) umgeschrieben werden.

$$\boldsymbol{M}\boldsymbol{x} = 0 \tag{2.19}$$

Dabei ist $\boldsymbol{M}$ die Konkatenation der Gleichungen (2.17) und (2.18) für alle Punkte und für $\boldsymbol{x}$ gilt $\boldsymbol{x} = [\boldsymbol{c}_1^{cT}, \boldsymbol{c}_2^{cT}, \boldsymbol{c}_3^{cT}, \boldsymbol{c}_4^{cT}]^T$. Dabei sind $\boldsymbol{c}_j^c$ die virtuellen Kontrollpunkte. Dementsprechend gehört $\boldsymbol{x}$ zum Kern von $\boldsymbol{M}$ und kann wie in Gleichung (2.20) beschrieben werden.

$$\boldsymbol{x} = \sum_{i=1}^{N} \beta_i \boldsymbol{q}_i \tag{2.20}$$

Die $\boldsymbol{q}_i$ sind die Rechts-Singulärvektoren der Matrix $\boldsymbol{M}$, die zu den N Singulärwerten gehören, die 0 sind. Diese lassen sich als die Eigenvektoren der Matrix $\boldsymbol{M}^T\boldsymbol{M}$ mit den Eigenwerten null bestimmen. Dann müssen nur noch die Koeffizienten $\{\beta_i\}_{i=1,\dots,N}$ bestimmt werden. Es ist auch möglich, dass das System unterbestimmt ist, wenn weniger als 6 Punkte gegeben sind, da jeder Punkt nur die 2 Gleichungen (2.17) und (2.18) für das Gleichungssystem erzeugt, dieses aber 12 Unbekannte hat. Der Kern der Matrix $\boldsymbol{M}^T\boldsymbol{M}$ hat eine Dimension von mindestens 1, da die Kontrollpunkte zusammen beliebig skaliert werden können. Außerdem hat der Kern maximal die Dimension 4, da die Translation der Kontrollpunkte in Richtung des Bildes die Position im Bild nicht verändert.

Aufgrund von Rauschen werden die Eigenwerte in der Regel, aber nicht genau 0, sondern nur klein. Folglich sind die in Frage kommenden Eigenwerte die 4 kleinsten Eigenwerte. Um zu entscheiden, wie viele dieser Eigenvektoren für die Gleichung (2.20) verwendet werden, werden die Kontrollpunkte für alle Möglichkeiten berechnet und die Auswahl von Kontrollpunkten gewählt, die die Distanz zwischen den ins Bild projizierten und den im Bild gefundenen Punkten minimiert. Zuletzt werden R und $\boldsymbol{t}$ aus den Kontrollpunkten berechnet.

3 Literaturübersicht

Bei der Objektposenschätzung in Bildern gibt es verschiedene Ansätze. Einerseits kann die 6D-Pose direkt aus dem Bild geschätzt werden, wie in Abschnitt 3.1 beschrieben. Andererseits können erst Merkmale der Objekte im Bild bestimmt werden und daraus die 6D-Pose berechnet werden. Darunter fallen auch die keypointbasierten Verfahren, auf die hier in Abschnitt 3.2 genauer eingegangen wird. In Abschnitt 3.3 wird auf OpenPose (Cao, Hidalgo u. a. 2019), auf dem diese Arbeit basiert, eingegangen. Außerdem wird er in dieser Arbeit verwendete Datensatz in Abschnitt 3.4 vorgestellt.

3.1 Objektposen direkt bestimmen

Bei der direkten Bestimmung von 6D-Posen wird die Pose als Rotation und Translation direkt vom Netzwerk ausgegeben. Ein Beispiel für diesen Ansatz ist PoseCNN (Xiang u. a. 2018). PoseCNN bestimmt erst das Zentrum des Objekts. Dafür wird das Bild pixelweise nach Objekten segmentiert und jeder Pixel stimmt mit einem Vektor, für eine Richtung zum Zentrum seines Objektes, ab. Dann wird die Entfernung zwischen dem Zentrum und der Kamera bestimmt und zusammen mit der Position des Zentrums die Translation berechnet. Um die Rotation zu bestimmen wird ein auf das Objekt begrenzter Teil des Bildes von einem konvolutionalem Netzwerk verarbeitet. Dabei wird die Rotation als Quaternion dargestellt (Xiang u. a. 2018). Der Nachteil dieses Verfahren ist, dass es nur für starre Objekte funktioniert.

3.2 keypointbasierte Verfahren

Die keypointbasierten Verfahren bestimmen die 6D-Objektpose, indem sie erst auf dem Bild charakteristische Punkte für alle Objekte bestimmen. Die Lage dieser Punkte wird im Vorhinein auf den Objekten allgemein definiert und dann ins jeweilige Bild projiziert. Danach wird mithilfe der charakteristischen Punkte und des PnP-Algorithmus die 6D-Pose berechnet. Diese Vorgehensweise ist aufgrund des PnP-Algorithmus auch nur für starre Objekte möglich. Die Bestimmung der

charakteristischen Punkte hingegen kann genauso für verformbare Objekte angewandt werden. Die unterschiedlichen keypointbasierten Ansätze unterscheiden sich vor allem darin, wie die charakteristischen Punkte gewählt werden, wie viele es pro Objekt gibt und wie sie bestimmt werden.

In „BB8“ (Rad und Lepetit 2018) wurde erst für jedes Objekt eine Segmentierungsmaske und dann die charakteristischen Punkte bestimmt. Die charakteristischen Punkte sind die 8 Ecken der Bounding Box eines Objektes, wodurch sie häufig außerhalb des Objektes liegen. Die Koordinaten der charakteristischen Punkte wurden hier vom Netzwerk direkt bestimmt.

Bei Pavlakos u. a. 2017 sind die charakteristischen Punkte auf der Objektoberfläche und wurden als Heatmaps von dem Netzwerk bestimmt. Die genaue Position eines charakteristischen Punktes ist das Maximum seiner zugehörigen Heatmap.

Bei „PVNet“ (Peng u. a. 2019) sind die charakteristischen Punkte auch auf der Objektoberfläche. Allerdings wurden die charakteristischen Punkte nicht direkt bestimmt. Stattdessen wurde für jeden Pixel bestimmt, ob und zu welchem Objekt dieser gehört. Jeder Pixel eines Objekts hat für jeden charakteristischen Punkt dieses Objekts einen 2D-Vektor, der zu dem charakteristischen Punkt zeigt. Der Punkt, in dem sich die meisten Vektoren des Objektes für diesen charakteristischen Punkt schneiden, wurde ausgegeben.

3.2.1 Verknüpfung von charakteristischen Punkten

Wenn Objekte nach Klassen bestimmt werden sollen und dementsprechend mehrere Objekte der selben Klasse im selben Bild auftauchen können, müssen die charakteristischen Punkte einzelnen Instanzen zugeordnet werden. Das kann entweder durch einen top-down-Ansatz oder einen bottom-up-Ansatz erzielt werden.

top-down-Ansatz

Beim top-down-Ansatz werden zuerst einzelne Objekte im Bild bestimmt und dann bei diesen die charakteristischen Punkte geschätzt. Ein häufiger top-down-Ansatz ist der Anchor Box Ansatz (Cao, Hidalgo u. a. 2019). Beim Anchor Box Ansatz werden über das ganze Bild Rechtecke verteilt und das Netzwerk muss für jede Anchor Box einschätzen, ob sich darin ein Objekt der gesuchten Klasse befindet. Für alle Boxen, die ein Objekt enthalten, wird dann die Pose durch ein Netzwerk erkannt, das für die Posenschätzung einzelner Objekte trainiert ist. Das Problem der Posenschätzung mehrerer Objektinstanzen im Bild wird dadurch auf die Posenschätzung einzelner Objektinstanzen reduziert.

Der Nachteil dieses top-down-Ansatzes ist, dass er stark davon abhängig ist,

dass die richtigen Anchor Boxes erkannt werden. Dieses Problem ist als early commitment bekannt. Außerdem steigt die Laufzeit linear mit der Anzahl der Objekte im Bild (Cao, Hidalgo u. a. 2019). Allerdings liefern top-down-Ansätze häufig bessere Ergebnisse als bottom-up-Ansätze (Cao, Hidalgo u. a. 2019).

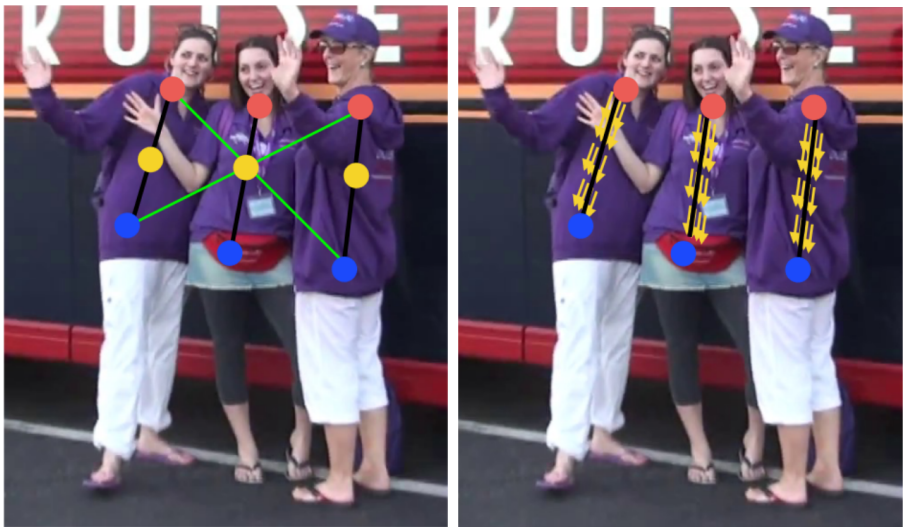
bottom-up-Ansatz

Der bottom-up-Ansatz bestimmt erst die charakteristischen Punkte und verknüpft diese dann zu Instanzen. Ein Beispiel für einen bottom-up-Ansatz ist die midpoint-representation. Bei dieser werden alle Zwischenpunkte möglicher Verbindungen darauf geprüft, ob sie die Verbindungen plausibel erscheinen lassen. Dieser Ansatz ist allerdings anfällig für Fehlzuordnungen bei sich nahe beieinander befindliche Objekten. Das ist in der Abbildung 3.1a zu erkennen. Die roten und blauen Punkte sind charakteristische Punkte, die miteinander verknüpft werden sollen. Die gelben Punkte sind Mittelpunkte, die die zugehörige Verknüpfung möglich erscheinen lassen. Die schwarzen Verbindungen sind die korrekten Verknüpfungen. Die grünen Verknüpfungen sind fehlerhafte Verknüpfungen, die durch den mittleren gelben Punkt plausibel erscheinen.

Bei Abbildung 3.1b sieht man, dass die PAFs die korrekten blauen und roten charakteristischen Punkte miteinander verbinden und so die Problematik nah beieinander liegender Instanzen lösen. Die genaue Funktionsweise der PAFs wird im Abschnitt 3.3 erklärt. Ältere bottom-up-Ansätze sind bei der Berechnung der Posen nicht schneller als top-down-Ansätze, da sie zwar schnell die Merkmale extrahieren, am Ende aber bei der finalen Zuordnung für die globale Inferenz viel Zeit benötigen (Cao, Hidalgo u. a. 2019).

3.2.2 Erweiterung auf 6D-Posen

Bei vielen Ansätzen wird nach der Bestimmung der charakteristischen Punkte der PnP-Algorithmus (Lepetit, Moreno-Noguer und Fua 2008) angewandt, um aus den charakteristischen Punkten, die 6D-Pose zu berechnen. Um Ausreißer herauszufiltern, wird der PnP-Algorithmus für die Objektposenschätzung teilweise mit dem RANSAC-Algorithmus (Fischler und Bolles 1981) kombiniert. Dazu wird der PnP-Algorithmus verwendet, um die Pose für den RANSAC-Algorithmus zu berechnen. Der RANSAC-Algorithmus bestimmt danach aus den gefundenen charakteristischen Punkten die Punkte, die für die finale Berechnung der Pose verwendet werden. Diese finale Berechnung wird ebenfalls vom PnP-Algorithmus durchgeführt. Die genauen Funktionsweisen des PnP-Algorithmus und RANSAC-Algorithmus sind in den Abschnitten 2.4 und 2.3 erklärt.



(a) mehrdeutige und fehlerhafte Zuordnung bei der midpoint-representation (Cao, Hidalgo u. a. 2019)

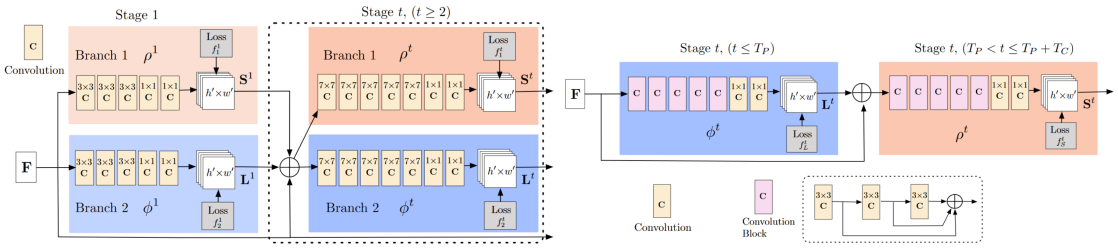
(b) eindeutige korrekte Zuordnung PAFs (Cao, Hidalgo u. a. 2019)

Abbildung 3.1: Zuordnung von charakteristischen Punkten zu Instanzen bei nahe beieinander liegenden Instanzen

3.3 OpenPose

OpenPose (Cao, Hidalgo u. a. 2019) ist ein keypointbasierter bottom-up-Ansatz zur Bestimmung von charakteristischen Punkten für eine beliebige Anzahl von Menschen im Bild. Diese charakteristischen Punkte geben die Lage und Pose der Menschen im Bild an. Dabei verwendet OpenPose für die Zuordnung der charakteristischen Punkten zu einzelnen Personen PAFs. PAFs sind 2D-Vektorfelder in der Größe des Bildes für jede Verknüpfung der charakteristischen Punkte. Dabei soll idealerweise in einem PAF nur dann zwischen den charakteristischen Punkten der entsprechenden Verknüpfung eine breitere Linie von Vektoren sein, wenn diese beiden Punkte zur selben Person gehören. Die Vektoren auf dieser Linie zeigen vom Anfangspunkt der Verknüpfung zum Endpunkt.

Der Vorteil der PAFs ist, dass sie sowohl Orts- als auch Richtungsinformationen kodieren. OpenPose vermeidet sowohl early commitment als auch die lineare Abhängigkeit der Laufzeit von der Anzahl der Personen im Bild durch die Nutzung von PAFs (Cao, Hidalgo u. a. 2019). Außerdem kann OpenPose Posen in Echtzeit schätzen und ist dabei deutlich zuverlässiger als die bisherigen bottom-up-Ansätze, insbesondere als der Ansatz der midpoint-respresentation. Allerdings sind die Ergebnisse der besten top-down-Ansätze besser als die von OpenPose, diese werden aber nicht in Echtzeit berechnet (Cao, Hidalgo u. a. 2019).



(a) Vorgängerversion von OpenPose (b) OpenPose

Abbildung 3.2: Netzwerkarchitekturen der unterschiedlichen Versionen von OpenPose

Es gibt zwei Versionen von OpenPose, die auf unterschiedliche Veröffentlichungen zurückgehen. Bei der ersten Veröffentlichung handelt es sich um Cao, Simon u. a. 2017 und bei der zweiten um Cao, Hidalgo u. a. 2019. Hierbei sprach nur Cao, Hidalgo u. a. 2019 offiziell von OpenPose. Allerdings baut diese klar auf der Ersten auf. Die Netzwerkarchitektur der ersten Veröffentlichung ist in Abbildung 3.2a dargestellt. In Abbildung 3.2b ist die Netzwerkarchitektur von OpenPose dargestellt.

Beide Versionen nutzen zur Vorverarbeitung des Bildes die ersten 10 Schichten von VGG-19. Beide Netzwerke sind zwar in Stufen aufgebaut, unterscheiden sich aber in deren Aufbau. Bei beiden wird der Loss nach jeder Stufe bestimmt. Beide Netzwerke nutzen als Eingabe die Ausgabe der Vorverarbeitung, in Abbildung 3.2 als $\boldsymbol{F}$ bezeichnet und die Ausgabe der vorherigen Stufe. Beide Netzwerke verbessern die Schätzung für die charakteristischen Punkte und die PAFs sukzessiv. Die Hintergrundfarbe in der Abbildung 3.2 veranschaulicht die Berechnung in den entsprechenden Bereichen. Rot hinterlegte Bereiche berechnen neue Heatmaps für die charakteristischen Punkte und blau hinterlegte Bereiche berechnen neue PAFs. Bei der ersten Variante (Cao, Simon u. a. 2017) werden charakteristische Punkte und PAFs gleichzeitig berechnet. Dementsprechend greift jede Stufe auf die Ausgabe beider Berechnungen zu. Diese bestehen in der ersten Stufe aus drei 3×3 und zwei 1×1 Konvolutionen. Danach besteht jede Berechnung jeder Stufe aus fünf 7×7 und zwei 1×1 Konvolutionen.

Bei OpenPose (Cao, Hidalgo u. a. 2019) werden zuerst die PAFs komplett berechnet und danach die charakteristischen Punkte. Außerdem ist die erste Stufe nun genauso aufgebaut wie die restlichen und die 7×7 Konvolutionen werden durch Konvolutionsblöcke ersetzt. Jeder Block besteht aus drei 3×3 Konvolutionen. Dadurch bleibt das rezeptive Feld gleich, während die Anzahl der Berechnungen sinkt. Die Ausgabe der Konvolutionsblöcke ist die Konkatenation der Ausgaben der drei einzelnen Konvolutionen. OpenPose hat daher deutlich mehr Stufen und liefert bessere Ergebnisse bei einer kürzeren Berechnungszeit. (Cao, Hidalgo u. a. 2019)

Die ersten Konvolutionsblöcke, für die Stufen, die die PAFs berechnen, haben 128 Ein- und 288 Ausgabekanäle . Die Stufen, die die Heatmaps berechnen haben $128+Pa$ Eingabekanäle, wobei Pa die Anzahl der Ausgabekanäle für die PAFs ist. Alle anderen Konvolutionsblöcke haben 288 Ein- und Ausgabekanäle. Die 3×3 Konvolutionen der Konvolutionsblöcke haben 96 Ein- und Ausgabekanäle. Eine Ausnahme davon bildet die erste Konvolution, die 288 Eingabekanäle hat. Die erste 1×1 Konvolution hat 288 Ein- und 256 Ausgabekanäle. Die zweite 1×1 Konvolution hat 256 Eingabekanäle und $C \cdot 12 \cdot 2$ Ausgabekanäle für die Stufen, die PAFs berechnen und $C \cdot 8 + 1$ Ausgabekanäle für Stufen, die Heatmaps berechnen. C ist die Anzahl der Objekte, die von einem Netzwerk bestimmt werden.

Sobald die Heatmaps und PAFs bestimmt sind, müssen diese zu Objektinstanzen gruppiert werden. Das funktioniert bei OpenPose folgendermaßen. Es werden aus den Heatmaps die möglichen charakteristischen Punkte bestimmt. Ein Punkt in der Heatmap gilt als möglicher charakteristischer Punkt, wenn er den maximalen Wert in seiner 4-Nachbarschaft hat und sein Wert über einem Schwellwert liegt. In dieser Arbeit liegt dieser Schwellwert bei 0.1 . Danach werden die PAFs verwendet, um die möglichen charakteristischen Punkte zu Instanzen zu verknüpft. Dafür wird nacheinander für jedes mögliche Kandidatenpaar einer Verknüpfung geprüft, ob die zwei Kandidaten zum selben Objekt gehören könnten. Ob zwei Kandidaten zum selben Objekt gehören wird auf Basis der Gleichung (3.1) entschieden.

$$e = \int_{u=0}^{u=1} \boldsymbol{L}_c(q(u)) \cdot \frac{\boldsymbol{d}_{j_2} - \boldsymbol{d}_{j_1}}{\|\boldsymbol{d}_{j_2} - \boldsymbol{d}_{j_1}\|_2} du \tag{3.1}$$

Dabei sind $\boldsymbol{d}_{j_1}$ und $\boldsymbol{d}_{j_2}$ die beiden Kandidaten, deren Zusammengehörigkeit überprüft wird. Die Funktion $q(u)$ ist in der Gleichung (3.2) genauer ausgeführt. $\boldsymbol{L}_c(\cdot)$ ist der Vektor der PAF für die Klasse c an dem Punkt $\cdot$.

$$q(u) = (1-u)\boldsymbol{d}_{j_1} + u\boldsymbol{d}_{j_2} \tag{3.2}$$

In der Implementation wird statt des Integrals eine Summe verwendet, indem eine festgelegte Anzahl an gleichmäßig zwischen 0 und 1 verteilten Punkten für u ausgewählt wird. In dieser Arbeit werden 10 Punkte zwischen 0 und 1 überprüft. Die Zuordnung der Kandidaten wird durch die Gleichung (3.6) festgelegt. Dabei wird e_c mithilfe der Gleichungen (3.3), (3.4) und (3.5) berechnet.

$$\max_{Z_c} e_C = \max_{Z_c} \sum_{m \in D_{j_1}} \sum_{n \in D_{j_2}} e_{mn} \cdot z^{mn}_{j_1 j_2} \tag{3.3}$$

$$s.t. \quad \forall m \in D_{j_1}, \sum_{n \in D_{j_2}} z^{mn}_{j_1 j_2} \leq 1 \tag{3.4}$$

$$\forall n \in D_{j_2}, \sum_{m \in D_{j_1}} z^{mn}_{j_1 j_2} \leq 1 \tag{3.5}$$

In diesen Gleichungen ist $z^{mn}_{j_1 j_2} \in \{0, 1\}$ eine Variable, die angibt, ob die Kandidaten $\boldsymbol{d}_{j_1}$ und $\boldsymbol{d}_{j_2}$ verknüpft sind und $Z = \{z^{mn}_{j_1 j_2} : für\ j_1, j_2 \in \{1...J\}, m \in \{1...N_{j_1}\}, n \in \{1...N_{j_2}\}\}$ ist die optimale Zuordnung der Kandidaten für charakteristische Punkte zu Instanzen. e_c ist das Gesamtgewicht der Zuordnung für die Verbindung c, Z_c ist die Teilmenge von Z, die die Verbindung c enthält und e_{mn} ist das Gewicht der Verbindung zwischen den Kandidaten $\boldsymbol{d}^m_{j_1}$ und $\boldsymbol{d}^n_{j_2}$ nach der Formel (3.1).

$$\max_{Z} e = \sum_{c=1}^{C} \max_{Z_c} e_c \tag{3.6}$$

Wenn zwei Punkte durch Z verknüpft werden sollen, wird mit absteigendem Gewicht der Verknüpfung überprüft, ob die beiden Punkte schon zu einer Instanz gehören. Wenn beide Punkte noch zu keiner Instanz gehören, werden sie einer neu erstellten Instanz zugeordnet. Wenn nur einer zu einer Instanz gehört, wird der zweite Punkt ebenfalls dieser Instanz zugeordnet. Wenn beide Punkte schon unterschiedlichen Instanzen zugeordnet sind, wird geprüft, ob es in den Instanzen mindestens einen charakteristischen Punkt gibt, den beide schon besitzen. In dem Fall wird die aktuelle Verknüpfung ignoriert. Falls es keine Dopplungen bei den charakteristischen Punkten gibt, werden die beiden Instanzen verschmolzen.

3.4 Datensatz

Zur Evaluation der Arbeit wird der YCB-V Datensatz (Xiang u. a. 2018) verwendet. Der Datensatz besteht aus RGB- und RGB-D-Bildern, geometrischen Modellen der Objekte, verschiedenen physikalischen Eigenschaften und Annotationen, die die exakte Pose aller Objekte im Bild angeben. Der YCB-V Datensatz ist für Benchmarks der Posenschätzung und Robotermanipulation verbreitet. Er zeichnet sich dadurch aus, dass er viele unterschiedliche, günstige, haltbare und leicht erwerbliche Objekte enthält, sodass auch in der Realität Tests mit Robotern durch-

(a) 01 (b) 02 (c) 03 (d) 04 (e) 05

(f) 06 (g) 07 (h) 08 (i) 09 (j) 10

(k) 11 (l) 12 (m) 13 (n) 14 (o) 15

(p) 16 (q) 17 (r) 18 (s) 19 (t) 20 (u) 21

Abbildung 3.3: Objekte aus dem YCB-V Datensatz

geführt werden können (Calli u. a. 2015). Die Objekte des YCB-V Datensatzes sind im Bild 3.3 dargestellt. Die Objekte sind im Bild 3.3 von links oben nach rechts unten nach ihrer Nummer sortiert. Die Nummer jedes Objektes befindet sich auch in dem jeweiligen Untertitel.

Jedes Objekt befindet sich in jedem Bild maximal einmal. Dadurch können die PAFs ihren ursprünglichen Zweck, nämlich die Zuordnung von Punkten zu Instanzen nicht mehr erfüllen. Sie können aber bei fehlerhaften Maxima das richtige Maximum detektieren. Zudem helfen sie die charakteristischen Punkte zu finden, falls diese, beispielsweise durch Verdeckung, schwierig zu finden sind. Bei dieser Arbeit werden, wie im ursprünglichen OpenPose-Ansatz (Cao, Hidalgo u. a. 2019), nur die RGB-Bilder betrachtet. Die Tiefeninformation bleibt unbeachtet.

3.4.1 Objektsymmetrie

Der YCB-V Datensatz beinhaltet 5 symmetrische Objekte. Diese sind die Schüssel, der Holzblock, die Zangen und der Ziegel und sind in der Abbildung 3.3 mit den Objektnummern 13, 16, 19, 20 und 21 zu sehen. Die Schüssel ist rotationssymmetrisch bezüglich der Z-Achse des Objektkoordinatensystems. Die Rotationssymmetrie der Schüssel ist eine kontinuierliche Symmetrie. Die Schüssel ist bei einer Rotation um einen beliebigen Winkel um die Symmetrieachse symmetrisch. Die restlichen vier Objekte sind symmetrisch bezüglich einer oder mehrerer Symmetrietransformationen. Der Holzblock ist das einzige Objekt mit mehreren Symmetrietransformationen. Die Symmetrietransformationen des Holzblocks sind Drehungen um ein Vielfaches von 90° jeweils einmal mit und ohne Spiegelung von oben nach unten. Die beiden Zangen und der Ziegel haben jeweils nur eine Symmetrietransformation, die sie von rechts nach links spiegeln.

4 Methodik

Im Folgenden wird die in dieser Arbeit entwickelte Methodik zur 6D-Posenschätzung im Detail erläutert. Es wird zuerst im Abschnitt 4.1 die Berechnung der charakteristischen Punkte und PAFs erklärt. Danach wird im Abschnitt 4.2 auf das verwendete Netzwerk eingegangen und zuletzt wird im Abschnitt 4.3 die Übertragung zur 6D-Posenschätzung ausgeführt.

4.1 Berechnung der Heatmaps und PAFs

Es wurden für alle verwendeten Datensätze Annotationen im Common Objects in Context (COCO) Format mithilfe des BOP-Toolkits (Hodan 2020) erstellt. Dafür wurden zuerst charakteristische Punkte auf allen Objekten definiert.

4.1.1 charakteristische Punkte

Inspiriert durch PVNet (Peng u. a. 2019) wurden für jedes Objekt 8 charakteristische Punkte auf der Objektoberfläche definiert. In dieser Arbeit wurden die charakteristischen Punkte auf zwei verschiedene Arten gewählt, automatisch und händisch. Die Ergebnisse werden miteinander verglichen, sie sind im Unterabschnitt 5.3.1 zu sehen.

Bei der automatischen Auswahl der Punkte wurde folgendermaßen vorgegangen. Es wurde mit einer Menge von Punkten gestartet, die nur den Ursprung des Objektkoordinatensystems enthält. Dann wurde für jeden Punkt des Meshes, also der Objektoberfläche, die minimale Distanz zu allen Punkten der Menge bestimmt. Der Punkt, der die größte minimale Distanz hat, wurde der Menge hinzugefügt. Dieser Prozess wurde wiederholt bis die Menge 8 Punkte enthält, wobei der Ursprung nicht mitgezählt wurde. Diese sind nun die charakteristischen Punkte für das Objekt. Die zweite Art der Auswahl der charakteristischen Punkte erfolgte von Hand. Die Punkte wurden so gewählt, dass sie auf möglichst markanten Punkten liegen und in ihrer Lage die Oberflächenstruktur des Objektes abbilden. Außerdem wurde versucht vier Punkte am oberen Ende und vier Punkte am unteren Ende des Objektes zu platzieren, sodass die Form der verknüpften Punkte nachher ein Quader bildet.

Danach wurden die charakteristischen Punkte mithilfe der annotierten 6D-Pose in die Bilder projiziert und in den Annotationen abgespeichert. Die Funktion charakteristische Punkte zu annotieren wurde dem BOP-Toolkit im Rahmen dieser Arbeit hinzugefügt.

Die charakteristischen Punkte werden wie bei OpenPose (Cao, Hidalgo u. a. 2019) als Heatmaps vom Netzwerk erkannt. Für jeden Punkt einer Objektklasse gibt es eine Heatmap in der Größe des Bildes. Allerdings sind die Heatmaps mit einem Faktor von 8 im Vergleich zum Bild runter skaliert. Für jeden Punkt j jeder Objektinstanz k wird eine Ground Truth Heatmap $\boldsymbol{S}^*_{j,k}$ erstellt. Die Berechnung von $\boldsymbol{S}^*_{j,k}$ für einen Pixel $\boldsymbol{u}$ ist in Gleichung (4.1) zu sehen.

$$\boldsymbol{S}^*_{j,k}(\boldsymbol{u}) = \exp\left(- \frac{\|\boldsymbol{u} - \boldsymbol{x}_{j,k}\|_2^2}{\sigma^2} \right) \tag{4.1}$$

Dabei ist $\boldsymbol{x}_{j,k}$ die Ground Truth Position des charakteristischen Punktes j der Objektinstanz k. σ bestimmt wie breit die Punkte in der Heatmap sind. Die endgültige Ground Truth Heatmap für alle Objektinstanzen wird wie in Formel (4.2) berechnet. Die Visualisierung der Heatmaps ist im Unterabschnitt 4.1.4 erklärt.

$$\boldsymbol{S}^*_j(\boldsymbol{u}) = \max_k \boldsymbol{S}^*_{j,k}(\boldsymbol{u}) \tag{4.2}$$

4.1.2 Part Affinity Fields (PAFs)

Für beide Varianten von Punkten mussten PAFs zur Verknüpfung der charakteristischen Punkte definiert werden. Die PAFs wurden so gewählt, dass sie möglichst entlang markanter Merkmale verlaufen. Dabei wurde versucht eine oben und eine unten liegende Gruppe von Punkten zu bilden, die ein geschlossenes Polygon ergeben. Zusätzlich wurden diese Polygone möglichst vertikal von oben nach unten verknüpft. Die Auswahl der PAFs wurde im OpenPose-Framework festgelegt. Die Visualisierung der PAFs ist im Unterabschnitt 4.1.4 erklärt.

4.1.3 Symmetriebehandlung

Bei symmetrischen Objekten gibt es Posen, die zwar gleich aussehen, aber unterschiedliche Ground Truth Posen zur Grundlage haben. Diese Posen sind für das Netzwerk nicht zu unterscheiden, wodurch das Netzwerk diese Objekte nicht oder nur schwierig erlernen kann. Aus diesem Grund wird die 6D-Pose symmetrischer Objekte in dieser Arbeit eingegrenzt. Das Ziel dieser Symmetriebehandlung

(a) 01 (b) 02 (c) 03 (d) 04 (e) 05

(f) 06 (g) 07 (h) 08 (i) 09 (j) 10

(k) 11 (l) 12 (m) 13 (n) 14 (o) 15

(p) 16 (q) 17 (r) 18 (s) 19 (t) 20 (u) 21

Abbildung 4.1: Objekte aus dem YCB-V Datensatz mit eingezeichneten Heatmaps der charakteristischen Punkte

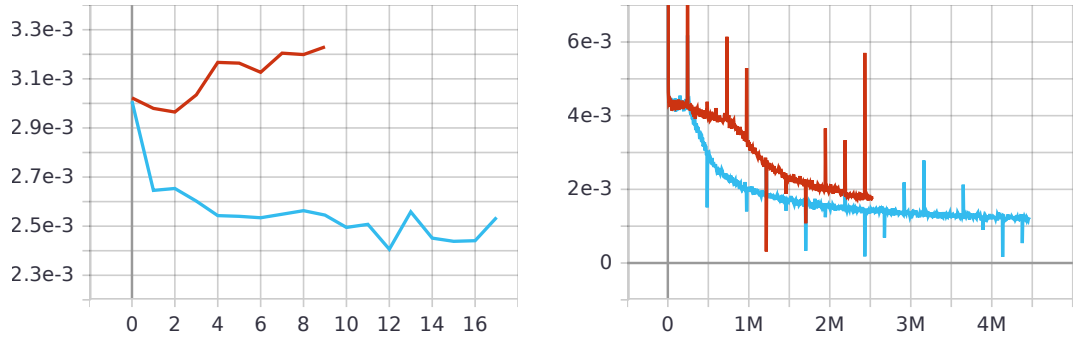
(a) 01 (b) 02 (c) 03 (d) 04 (e) 05

(f) 06 (g) 07 (h) 08 (i) 09 (j) 10

(k) 11 (l) 12 (m) 13 (n) 14 (o) 15

(p) 16 (q) 17 (r) 18 (s) 19 (t) 20 (u) 21

Abbildung 4.2: Objekte aus dem YCB-V Datensatz mit eingezeichneten PAFs



(a) Testloss (b) Trainingsloss

Abbildung 4.3: Verlauf des Trainings- und Testlosses für die Schüssel mit (blau) und ohne (rot) Symmetriebehandlung. Das Training ohne Symmetriebehandlung wurde früher beendet, weil keine Aussicht auf einen Trainingserfolg bestand. Die x-Achse ist in Epochen (Testloss) bzw. Bildern (Trainingsloss) angegeben. Eine Trainingsepoche besteht aus 243198 Bildern.

ist es unterschiedliche äquivalente Ground Truth Posen zu eliminieren. Die Posen mit behobener Symmetrie werden für das Training, aber nicht für die Evaluation verwendet.

Im Bild 4.3 ist veranschaulicht, weswegen diese Symmetriebehandlung notwendig ist. Dort ist der Traings- und Testloss für die Schüssel (Objekt 13, Abbildung 4.1m) im Verlauf des Trainings einmal mit und einmal ohne behobene Symmetrie zu sehen. In beiden Fällen fällt der Trainingsloss logistisch ab, wobei der Abfall beim Training mit Symmetriebehandlung ca. 500000 Bilder früher geschieht und deutlich steiler ausfällt. Beim Testloss ist es aber so, dass dieser ohne Symmetriebehandlung linear ansteigt und mit Symmetriebehandlung exponentiell abfällt. Folglich wird in dieser Arbeit die symmetrieäquivalenten Posen bei den Ground Truth Posen eliminiert, um das Training für symmetrische Objekte zu erleichtern.

Bei der Symmetriebehandlung wird folgendermaßen vorgegangen. Da die Schüssel (Objekt 13, Abbildung 4.1m) rotationssymmetrisch bezüglich der objektinternen z-Achse ist und bei einer Rotation um einen beliebigen Winkel gleich aussieht, wird die Rotation um diese Achse in den Ground Truth Posen auf 0 gesetzt. Die objektinterne z-Achse der Schüssel verläuft senkrecht und mittig durch den Boden der Schüssel. Für die anderen Objekte (Objekte 16, 19, 20 und 21, Abbildungen 4.1p, 4.1s, 4.1t und 4.1u) werden alle symmetrischen Posen gebildet, indem die Symmetrietransformationen auf die 6D-Pose angewandt werden. Danach muss aus den symmetrischen Posen und der ursprünglichen Pose eine ausgewählt werden, die als Ground Truth Pose für das Training des Netzwerks verwendet wird.

Diese Auswahl soll dafür sorgen, dass bei ähnlich aussehenden Posen die relative Anordnung der charakteristischen Punkte gleich ist. Das wird erreicht, indem ein

Testpunkt $\boldsymbol{r}$ und eine Menge B von ein bis drei Referenzpunkten auf den charakteristischen Punkten definiert werden. Jeder Referenzpunkt soll die Einhaltung der relativen Anordnung für eine Dimension sicherstellen. Wenn die relative Anordnung für alle Dimensionen korrekt ist gilt eine Pose als akzeptiert. Diese Bedingung ist in Gleichung (4.3) ausgeführt. Aufgrund der geometrischen Eigenschaften und der händischen Auswahl der Test- und Referenzpunkte wird manchmal keine Pose akzeptiert.

$$\bigwedge_{\boldsymbol{b}^i \in B} r_i < b_i^i \tag{4.3}$$

$\boldsymbol{r}$ ist der Testpunkt im Kamerakoordinatensystem und $\boldsymbol{b}^i$ ist der Referenzpunkt bezüglich der i-Achse im Kamerakoordinatensystem, wobei $i \in \{x, y, z\}$ gilt. Die tiefer gestellten Buchstaben geben an, welche Dimension eines Punktes ausgewählt wird. Das Kamerakoordinatensystem ist so ausgerichtet, dass die x-Achse nach rechts, die y-Achse nach unten und die z-Achse nach hinten zeigt.

Da die Zangen (Objekte 19 und 20, Abbildungen 4.1s und 4.1t) und der Ziegel (Objekt 21, Abbildung 4.1u) jeweils nur eine Symmetrietransformation haben, gibt es auch nur zwei Posen, zwischen denen entschieden werden muss. Die Test- und Referenzpunkte der Zangen liegen mittig auf der Ober- bzw. Unterseite des Gelenks. Dabei ist die Oberseite der Zange, die Seite in Richtung der positiven z-Achse des Objektkoordinatensystems. Bei den Zangen gilt $B = \{\boldsymbol{b}^y\}$. Beim Ziegel ist der Testpunkt auf der oberen, linken, vorderen Ecke und der Referenzpunkt auf der oberen, rechten, vorderen Ecke. Hier gilt $B = \{\boldsymbol{b}^x\}$. Der Holzblock (Objekt 16, Abbildung 4.1p) hat 7 Symmetrietransformationen und dementsprechend auch 8 mögliche Posen. Es gilt $B = \{\boldsymbol{b}^x, \boldsymbol{b}^y, \boldsymbol{b}^z\}$. Der Testpunkt ist die obere, linke, vordere Ecke. Die Referenzpunkte sind die 3 angrenzenden Ecken.

Wenn genau eine Pose akzeptiert wird, wird diese zur Ground Truth Pose. Wenn keine Pose akzeptiert wird, dann wird die Pose gewählt, die für Gleichung (4.4) den maximalen Wert annimmt.

$$\sum_{\boldsymbol{b}^i \in B} min(0, b_i^i - r_i) \tag{4.4}$$

Die Punkte $\boldsymbol{r}$ und $\boldsymbol{b}^i$ sind so definiert wie in Gleichung (4.3). Die Gleichung ist so geschrieben, dass ein Summand 0 ist, wenn die entsprechende Relation aus Gleichung (4.3) erfüllt ist, und dass der Summand negativ ist, wenn die Relation nicht erfüllt ist. Die Motivation hinter dieser Formel ist, dass die Pose ausgewählt

werden sollte, die am wenigsten gegen die Gleichung (4.3) verstößt. Dafür ist es aber nicht wichtig, ob für die erfüllten Bedingungen der Abstand groß ist, weswegen die Differenz durch $min(\cdot)$ nach oben durch 0 begrenzt ist. Wenn mehrere Posen akzeptiert sind, wird für die Zangen die Pose als Ground Truth Pose ausgewählt, bei der r_y den minimalen Wert hat. Für den Holzblock und den Ziegel wird die Pose ausgewählt, die für die Gleichung $r_x + r_y$ den minimalen Wert annimmt. Diese finale Auswahl ist heuristisch begründet.

4.1.4 Visualisierung

Im Bild 4.1 wurden die Heatmaps der charakteristischen Punkte auf den Objekten eingezeichnet. Dabei wurden auf diesen vorher die Skelette eingezeichnet, um die Einordnung der charakteristischen Punkte zu erleichtern. Die Skelette wurden eingezeichnet, indem man die Punkte einzeichnet und diese entlang der PAFs mit Linien verbindet. Im Bild 4.2 wurden die PAFs auf den Objekten eingezeichnet.

Es gibt insgesamt 24 Farben für die Linien der Skelette und die PAFs. Diese wurden so gewählt, dass sie sich voneinander bestmöglich unterscheiden. Diese 24 Farben wurden in zwei Gruppen geteilt. Die Farbgruppen werden von den Objektklassen abwechselnd verwendet. Die Farben der Heatmaps sind bei allen Objekten gleich. Es wurden die 8 Farben, die sich als Extrema des RGB Spektrums ergeben, für die Heatmaps ausgewählt, damit sich die unterschiedlichen charakteristischen Punkte voneinander unterscheiden. Für die Heatmaps werden die Werte der Heatmaps direkt eingezeichnet, während für die PAFs die Länge des Vektors eingezeichnet und die Richtung ignoriert wird. Die Werte werden vorher jeweils auf den Bereich $[0, 1]$ begrenzt. Die Überlagerung von Bild und Heatmaps oder Bild und PAFs wird dabei pixelweise mit der Formel (4.5) berechnet. Die Überlagerung von mehreren Heatmaps bzw. mehreren PAFs wird pixelweise mit der Formel (4.6) berechnet.

$$\boldsymbol{f}_{neu}(m) = \boldsymbol{f}_b \cdot (1 - m) + \boldsymbol{f}_h \cdot m \tag{4.5}$$

$\boldsymbol{f}_b$ ist der Farbwert des ursprünglichen Bildes und $\boldsymbol{f}_h$ ist der Farbwert der Heatmaps oder PAFs. Der maximale Wert aller Heatmaps oder PAFs an dem zu berechnenden Pixel wird mit m bezeichnet.

$$\boldsymbol{f}_h = \sum_{i=1}^{21} \boldsymbol{f}_{h_i} \cdot \frac{h_i}{\sum_{i=1}^{21} h_i} \tag{4.6}$$

Dabei ist $\boldsymbol{f}_{h_i}$ die oben erwähnte ausgewählte Farbe der i-ten Heatmap bzw. des i-ten PAF. h_i ist der Wert den die i-te Heatmap bzw. das i-te PAF am zugehörigen Pixel hat.

Wie aus den Formeln (4.5) und (4.6) zu entnehmen ist, werden erst die Heatmaps bzw. PAFs untereinander überlagert, bevor diese mit dem ursprünglichen Bild überlagert werden.

4.2 Netzwerk

Zum Training auf dem Datensatz wurde ein Framework[1], das OpenPose (Cao, Hidalgo u. a. 2019) und weitere Netzwerke zur Posenschätzung implementiert, als Basis verwendet. Die Funktionsweise von OpenPose wird im Abschnitt 3.3 bereits erklärt. Außerdem implementiert das Framework Verfahren zum Training und zur Evaluation des Netzwerks für das Vorgängernetzwerk von OpenPose. Darin inbegriffen sind die Vorverarbeitung der Eingabedaten und die Nachverarbeitung der Ausgabedaten, zum Generieren von Bildern mit Objektskeletten, für den COCO Datensatz (Lin u. a. 2015).

Es wurde die Vor- und Nachverarbeitung an die neue Datenmenge und die neuen charakteristischen Punkte und PAFs angepasst. Bei einem Eingabebild der Form $H \times W$ und C Objektklassen hat die Ausgabe für die Heatmaps die Form $\frac{H}{8} \times \frac{W}{8} \times (C \cdot 8 + 1)$ und die Ausgabe für die PAFs hat die Form $\frac{H}{8} \times \frac{W}{8} \times (C \cdot 12 \cdot 2)$. Damit das Netzwerk mehrere Objekte erkennen kann muss die in Abschnitt 3.3 genannte Anzahl der Ein- und Ausgabekanäle in jeder Stufe von OpenPose vergrößert werden. Da ein Netzwerk für alle Objekte zu groß für den Speicher der meisten Grafikkarten ist, wird in der Implementierung für jedes Objekt ein eigenes Netzwerk trainiert. Diese Modelle werden als 1-Objekt Modelle bezeichnet, da sie nur eine Objektklasse erkennen sollen. 2-Objekt Modelle sollen dementsprechend 2 verschiedene Objektklassen erkennen. Das Ausgabeformat des Netzwerkes wurde entsprechend angepasst. Um für jedes Bild alle Objekte erkennen zu können wurde die kombinierte Evaluation für mehrere Modelle implementiert. Des Weiteren wurde die Möglichkeit das Training für einzelne Objekte durchzuführen eingeführt.

Außerdem wurden das Training und die Evaluation an den YCB-V Datensatz und OpenPose angepasst. Zusätzlich wurde die Funktionalität der Posenschätzung mehrerer verschiedener Objekte in das Framework eingefügt. Darin inbegriffen sind die Funktion alle nötigen Metriken zu berechnen und die erkannten 6D Posen,

[1] *Tensorboy/Pytorch_Realtime_Multi-Person_Pose_Estimation*–URL: `https://github.com/tensorboy/pytorch_Realtime_Multi-Person_Pose_Estimation`.

sowie die Zwischenergebnisse als Bilder abzuspeichern. Zudem werden während des Trainings der durchschnittliche Trainings- und Testloss, die Speicherauslastung der GPU und CPU, die Auslastung der GPU, die durchschnittliche Zeit zum Laden von Daten und Zwischenergebnisse in einem tensorboard Server gespeichert.

4.2.1 Loss

Die Funktion für den Loss wird aus OpenPose (Cao, Hidalgo u. a. 2019) übernommen. Die Lossfunktion besteht dabei aus den Gleichungen (4.7) und (4.8). Diese werden wie in Gleichung (4.9) kombiniert.

$$\boldsymbol{l}_{\boldsymbol{L}}^{t_i} = \sum_{c=1}^{C} \sum_{p} \boldsymbol{W}(\boldsymbol{u}) \cdot \|\boldsymbol{L}_c^{t_i}(\boldsymbol{u}) - \boldsymbol{L}_c^*(\boldsymbol{u})\|_2^2 \tag{4.7}$$

$$\boldsymbol{l}_{\boldsymbol{S}}^{t_k} = \sum_{j=1}^{J} \sum_{p} \boldsymbol{W}(\boldsymbol{u}) \cdot \|\boldsymbol{S}_j^{t_k}(\boldsymbol{u}) - \boldsymbol{S}_j^*(\boldsymbol{u})\|_2^2 \tag{4.8}$$

Dabei ist $\boldsymbol{L}_c^*$ das Ground Truth PAF und $\boldsymbol{S}_j^*$ die Ground Truth Heatmap. $\boldsymbol{W}(\boldsymbol{u})$ ist eine binäre Maske, die angibt, ob ein Pixel annotiert ist oder nicht. $\boldsymbol{W}(\boldsymbol{u}) = 0$ bedeutet, dass der Pixel $\boldsymbol{u}$ nicht annotiert ist. J ist die Anzahl der Heatmaps und C ist die Anzahl der PAFs. Mit t_i bzw. t_k werden die i-te bzw. k-te Schicht des Netzwerks gekennzeichnet. Die Heatmaps, die die i-te Schicht des Netzwerks ausgibt, werden mit $\boldsymbol{L}_c^{t_i}$ bezeichnet, und die PAFs, die die k-te Schicht des Netzwerkes ausgibt, werden mit $\boldsymbol{S}_j^{t_k}$ bezeichnet.

$$\boldsymbol{l} = \sum_{t=1}^{T_P} \boldsymbol{l}_{\boldsymbol{L}}^{t} + \sum_{t=T_P+1}^{T_P+T_C} \boldsymbol{l}_{\boldsymbol{S}}^{t} \tag{4.9}$$

$\boldsymbol{l}_{\boldsymbol{L}}^{t_i}$ und $\boldsymbol{l}_{\boldsymbol{S}}^{t_k}$ sind in den Gleichungen (4.7) und (4.8) definiert. T_P ist die Anzahl der Schichten des Netzwerks, die die Ausgabe für die PAFs bestimmen, und T_C ist die Anzahl der Schichten des Netzwerks, die die Ausgabe für die Heatmaps bestimmen.

4.3 6D-Posenschätzung

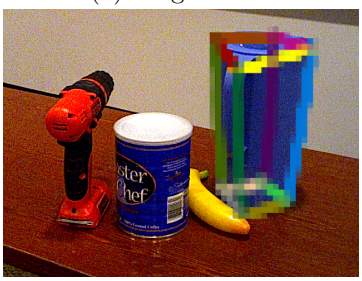
Der grundlegende Ablauf der Evaluation ist in Abbildung 4.4 dargestellt. Das Netzwerk erhält ein Bild als Eingabe. In Abbildung 4.4a ist ein solches Beispielbild zu sehen. Dann werden die Heatmaps und PAFs für das gegebene Bild durch das Netzwerk bestimmt. Die Bestimmung von Objektinstanzen aus den Heatmaps und PAFs ist wie in OpenPose (Cao, Hidalgo u. a. 2019). Die Vorgehensweise wird im Abschnitt 3.3 erklärt. Die Abbildungen 4.4b, 4.4c und 4.4d zeigen den Ablauf bis zu den Objektskeletten beispielhaft für das Objekt 11. Dieser Vorgang wird für alle Objekte im Bild durchgeführt, sodass ein Ergebnis wie Abbildung 4.4e erhalten wird. Als nächstes folgt die Berechnung der 6D-Pose. Die ins Bild projizierten 6D-Posen aller Objekte sind in Abbildung 4.4f zu sehen. Dafür werden der RANSAC-Algorithmus und der PnP-Algorithmus verwendet. Die Vorgehensweise dieser Algorithmen wird in den Abschnitten 2.3 und 2.4 erklärt. Falls eine gültige Pose gefunden wurde wird diese ausgegeben, sonst wird das gefundene Objekt ignoriert.

Bei 4 oder 5 gegebenen Punkten wird statt der Kombination aus RANSAC-Algorithmus und PnP-Algorithmus nur der PnP-Algorithmus verwendet, um nicht durch den RANSAC-Algorithmus zu viele Punkte zu entfernen, sodass der PnP-Algorithmus kein Ergebnis mehr finden kann. Bei 6 oder mehr gefundenen Punkten wird die Kombination aus RANSAC-Algorithmus und PnP-Algorithmus verwendet. Dabei gilt ein Punkt im RANSAC-Algorithmus als einer Pose zugehörig, wenn der 2D Projektionsfehler dieses Punktes kleiner als ein Grenzwert ist. Dieser Grenzwert ist als Parameter wählbar und beträgt in dieser Arbeit standardmäßig 5 Pixel.

Die Zuordnung der in einem Bild gefundenen und der annotierten Objekte erfolgt wie nachfolgend dargestellt. Alle Objektinstanzen von Objekten, die nicht im Bild annotiert sind, werden ignoriert. Wenn der Datensatz nur maximal eine Objektinstanz pro Bild enthält, wird jede gefundene Objektinstanz der ersten und einzigen annotierten Instanz zugeordnet. Nach der Berechnung der 6D-Pose wird allerdings nur die gefundene Instanz mit der geringsten Summe von ADD Metrik, ADD-S Metrik und 2D Projektionsmetrik beibehalten. Alle restlichen gefundenen Instanzen werden ignoriert. Bei Datensätzen, die mehrere Objektinstanzen im selben Bild haben können, wird jede gefundene Instanz nach der Berechnung der 6D-Pose der annotierten Objektinstanz zugeordnet, für die die ADD Metrik am kleinsten ist.

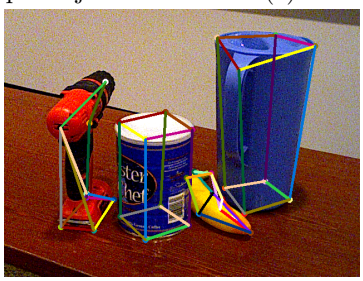
(a) Originalbild

(b) Heatmaps Obj. 11



(c) PAFs Obj. 11

(d) Objektskelett Obj. 11



(e) alle Objektskelette

(f) 6D-Posen

Abbildung 4.4: Ablauf der Bestimmung von 6D Posen in dieser Arbeit

5 Evaluation

Zuerst werden im Abschnitt 5.1 verschiedene Metriken erklärt. Im Abschnitt 5.2 wird genauer auf den BOP Wettbewerb eingegangen. Dafür werden in den Unterabschnitten 5.2.2, 5.2.3 und 5.2.4 die Metriken des BOP Wettbewerbs beschrieben, im Unterabschnitt 5.2.5 die Symmetriebehandlung und in Unterabschnitt 5.2.6 Bewertung des BOP Wettbewerbs erklärt. Außerdem wird in Unterabschnitt 5.2.7 der BOP Wettbewerb als state-of-the-art zum Vergleich mit dieser Arbeit verwendet. Die Evaluation der Arbeit wird auf dem YCB-V Datensatz durchgeführt, der auch zum Training und zur Validierung des Modells verwendet wird. Zuletzt folgen einige systematische Untersuchungen, um den Effekt ausgewählter Veränderungen auf die Ergebnisse einzuschätzen.

5.1 Metrik

Bei dieser Arbeit werden, inspiriert durch PoseCNN (Xiang u. a. 2018) und PVNet (Peng u. a. 2019), die ADD Metrik, ADD-S Metrik und die 2D Projektion Metrik zur Evaluation verwendet. Alle Metriken sind zur Evaluation von 6D-Posen. Dabei sind diese Metriken auf Basis der Meshes der Objekte definiert.

5.1.1 ADD Metrik

Für die ADD Metrik (Peng u. a. 2019) wird das Objekt, dessen Pose geschätzt wird, in den Bildraum transformiert und die geschätzte 6D-Pose besteht aus der geschätzten Rotation $\tilde{\boldsymbol{R}}$ und Translation $\tilde{\boldsymbol{t}}$ und die Ground Truth 6D-Pose besteht dementsprechend aus der Ground Truth Rotation $\boldsymbol{R}$ und Translation $\boldsymbol{t}$. Der Fehler nach ADD Metrik ist dann die durchschnittliche Distanz aller Punkte im Bildraum. Die Berechnung ist in Formel (5.1) dargestellt. Jedes Mesh hat eine Menge P von Punkten.

$$\epsilon_{ADD} = \frac{1}{|P|} \sum_{\boldsymbol{p} \in P} \| (\boldsymbol{R}\boldsymbol{p} + \boldsymbol{t}) - (\tilde{\boldsymbol{R}}\boldsymbol{p} + \tilde{\boldsymbol{t}}) \| \tag{5.1}$$

Ein Objekt und seine Pose gelten als richtig erkannt, wenn der Fehler nach ADD Metrik weniger als 10% des Durchmessers des Objekts beträgt (Peng u. a. 2019). Der Durchmesser eines Objekts ist in dieser Arbeit definiert als der maximale euklidische Abstand zweier Punkte des zugehörigen Meshes. Der Maßstab der ADD Metrik ist der gleiche Maßstab den auch die Meshes der Objekte nutzen. Im Fall des YCB-V Datensatz ist der Maßstab in mm.

5.1.2 ADD-S Metrik

Die ADD-S Metrik ist eine abgewandelte Version der ADD- Metrik für symmetrische Objekte. Hier wird nicht die Distanz zu dem zugehörigen Punkt des Meshes, sondern zu dem nächsten Punkt bestimmt. Die genaue Berechnung ist in Gleichung (5.2) zu sehen.

$$\epsilon_{ADD-S} = \frac{1}{|P|} \sum_{\boldsymbol{p}_1 \in P} \min_{\boldsymbol{p}_2 \in P} \| (\boldsymbol{R}\boldsymbol{p}_1 + \boldsymbol{t}) - (\tilde{\boldsymbol{R}}\boldsymbol{p}_2 + \tilde{\boldsymbol{t}}) \| \tag{5.2}$$

5.1.3 2D Projektion Metrik

Für die 2D Projektion Metrik (Peng u. a. 2019) wird das Objekt in den Bildraum transformiert und anschließend in die Bildebene projiziert. Danach wird für jeden Punkt des Objekts die Distanz zwischen der echten und der geschätzten Position in der Bildebene berechnet. Der Fehler nach 2D Projektion Metrik ist die durchschnittliche Distanz aller Punkte in der Bildebene. Die Berechnung ist in Formel 5.3 dargestellt. Jedes Mesh hat eine Menge P von Punkten. $proj(\cdot)$ ist die Projektion eines Punktes aus dem Bildraum in die Bildebene.

$$\epsilon_{2DProj.} = \frac{1}{|P|} \sum_{\boldsymbol{p} \in P} \| proj(\boldsymbol{R}\boldsymbol{p} + \boldsymbol{t}) - proj(\tilde{\boldsymbol{R}}\boldsymbol{p} + \tilde{\boldsymbol{t}}) \| \tag{5.3}$$

Ein Objekt gilt als korrekt erkannt, wenn der Fehler nach 2D Projektion Metrik kleiner als 5 ist (Peng u. a. 2019). Der Maßstab der 2D Projektion Metrik ist in Pixeln definiert.

5.1.4 Aufteilung

In der Evaluation werden für die gefundenen 6D-Posen die ADD Metrik, die ADD-S Metrik und die 2D Projektionsmetrik berechnet. Außerdem wird auch bestimmt,

welche Anteile der 6D-Posen nach den im Abschnitt 5.1 genannten Grenzwerten korrekt sind. Zusätzlich wird die Area under Curve für die ADD Metrik, die ADD-S Metrik und die 2D Projektionsmetrik, wie in PoseCNN (Xiang u. a. 2018), bestimmt. Die maximalen Grenzwerte für die Area under Curve sind für die ADD und ADD-S Metrik 10 cm und für die 2D Projektionsmetrik 40 Pixel. Des Weiteren wird der Anteil der Objekte, für die irgendeine 6D-Pose gefunden wurde, bestimmt. All diese Werte, mit Ausnahme der Area under Curve, werden nach dem sichtbaren Anteil des Objekts in 10% Schritten bzw. nach der Anzahl der gefundenen charakteristischen Punkte aufgeschlüsselt erhoben. Außerdem werden alle Werte für jedes Objekt einzeln und für alle Objekte gemeinsam berechnet.

5.2 Benchmark for 6D Object Pose Estimation (BOP) Wettbewerb

Ziel des BOP Wettbewerbs (Hodan u. a. 2020) ist es, den aktuellen Forschungsstand im Bereich der 6D Objektposenschätzung zu erfassen. Der erste Wettbewerb fand 2017 statt, der zweite Wettbewerb 2019 und der aktuellste 2020. Da die Art der Evaluation beim zweiten und dritten Wettbewerb gleich ist, sind im aktuellsten Wettbewerb Ergebnisse aus den Jahren 2019 und 2020 vorhanden.

Der BOP Wettbewerb beinhaltet 11 verschiedene Datensätze. Einer dieser Datensätze ist der YCB-V Datensatz. Dadurch können die state-of-the-art Ergebnisse der 6D-Objektposenschätzung auf dem YCB-V Datensatz mit den Ergebnissen in dieser Arbeit verglichen werden. Allerdings verwendet der BOP Wettbewerb andere Evaluationsmetriken als die bisher beschriebenen.

5.2.1 Aufgabenbeschreibung

Zum Trainingszeitpunkt ist für jedes Objekt mit dem Index $o \in 1, ..., k$ ein dreidimensionales Modell und echte und synthetische RGB-D Trainingsbilder mit annotierten 6D-Posen gegeben. Zum Testzeitpunkt liegen RGB-D Testbilder vor. Für jedes Bild I ist eine Liste $L = [(o_1, n_1), ..., (o_m, n_m)]$ vorgegeben. In dieser ist für jedes Objekt o_i die Anzahl der Instanzen n_i im Bild I angegeben. Das Netzwerk soll nun eine Liste $E = [E_1, ..., E_m]$ erzeugen. Dabei ist E_i eine Liste der n_i geschätzten 6D-Posen der Objektinstanzen von Objekt o_i. Jede 6D-Pose besteht aus einer 3×3 Rotationsmatrix, einem 3×1 Translationsvektor und einem Wert, der angibt, wie sicher das Netzwerk sich mit dieser 6D-Pose ist.

5.2.2 Visible Surface Discrepancy (VSD)

Die erste Metrik des BOP Wettbewerbs ist VSD. Die Berechnung dieser Metrik ist in Gleichung (5.4) beschrieben.

$$\epsilon_{VSD}(\hat{\boldsymbol{D}}, \bar{\boldsymbol{D}}, \hat{\boldsymbol{V}}, \bar{\boldsymbol{V}}, \tau) = avg_{\boldsymbol{p} \in \hat{\boldsymbol{V}} \cup \bar{\boldsymbol{V}}} \begin{cases} 0 & \text{wenn } \boldsymbol{p} \in \hat{\boldsymbol{V}} \cap \bar{\boldsymbol{V}} \wedge |\hat{\boldsymbol{D}}(\boldsymbol{p}) - \bar{\boldsymbol{D}}(\boldsymbol{p})| < \tau \\ 1 & \text{sonst} \end{cases} \tag{5.4}$$

$\hat{\boldsymbol{D}}$ und $\bar{\boldsymbol{D}}$ sind Distance Maps für das transformierte Objektmodell P. Dabei wurde das Objektmodell mit der geschätzten 6D-Pose $\hat{\boldsymbol{G}}$ bzw. mit der Ground Truth 6D-Pose $\bar{\boldsymbol{G}}$ transformiert. Eine Distance Map speichert für jedes Pixel $\boldsymbol{p}$ die Distanz vom Ursprung des Kamerakoordinatensystems zu einem dreidimensionalen Punkt $\boldsymbol{x_p}$, der auf das Pixel $\boldsymbol{p}$ projiziert wird. $\hat{\boldsymbol{V}}$ und $\bar{\boldsymbol{V}}$ sind Masken, die die sichtbaren Pixel des Objektmodells P angeben. Dabei beziehen $\hat{\boldsymbol{V}}$ und $\bar{\boldsymbol{V}}$ sich auch auf die geschätzte 6D-Pose $\hat{\boldsymbol{G}}$ und die Ground Truth 6D-Pose $\bar{\boldsymbol{G}}$. Der Parameter τ ist ein Toleranzgrenzwert.

5.2.3 Maximum Symmetry-Aware Surface Distance (MSSD)

MSSD ist die zweite Metrik des BOP Wettbewerbs. Diese Metrik ähnelt der ADD-S Metrik, verwendet aber das Maximum der Distanz statt des Durchschnitts. Die genaue Berechnung der MSSD ist in Gleichung (5.5) beschrieben.

$$\epsilon_{MSSD}(\hat{\boldsymbol{G}}, \bar{\boldsymbol{P}}, \boldsymbol{K}_P, \boldsymbol{V}_P) = \min_{\boldsymbol{K} \in \boldsymbol{K}_P} \max_{\boldsymbol{p} \in \boldsymbol{V}_P} \|\hat{\boldsymbol{G}}\boldsymbol{p} - \bar{\boldsymbol{G}}\boldsymbol{K}\boldsymbol{p}\|_2 \tag{5.5}$$

$\boldsymbol{K}_P$ ist eine Menge von Symmetrieoperationen für das Objektmodell P. Die Menge $\boldsymbol{V}_P$ enthält die Punkte des Objektmodells P. Im BOP Wettbewerb wird das Maximum statt des Durchschnitts verglichen, da dieses weniger von der Punktdichte und der Form des Objekts abhängt.

5.2.4 Maximum Symmetry-Aware Projection Distance (MSPD)

Die letzte Metrik des BOP Wettbewerbs ist die MSPD. Diese ähnelt der 2D Projektion Metrik, so wie die ADD-S Metrik der MSSD Metrik ähnelt. Die mathematische Beschreibung der MSPD ist in Gleichung (5.6) dargelegt.

$$\epsilon_{MSPD}(\hat{\boldsymbol{G}}, \bar{\boldsymbol{G}}, \boldsymbol{K}_P, \boldsymbol{V}_P) = \min_{\boldsymbol{K} \in \boldsymbol{K}_P} \max_{\boldsymbol{p} \in \boldsymbol{V}_P} \| proj(\hat{\boldsymbol{G}}\boldsymbol{p}) - proj(\bar{\boldsymbol{G}}\boldsymbol{K}\boldsymbol{p}) \|_2 \tag{5.6}$$

Die Funktion $proj(\cdot)$ ist die Projektion von dreidimensionalen Punkten aus dem Kamerakoordinatensystem in die Bildebene. Die restlichen Symbole haben die gleiche Bedeutung wie in der MSSD Metrik. Bei der MSPD wird aus dem gleichen Grund, wie bei der MSSD, das Maximum statt des Durchschnitts gewählt.

5.2.5 Symmetrieoperationen

Die Symmetrieoperationen für MSSD und MSPD müssen zwei Bedingungen erfüllen. Die erste ist in Gleichung (5.7) definiert und stellt sicher, dass die Symmetrie für die Objektform gilt. Außerdem muss die Symmetrie sich nicht durch die Objekttextur klären lassen. Diese zweite Bedingung wurde subjektiv durch die Veranstalter des BOP Wettbewerbs entschieden.

$$\boldsymbol{K}_P = \{\boldsymbol{K} : h(\boldsymbol{V}_P, \boldsymbol{K}\boldsymbol{V}_P) < \hat{\epsilon}\} \tag{5.7}$$

In der Gleichung 5.7 ist $h(\cdot)$ die Hausdorff Metrik. Außerdem gilt $\hat{\epsilon} = \max(15mm, 0.1d)$. Dabei ist d der Durchmesser des Objektmodells P. Der Toleranzwert $\hat{\epsilon}$ sorgt dafür, dass nahezu symmetrische Posen trotzdem als symmetrisch angesehen werden, da die sehr geringen Differenzen zu klein sind, um sinnvoll unterschieden werden zu können.

5.2.6 Ergebnisbewertung

Eine vom Netzwerk geschätzte 6D-Pose wird als korrekt bezüglich einer Fehlerfunktion ϵ gewertet, wenn $\epsilon < \theta_\epsilon$ für $\epsilon \in \{\epsilon_{VSD}, \epsilon_{MSSD}, \epsilon_{MSPD}\}$gilt. θ_ϵ ist der Grenzwert für korrekte 6D-Posen. Als Recall wird der Anteil der korrekt geschätzten Posen von allen annotierten Posen bezeichnet. Der durchschnittliche Recall bezüglich der Fehlerfunktion ϵ wird als AR_ϵ bezeichnet und als Durchschnitt des Recalls für verschiedene Kombinationen von θ_ϵ und τ berechnet. Für AR_{VSD} nimmt τ die Werte 5% bis 50% des Objektdurchmessers mit einer Schrittweite von 5% an. θ_{VSD} wird zwischen 0.05 und 0.5 mit einer Schrittweite von 0.05 gewählt. θ_{MSSD} reicht von 5% bis 50% des Objektdurchmessers mit einer Schrittweite von 5%. θ_{MSPD} nimmt Werte zwischen $5r$ und $50r$ mit einer Schrittweite von $5r$ an. Dabei gilt $r = \frac{w}{640}$ und w ist die Breite des Bildes in Pixeln.

Die Genauigkeit einer Methode auf Datensatz D wird wie in Gleichung (5.8) berechnet. Für diese Arbeit ist AR_{YCB-V} interessant. Für den BOP Wettbewerb hingegen ist AR_{Core} entscheidend. Dabei ist AR_{Core} der Durchschnitt aller AR_D für eine Methode.

$$AR_D = \frac{AR_{VSD} + AR_{MSSD} + AR_{MSPD}}{3} \tag{5.8}$$

5.2.7 Vergleich mit dem BOP Wettbewerb

In der Tabelle 5.1 sind die Ergebnisse dieser Arbeit nach den Metriken des BOP Wettbewerbs zu sehen. Um Vergleichbarkeit zu schaffen, sind in der Tabelle auch die 5 besten Kandidaten, die nur RGB Daten verwenden, aufgeführt. Diese Arbeit belegte unter diesen Kandidaten den dritten Platz. Außerdem verwendete dieses Netzwerk, wie drei der Kandidaten, sowohl synthetische als auch reale Trainingsbilder.

CosyPose (Labbé u. a. 2020) schneidet mit Abstand am besten im BOP Wettbewerb ab. Die Vorgehensweise von CosyPose in diesem Wettbewerb ist wie folgt. Erst wird zunächst eine grobe 6D Posenschätzung erzeugt. Das kann mit beliebigen Netzwerken zur Objektposenschätzung erreicht werden, wobei CosyPose ein eigenes Netzwerk dafür implementiert. Danach wird eine 3D Szene aus den geschätzten Objekten gerendert. Ein weiteres Netzwerk bekommt das gerenderte Bild und das Eingabebild als Eingabe und versucht die 6D-Posen der Objekte zu verbessern. CosyPose nutzt eine kontinuierliche Darstellung der Rotation und entkoppelt die x-Translation und y-Translation, die im Bild ist, von der z-Translation, die die Tiefe bestimmt. Diese Darstellung erleichtert das Lernen der 6D-Pose.

5.3 Ergebnisse auf dem Yale-CMU-Berkeley-Video (YCB-V) Datensatz

In der Tabelle 5.2 sind die Ergebnisse für die Area under Curve bezüglich der ADD Metrik, der ADD-S Metrik und der 2D Projektion Metrik aufgeführt. Zusätzlich sind in der Tabelle die Ergebnisse von PoseCNN (Xiang u. a. 2018) für die Area under Curve der ADD Metrik und ADD-S Metrik aufgeführt, um einen Vergleich zu ermöglichen. Die Zeichen der symmetrischen Objekte sind, in Anlehnung an PoseCNN (Xiang u. a. 2018), in rot geschrieben, um diese besser erkennbar zu machen.

Diese Arbeit lieferte bei den Objekten 2, 3, 7, 8, 11 und 15 deutlich bessere

Tabelle 5.1: Ergebnisse der besten 5 Kandidaten bei dem BOP Wettbewerb 2020, die nur RGB Daten verwendet haben im Vergleich mit dieser Arbeit

Platz	Name	AR	AR_{VSD}	AR_{MSSD}	AR_{MSPD}	Trainingstyp
1	CosyPose-ECCV20-SYNT+REAL	0.821	0.772	0.842	0.850	pbr+real
2	EPOS-CVPR20	0.696	0.626	0.677	0.783	pbr
3	**diese Arbeit**	0.575	0.506	0.567	0.654	pbr+syn+real
4	CosyPose-ECCV20-PBR	0.574	0.516	0.554	0.653	pbr
5	leaping from 2D to 6D	0.543	0.443	0.499	0.687	pbr+real
6	CDPNv2_BOP20-RGB	0.532	0.396	0.570	0.631	pbr+real

Ergebnisse als PoseCNN. Objekt 12 lieferte für die ADD Metrik bei dieser Arbeit bessere Ergebnisse, allerdings für die ADD-S Metrik bessere Ergebnisse bei PoseCNN. Insgesamt lieferte diese Arbeit bessere Ergebnisse als PoseCNN für die ADD Metrik, aber schlechtere Ergebnisse für die ADD-S Metrik. Umgekehrt ist zu erkennen, dass PoseCNN bei den symmetrischen Objekten bessere Ergebnisse lieferte. Eine Ausnahme davon ist das Ergebnis der ADD Metrik für Objekt 13. Außerdem wurden die Objekte 1, 9, 10, 14 und 18 von PoseCNN besser erkannt. Besonders stark waren die Unterschiede bei den Objekten 10, 13, 16, 18 und 21. Bei diesen lieferte PoseCNN deutlich bessere Ergebnisse für die ADD und ADD-S Metrik.

Die Objekte, die von diesem Netzwerk schwierig erkannt wurden, sind vor allem symmetrische Objekte als auch Objekte, die kaum markante Punkte und Kanten haben, wie die Objekte 10 und 18. Das schlechte Abschneiden symmetrischer Objekte bei der ADD Metrik ist auch darauf zurückzuführen, dass die Symmetriebehandlung dem Netzwerk teilweise falsche aber äquivalente Posen vorgibt, um die Symmetrie zu eliminieren. Diese Symmetriebehandlung wird selbstverständlich nicht auf die Posen für Evaluation angewandt, da sonst die Ergebnisse nicht mit denen anderer Arbeiten vergleichbar wären. Dadurch haben aber auch die Ground Truth Posen einen gewissen Fehler in der ADD Metrik und dementsprechend ist ein solcher auch beim Netzwerk zu erwarten. Besonders gut wurden quaderförmige Objekte, wie die Objekte 2, 3, 7 und 8, erkannt. Ausgenommen davon sind die symmetrischen Objekte 16 und 21.

In der Tabelle 5.3 ist ein Vergleich von PoseCNN, PVNet (Peng u. a. 2019) und dieser Arbeit aufgeführt. Verglichen werden die kombinierte ADD(-S) Metrik, in der für nicht symmetrische Objekte die ADD Metrik und für symmetrische Objekte

die ADD-S Metrik verwendet wird und der Anteil der korrekt erkannten Posen nach 2D Projektionsmetrik. Beide Metriken und die Ergebnisse für PVNet und PoseCNN sind aus PVNet (Peng u. a. 2019) übernommen worden. Diese Arbeit übertrifft PoseCNN bei der ADD(-S) leicht und bei dem Anteil korrekter Posen nach der 2D Projektionsmetik bei weitem. PVNet übertrifft hingegen diese Arbeit in den beiden genannten Werten.

In Abbildung 5.1 sind Bilder für die das Netzwerk gute Ergebnisse liefert. Dabei ist in den Abbildungen 5.1a, 5.1b und 5.1c zu sehen, dass das Netzwerk gut mit Verdeckung umgehen kann. Zudem sind die linken unteren Ecken der Boxen in Abbildung 5.1f falsch platziert. Allerdings ist in Abbildung 5.1g zu sehen, dass die 6D Posen sehr gut sind. Das lässt darauf schließen, dass der RANSAC-Algorithmus die falsch lokalisierten Punkte als Ausreißer klassifiziert haben muss. Die charakteristischen Punkte der Schere in Abbildung 5.1h wurden nur teilweise gefunden, während die PAFs in Abbildung 5.1i komplett lokalisiert werden. Dennoch sind die gefundenen Punkte ausreichend, um eine gute 6D Pose zu schätzen, wie in Abbildung 5.1j zu sehen ist. In den Abbildungen 5.1d und 5.1e sind die Visualisierung und 6D Pose eines weiteren gut erkannten Bildes zu sehen.

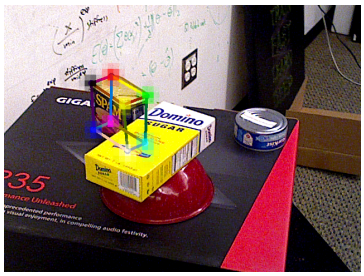
Es treten aber auch einige Schwierigkeiten auf, auf die in Abbildung 5.2 eingegangen wird. Ein Problem ist, dass die PAFs zwar gut bestimmt werden, die Heatmaps aber keine Punkte erzeugen und so die PAFs auch keine Punkte verknüpfen können. Dieses Problem ist in den Abbildungen 5.2a und 5.2b aufgeführt. Das weniger gute Auffinden der Heatmaps ist vermutlich darauf zurückzuführen, dass 4 Stufen zur Bestimmung der PAFs, aber nur 2 Stufen zur Bestimmung der Heatmaps verwendet werden. Aufgrund der fehlenden Punkte kommt es dann zu mehrdeutigen Objektskeletten, die zu falschen 6D Posen führen können, wie es in Abbildung 5.2c zu erkennen ist. Die 6D Posen der anderen Objekte wurden hingegen gut erkannt und ebenfalls eingezeichnet. Wenn stattdessen manche charakteristische Punkte mehrfach erkannt werden, kann es dazu kommen, dass diese die Objekte zerteilen. Das ist damit zu begründen, dass die Punkte, wie in Abbildung 5.2d, auf einer Verbindungslinie mit einem im Objektskelett benachbarten Punkt liegen. Dadurch lassen die PAFs, die die korrekte Verknüpfung plausibel erscheinen lassen, auch die kürzere zerteilte Verknüpfung plausibel erscheinen. Das ist in Abbildung 5.2e zu sehen. Sobald der korrekte und der falsche Punkt unterschiedlichen Objekten zugeordnet sind, werden alle Verknüpfungen zwischen diesen ignoriert und das Objekt bleibt zerteilt. In Abbildung 5.2f ist zu sehen, dass die anderen Objekte hingegen gut erkannt wurden.

In den Tabellen 5.4, 5.5, 5.6 und 5.7 sieht man den Anteil der erkannten Objekte für die ADD(-S) Metrik und 2D Projektionsmetrik aufgetragen nach der Sichtbarkeit des Objektes bzw. der Anzahl der gefundenen charakteristischen Punkte. Die

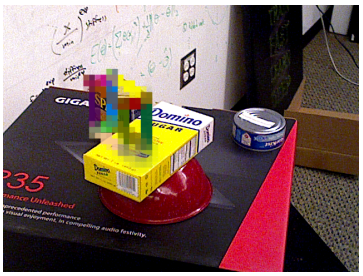
Tabelle 5.2: Area under Curve für verschiedene Metriken für dieses Modell und für PoseCNN (Xiang u. a. 2018)

Objekt	diese Arbeit			PoseCNN	
	ADD	ADD-S	2D Proj.	ADD	ADD-S
1	49.9	80.7	54.8	**50.9**	**84.0**
2	**80.5**	**88.4**	84.3	51.7	76.9
3	**85.5**	**92.4**	88.8	68.6	84.3
4	**68.5**	**81.4**	84.8	66.0	80.9
5	**87.0**	**93.3**	89.8	79.9	90.2
6	**79.3**	**89.7**	81.7	70.4	87.9
7	**81.8**	**89.5**	88.7	62.9	79.0
8	**89.4**	**94.0**	92.9	75.2	87.1
9	**59.6**	70.0	69.0	**59.6**	**78.5**
10	36.5	58.3	55.0	**72.3**	**85.9**
11	**78.1**	**86.9**	78.0	52.5	76.8
12	**56.7**	67.1	66.2	50.5	**71.9**
13	**12.2**	23.5	4.1	6.5	**69.7**
14	54.0	76.9	75.2	**57.7**	**78.0**
15	**82.8**	**91.0**	88.2	55.1	72.8
16	16.7	29.6	29.5	**31.8**	**65.8**
17	**46.0**	**64.1**	76.7	35.8	56.2
18	9.8	11.9	20.8	**58.0**	**71.4**
19	20.0	47.4	8.9	**25.0**	**49.9**
20	14.1	45.5	3.5	**15.8**	**47.0**
21	12.1	29.7	2.3	**40.4**	**87.8**
insgesamt	**59.0**	72.7	65.0	53.7	**75.9**

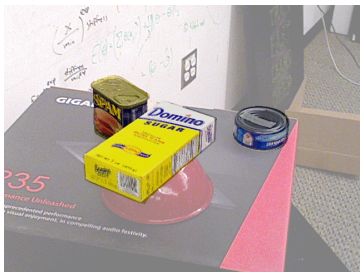
Sichtbarkeit eines Objektes ist aus dem YCB-V Datensatz direkt entnommen und ist definiert durch den Anteil der Pixel des Objektes in der gegebenen 6D Pose, die nicht verdeckt sind. Es ist zu erkennen, dass der Anteil der gefundenen Objekte mit der Sichtbarkeit steigt. Es gibt einige Ausreißer, diese sind aber dadurch zu erklären, dass nur wenige Objekte der entsprechenden Zelle zugeordnet sind. Der steigende Trend ist ebenfalls in den Tabellen 5.6 und 5.7 zu sehen. In diesen Tabellen sind weniger Ausreißer. Diese lassen sich mit dem gleichen Grund erklären. Für die symmetrischen Objekte ist das Ergebnis der 2D Projektionsmetrik weniger gut, als für die restlichen Objekte. Das ist dadurch zu erklären, dass in der 2D Projektionsmetrik die Symmetrie nicht beachtet wird. Die projizierte Ground Truth Pose weicht mehr als 5 Pixel von den projizierten symmetrieäquivalenten Posen ab, wodurch alle diese Posen als falsch klassifiziert werden.



(a) Heatmap (Obj. 09) bei Verdeckung



(b) PAFs (Obj. 09) bei Verdeckung



(c) 6D Pose bei Verdeckung

(d) Visualisierung

(e) 6D Pose

(f) Visualisierung mit Ausreißern

(g) 6D Pose mit Ausreißern

(h) Heatmaps (Obj. 17) für fehlende Punkte

(i) PAFs (Obj. 17) für fehlende Punkte

(j) 6D Pose für fehlende Punkte

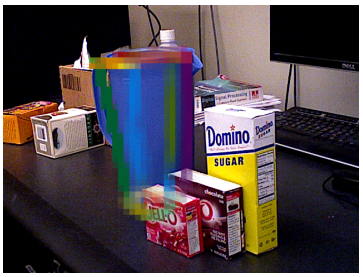
Abbildung 5.1: Beispielbilder mit gutem Ergebnis

(a) Heatmaps (Obj. 12/ 16) für Mehrdeutigkeit aufgrund fehlender Punkte

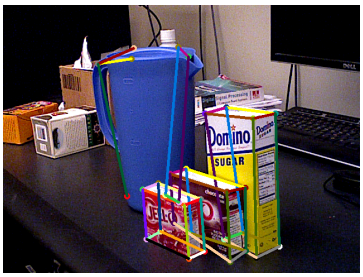
(b) PAFs (Obj. 12/ 16) für Mehrdeutigkeit aufgrund fehlender Punkte

(c) 6D Pose für Mehrdeutigkeit aufgrund fehlender Punkte

(d) Heatmaps (Obj. 11) für zerteilte Objekte



(e) PAFs (Obj. 11) für zerteilte Objekte



(f) Visualisierung für zerteilte Objekte

Abbildung 5.2: Beispielbilder mit weniger gutem Ergebnis

Tabelle 5.3: Vergleich mit PVNet (Peng u. a. 2019) mit den in PVNet verwendeten Metriken

Metrik	PoseCNN	diese Arbeit	PVNet
ADD(-S) AuC	61.0	62.4	**73.4**
2D Proj.	3.7	41.2	**47.4**

Tabelle 5.4: Anteil, der nach der ADD(-S) Metrik, richtigen 6D Posen, aufgetragen nach Sichtbarkeit der Objekte

Objekt	30%	40%	50%	60%	70%	80%	90%	100%	insgesamt
1	-	-	-	-	9.3	30.4	18.0	23.0	21.3
2	45.9	82.6	75.9	92.2	72.5	71.0	84.3	89.7	80.6
3	-	-	-	-	44.9	69.5	88.2	87.1	79.4
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	48.0	39.6
5	-	-	-	-	-	-	-	87.6	87.6
6	-	-	-	-	-	-	-	28.6	28.6
7	-	-	-	21.8	57.1	88.7	-	-	56.1
8	-	-	-	-	-	85.7	82.9	64.8	65.2
9	-	9.4	2.3	8.0	10.9	15.3	32.9	63.0	36.5
10	-	-	-	0.0	0.0	0.0	0.0	26.0	20.8
11	-	-	-	-	-	-	28.4	78.0	75.2
12	-	-	-	0.0	12.6	25.9	68.7	58.6	47.8
13	-	-	1.8	0.0	0.8	0.4	54.4	81.6	36.7
14	-	-	-	-	-	-	0.0	11.2	11.2
15	-	-	-	-	8.3	54.1	69.7	77.1	75.6
16	-	-	-	-	1.9	36.2	0.0	-	29.4
17	-	-	-	7.1	56.3	6.5	0.3	-	22.9
18	-	-	-	-	-	0.0	0.0	0.3	0.3
19	-	-	-	0.0	0.0	18.1	15.7	66.0	53.3
20	-	-	0.0	0.0	64.6	62.0	55.6	76.2	75.2
21	-	0.0	0.0	0.0	16.7	3.6	11.0	10.4	9.8

Tabelle 5.5: Anteil, der nach der 2D Projektionsmetrik, richtigen 6D Posen, aufgetragen nach Sichtbarkeit der Objekte

Objekt	30%	40%	50%	60%	70%	80%	90%	100%	insgesamt
1	-	-	-	-	2.1	3.2	14.7	7.5	9.7
2	2.7	13.0	60.7	70.1	32.0	65.3	85.0	86.8	64.8
3	-	-	-	-	19.1	80.4	71.6	79.7	74.9
4	0.0	5.2	6.4	16.9	2.1	2.9	31.2	59.0	50.3
5	-	-	-	-	-	-	-	91.4	91.4
6	-	-	-	-	-	-	-	46.6	46.6
7	-	-	-	66.7	63.0	83.9	-	-	71.6
8	-	-	-	-	-	85.7	97.1	95.6	95.6
9	-	31.8	4.9	5.8	38.5	67.8	63.4	86.3	61.7
10	-	-	-	0.0	0.0	0.0	0.0	14.6	11.7
11	-	-	-	-	-	-	9.5	55.6	52.9
12	-	-	-	0.0	1.2	2.6	46.0	40.6	28.5
13	-	-	0.0	0.0	0.0	0.0	0.0	0.0	0.0
14	-	-	-	-	-	-	0.0	34.1	34.1
15	-	-	-	-	23.3	63.9	71.6	72.8	72.1
16	-	-	-	-	0.0	1.4	0.0	-	1.1
17	-	-	-	12.4	54.4	9.4	4.7	-	24.8
18	-	-	-	-	-	0.0	0.0	0.0	0.0
19	-	-	-	0.0	0.0	0.0	0.0	0.0	0.0
20	-	-	0.0	0.0	0.0	0.0	11.1	0.0	0.0
21	-	0.0	0.0	0.0	8.3	0.0	0.3	1.6	1.2

Tabelle 5.6: Anteil, der nach der ADD(-S) Metrik, richtigen 6D Posen, aufgetragen nach Anzahl der gefundenen charakteristischen Punkte

Objekt	4	5	6	7	8	insgesamt
1	5.3	1.5	15.6	18.7	22.5	21.3
2	76.7	62.1	84.4	81.5	81.0	80.6
3	30.8	55.4	53.8	68.6	81.1	79.4
4	12.5	0.0	3.6	5.6	40.2	39.6
5	0.0	41.7	53.1	73.9	89.0	87.6
6	0.0	4.3	24.9	33.5	28.9	28.6
7	0.0	45.0	39.5	56.5	59.8	56.1
8	0.0	0.0	87.2	44.7	65.2	65.2
9	8.8	7.9	13.0	14.1	41.4	36.5
10	0.0	0.0	3.4	8.1	25.5	20.8
11	12.5	15.6	72.4	48.4	85.0	75.2
12	0.4	8.0	13.8	18.6	60.3	47.8
13	0.9	0.4	0.0	0.0	75.6	36.7
14	4.6	1.5	16.5	5.5	11.4	11.2
15	25.0	8.3	40.9	73.6	76.6	75.6
16	9.6	21.5	48.3	31.2	39.3	29.4
17	0.0	0.0	3.8	30.5	6.8	22.9
18	0.0	0.0	1.0	0.0	0.3	0.3
19	15.9	30.6	65.1	32.5	58.4	53.3
20	29.1	25.2	65.7	69.6	83.3	75.2
21	9.9	7.4	15.3	15.9	7.5	9.8

Tabelle 5.7: Anteil, der nach der 2D Projektionsmetrik, richtigen 6D Posen, aufgetragen nach Anzahl der gefundenen charakteristischen Punkte

Objekt	4	5	6	7	8	insgesamt
1	0.0	0.0	8.1	5.5	10.6	9.7
2	14.9	24.5	61.0	70.5	75.1	64.8
3	46.2	28.6	42.4	46.3	77.4	74.9
4	16.7	7.7	14.3	13.0	51.0	50.3
5	0.0	8.3	63.3	33.3	94.0	91.4
6	3.2	11.4	26.7	33.7	52.2	46.6
7	0.0	45.0	60.9	66.8	75.8	71.6
8	0.0	0.0	97.9	93.6	95.6	95.6
9	8.8	15.1	30.0	22.8	70.2	61.7
10	0.0	0.0	0.9	2.6	14.9	11.7
11	9.4	4.2	31.6	30.7	61.4	52.9
12	0.0	1.3	3.2	3.6	36.9	28.5
13	0.0	0.0	0.0	0.0	0.0	0.0
14	1.1	8.3	21.0	16.9	41.8	34.1
15	0.0	4.2	40.9	67.4	73.3	72.1
16	0.0	0.0	2.5	2.9	0.7	1.1
17	0.0	0.0	3.8	33.4	6.8	24.8
18	0.0	0.0	0.0	0.0	0.0	0.0
19	0.0	0.0	0.0	0.0	0.0	0.0
20	0.0	0.9	0.0	0.0	0.0	0.0
21	5.5	0.0	0.0	0.0	0.0	1.2

5.3.1 systematische Untersuchungen

Zusätzlich wurden in dieser Arbeit einige systematische Untersuchungen gemacht, um den Einfluss gewisser Aspekte genauer zu untersuchen.

PAFs

Es wurde untersucht, welchen Nutzen PAFs bei Datensätzen bieten, die nur maximal eine Instanz eines Objektes pro Bild haben. Dabei wurde hier der Nutzen der PAFs in der Nachverarbeitung überprüft. Die PAFs wurden normal berechnet und fließen auch ganz normal in die Berechnung der Heatmaps ein. Allerdings wurde die Zuordnung von charakteristischen Punkten zu Instanzen nicht mehr mithilfe der PAFs berechnet. Stattdessen wurde das Maximum jeder Heatmap bestimmt. Falls das Maximum größer als ein festgelegter Schwellwert ist, liegt der zugehörige charakteristische Punkt auf diesem Maximum und ist Teil der einzigen Instanz des zugehörigen Objektes im Bild.

In der Tabelle 5.8 sind die Ergebnisse dieses Vergleichs zu sehen. Dabei sind die Ergebnisse einmal für jedes Objekt einzeln und einmal für alle Objekte zusammengefasst aufgeführt. Die Verwendung von PAFs lieferte in fast allen Fällen ein besseres Ergebnis. Eine Ausnahme bildet Objekt 18, das mit der Verwendung reiner Heatmaps erheblich besser erkannt wurde. Das liegt vermutlich daran, dass die PAFs, die sich an der Spitze und dem Ende des Stiftes befinden kurz sind und aufgrund der geringen Größe des Stiftes häufig nicht ausreichend gefunden werden, sodass das Objektskelett des Stiftes nicht vollständig verknüpft wird. Außerdem lieferten die symmetrischen Objekte 13, 19 und 20 einen besseren Wert bei der 2D Projektionsmetrik, wenn nur die Heatmaps verwendet werden.

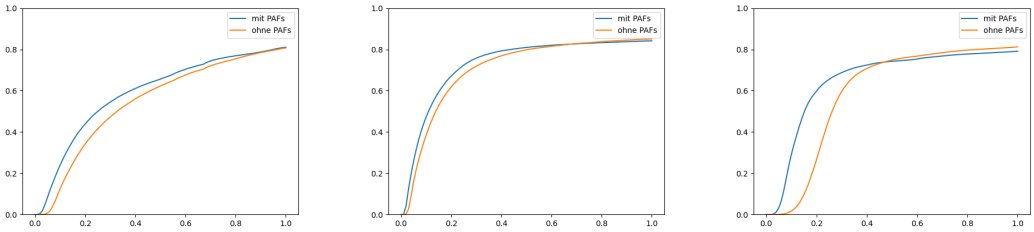
In der Abbildung 5.3 sind die Graphen für die Berechnung der Area under Curve aufgeführt. Man sieht, dass der Graph für die Ergebnisse mit Verwendung von PAFs bei allen Metriken bei kleinen Fehlertoleranzen höher liegt.

Anzahl an Objekten pro Modell

Des Weiteren wurde untersucht, wie die Ergebnisse dadurch beeinflusst werden, ob ein Netzwerk nur für ein Objekt trainiert wurde oder ob ein Modell für mehrere verschiedene Objekte trainiert wurde. In der Tabelle 5.9 sind die Area under Curve Ergebnisse für diesen Vergleich aufgeführt. Dabei ist zu erkennen, dass die Ergebnisse für Modelle, die nur ein Objekt erkennen sollten, deutlich besser sind. Dementsprechend sollten ausschließlich Modelle verwendet werden, die nur ein Objekt erkennen. In Tabelle 5.10 ist der Vergleich zweier 2-Objekt Modelle zu sehen. Das erste Modell hat, wie in Abschnitt 4.2 angesprochen, die Anzahl der Ein- und

Tabelle 5.8: Area under Curve für verschiedene Metriken unter Verwendung von PAFs und unter reiner Verwendung von Heatmaps

Objekt	mit PAFs			nur Heatmaps		
	ADD	ADD-S	2D Proj.	ADD	ADD-S	2D Proj.
1	**49.9**	**80.7**	**54.8**	48.0	80.1	50.2
2	**80.5**	**88.4**	**84.3**	73.1	83.3	72.7
3	**85.5**	**92.4**	**88.8**	79.2	89.4	76.6
4	**68.5**	**81.4**	**84.8**	59.5	75.6	75.0
5	**87.0**	**93.3**	**89.8**	80.4	90.8	79.3
6	**79.3**	**89.7**	**81.7**	68.3	84.5	72.4
7	**81.8**	**89.5**	**88.7**	74.0	84.3	81.2
8	**89.4**	**94.0**	**92.9**	81.8	90.3	81.0
9	**59.6**	**70.0**	**69.0**	54.3	66.3	60.4
10	**36.5**	**58.3**	**55.0**	35.2	55.3	53.0
11	**78.1**	**86.9**	**78.0**	72.9	84.3	67.4
12	**56.7**	**67.1**	**66.2**	51.7	64.3	57.8
13	**12.2**	**23.5**	**4.1**	11.5	23.2	**4.1**
14	**54.0**	**76.9**	**75.2**	49.1	72.0	63.3
15	**82.8**	**91.0**	**88.2**	76.8	88.3	77.1
16	**16.7**	**29.6**	**29.5**	15.7	27.4	23.2
17	**46.0**	**64.1**	**76.7**	37.3	56.2	65.1
18	9.8	11.9	20.8	**36.1**	**43.1**	**65.1**
19	**20.0**	**47.4**	8.9	17.8	45.1	**10.1**
20	**14.1**	**45.5**	3.5	11.8	43.0	**4.7**
21	**12.1**	**29.7**	**2.3**	6.2	18.3	0.2
insgesamt	**59.0**	**72.7**	**65.0**	54.5	70.4	58.9

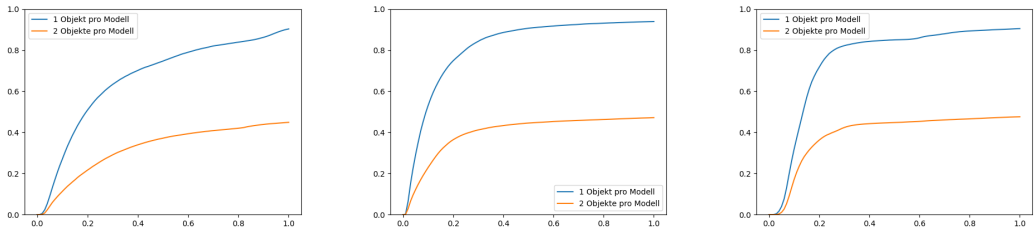


(a) ADD (b) ADD-S (c) 2D Projektion

Abbildung 5.3: Vergleich der Area under Curve für alle Objekte des YCB-V Datensatz unter Verwendung von PAFs und unter reiner Verwendung der Heatmaps

Tabelle 5.9: Area under Curve für verschiedene Metriken unter Verwendung von Modellen, die 1 Objekt erkennen und Modellen, die 2 Objekte erkennen

Objekt	1-Objekt Modelle			2-Objekt Modelle		
	ADD	ADD-S	2D Proj.	ADD	ADD-S	2D Proj.
1	**49.9**	**80.7**	**54.8**	18.0	31.5	23.0
4	**68.5**	**81.4**	**84.8**	38.5	43.6	46.8
5	**87.0**	**93.3**	**89.8**	20.3	23.8	23.9
6	**79.3**	**89.7**	**81.7**	53.6	63.1	60.3
7	**81.8**	**89.5**	**88.7**	38.5	42.3	42.5
8	**89.4**	**94.0**	**92.9**	39.8	44.4	45.3
9	**59.6**	**70.0**	**69.0**	25.2	33.3	35.5
10	**36.5**	**58.3**	**55.0**	8.3	12.4	16.9
insgesamt	**66.2**	**81.3**	**75.2**	32.2	39.8	39.2



(a) ADD (b) ADD-S (c) 2D Projektion

Abbildung 5.4: Vergleich der Area under Curve für alle Objekte des YCB-V Datensatz unter Verwendung von Modellen, die 1 Objekt erkennen und Modellen, die 2 Objekte erkennen

Ausgabekanäle jeder Stufe verdoppelt und ist damit doppelt so groß. Das zweite Modelle hat die normale Größe. Allerdings ist die Verbesserung bei einer der Objektanzahl entsprechenden Vergrößerung des Netzwerkes nur gering.

In Abbildung 5.4 sind die Graphen für die Area under Curve Berechnung für diese Untersuchung zu sehen. Es ist erkennbar, dass der Verlauf der beiden Graphen ähnlich aussieht, die Graphen der 1-Objekt Modelle aber generell höhere Werte haben.

Auswahl der charakteristischen Punkte

Zudem werden die Ergebnisse, die die automatisch generierten charakteristischen Punkte und die von Hand ausgewählten charakteristischen Punkte liefern, verglichen. Die Objekte 1 und 2 sind exemplarisch in Abbildung 5.5 zu sehen. Da nicht

Tabelle 5.10: Area under Curve für verschiedene Metriken unter Verwendung von normal großen und doppelt so großen 2-Objekt Modellen

Objekt	doppelte Größe			normale Größe		
	ADD	ADD-S	2D Proj.	ADD	ADD-S	2D Proj.
1	**19.1**	**31.5**	**25.8**	18.0	**31.5**	23.0
4	**41.5**	**47.1**	**47.4**	38.5	43.6	46.8
insgesamt	**32.2**	**40.7**	38.5	**32.2**	39.8	**39.2**

Tabelle 5.11: Area under Curve für verschiedene Metriken unter Verwendung automatisch generierten und händisch gewählten charakteristischen Punkten

	händisch gewählte Punkte			automatisch gewählte Punkte		
Objekte pro Modell	ADD	ADD-S	2D Proj.	ADD	ADD-S	2D Proj.
1	**49.9**	**80.7**	**54.8**	26.8	53.7	40.3
2	**18.0**	**31.5**	**23.0**	14.0	26.1	20.1

viele 1-Objekt und 2-Objekt Modelle für die automatisch generierten charakteristischen Punkte existieren, werden diese stichprobenartig verglichen. Dafür werden die Ergebnisse eines 1-Objekt Modells für Objekt 1 und eines 2-Objekt Modells für Objekt 1 mit den Ergebnissen der händisch gewählten Punkte verglichen.

In Tabelle 5.11 sind die Ergebnisse dieses Vergleichs aufgeführt. Die händisch gewählten charakteristischen Punkte lieferten demnach deutlich bessere Ergebnisse als die automatisch generierten charakteristischen Punkte. Der Grund dafür ist vermutlich, dass die charakteristischen Punkte und die PAFs leichter gefunden werden können, wenn sie von Hand auf herausstechenden Merkmalen und Positionen platziert sind. Bei den Boxen in den Abbildungen 5.5b und 5.5b ist es beispielsweise so, dass die automatisch generierten charakteristischen Punkte auf den Kanten liegen und dadurch schwieriger zu erkennen sind als die händisch gewählten charakteristischen Punkte, die auf den Ecken platziert sind. Außerdem liegen bei den Dosen in den Abbildungen 5.5a und 5.5d, sowie bei der Flasche in Abbildung 5.5e einige Punkte auch auf Flächen, statt auf Kanten, und sind dadurch in zwei Dimensionen schwierig zu bestimmen.

RANSAC Grenzwert

Zuletzt wurde überprüft, wie das Ändern des RANSAC Grenzwertes die Ergebnisse beeinflusst. Der RANSAC Grenzwert ist der maximale 2D Projektionsfehler, den ein Punkt im Vergleich zu einer geschätzten Pose haben darf, um noch als Teil dieser Pose zu gelten. Untersucht wurden die Grenzwerte 3, 5, 8, 10 und 12 Pixel.

(a) 01 (b) 02 (c) 03 (d) 04 (e) 05

Abbildung 5.5: Objekte aus dem YCB-V Datensatz mit eingezeichneten Heatmaps der automatisch gewählten charakteristischen Punkte

Alle Grenzwerte lieferten bei allen Metriken die gleichen Area under Curve Werte. Dementsprechend ist die Wahl dieses Grenzwertes für das Ergebnis bei dieser Methode irrelevant.

6 Fazit

In dieser Arbeit wurde OpenPose adaptiert und auf die 6D Objektposenschätzung erweitert. Für diesen zweistufigen Ansatz wurden der PnP- und RANSAC-Algorithmus zur Berechnung der 6D-Pose verwendet. Es wurde ein Framework, das OpenPose implementiert, genutzt. Der YCB-V Datensatz wurde um zwei Arten von charakteristischen Punkten und zugehörige Verknüpfungen dieser zu Objektskeletten erweitert. Außerdem wurde eine einfache Symmetriebehandlung eingesetzt. Die Ergebnisse dieser Arbeit wurden auf einer Vielzahl von Metriken erfasst und mit dem state-of-the-art verglichen.

Dabei lieferte diese Arbeit das drittbeste Ergebnis des BOP Wettbewerbs für den YCB-V Datensatz ohne Verwendung von Tiefeninformationen. Zudem schnitt diese Arbeit bei der ADD(-S) Metrik und bei der 2D Projektionsmetrik besser ab als PoseCNN. Es wurde gezeigt, dass PAFs auch bei Datensätzen mit nur einer Objektinstanz im Bild die Ergebnisse verbessern und dass Modelle, die nur 1 Objekt erkennen sollen, deutlich besser abschneiden als Modelle, die 2 Objekte erkennen sollen. Außerdem liefern händisch gewählte charakteristische Punkte bessere Ergebnisse als automatisch gewählte. Des Weiteren ist die Wahl des RANSAC Grenzwertes für die Ergebnisse dieser Arbeit irrelevant. Vor allem quaderförmige Objekte konnten gut erkannt werden. Hingegen werden die 6D-Pose symmetrischer Objekte und von Objekten ohne markante Punkte und Textur schlecht erkannt. PVNet lieferte bessere Ergebnisse für die ADD(-S) Metrik und die 2D Projektionsmetrik als diese Arbeit.

6.1 Ausblick

Eine automatische Auswahl der charakteristischen Punkte und PAFs, die ähnlich gute Ergebnisse liefert sollte konstruiert werden, um die Ausweitung der Methodik auf neue Datensätze zu vereinfachen. Es wäre interessant zu sehen, welche Ergebnisse mit einer besseren Symmetriebehandlung erreicht werden können. Die Ergebnisse sollten auf den symmetrischen Objekten verbessert werden. Besonders die Größe des Netzwerkes ist ein Hyperparameter dessen Belegung bisher kaum überprüft wurde und der einen entscheidenden Einfluss auf das Ergebnis hat.

Abkürzungsverzeichnis

PAF	Part Affinity Field
BOP	Benchmark for 6D Object Pose Estimation
COCO	Common Objects in Context
MSPD	Maximum Symmetry-Aware Projection Distance
MSSD	Maximum Symmetry-Aware Surface Distance
PnP	Perspective-n-Point
RANSAC	Random Sample Consensus
VSD	Visible Surface Discrepancy
YCB-V	Yale-CMU-Berkeley-Video

Abbildungsverzeichnis

Tabellenverzeichnis

Literatur

An, Wangpeng (31. Juli 2020). *Tensorboy/pytorch_Realtime_Multi-Person_Pose_Estimation*. URL: https://github.com/tensorboy/pytorch_Realtime_Multi-Person_Pose_Estimation (besucht am 01. 08. 2020).

Calli, Berk, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel und Aaron M. Dollar (Sep. 2015). „Benchmarking in Manipulation Research: Using the Yale-CMU-Berkeley Object and Model Set". In: *Ieee robotics automation magazine* 22.3, S. 36–52. ISSN: 1558-223X.

Cao, Zhe, Gines Hidalgo, Tomas Simon, Shih-En Wei und Yaser Sheikh (30. Mai 2019). *OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields*. arXiv: 1812.08008 [cs]. URL: http://arxiv.org/abs/1812.08008 (besucht am 09. 06. 2020).

Cao, Zhe, Tomas Simon, Shih-En Wei und Yaser Sheikh (13. Apr. 2017). *Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields*. Version 2. arXiv: 1611.08050 [cs]. URL: http://arxiv.org/abs/1611.08050 (besucht am 02. 08. 2020).

Fischler, Martin A. und Robert C. Bolles (1. Juni 1981). „Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography". In: *Communications of the acm* 24.6, S. 381–395. ISSN: 0001-0782. URL: https://doi.org/10.1145/358669.358692 (besucht am 08. 08. 2020).

Hodan, Tomas (21. Juli 2020). *Thodan/bop_toolkit*. URL: https://github.com/thodan/bop_toolkit (besucht am 01. 08. 2020).

Hodan, Tomas, Martin Sundermeyer, Bertram Drost, Yann Labbe, Eric Brachmann, Frank Michel, Carsten Rother und Jiri Matas (13. Okt. 2020). *BOP Challenge 2020 on 6D Object Localization*. arXiv: 2009.07378 [cs]. URL: http://arxiv.org/abs/2009.07378 (besucht am 01. 01. 2021).

Labbé, Yann, Justin Carpentier, Mathieu Aubry und Josef Sivic (2020). „CosyPose: Consistent Multi-view Multi-object 6D Pose Estimation". In: *Computer Vision – ECCV 2020*. Hrsg. von Andrea Vedaldi, Horst Bischof, Thomas Brox und Jan-Michael Frahm. Lecture Notes in Computer Science. Cham: Springer International Publishing, S. 574–591. ISBN: 978-3-030-58520-4.

Lepetit, Vincent, Francesc Moreno-Noguer und Pascal Fua (19. Juli 2008). „EPnP: An Accurate O(n) Solution to the PnP Problem". In: *International journal of computer vision* 81.2, S. 155. ISSN: 1573-1405. URL: https://doi.org/10.1007/s11263-008-0152-6 (besucht am 08. 08. 2020).

Lin, Tsung-Yi, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick und Piotr Dollár (20. Feb. 2015). *Microsoft COCO: Common Objects in Context*. arXiv: 1405.0312 [cs]. URL: http://arxiv.org/abs/1405.0312 (besucht am 21. 12. 2020).

Pavlakos, Georgios, Xiaowei Zhou, Aaron Chan, Konstantinos G. Derpanis und Kostas Daniilidis (14. März 2017). *6-DoF Object Pose from Semantic Keypoints*. arXiv: 1703.04670 [cs]. URL: http://arxiv.org/abs/1703.04670 (besucht am 06. 08. 2020).

Peng, Sida, Yuan Liu, Qixing Huang, Xiaowei Zhou und Hujun Bao (2019). „PVNet: Pixel-Wise Voting Network for 6DoF Pose Estimation". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, S. 4561–4570. URL: https://openaccess.thecvf.com/content_CVPR_2019/html/Peng_PVNet_Pixel-Wise_Voting_Network_for_6DoF_Pose_Estimation_CVPR_2019_paper.html (besucht am 01. 08. 2020).

Rad, Mahdi und Vincent Lepetit (26. März 2018). *BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects without Using Depth*. arXiv: 1703.10896 [cs]. URL: http://arxiv.org/abs/1703.10896 (besucht am 06. 08. 2020).

Xiang, Yu, Tanner Schmidt, Venkatraman Narayanan und Dieter Fox (26. Mai 2018). *PoseCNN: A Convolutional Neural Network for 6D Object Pose Estimation in Cluttered Scenes*. arXiv: 1711.00199 [cs]. URL: http://arxiv.org/abs/1711.00199 (besucht am 01. 08. 2020).

Zhang, Z. (Nov. 2000). „A flexible new technique for camera calibration". In: *Ieee transactions on pattern analysis and machine intelligence* 22.11, S. 1330–1334. ISSN: 1939-3539.