Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Andreas Neumann

Vergleich und Analyse von kamerabasierten Tracking-Algorithmen für die Implementation einer Personenfolge-Funktionalität eines Roboters im Healthcare Sektor

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Andreas Neumann

Vergleich und Analyse von kamerabasierten Tracking-Algorithmen für die Implementation einer Personenfolge-Funktionalität eines Roboters im Healthcare Sektor

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Informatik Technischer Systeme
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Peer Stelldinger
Zweitgutachter: Prof. Dr. Tim Tiedemann

Eingereicht am: 5. März 2024

Andreas Neumann

Thema der Arbeit

Vergleich und Analyse von kamerabasierten Tracking-Algorithmen für die Implementation einer Personenfolge-Funktionalität eines Roboters im Healthcare Sektor

Stichworte

Tracking, Robotik, Personenfolge, Healthcare Sektor

Kurzzusammenfassung

Diese Thesis beschäftigt sich mit dem Vergleich und der Analyse verschiedener Tracking-Algorithmen zur Implementierung einer Personenfolge-Funktionalität auf einem Roboter und bedient sich zur Evaluation bei typischen Szenarien aus dem Healthcare Sektor. Hierzu wird die Personenfolge auf einem autonomen Roboter simuliert, um dann die modular austauschbaren Tracking-Algorithmen miteinander zu vergleichen.

Andreas Neumann

Title of the paper

Comparison and analysis of camerabased tracking algorithms for the implementation of a person follow functionality of a robot in the healthcare sector

Keywords

tracking, robotics,person follow, healthcare sector

Abstract

This thesis deals with the comparison and analysis of different tracking algorithms for the implementation of a person tracking functionality on a robot and uses typical scenarios from the healthcare sector for the evaluation. For this purpose, the person follow is simulated on an autonomous robot in order to then compare the modular interchangeable tracking algorithms with each other.

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung

Durch die wachsende Anzahl mobiler Roboter in zahlreichen Bereichen (Zhang u. a., 2022b) ist die Implementierung einer Personenfolge-Funktionalität von zunehmendem Interesse. Dabei stellt insbesondere der Healthcare Sektor spezifische Herausforderungen an kamerabasierte Lösungen für die Ziellokalisierung. Um einen geeigneten Kandidaten für eine kontinuierliche, kamerabasierte Ziellokalisierung zu finden, werden in dieser Arbeit diverse State-of-the-Art Tracking-Algorithmen miteinander verglichen. Da derzeit keine öffentlich verfügbaren Benchmarks zur Evaluation von Tracking-Algorithmen im Kontext einer Personenfolge-Funktionalität auf einem mobilen Roboter im Healthcare Sektor existieren, soll sich diese Arbeit mit der Erstellung einer Sammlung von gelabelten Szenarien, der Festlegung von Evaluationmetriken und schließlich mit der Bewertung der ausgewählten Tracking-Algorithmen anhand des erarbeiteten Benchmarks befassen. Das Ziel besteht dabei darin, den bestmöglichen Tracking-Algorithmus für die Implementation einer Personenfolge-Funktionalität eines mobilen Roboters im Healthcare Sektor zu finden.

2 Aufbau der Thesis

Die Thesis beginnt mit der Einführung in die Grundlagen des visuellen Objekttrackings, um sich dann mit der begründeten Auswahl mehrerer zur Evaluation selektierten Tracking-Algorithmen befassen, welche im darauffolgenden Abschnitt genauer in ihrer Funktionsweise beschrieben werden. Danach schließt das Grundlagenkapitel mit der Auswahl und Erklärung der Evaluationsmetriken ab. Zunächst wird dann die Hardwareplattform vorgestellt, die für die Aufnahme der Szenarien verwendet wird. Dann wird, um die Rahmenbedinungen noch ausführlicher darzustellen, der Anwendungskontext und seine spezifischen Herausforderungen erläutert. Nachdem der Kontext definiert und die technischen Grundlagen geschildert wurden, werden darauf die durchgeführten Experimente beschrieben. Die Resultate der Experimente werden dann in der Evaluation ausgewertet. Zum Abschluss werden im Fazit die Evaluationsergebnisse diskutiert und die gewonnenen Erkenntnisse zusammengefasst und im Ausblick ein Überblick über mögliche Verbesserungen und Alternativen zu den getesteten Ansätzen gegeben.

3 Grundlagen

In den Grundlagen findet zuerst eine kurze Einführung in das Themengebiet des Visuellen Objekt-Trackings statt, wobei die Ansätze grob in Single Object Tracking (SOT) und Multi Object Tracking (MOT) unterteilt werden. Der darauffolgende Teil beschäftigt sich zuerst mit der Auswahl der in dieser Arbeit verwendeten Tracking-Algorithmen, worauf diese dann in ihrer Funktionsweise beschrieben werden. Zum Schluss werden die für die Evaluation verwendeten Metriken behandelt.

3.1 Visuelles Objekttracking

Das Ziel des Visuellen Objekttrackings wird definiert als die Verfolgung eines beliebigen Ziels in jedem Bild eines Videos, wobei nur sein anfängliches Aussehen gegeben ist (Ye u. a., 2022). Um dies zu erreichen, werden automatisch oder manuell eine meist beliebige Anzahl an Zielobjekten ausgewählt, deren Position über eine Menge aufeinanderfolgender Bilder verfolgt werden soll. Die Lösung dieses Problems ist auch heute noch eine der wichtigsten Aufgaben im Bereich der Computer Vision und hat zahlreiche unterschiedliche Anwendungsgebiete wie Verkehrsüberwachung, Robotik, autonomes Fahren und vieles mehr (Soleimanitaleb und Keyvanrad, 2022). Die angewandten Methoden lassen sich grob in vier Kategorien unterteilen: Eigenschaftsbasiert, Segmentationsbasiert, Schätzungsbasiert und Lernbasiert. Diese Arbeit fokussiert sich auf Lernbasierte Methoden, welche in den letzten Jahren große Fortschritte gemacht haben (Soleimanitaleb und Keyvanrad, 2022) und die Ranglisten in allen betrachteten Single-Object und Multi-Object Tracking Benchmarks (Kristan u. a. (2016), Valmadre u. a. (2018), Moudgil und Gandhi (2019), Zhang u. a. (2022a) und Dendorfer u. a. (2020)) anführen.

3.1.1 Single-Object Tracking

Beim Single-Object Tracking (SOT) wird nach Festlegung einer initialen Bounding Box, das in der Bounding Box befindliche Objekt über die folgenden Bilder verfolgt. SOT verzichtet dabei auf eine Klassifizierung des Zielobjekts in Kategorien und konzentriert sich auf die Unterscheidung zwischen Zielobjekt und Hintergrund (Soleimanitaleb und Keyvanrad, 2022).

3.1.2 Multi-Object Tracking

Das Ziel des Multi-Object Tracking (MOT) ist es, alle Objekte in einem Bild zu detektieren und zu verfolgen und dabei eine einmalige ID für jedes Objekt zu halten (Aharon u. a., 2022). Das effektivste Paradigma ist dafür das tracking-by-detection (Aharon u. a., 2022), welches ein Detektionsmodell verwendet, um zuerst alle Objekte einer Kategorie im Bild zu finden, damit diese dann bestehenden oder neuen identifizierbaren Tracklets zugeordnet werden können. Die beliebtesten Ansätze für die Zuordnung sind zum einen die Berechnung des Überlappungswerts (IoU), welcher auf der Annahme basiert, dass sich die Position eines Zielobjekts und damit seine Bounding Box zwischen Bildern nur minimal verschiebt und zum anderen die Erstellung von Erscheinungsbildmodellen, welche für eine Zuordnung durch Wiedererkennung genutzt werden können (Aharon u. a., 2022).

3.2 Tracking-Algorithmen

In diesem Abschnitt wird nun zuerst die Auswahl der zur Evaluation ausgewählten Tracking-Algorithmen begründet, worauf die Funktionsweisen der einzelnen Algorithmen beschrieben werden.

3.2.1 Auswahl

Die Auswahl des optimalen Trackers für die Implementierung einer Personenfolge auf einem mobilen Roboter wird durch grundlegende Hardwarebeschränkungen beeinflusst, da der Fokus meist auf den Einsatz energieeffizienter Hardware zur Maximierung der Betriebszeit gesetzt wird. Daher ist es notwendig Tracker zu wählen, die auch auf Entwicklerboards, wie zum Beispiel dem von NVIDIA hergestellten Jetson Board effizient laufen und eine ausreichende Bildrate liefern. Zudem sollten die Tracker in der Lage sein, Ziele über einen längeren Zeitraum zu verfolgen, weswegen sich zur Auswahl der Tracker zuerst auf "Long-term Visual Tracking: Review and Experimental Comparison" von Liu u. a. (2022) bezogen wird, worauf dann aktuellere Veröffentlichungen im Bereich des visuellen Trackings in Betracht gezogen werden.

In "Long-term Visual Tracking: Review and Experimental Comparison" von Liu u. a. (2022) stechen insbesondere die Tracker KeepTrack (Mayer u. a., 2021) und STARK (Yan u. a., 2021) hervor, wobei KeepTrack durch den innovativen Ansatz der Zielkandidatenzuordnung zur Unterdrückung von Ablenkern und herausragende Benchmark-Ergebnisse auf den Benchmarks OxUvA, VOTLT2018, LaSOT und TLP (Liu u. a., 2022) seine Eignung beweist. Wohingegen STARK durch die Verwendung der Transformer-Architektur und ebenfalls herausragende Benchmark-Ergebnisse, insbesondere auf der Langzeit-Teilmenge des VTUAV-V Benchmarks auffällt (Liu u. a., 2022). Ein gemeinsamer Nachteil der beiden Algorithmen ist die geringe Bildrate von 19 Bildern pro Sekunde bei KeepTrack und 42 Bildern pro Sekunde bei der verwendeten Variante von STARK (STARK-ST50). Aus diesem Grund wurden zusätzlich die neueren Tracker MixFormerV2-B (Cui u. a., 2023) mit 165 Bildern pro Sekunde und OSTrack-256 (Ye u. a., 2022) mit 105 Bildern pro Sekunde und mit besseren Benchmark-Ergebnissen auf dem LaSOT Benchmark ausgewählt (Hinweis: die genannten Bildraten wurden Cui u. a. (2023) entnommen, da die Arbeit keine Angaben zur genutzten Hardware macht, sind die Werte nur für den groben Vergleich der Effizienz gedacht). Wie STARK basiert sowohl OSTrack als auch MixFormerV2 auf der Transformer-Architektur. OSTrack erreicht hierbei insbesondere durch die Verwendung von "Early Candidate Elimination"-Modulen eine hohe Effizienz, da

Bildbereiche, die schon in den anfänglichen Schichten des neuronalen Netzes dem Hintergrund zugeordnet werden können, von darüberliegenden Schichten ignoriert werden. Beim MixFormerV2 wird die Effizienz im Vergleich zum Vorgänger MixFormer (25 Bilder pro Sekunde) von Cui u. a. (2022) vor allem durch die Anwendung von Modelldestillation und den Verzicht auf Convolutional Neural Networks erreicht.

Zusätzlich zur Auswahl von Trackern aus der Kategorie des Single Object Trackings wurde der Multi Object Tracker BoT-SORT (Aharon u. a., 2022) mit dem Detektionsmodell YOLOv8 der Firma ultralytics ausgewählt, welcher zu seiner Veröffentlichung State-of-the-Art-Ergebnisse auf den MOT17 und MOT20 Testdatensätze lieferte. Außerdem erhöht ein eingebautes Kamerabewegungskompensations-Feature (Aharon u. a., 2022) die Qualität der Bounding Boxes bei Bewegung der Kamera, wie sie bei einer auf einem mobilen Roboter installierten Kamera auftreten.

3.2.2 KeepTrack

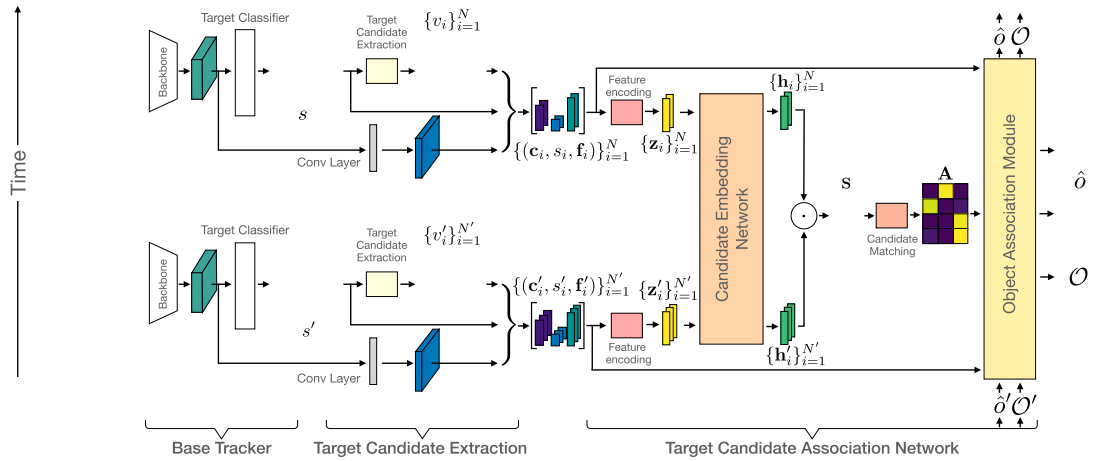

Abbildung 3.1: Überblick über die gesamte Online-Tracking-Pipeline, die die vorherigen und aktuellen Bilder gemeinsam verarbeitet, um das Zielobjekt vorherzusagen (Mayer u. a., 2021).

Der Tracker KeepTrack (Mayer u. a., 2021) basiert auf dem Konzept, Kandidaten, die von einem Basetracker im aktuellen und vorherigen Frame identifiziert wurden, miteinander zu verknüpfen. Das Ziel dabei ist, die Identifikation des Zielobjekts zu erleichtern, indem sogenannte Ablenkerobjekte parallel gehalten werden. Für jeden Frame wird ein Eigenschaftsvektor für jeden Kandidaten erstellt, basierend auf dem Classifier-Score, der Position und dem generierten

Appearance Cue. Diese Vektoren werden durch das Candidate Embedding Network weiter optimiert, indem die Eigenschaftsvektoren aus dem vorherigen Frame zusammen mit den neuen Vektoren in das Netzwerk eingespeist werden. Die Ähnlichkeit der Eigenschaftsvektoren aus beiden Frames wird dann durch die Berechnung ihrer Skalarprodukte ermittelt. Die resultierenden Skalarprodukte dienen dazu, eine Zuordnungsmatrix zu erstellen. Neue und verschwundene Kandidaten werden dem sogenannten Dustbin (Abfalleimer) zugeordnet, während bekannte Kandidaten eindeutig zwischen den Frames zugeordnet werden. Nach Abschluss der Inferenz des Zuordnungsprozesses durch das Objekt-Assoziations-Modul wird der Ziel-Kandidat für den Frame durch die Kombination des Target-Classifier-Scores und der Konfidenz des Objekt-Assoziations-Moduls bestimmt.

3.2.3 STARK

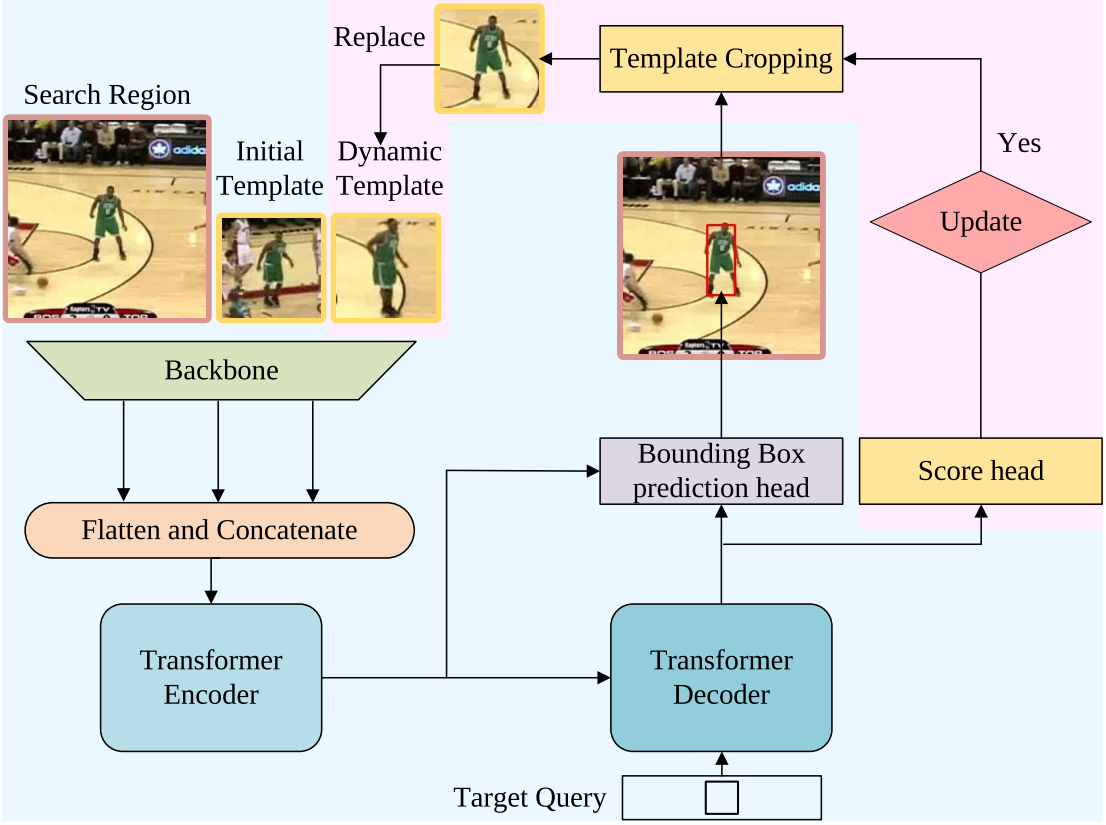

Abbildung 3.2: Netzwerk-Architektur von STARK (Yan u. a., 2021).

Bei STARK (**S**patio-**T**emporal Tr**a**nsfo**r**mer for Visual Trac**k**ing) (Yan u. a., 2021) werden zyklisch aus der Suchregion, dem initialen Suchtemplate und dem dynamischen Suchtemplate durch ein ResNet-Backbone Merkmale extrahiert. Die resultierenden zweidimensionalen Merkmalsmatrizen werden danach für die Eingabe in einen Encoder abgeflacht und miteinander konkateniert. Der Encoder wird dann genutzt, um Zusammenhänge innerhalb der des Eingabevektors hervorzuheben. Dazu werden mehrere Encoderschichten bestehend aus Selbstaufmerksamkeitsmodulen und einfachen feed-forward Netzwerken verwendet. Die Ausgabe des Encoders dient einem Decoder und einem Bounding Box Prediction Head als Eingabe. Während der Decoder die Daten nutzt, um Aufmerksamkeit auf die räumliche Lage des Zielobjekts anzuwenden, erzeugt der Bounding Box Prediction Head auf Grundlage der Outputs vom Encoder und dem Decoder eine Wahrscheinlichkeitskarte für die obere linke und die untere rechte Ecke der vorherzusagenden Bounding Box. Die Koordinaten der Bounding Box werden dann durch die Berechnung der Erwartungswerte der Wahrscheinlichkeitskarten berechnet. Parallel dazu wird ein Score Head verwendet, um die Konfidenz zu berechnen, ob sich das Ziel im Suchgebiet befindet. Wenn bei der Berechnung ein gewisser Threshold überschritten wird, wird das dynamische Suchtemplate aktualisiert. Dazu wird das dynamische Suchtemplate mit dem Inhalt der aktuellen Bounding Box ersetzt.

3.2.4 OSTrack

OSTrack (Ye u. a., 2022) verwirft den weitverbreiteten Ansatz, Tracking in zwei Datenflüsse und zwei Stufen aufzuteilen, wie es beispielsweise STARK macht. Die zwei Datenflüsse bedeuten dabei, dass die Merkmale vom Zieltemplate und der Suchregion separat extrahiert werden. Die zwei Stufen beziehen sich auf die voneinander getrennt durchgeführte Merkmalsextraktion und Abhängigkeitsmodellierung. Entgegen diesen Schemas führt OSTrack die Merkmalsextraktion und die Abhängigkeitsmodellierung in einem gemeinsamen Schritt durch, indem die aus Zieltemplate und Suchregion generierten Eingabetokens ohne komplexe Vorverarbeitungsschritte in einen Encoder gespeist werden. Der Encoder ist in der Lage auf Basis der gegebenen Eingabe den globalen Kontext und damit die Merkmale und Abhängigkeiten zu modellieren.

Wie 3.3 zu entnehmen ist, werden bei OSTrack sowohl das Zieltemplate als auch die Suchregion in Patches unterteilt. Die Patches werden zuerst abgeflacht und durch eine lineare Projektion auf die vom Encoder vorgegebene Länge gebracht. Die resultierenden Patch-Einbettungen werden zusätzlich jeweils mit einer positionellen Einbettung versehen. Die dabei entstandenen Tokens bilden durch Konkatenation die finale Eingabe für den mehrschichtigen Encoder, wobei die frühen Schichten des Encoders über sogenannte "Early Candidate Elimination"-Module

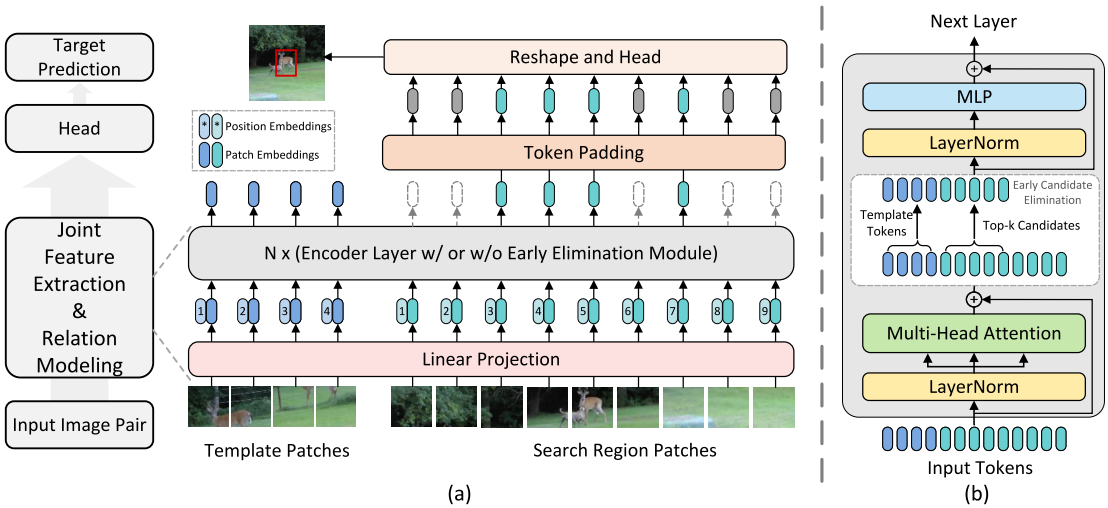

Abbildung 3.3: **(a)** Überblick über das Rahmenwerk von OSTrack (Ye u. a., 2022). **(b)** Aufbau der Encoder-Schicht mit "Early Candidate Elimination"-Modul (Ye u. a., 2022).

verfügen. Diese Module dienen dem Tracker zur Effizienzsteigerung, da Bereiche die als Teil des Hintergrunds identifiziert wurden, im Rest des Netzes nicht weiter betrachtet werden. Nachdem die Eingabedaten den Encoder durchlaufen haben, werden die durch die Early Candidate Elimination entstandenen Lücken mit Nullen wiederaufgefüllt, um die ursprüngliche Anordnung der Tokens wiederherzustellen. Die aufgefüllte Sequenz wird dann als zweidimensionale Merkmalskarte reinterpretiert und in ein Fully Convolutional Network (FCN) gespeist, aus dessen Ausgaben die finale Bounding Box und der Klassifizierungswert ermittelt werden kann. Im Gegensatz zu den anderen verwendeten Single-Object Trackern implementiert OSTrack keinen Mechanismus zur Aktualisierung des Zieltemplates.

3.2.5 MixFormerV2

Der Tracker MixFormerV2 (Cui u. a., 2023) greift den Ansatz von OSTrack (Ye u. a., 2022) auf und verwendet ein Transformer-Backbone für die Merkmalsextraktion und die Abhängigkeitsmodellierung. Allerdings verzichtet MixFormerV2 im Gegensatz zu OSTrack auf ein Fully Convolutional Network (FCN) zur Verarbeitung der Transformerausgabe und führt stattdessen spezielle Vorhersagetokens ein, welche in der Lage sind komplexe Zusammenhänge zwischen dem Zieltemplate und der Suchregion einzufangen (Cui u. a., 2023). Aus diesen können dann mit Hilfe mehrschichtiger Perzeptronen die Koordinaten und der Konfidenzwert der Bounding Boxes berechnet werden. Zusätzlich wird durch Modelldestillation eine Parametereduktion und damit eine Effizienzsteigerung erreicht.

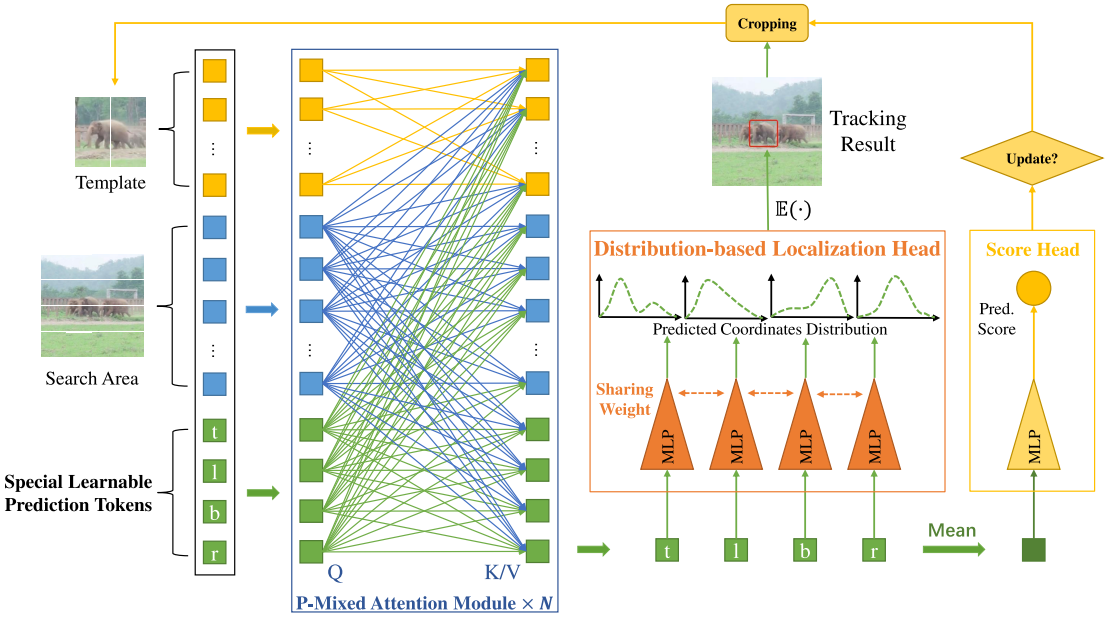

Abbildung 3.4: Überblick über das Rahmenwerk von MixFormerV2 (Cui u. a., 2023).

Bei MixFormerV2 werden wie bei OSTrack zuerst aus dem Zieltemplate und der Suchregion Tokens generiert. Die Tokens werden dann gemeinsam mit den lernbaren Vorhersagetokens konkateniert. Die resultierende Tokensequenz dient einem aus mehreren "Prediction-Token-Involved Mixed AttentionModulen bestehenden Transformer als Eingabe. Die Mixed Attention wird dabei verwendet, um wichtige Informationen für die Lokalisierung des Zielobjekts zu extrahieren. Nach Durchlaufen des Transformers werden die Vorhersagetokens genutzt, um mit gleichgewichteten mehrschichtigen Perzeptronen die Wahrscheinlichkeitsverteilung für die Koordinaten der Bounding Box zu berechnen. Durch den Erwartungswert kann dann die finale Bounding Box ermittelt werden. Zur Berechnung der Konfidenz wird der Mittelwert der Vorhersagetokens gebildet und dann ebenfalls in einen mehrschichtigen Perzeptron gespeist, welcher die Konfidenz ausgibt.

3.2.6 BoT-SORT

Der Multi Objekt Tracker BoT-SORT (ausgeschrieben **B**ag-**o**f-**T**ricks **S**imple **O**nline and **R**ealtime **T**racking) verwendet für das Tracking von Zielobjekten einen tracking-by-detection Ansatz. Beim tracking-by-detection werden im ersten Schritt durch einen Objektdetektor alle Objekte einer oder mehrerer bestimmter Klassen wie zum Beispiel Menschen, Hunde oder Katzen detektiert. Im zweiten Schritt wird zuerst das Bewegungsmodell des verwendeten Kalman-

Filters aktualisiert, um dann eine Vorhersage für die Bounding Boxes der bekannten Tracklets zu treffen. Danach wird versucht die vorhergesagten Bounding Boxes der Tracklets den vom Objektdetektor gegebenen Bounding Boxes zuzuordnen. Für die Zuordnung wird die Intersection-over-Union(IoU) zwischen den vorhergesagten und den detektierten Bounding Boxes berechnet, wobei die Paare mit den höchsten IoU-Werten einander zugeordnet werden. Zusätzlich werden für Detektionen mit hohen Konfidenzwerten und ohne Zuordnung zu einem der bestehenden Tracklets, neue Tracklets erstellt und Tracklets ohne Zuordnung gelöscht.

Zusätzlich wird die beschriebene Funktionsweise durch zwei Features unterstützt. Zum einen eine Kamerabewegungskompensation und zum anderen eine leichte Veränderung des verwendeten Kalman-Filters. Dabei wird die Kamerabewegungskompensation genutzt, um durch die Verfolgung von Bildschlüsselpunkten die Bewegung der Kamera nachzuvollziehen und basierend auf der errechneten Kamerabewegung den Zustandsvektor des Kalman-Filters zu korrigieren. Außerdem wurde der Kalman-Filter so verändert, dass er anstelle von Seitenverhältnisssen und Flächen der Bounding Boxes, wie im ursprünglichen SORT (Bewley u. a., 2016), direkt ihre Breite und Höhe vorhersagt. Dadurch entstehen Bounding Boxes die besser an ihr Ziel angespasst sind, was zu besserer Trackinggenauigkeit führt.

3.3 Evaluationsmetriken

Um die Qualität der erzeugten Ergebnisse zu bewerten und um abschätzen zu können wie zuverlässig die Tracker bei Anwesenheit der Zielperson Ergebnisse liefern, wurden die Metriken Precision (Pr) und Recall (Re) ausgewählt (Lukežič u. a., 2018). Dabei deutet eine hohe Precision auf eine hohe Anzahl an True Positives und eine geringe Zahl an False Positives hin. Wohingegen der Recall bemisst, wie viele von den möglichen Bounding Boxes der Grundwahrheit erfolgreich vom Tracking-Algorithmus erkannt wurden, somit steht ein hoher Recall für eine geringe Anzahl an False Negatives und eine hohe Anzahl an True Positives. Im Kontext einer Personenfolge sollte stets eine hohe Precision angestrebt werden, da False Positives dazu führen können, dass der Roboter vom Pfad abkommt. Dagegen ist es tolerierbar, wenn der Tracking-Algorithmus die Zielperson in einzelnen Bildern trotz Anwesenheit nicht erkennt, was bei einem geringen Recall der Fall wäre.

3.3.1 Precision

$$\mathrm{Pr}(\tau\theta) = \frac{1}{N_p} \sum_{t \in \{t : A_t(\theta_t) \neq \emptyset\}} \Omega(A_t(\theta_t), G_t) \tag{3.1}$$

Berechnet wird die Precision mit Klassifizierungsthreshold $\tau\theta$, indem die Summe der Überlappungsverhältnisse (IoU Werte) $\Omega(A_t(\theta_t), G_t)$ für alle Bilder mit erfolgreichen Messungen $t \in \{t : A_t(\theta_t) \neq \emptyset\}$, also Trackingresultate über dem Klassifizierungsthreshold $\tau\theta$, gebildet und dann durch ihre Anzahl N_p geteilt wird.

3.3.2 Recall

$$\mathrm{Re}(\tau\theta) = \frac{1}{N_g} \sum_{t \in \{t : G_t \neq \emptyset\}} \Omega(A_t(\theta_t), G_t) \tag{3.2}$$

Der Recall mit Klassifizierungsthreshold $\tau\theta$ wird berechnet, indem die Summe der Überlappungsverhältnisse (IoU Werte) $\Omega(A_t(\theta_t), G_t)$ für alle Bilder mit vorhandenen Grundwahrheitswert $t \in \{t : G_t(\theta_t) \neq \emptyset\}$ gebildet und dann durch ihre Anzahl N_g geteilt wird.

4 Hardwareplattform

Für die Aufnahme der Szenarien wurde eine OAK-D S2 vom Hersteller Luxonis verwendet, welche in einem Aufbau auf einer Robotnik Theron Roboterplattform integriert wurde (siehe 4.1). Die OAK-D S2 verfügt über drei Kameras, aufgeteilt in ein Paar links und rechts für die Aufnahme von Grayscale Bildern zur Erzeugung von Tiefenbildern und eine zentrale RGB Kamera. Bei der Aufnahme der Szenarien werden die Datenströme der Kameras mit einer Rate von 25 BpS aufgezeichnet. Zusätzlich wird zur Simulation der Personenfolge-Funktionalität die Möglichkeit genutzt, die Roboterplattform mit einem Controller manuell zu steuern.

Abbildung 4.1: Roboteraufbau mit markierter OAK-D S2

5 Anwendungskontext

Die Personenfolge-Funktionalität soll in erster Linie in Krankenhäusern und Pflegeheimen eingesetzt werden, welche sich insbesondere durch lange Flure, ähnlich bis gleich gekleidetes Personal und eine Vielzahl von potentiellen Ablenkern in Form von Patienten, Besuchern und Personal auszeichnen. Diese Umgebung stellt einige Herausforderungen an die Tracking-Algorithmen. Dazu gehören wechselhafte Lichtbedingungen, die beispielsweise durch kaputte Lampen auf einem ansonsten gut beleuchteten Flur, temporäre Verdeckung von Lichtquellen durch Personal oder Patienten oder einfallendes Sonnenlicht, welches aus Zimmern auf den Flur tritt, entstehen können. Auch Größenveränderungen der Zielperson, verursacht durch ihre Bewegung weg oder hin zum Roboter, sind eine Herausforderung. Die Verdeckung der Zielperson durch andere Personen ist ebenfalls problematisch, weil Mitglieder des Personals aufgrund ihrer ähnlichen Uniformen leicht mit der Zielperson verwechselt werden können.

Die Szenarien für diese Arbeit wurden in einem leerstehenden Abschnitt eines Hospitals (5.1) und auf einem Büroflur aufgenommen (5.2).

Abbildung 5.1: Roboter auf dem Flur des leerstehenden Krankenhausabschnitts

Abbildung 5.2: Büroflur

6 Experimente

In diesem Kapitel werden die durchgeführte Effizienzmessung und die zur Evaluation der Tracking-Algorithmen aufgezeichneten Szenarien beschrieben.

6.1 Effizienzmessung

Zur Messung der Effizienz wurde die durchschnittliche Bildrate der verwendeten Tracker in den aufgenommenen Szenarien gebildet. Dabei wurde eine NVIDIA GeForce GTX 1650 Ti Mobile verwendet. Zum Vergleich und zur Evaluation der Realtimefähigkeit auf Entwicklerboards wurde die Messung zusätzlich auf dem NVIDIA Jetson Orin Nano durchgeführt, allerdings ausschießlich mit dem Tracker OSTrack.

6.2 Szenarien

Im Folgenden werden die aufgezeichneten Szenarien vorgestellt und ihre Herausforderungen im Kontext des visuellen Objekttrackings erläutert. Die Szenarien sind von simpel bis komplex sortiert, wobei es sich bei den ersten drei Szenarien um simplere Szenarien mit zwei unterschiedlich gekleideten Personen und bei den restlichen vier Szenarien um komplexere Szenarien mit fünf in OP-Kassacks gekleideten Personen handelt. In allen Szenarien startet die Zielperson mit dem Gesicht zur Kamera und befindet sich in einer Entfernung zum Roboter, die es ermöglicht, den gesamten Kopf sowie den Großteil des Oberkörpers zu erkennen.

6.2.1 Szenario ”Ablenker geht vorbei”

In diesem Szenario geht ein Ablenker an der Zielperson vorbei, während der Roboter der Zielperson hinterherfährt. Die Herausforderungen bei diesem Szenario sind die Anwesenheit eines Ablenkers und die Veränderung der Position und Ausrichtung der Zielperson.

Abbildung 6.1: Szenario ”Ablenker geht vorbei”

6.2.2 Szenario ”Ablenker kreuzt”

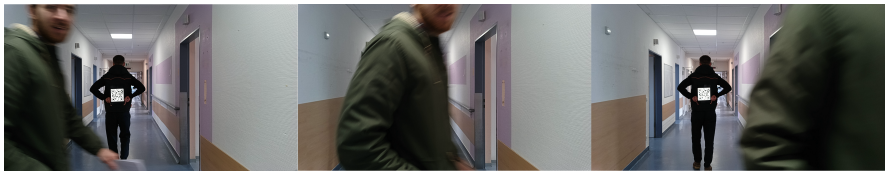

Abbildung 6.2: Szenario ”Ablenker kreuzt”

In diesem Szenario kreuzt ein Ablenker den Weg zwischen Roboter und Zielperson, wodurch die Zielperson für eine kurze Zeit (11 Bilder) verdeckt wird. Die Herausforderungen bei diesem Szenario sind die Anwesenheit eines Ablenkers, die Veränderung der Position und Ausrichtung der Zielperson und die teilweise bis komplette Verdeckung der Zielperson für einen kurzen Zeitraum.

6.2.3 Szenario ”Ablenker verdeckt”

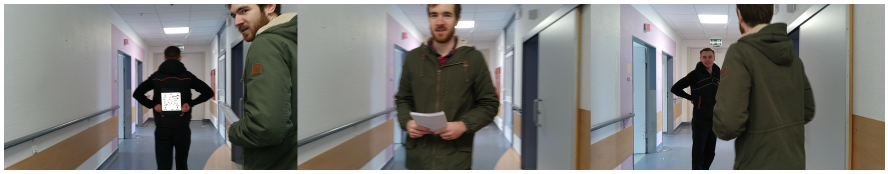

Abbildung 6.3: Szenario ”Ablenker verdeckt”

In diesem Szenario bewegt sich ein Ablenker vor den Roboter und bleibt zwischen Roboter und Zielperson stehen, wodurch die Zielperson für eine längere Zeit (74 Bilder) verdeckt wird. Danach bewegt sich der Ablenker weiter an der rechten Seite des Flurs in Richtung Zielperson und durchquert den Flur hinter der Zielperson, um durch die hinterste Tür der linken Seite des Flurs zu gehen. Die Herausforderungen bei diesem Szenario sind die Anwesenheit eines

Ablenkers, die Veränderung der Position und Ausrichtung der Zielperson und die teilweise bis komplette Verdeckung der Zielperson für einen längeren Zeitraum.

6.2.4 Szenario ”Ohne Interaktion durch Gruppe gehen”

Abbildung 6.4: Szenario ”Ohne Interaktion durch Gruppe gehen”

In diesem Szenario geht die Zielperson auf eine Gruppe von vier Personen zu, welche sich dann in je zwei Personen aufteilt, um den Weg für die Zielperson und den Roboter freizumachen. Durch den Geschwindigkeitsunterschied zwischen Zielperson und Roboter entsteht währenddessen eine wachsende Lücke zwischen ihnen, weshalb die Zielperson am Ende des Flurs anhält, sich umdreht und dem Roboter Zeit gibt zu ihr aufzuschließen. Die Herausforderungen sind hierbei die starken Veränderungen der Lichtverhältnisse, die Anwesenheit ähnlich gekleideter Ablenker, die Veränderung der Position und Ausrichtung der Zielperson und die Größenvariationen durch die variierende Distanz zwischen Roboter und Zielperson.

6.2.5 Szenario ”Durch sich bewegende Gruppe gehen”

Abbildung 6.5: Szenario ”Durch sich bewegende Gruppe gehen”

In diesem Szenario geht die Zielperson auf eine sich auf die Zielperson zu bewegende Gruppe von vier Personen zu, welche sich dann in je zwei Personen aufteilt, um den Weg für die Zielperson und den Roboter freizumachen. Durch den Geschwindigkeitsunterschied zwischen Zielperson und Roboter entsteht währenddessen eine wachsende Lücke zwischen ihnen, weshalb die Zielperson am Ende des Flurs anhält, sich umdreht und dem Roboter Zeit gibt zu ihr aufzuschließen. Die Herausforderungen sind hierbei die starken Veränderungen der Lichtverhältnisse, die Anwesenheit ähnlich gekleideter Ablenker, welche in diesem Szenario

besonders nah an der Zielperson vorbeigehen, die Veränderung der Position und Ausrichtung der Zielperson und die Größenvariationen durch die variierende Distanz zwischen Roboter und Zielperson.

6.2.6 Szenario ”Zwei Personen verdecken und Konversation”

Abbildung 6.6: Szenario ”Zwei Personen verdecken und Konversation”

In diesem Szenario geht die Zielperson auf zwei im Flur stehende Personen zu, währenddessen stellen sich zwei Ablenker zwischen Roboter und Zielperson, wodurch sie den Körper der Zielperson verdecken. Während die Verdeckung der Zielperson stattfindet, beginnt die Zielperson eine Konversation und dreht ihren Körper dabei um 90 Grad, um sich zu den anderen beiden Konversationsteilnehmern auszurichten. Nachdem die Ablenker den Weg zur Zielperson frei machen, beginnt der Roboter zur Zielperson aufzuschließen und die Zielperson beginnt ihren Weg entlang des Flurs fortzusetzen. Die Herausforderungen sind hierbei die starken Veränderungen der Lichtverhältnisse, die Anwesenheit ähnlich gekleideter Ablenker, die teilweise bis komplette Verdeckung der Zielperson, die Veränderung der Position und Ausrichtung der Zielperson und die Größenvariationen durch die variierende Distanz zwischen Roboter und Zielperson.

6.2.7 Szenario ”Mit Gruppe gehen”

Abbildung 6.7: Szenario ”Mit Gruppe gehen”

In diesem Szenario startet die Zielperson umgeben von Ablenkern, worauf sie gemeinsam ans Ende des Flurs gehen. Während des Szenarios wird das Ziel mehrfach von den Zielen teilweise bis komplett verdeckt. Die Herausforderungen sind hierbei die starken Veränderungen der

Lichtverhältnisse, die Anwesenheit ähnlich gekleideter Ablenker, die teilweise bis komplette Verdeckung der Zielperson, die Veränderung der Position und Ausrichtung der Zielperson und die Größenvariationen durch die variierende Distanz zwischen Roboter und Zielperson.

7 Evaluation

Um die Eignung der ausgewählten Tracking-Algorithmen festzustellen, werden diese auf Basis der durch die Experimente gewonnenen Daten nach den Kriterien Effizienz, Precision und Recall evaluiert.

7.1 Effizienzbewertung

Tabelle 7.1: Durchschnittliche Bildrate der Tracker

Tracker	Bildrate
KeepTrack	11 BpS
STARK	11.7 BpS
OSTrack	**14.4** BpS
MixFormerV2	12.5 BpS

Mit einem maximalen Unterschied von ca. 30 Prozent (siehe 7.1) liegen die Tracking-Algorithmen in ihrer Effizienz nah beieinander und können alle als Realtime-fähig eingestuft werden. Um zu überprüfen, ob die Tracking-Algorithmen auch auf einem Entwicklerboard Ergebnisse in Realtime produzieren können, wurde der Test auf einem NVIDIA Jetson Orin Nano, allerdings nur mit OSTrack, reproduziert. Dabei ist der Tracking-Algorithmus auf eine Bildrate von durchschnittlich 10.7 BpS gekommen und liegt damit zwar ca. 35 Prozent hinter der Leistung auf der NVIDIA GeForce GTX 1650 Ti Mobile, aber liefert immer noch ausreichend BpS, um in Realtime-Szenarien eingesetzt zu werden.

7.2 Szenarien

Für die Evaluation der Tracker auf den Szenarien wurde zuerst eine Grundwahrheit für alle verwendeten Szenarien ermittelt, indem vom BoT-SORT Algorithmus erzeugte Tracks der Zielperson zugeordnet wurden. Danach wurden die Trackingresultate der Szenarien nach den Metriken Precision und Recall bei einem Klassifizierungsthreshold von 0.8 bewertet. Das

Klassifizierungsthreshold wurde für alle Tracker gleichgesetzt, um vergleichbarere Ergebnisse zu erhalten und mit 0.8 relativ hoch gewählt, da für die Implementation einer Personenfolge-Funktionalität auf einem mobilen Roboter eine höhere Priorität auf der Precision als auf dem Recall liegt. Im Folgenden werden die Ergebnisse analysiert und besondere Auffälligkeiten im Verhalten der Tracking-Algorithmen beschrieben. Obwohl BoT-SORT einzelnd betrachtet nahezu perfekte Tracks liefert, wird der Tracker nicht weiter berücksichtigt, da seine schwache Wiedererkennungsfähigkeit es ihm in keinem der Fälle mit längeren Verdeckungen ermöglichte, die Zielperson wieder zu identifizieren.

7.2.1 Szenario ”Ablenker geht vorbei”

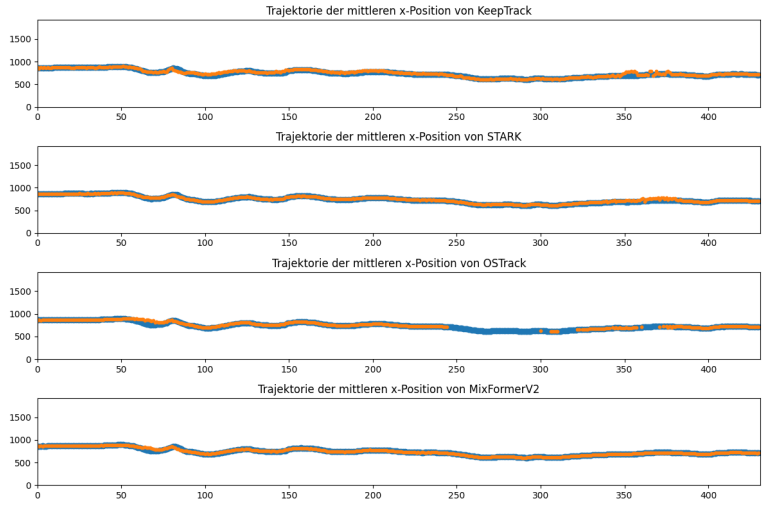

Abbildung 7.1: Trajektorien der mittleren x-Position der verwendeten Tracker für das Szenario ”Ablenker geht vorbei”. Die Ergebnisse der Tracker sind in Orange und die Groundtruth ist in Blau dargestellt.

Tabelle 7.2: Evaluationsresultate für das Szenario ”Ablenker geht vorbei”

Tracker	Precision	Recall
KeepTrack	0.9203	0.9203
STARK	0.9436	0.9436
OSTrack	0.956	0.763
MixFormerV2	**0.9663**	**0.9663**

Beim Durchlaufen des Szenarios lässt sich erkennen, dass die Bounding Boxes der Tracker KeepTrack und STARK unter einer konstanten leichten Störung leiden, also ihre Bounding Boxes einen gewissen Grad an Schwingung aufweisen. Außerdem haben die beiden Tracker jeweils einen kurzen Moment in dem die Bounding Box komplett auf den Ablenker springt (7.1 Bilder ca. 350 bis ca. 370), die Zielperson jedoch nach kurzer Zeit wiedererfasst werden kann. OSTrack erzeugt wie MixFormerV2 konstant die korrekte Ausgabe, leidet jedoch bei Anwesenheit des Ablenkers unter einer geringen Konfidenz (7.1 Bilder ca. 245 bis ca. 370).

7.2.2 Szenario ”Ablenker kreuzt”

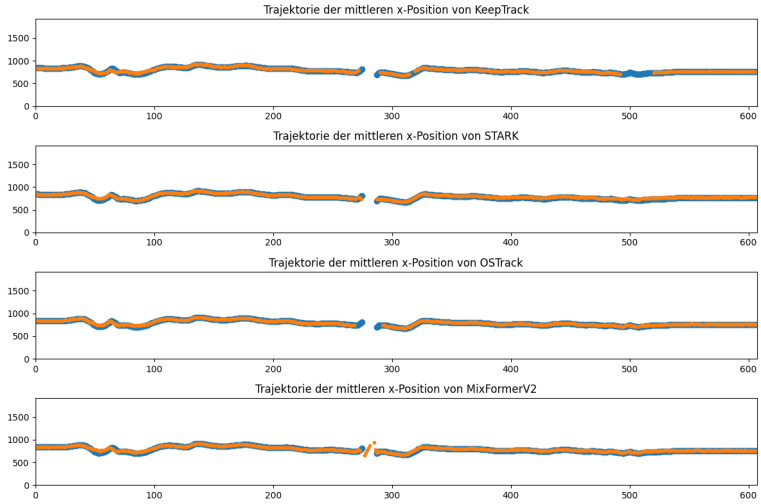

Abbildung 7.2: Trajektorien der mittleren x-Position der verwendeten Tracker für das Szenario ”Ablenker kreuzt”. Die Ergebnisse der Tracker sind in Orange und die Groundtruth ist in Blau dargestellt.

Tabelle 7.3: Evaluationsresultate für das Szenario ”Ablenker kreuzt”

Tracker	Precision	Recall
KeepTrack	0.9283	0.8676
STARK	0.9153	0.9137
OSTrack	**0.949**	0.9299
MixFormerV2	0.9252	**0.9329**

An den Ergebnissen lässt sich erkennen, dass die Tracker gut mit kurzen Verdeckungen umgehen können. Außerdem lässt sich beobachten, dass alle Tracker in der Lage sind, das Ziel nach der Verdeckung schnell wiederzuerkennen. MixFormerV2 erreicht in diesem Szenario den höchsten Recall, indem er alle seine Ergebnisse mit einer hohen Konfidenz bewertet und so Bounding Boxes der Grundwahrheit trifft, welche die anderen Tracker verpassen. OSTrack erreicht durch die hohe Qualität seiner Bounding Boxes und seine Selektivität die höchste Precision, verpasst aber mögliche Bounding Boxes, während der Ablenker das Bild kreuzt.

7.2.3 Szenario ”Ablenker verdeckt”

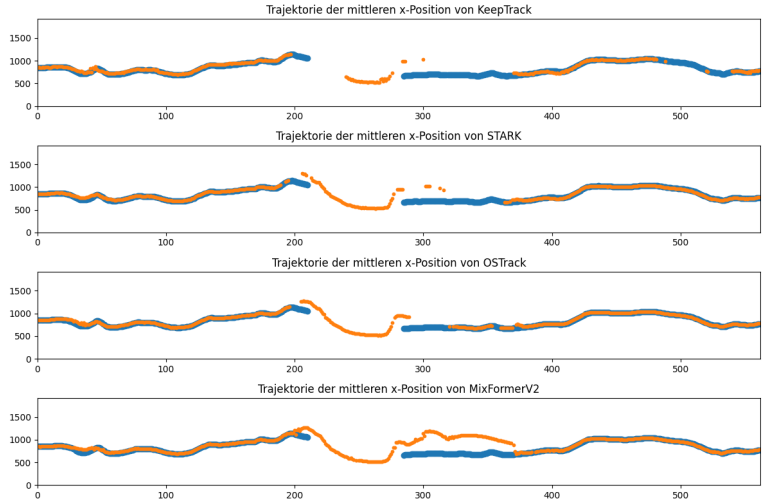

Abbildung 7.3: Trajektorien der mittleren x-Position der verwendeten Tracker für das Szenario ”Ablenker verdeckt”. Die Ergebnisse der Tracker sind in Orange und die Groundtruth ist in Blau dargestellt.

Tabelle 7.4: Evaluationsresultate für das Szenario ”Ablenker verdeckt”

Tracker	Precision	Recall
KeepTrack	**0.813**	0.6231
STARK	0.785	0.756
OSTrack	0.7941	**0.8315**
MixFormerV2	0.6749	0.7675

Wie den Daten in 7.1 zu entnehmen ist, hat die Verdeckung der Zielperson bei allen Trackern zu einem Zielwechsel geführt. Nach der Verdeckung ist MixFormerV2 der einzige Tracker der den Ablenker bis zum Verlassen des Bildes mit hoher Konfidenz verfolgt. Nach Verlassen des Ablenkers verfolgen KeepTrack, STARK und MixFormerV2 die Zielperson wieder mit einer hohen Konfidenz. Der einzige Tracker der in der Lage ist, die Zielperson vorm Verlassen des Ablenkers wiederzuerfassen, ist in diesem Szenario OSTrack, welcher mit dieser Leistung den höchsten Recall erreicht. Die höchste Precision erreicht KeepTrack, weil zum einen der Zielwechsel von der Zielperson zum Ablenker nicht so schnell passiert wie bei den anderen Trackern und weil zum anderen die Konfidenz nach der Verdeckung, während der gemeinsamen Anwesenheit von Zielperson und Ablenker, zu gering für eine Ausgabe ist.

7.2.4 Szenario ”Ohne Interaktion durch Gruppe gehen”

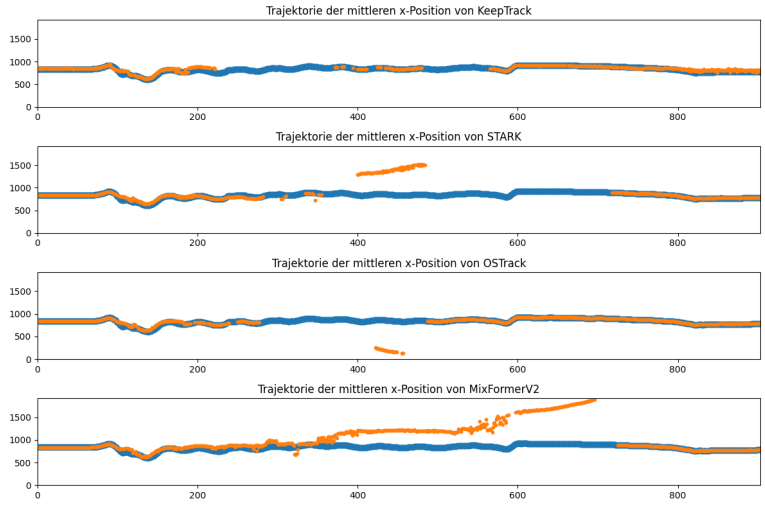

Abbildung 7.4: Trajektorien der mittleren x-Position der verwendeten Tracker für das Szenario ”Ohne Interaktion durch Gruppe gehen”. Die Ergebnisse der Tracker sind in Orange und die Groundtruth ist in Blau dargestellt.

Tabelle 7.5: Evaluationsresultate für das Szenario ”Ohne Interaktion durch Gruppe gehen”

Tracker	Precision	Recall
KeepTrack	**0.9108**	0.6112
STARK	0.7916	0.4944
OSTrack	0.8993	**0.6932**
MixFormerV2	0.5285	0.5045

In diesem Szenario findet bei allen Tracker bis auf KeepTrack ein Zielwechsel statt, wodurch KeepTrack die höchste Precision erreicht. OSTrack liefert vor und nach dem Durchfahren der Gruppensituation gute Ergebnisse und erreicht so den höchsten Recall. Allerdings springt er während der Gruppensituation auf einen Ablenker zur Rechten des Roboters über und liefert nur wenige Ergebnisse. Am schlechtesten schneiden STARK und MixFormerV2 ab, welche sich während des Passierens der Gruppe auf einige Muster auf der hinter ihr liegenden Wand fokussieren, bis die Wand aus dem Sichtfeld verschwindet und das Ziel wiedererfasst wird.

7.2.5 Szenario ”Durch sich bewegende Gruppe gehen”

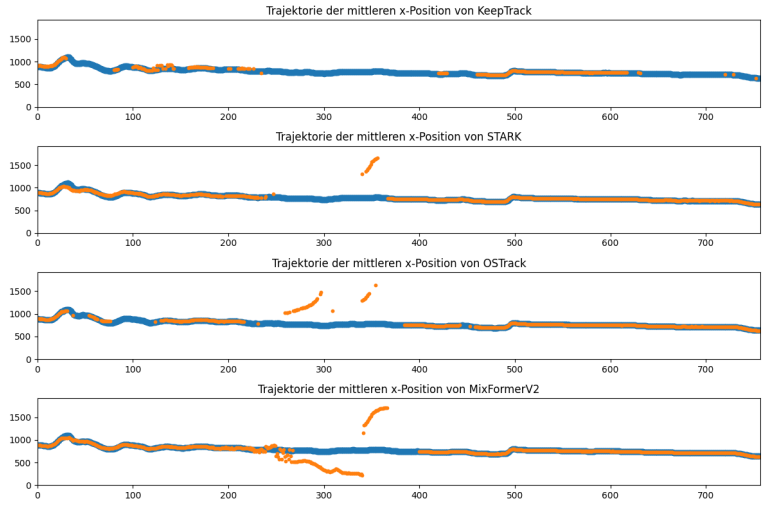

Abbildung 7.5: Trajektorien der mittleren x-Position der verwendeten Tracker für das Szenario ”Durch sich bewegende Gruppe gehen”. Die Ergebnisse der Tracker sind in Orange und die Groundtruth ist in Blau dargestellt.

Tabelle 7.6: Evaluationsresultate für das Szenario ”Durch sich bewegende Gruppe gehen”

Tracker	Precision	Recall
KeepTrack	0.849	0.3241
STARK	**0.9045**	**0.7611**
OSTrack	0.8636	0.6092
MixFormerV2	0.7917	0.7572

Obwohl in diesem Szenario zu keinem Zeitpunkt eine komplette Verdeckung der Zielperson stattfindet, ist keiner der Tracker in der Lage, die Zielperson durchgehend zu verfolgen. Eine möglich Erklärung für die Zielwechsel bei den Transformer-basierten Trackern ist, dass das initiale Zieltemplate, welches mit der Zielperson zur Kamera gerichtet aufgenommen wird, den entgegenkommenden Ablenkern ähnlicher ist als der Rückansicht der Zielperson. Zusätzlich wird der Zielwechsel bei den Transformer-basierten Trackern, welche das Ziel in einer Suchregion anstelle vom ganzen Bild suchen, durch die Nähe zwischen Zielperson und Ablenkern begünstigt. KeepTrack ist in diesem Szenario der einzige Tracker der keinen Zielwechsel durchführt, liefert dafür aber kaum Ergebnisse auf Grund niedriger Konfidenzwerte. STARK

dagegen hat nur während der Bewegung durch die Gruppe niedrige Konfidenzwerte, erzeugt aber dennoch bis auf einen kurzen Zielwechsel gute Ergebnisse.

7.2.6 Szenario ”Zwei Personen verdecken und Konversation”

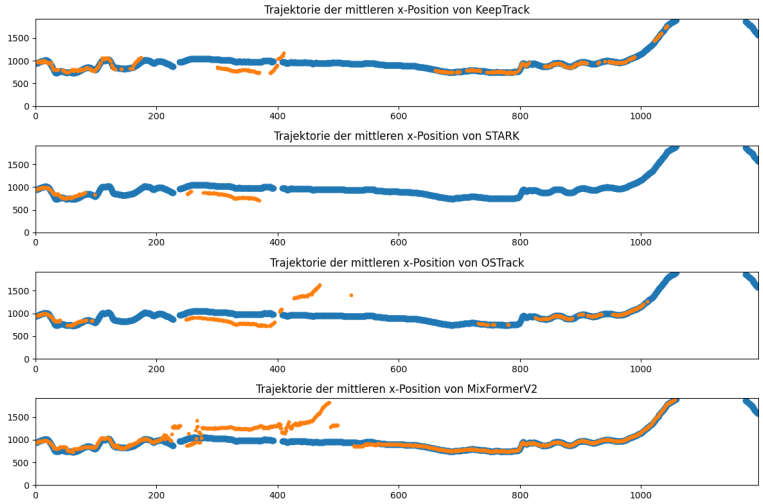

Abbildung 7.6: Trajektorien der mittleren x-Position der verwendeten Tracker für das Szenario ”Zwei Personen verdecken und Konversation”. Die Ergebnisse der Tracker sind in Orange und die Groundtruth ist in Blau dargestellt.

Tabelle 7.7: Evaluationsresultate für das Szenario ”Zwei Personen verdecken und Konversation”

Tracker	Precision	Recall
KeepTrack	**0.714**	0.3022
STARK	0.4263	0.0702
OSTrack	0.5107	0.1862
MixFormerV2	0.6656	**0.6423**

In diesem Szenario war keiner der Tracker fähig, die zwei verdeckenden Ablenker sicher als solche zu erkennen, wodurch in jedem Durchlauf mindestens ein Zielwechsel stattfand. Außerdem ist nur MixFormerV2 in der Lage gewesen, die Zielperson nach der Verdeckung mit hoher Konfidenz schnell wiederzuerfassen. Bei KeepTrack und OSTrack muss der Roboter erst wieder näher an die Zielperson heranfahren, um eine Wiedererfassung ermöglichen. Allerdings erreichen sie durch Konfidenzwerte nahe des Thresholds keine kontinuierliche Verfolgung. STARK liefert in diesem Szenario nahezu keine korrekten Ergebnisse un erreicht damit mit einem Recall von unter 0.1 den geringsten Recall im gesamten Experiment.

7.2.7 Szenario ”Mit Gruppe gehen”

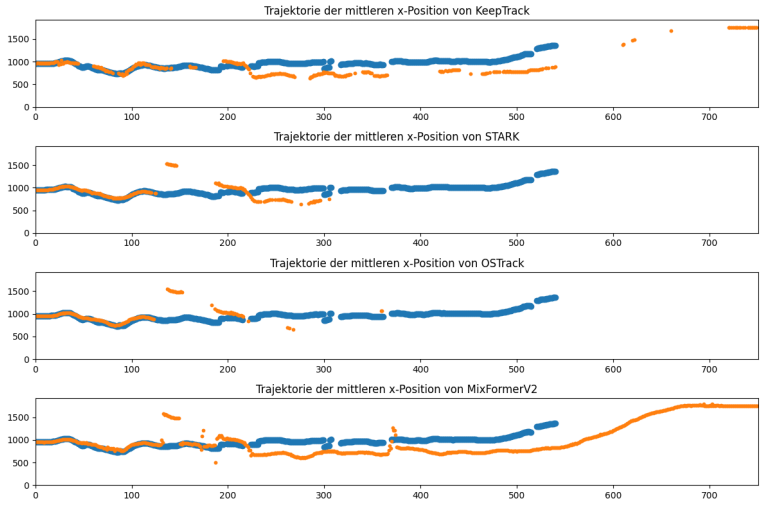

Abbildung 7.7: Trajektorien der mittleren x-Position der verwendeten Tracker für das Szenario ”Mit Gruppe gehen”. Die Ergebnisse der Tracker sind in Orange und die Groundtruth ist in Blau dargestellt.

Tabelle 7.8: Evaluationsresultate für das Szenario ”Mit Gruppe gehen”

Tracker	Precision	Recall
KeepTrack	0.3304	0.2398
STARK	0.5533	0.2357
OSTrack	**0.663**	0.2288
MixFormerV2	0.2276	**0.3371**

Bei diesem Szenario springen alle Tracker bis auf KeepTrack schon vor der Verdeckung der Zielperson auf einen Ablenker über. Nach der ersten Verdeckung war kein Tracker in der Lage, die Zielperson bei teilweiser Verdeckung wiederzuerfassen. Allerdings erzeugt OSTrack nur wenige Ergebnisse mit ausreichender Konfidenz und erreicht dadurch die höchste Precision. MixFormerV2 liefert dagegen durchgehend Ergebnisse und erreicht dadurch den höchsten Recall gepaart mit der geringsten Precision.

8 Fazit

Alle ausgewählten Tracker waren in der Lage, in den ersten beiden einfachen Szenarien ohne längere Verdeckungen die Zielperson kontinuierlich zu verfolgen und dabei Precision-Werte von über 0.9 zu erreichen. Allerdings kam es in allen Szenarien mit Verdeckungen bei allen Trackern zu Zielwechseln (siehe 7.3, 7.6 und 7.7). Zusätzlich war bis auf KeepTrack keiner der ausgewählten Tracker fähig, die Zielperson zuverlässig von nahe stehenden Ablenkern zu unterscheiden, was ebenfalls zu Zielwechseln führte (siehe 7.4 und 7.5). Dieses Verhalten ist für die Implementation einer Personenfolge-Funktionalität auf einem mobilen Roboter im Healthcare Sektor nicht tolerierbar, da Zielwechsel dazu führen, dass andere Personen als die Zielperson verfolgt beziehungsweise hinterhergefahren werden würden, was die Personenfolge-Funktionalität in Szenarien mit mehreren Personen unbrauchbar machen würde.

9 Aussicht

Die Arbeit hat aufgezeigt, dass insbesondere die zuverlässige Wiedererkennung der Zielperson und die Differenzierung zwischen Zielperson und Ablenkern eine große Schwierigkeit darstellen. Um diese Schwierigkeiten zu lösen wäre es sinnvoll, eine robuste Wiedererkennung zu implementieren.

Ein Tracker der diesen Ansatz realisiert, ist CARPE-ID von Rollo u. a. (2023), welcher zwar aufgrund der Unverfügbarkeit der Implementation nicht zur Evaluation durch diese Arbeit ausgewählt wurde, aber dennoch vielversprechende Ergebnisse im Bereich der personalisierten Roboterassistenz liefert. CARPE-ID basiert auf einer Kombination aus dem Multi-Object Tracker StrongSORT und einem Wiedererkennungssystem, wobei StrongSORT genutzt wird, um aus den Bounding Boxes eines YOLO-Detektionsmodells Tracklets zu erzeugen, während das Wiedererkennungssystem bei Verlust des Zieltracklets versucht, dem Zielobjekt ein neues Tracklet zuzuordnen. Die Besonderheit des Wiedererkennungssystems von CARPE-ID liegt dabei in der kontinuierlichen Anpassung der Zielrepräsentation. In Rollo u. a. (2023) wurde die Robustheit von CARPE-ID in Szenarien mit Ablenkern und teilweiser bis kompletter Verdeckung der Zielperson getestet. Hierzu wurde eine Personfolge-Funktionalität mit CARPE-ID als Tracker auf einem mobilen Roboter implementiert. In allen Szenarien war CARPE-ID in der Lage, die Zielperson korrekt zu tracken und bei Bedarf wiederzuerkennen.

Ein weiterer Ansatz wäre es, die Tracker mit anderen Identifikationslösungen, wie QR-Codes, RFIDs, etc. zu kombinieren. Zusätzlich können Risiken durch die Beschränkung des Einsatzgebietes des Roboters sowie die Anfrage auf manuelle Reidentifikation bei erkannten Unsicherheiten im Tracking reduziert werden.

Literaturverzeichnis

[Aharon u. a. 2022] Aharon, Nir ; Orfaig, Roy ; Bobrovsky, Ben-Zion: *BoT-SORT: Robust Associations Multi-Pedestrian Tracking*. 2022

[Bewley u. a. 2016] Bewley, Alex ; Ge, ZongYuan ; Ott, Lionel ; Ramos, Fabio ; Upcroft, Ben: Simple Online and Realtime Tracking. In: *CoRR* abs/1602.00763 (2016). – URL http://arxiv.org/abs/1602.00763

[Cui u. a. 2022] Cui, Yutao ; Jiang, Cheng ; Wang, Limin ; Wu, Gangshan: *MixFormer: End-to-End Tracking with Iterative Mixed Attention*. 2022

[Cui u. a. 2023] Cui, Yutao ; Song, Tianhui ; Wu, Gangshan ; Wang, Limin: *MixFormerV2: Efficient Fully Transformer Tracking*. 2023

[Dendorfer u. a. 2020] Dendorfer, Patrick ; Ošep, Aljoša ; Milan, Anton ; Schindler, Konrad ; Cremers, Daniel ; Reid, Ian ; Roth, Stefan ; Leal-Taixé, Laura: *MOTChallenge: A Benchmark for Single-Camera Multiple Target Tracking*. 2020

[Kristan u. a. 2016] Kristan, Matej ; Matas, Jiri ; Leonardis, Aleš ; Vojir, Tomas ; Pflugfelder, Roman ; Fernandez, Gustavo ; Nebehay, Georg ; Porikli, Fatih ; Čehovin, Luka: A Novel Performance Evaluation Methodology for Single-Target Trackers. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38 (2016), Nov, Nr. 11, S. 2137–2155. – ISSN 0162-8828

[Liu u. a. 2022] Liu, Chang ; Chen, Xiao-Fan ; Bo, Chun-Juan ; Wang, Dong: Long-term Visual Tracking: Review and Experimental Comparison. In: *Machine Intelligence Research* 19 (2022), Dec, Nr. 6, S. 512–530. – URL https://doi.org/10.1007/s11633-022-1344-1. – ISSN 2731-5398

[Lukežič u. a. 2018] Lukežič, Alan ; Zajc, Luka Čehovin ; Vojíř, Tomáš ; Matas, Jiří ; Kristan, Matej: *Now you see me: evaluating performance in long-term visual tracking*. 2018

[Mayer u. a. 2021] Mayer, Christoph ; Danelljan, Martin ; Paudel, Danda P. ; Gool, Luc V.: *Learning Target Candidate Association to Keep Track of What Not to Track*. 2021

[Moudgil und Gandhi 2019] Moudgil, Abhinav ; Gandhi, Vineet: *Long-Term Visual Object Tracking Benchmark.* 2019

[Rollo u. a. 2023] Rollo, Federico ; Zunino, Andrea ; Tsagarakis, Nikolaos ; Hoffman, Enrico M. ; Ajoudani, Arash: *CARPE-ID: Continuously Adaptable Re-identification for Personalized Robot Assistance.* 2023

[Soleimanitaleb und Keyvanrad 2022] Soleimanitaleb, Zahra ; Keyvanrad, Mohammad A.: *Single Object Tracking: A Survey of Methods, Datasets, and Evaluation Metrics.* 2022

[Valmadre u. a. 2018] Valmadre, Jack ; Bertinetto, Luca ; Henriques, João F. ; Tao, Ran ; Vedaldi, Andrea ; Smeulders, Arnold ; Torr, Philip ; Gavves, Efstratios: *Long-term Tracking in the Wild: A Benchmark.* 2018

[Yan u. a. 2021] Yan, Bin ; Peng, Houwen ; Fu, Jianlong ; Wang, Dong ; Lu, Huchuan: *Learning Spatio-Temporal Transformer for Visual Tracking.* 2021

[Ye u. a. 2022] Ye, Botao ; Chang, Hong ; Ma, Bingpeng ; Shan, Shiguang ; Chen, Xilin: *Joint Feature Learning and Relation Modeling for Tracking: A One-Stream Framework.* 2022

[Zhang u. a. 2022a] Zhang, Pengyu ; Zhao, Jie ; Wang, Dong ; Lu, Huchuan ; Ruan, Xiang: *Visible-Thermal UAV Tracking: A Large-Scale Benchmark and New Baseline.* 2022

[Zhang u. a. 2022b] Zhang, Yang ; Zhou, Yanjun ; Li, Hehua ; Hao, Hao ; Chen, Weijiong ; Zhan, Weiwei: The Navigation System of a Logistics Inspection Robot Based on Multi-Sensor Fusion in a Complex Storage Environment. In: *Sensors* 22 (2022), Nr. 20. – URL https://www.mdpi.com/1424-8220/22/20/7794. – ISSN 1424-8220

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 5. März 2024 Andreas Neumann