

OpenTrafficCam_live

Schlussbericht

Förderkennzeichen platomo: 01F2197A

platomo
planning. tools. mobility.

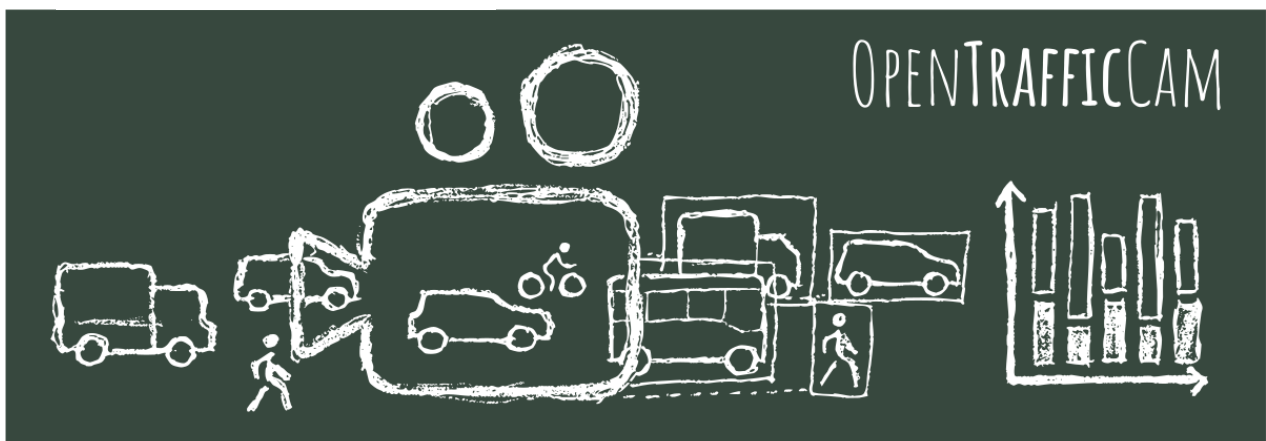
Förderkennzeichen TUD: 01F2197B



Gefördert durch:



aufgrund eines Beschlusses
des Deutschen Bundestages



Inhalt

1	Aufgabenstellung.....	4
2	Vorraussetzungen, unter denen das Vorhaben durchgeführt wurde.....	5
3	Planung und Ablauf des Vorhabens	5
3.1	Planung	5
3.2	Ablauf	6
4	Stand der Wissenschaft, Technik (und des Wettbewerbs)	7
4.1	Gesamtsysteme zur videobasierten Verkehrsanalyse	7
4.2	Videoerfassung	10
4.3	Computer Vision	11
4.4	Verkehrsanalyse.....	15
5	Zusammenarbeit mit Dritten.....	17
5.1	Workshop mit Verkehrsfachleuten zum Thema Datenschutz	17
5.2	Austausch mit fachlich ähnlich gelagerten mFUND-Projekten	18
5.3	Teilnahme an mFUND-Formaten	19
5.4	Öffentlichkeitsarbeit.....	19
6	Verwendung der Zuwendung und erzielttes Ergebnis.....	20
6.1	AP 1 Datenschutz (platomo).....	20
6.2	AP 2 OTCamera (platomo).....	20
6.3	AP 3 (PHOTO).....	25
6.4	AP 4 OTLabels (platomo)	35
6.5	AP 5 OTVision (platomo)	38
6.6	AP 6 OTAnalytics (platomo)	45
6.7	AP 7 OTCloud (platomo)	55
6.8	AP 8 Testfeld (MSP).....	61
6.9	AP 9 Projektmanagement und Öffentlichkeitsarbeit (platomo)	63
7	Zahlenmäßiger Nachweis.....	65
7.1	platomo GmbH.....	65
7.2	TUD (MSP und PHOTO)	66
8	Notwendigkeit und Angemessenheit der geleisteten Arbeit	66
9	Voraussichtlicher Nutzen und Verwertbarkeit.....	67
10	Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen.....	67
11	Veröffentlichungen	69
	Literaturverzeichnis.....	70

1 Aufgabenstellung

Im Projekt OpenTrafficCam_live stand die Weiterentwicklung eines offenen Frameworks zur automatisierten, datenschutzkonformen Erhebung und Analyse von Verkehrsdaten im Vordergrund. In einer mFUND-Vorstudie wurde bereits die Machbarkeit der angestrebten OpenSource Lösung gezeigt. Es existierten daher vor Projektbeginn bereits rudimentäre Prototypen für die Videoerfassung (OTCamera), Detektion und Tracking (OTVision) sowie für die Auswertung (OTAnalytics), die im Rahmen dieses Folgeprojekts weiter verbessert werden sollten.

Im ersten Arbeitsschritt sollte das Kamerasystem OTCamera inklusive der zugehörigen Hardware- und Software-Plattform so optimiert und modularisiert werden, dass es sowohl für den mobilen Kurzzeiteinsatz als auch für Dauerinstallationen eingesetzt werden konnte. Dazu sollten wiederkehrende Performance- und Zuverlässigkeitsverbesserungen implementiert, die automatisierte Installation und Onlinefähigkeit gewährleistet, Weitwinkelmodule integriert sowie alternative Hardwarekomponenten und eine kompakte Systemvariante erprobt worden sein.

Parallel dazu sollte ein umfassendes Datenschutzkonzept erarbeitet werden. Es war vorgesehen, DSGVO-Verzeichnisse zu erstellen, technische Funktionen zur Vermeidung personenbezogener Daten zu integrieren und einen Workshop mit potentiellen Nutzenden und Anwendern im Rahmen des mFUND-Programms durchzuführen.

Ein weiterer zentraler Punkt war die Entwicklung eines Multi-Kamerasystems zur Überwindung von Objektverdeckungen. Hierbei sollte die zeitliche und räumliche Synchronisierung mehrerer Kameras umgesetzt, Testdatensätze inklusive Ground-Truth-Aufnahmen per Drohne erhoben, die Fusion und Mittelung der Trajektorien realisiert und entsprechende Algorithmen in das Modul OTMulticamera integriert werden.

Für die automatische Erkennung und Verfolgung von Verkehrsobjekten sollte eine Bilddatenbank mit mindestens 10.000 Bildern aufgebaut und teilautomatisch annotiert werden. Auf dieser Basis sollten Modelle für Fuß- und Radverkehr sowie für alle gängigen Fahrzeugklassen trainiert, validiert und in OTVision eingebunden werden.

Die Softwarearchitekturen von OTVision und OTAnalytics sollten refaktoriert und modularisiert werden, um den Betrieb auf Hochleistungsrechnern und in GPU-Clustern zu ermöglichen. OTAnalytics sollte um Funktionen zur Verkehrsflussanalyse, Geschwindigkeits- und Wartezeitmessung, Sicherheitskennwerte sowie um interaktive und automatisierte Berichts- und Exportfunktionen erweitert werden.

Für die Orchestrierung der Verarbeitungsschritte sollte OTCloud als zentrale Plattform entstehen, die sowohl On-Premise als auch On-Demand betrieben werden kann. Dies umfasste die Implementierung einer Web-UI zur Jobverwaltung, einer API-gestützten Workflow-Automatisierung, Ausfallschutzmechanismen und CI/CD-Pipelines für die kontinuierliche Integration und Bereitstellung.

Abschließend sollte das Gesamtsystem auf einem realen Testfeld – einem innerstädtischen Knotenpunkt in Dresden – erprobt werden, um die Praxistauglichkeit zu belegen und mittels Ground-Truth-Auswertungen die Detektionsgenauigkeit der Fahrzeugzählung zu verifizieren. Über alle Arbeitspakete hinweg wurden Projektmanagement, Dokumentation und Öffentlichkeitsarbeit die Koordination und Nachvollziehbarkeit der Entwicklungen im Austausch mit Fachkreisen sichergestellt.

2 Vorraussetzungen, unter denen das Vorhaben durchgeführt wurde

Das Vorhaben *OpenTrafficCam_live* baute auf der zuvor abgeschlossenen mFUND-Vorstudie *OpenTrafficCam* auf, in der bereits ein funktionierender Prototyp der Kamerahardware (OTCamera), ein erster Workflow zur Objekterkennung und -verfolgung (OTLabels, OTVision) sowie eine Grundversion der Verkehrsanalysesoftware (OTAnalytics) entwickelt und erfolgreich erprobt worden waren. Ausgangspunkt für das Projekt war die Erkenntnis, dass für eine automatisierte, videobasierte Verkehrserfassung leistungsfähige und frei verfügbare Algorithmen existieren (z. B. YOLOv5), deren Einsatz jedoch bisher hohen manuellen Aufwand oder fehlende technische Infrastruktur voraussetzte. Um diese Algorithmen reproduzierbar und betriebssicher einzusetzen, standen am Konsortium GPU-Cluster an der TU Dresden sowie Workstations mit entsprechender Rechenleistung bereit, auf denen Trainingsläufe für die neuronalen Netze und Echtzeitanalysen durchgeführt werden konnten.

Für die operativen Tests im realen Umfeld wurden Kooperationen mit der Stadt Dresden als Testfeldpartner sowie mit der Bundesanstalt für Straßenwesen (BASt) und mehreren Ingenieurbüros angestrebt, die Genehmigungen, Infrastruktur und fachliche Begleitung sicherstellten. Technisch basierte die OTCamera auf erprobten Low-Cost-Komponenten wie Raspberry Pi Zero W, Raspberry Pi Camera Modul v2 NoIR und einer eigens entwickelten Platine zur Integration von Stromversorgung und Real-Time-Clock, die in einem wetterfesten Gehäuse montiert wurden. Diese Hardwarekonfiguration erlaubte mobile Kurzzeitzählungen ebenso wie kurzfristige Feldinstallationen.

Datenschutzrechtliche Vorgaben nach DSGVO und BDSG bildeten eine weitere zentrale Voraussetzung: Aus der Machbarkeitsstudie ging bereits ein Entwurf für ein FAQ zum Thema Datenschutz hervor, aus dem auch einigen potenziell sinnvollen technischen Maßnahmen (z. B. eine bewusste Defokussierung der Kameraoptik und gering aufgelöste Videoaufzeichnung) identifiziert wurden.

Schließlich bildete die interdisziplinäre Zusammensetzung des Konsortiums – platomo GmbH mit Expertise in Softwareentwicklung und Verkehrsberatung, die Professur für Mobilitätssystemplanung (MSP) (zu Beginn Professur für Integrierte Verkehrsplanung und Straßenverkehrstechnik-IVST) sowie die Professur PHOTO für Photogrammetrie – eine unverzichtbare Voraussetzung. Nur so konnten anspruchsvolle methodische Ansätze wie synchronisierte Mehrkamera-Fusion, 3D-Rekonstruktion von Trajektorien und differenzierte Klassifikation von Fahrzeug-, Fuß- und Radverkehr erfolgreich umgesetzt und in die OpenTrafficCam-Pipeline integriert werden.

3 Planung und Ablauf des Vorhabens

3.1 Planung

Das Vorhaben *OpenTrafficCam_live* war auf eine dreijährige Laufzeit von Januar 2022 bis Dezember 2024 ausgelegt. Die Arbeits- und Zeitplanung sah neun Arbeitspakete vor, die sich über den gesamten Zeitraum verteilten. Datenschutz, OTCamera und OTMultiCamera starteten bereits 2022, während OTCloud und die verkehrliche Analysekomponente OTAnalytics sukzessive bis 2024 aufgebaut wurden.

Die Zielerreichung wurde durch drei Meilensteine abgesichert: Meilenstein 1 zum 15. Dezember 2022, Meilenstein 2 zum 15. Dezember 2023 und der Projektabschluss zum 31. Dezember 2024. Für jeden Meilenstein legte das Konsortium klar umrissene Ergebnisse und abzuschließende Arbeitspakete fest.

Für Meilenstein 1 waren neben den datenschutzrechtlichen Grundlagen vor allem die Bereitstellung der Baupläne für Platine und Gehäuse der OTCamera, die Modularisierung der aus der Vorstudie stammenden Software-Komponenten sowie ein erster mFUND-Fachaustausch vorgesehen. Ergänzend sollten FAQ-Vorlagen zum Datenschutz veröffentlicht und ein Algorithmus zur Key-Feature-Detektion in OTVision integriert sein, begleitet von einer initialen Anforderungsanalyse für die künftige OTCloud.

Meilenstein 2 knüpfte daran an und stellte die funktionale Reife der hardware-nahen Komponenten sowie wesentliche Fortschritte in der Datenverarbeitung sicher. Geplant waren ein Workshop zu Datenschutzfragen, die elektrotechnische Prüfung und Optimierung der OTCamera, die Synchronisierung mehrerer Kameras über OTMultiCamera, die Fertigstellung eines Ground-Truth-Tools zur Parametrisierung der Detektionsalgorithmen, die Annotation umfangreicher Trainingsdatensätze sowie der Aufbau der Backend-Strukturen der OTCloud und die Einrichtung eines Testfelds.

Der abschließende Meilenstein 3 umfasste die vollständige Integration und Übergabe des Gesamtsystems. Dies beinhaltete das Verfahrensverzeichnis samt Datenschutzfolgeabschätzung, die Einbettung der Multi-Camera-Funktionalität in OTVision, die Veröffentlichung kuratierter Trainingsdatensätze, die Portierung von OTVision auf Hochleistungsrechner, die Fertigstellung der verkehrsanalytischen Auswertungen in OTAnalytics sowie die Bereitstellung von Frontend- und Quality-Gate-Funktionen in OTCloud. Zwei Workshops zur Evaluation der Nutzererfahrung sollten den Abschluss begleiten und die gesammelten Erkenntnisse für eine potenzielle Weiterentwicklung bündeln.

3.2 Ablauf

Der tatsächliche Projektverlauf stimmt in den Kernpunkten mit der ursprünglichen Planung überein, weist jedoch in einzelnen Aufgaben Verschiebungen und Erweiterungen auf. Im Jahr 2022 konzentrierten sich die Arbeiten wie vorgesehen auf die Weiterentwicklung der OTCamera, die Vorbereitung des Modelltrainings in OTLabels und erste Optimierungen in OTVision und OTAnalytics. Der erste Meilenstein konnte weitgehend erreicht werden, lediglich der geplante Workshop zum Open-Source-Geschäftsmodell musste auf 2023 verschoben werden.

Im zweiten Projektjahr wurden OTCamera und OTVision in einen marktreifen MVP-Status überführt, OTAnalytics komplett neu implementiert und die Arbeiten an OTCloud aufgenommen. Außerdem wurden Trainingsdaten für Genauigkeitsprüfungen erstellt und der zuvor verschobene Workshop im Dezember 2023 nachgeholt.

Ab 2024 lag der Schwerpunkt auf der Inbetriebnahme und fortlaufenden Weiterentwicklung von OTCloud. Die Plattform ging im Frühjahr 2024 produktiv, erhielt Quality-Gates zur Vor- und Nachprüfung der Videoverarbeitung und wurde hinsichtlich Performance und Skalierbarkeit optimiert. Parallel dazu wurden die Hardware- und Softwarekomponenten der OTCamera weiter verbessert, was zu einer deutlichen Steigerung der Einsatzzahlen im Feld führte. Somit konnte das Konsortium den geplanten Arbeitsumfang trotz einzelner terminlicher Anpassungen vollständig umsetzen und die Zielerreichung kontinuierlich absichern.

Hinsichtlich der Meilensteine lässt sich zusammenfassend feststellen, dass der erste Meilenstein zum 15. Dezember 2022 bis auf den verschobenen Workshop vollständig erfüllt wurde. Die fehlende Veranstaltung beeinflusste den weiteren Projektfortschritt nicht, da alle technischen Ergebnisse – darunter die Baupläne der OTCamera und die ersten Datenschutzvorlagen – termingerecht

vorlagen. Der zweite Meilenstein zum 15. Dezember 2023 wurde nach Nachholung des Workshops und Abschluss aller vorgesehenen Entwicklungsarbeiten im Bereich Datenschutz, Multikamerasystem und Cloud-Backend erreicht; lediglich die Feinabstimmung der OTCloud-Oberfläche wurde in das Folgejahr verlagert. Der abschließende Meilenstein zum 31. Dezember 2024 konnte fristgerecht realisiert werden, da OTCloud intern produktiv geschaltet, OTAnalytics funktionsfähig integriert und die Multi-Camera-Fusion in OTVision übergeben wurden; damit liegen alle im Antrag definierten Endergebnisse vor, sodass das Gesamtvorhaben ohne Terminüberschreitung abgeschlossen werden konnte.

4 Stand der Wissenschaft, Technik (und des Wettbewerbs)

Videobasierte Verkehrsanalyse benötigt in der Regel ein End-to-End-System von der Videoerfassung über die automatisierte Auswertung bis zur Gewinnung verkehrsrelevanter Kennwerte. Traditionell mussten Verkehrsingenieure Videoaufnahmen aufwändig manuell auszählen, doch moderne Ansätze nutzen KI-gestützte *Computer Vision*, um Fahrzeuge, Fußgänger und Radfahrende automatisch in Videos zu erkennen und zu verfolgen. Ein solches Gesamtsystem besteht typischerweise aus der Hardware (z. B. vernetzte Kamera(s) mit Aufzeichnungseinheit) und einer Software-Pipeline für Objekterkennung, -verfolgung und Datenanalyse.

4.1 Gesamtsysteme zur videobasierten Verkehrsanalyse

Der Markt für videobasierte Verkehrsanalyse ist in den letzten Jahren stark gewachsen und umfasst sowohl proprietäre kommerzielle Systeme als auch einige wenige Open-Source-Lösungen. Im Folgenden werden einige wichtige nationale und internationale Akteure und Projekte vorgestellt und hinsichtlich Funktionsumfang, Lizenzmodell, API-Verfügbarkeit und Datenqualität verglichen. Außerdem werden aktuelle Markttrends und Lücken beleuchtet.

4.1.1 Anbieter

4.1.1.1 DataFromSky (RCE Systems, Tschechien)

DataFromSky ist eine umfangreiche Plattform für Verkehrsanalysen aus Videos, entstanden ursprünglich aus der Auswertung von Drohnenaufnahmen [1]. Inzwischen bietet das Unternehmen mit TrafficSurvey einen Cloud-Dienst zur Videoprozessierung inkl. Offline-Lösungen zur Analyse der prozessierten Videos und mit FLOW eine Echtzeit-Lösung für Smart Cities. Der Funktionsumfang ist sehr breit: automatisierte Zählungen (auch O/D-Matrizen), Trajektorien und Geschwindigkeiten, Verkehrsfluss- und Sicherheitsanalysen (Erkennen von Beinahe-Unfällen), Parkplatz- und Parkraumüberwachung, sowie sogar Detektion von Verkehrsverstößen (Geisterfahrer, Rotlicht). DataFromSky setzt auf KI-Modelle mit Edge-Prozessierung und bietet auch eigene Hardware an (TrafficCamera mit eingebauter AI). Die Plattform ist proprietär; Ergebnisse können über offene Schnittstellen (z. B. JSON/REST-API) integriert.

4.1.1.2 Miovision (Kanada)

Miovision ist ein etablierter Anbieter, spezialisiert auf Video-basierte Verkehrsdaten für Verkehrsplanung und Signalsteuerung [2]. Kernprodukt war der Miovision Scout, eine mobile Kameraeinheit zur manuellen Aufstellung, die 24-48h-Verkehrszählungen durchführt. Kunden laden die Videos in eine Cloud hoch, wo sie automatisch ausgewertet werden (Knotenstromzählungen, Klassifizierungen etc.). Miovision gibt eine Genauigkeit von ~95 % für Klassenzählungen an. Neben Offline-Studien hat Miovision mit TrafficLink auch ein System für Dauerinstallationen an Ampeln, welches Kamera-Detektion für die Lichtsignalsteuerung (Ersatz für Induktionsschleifen) nutzt. Miovision's Funktionsumfang umfasst hauptsächlich: Kfz-/Rad-/Fußgänger-Zählungen (auch gleichzei-

tig), Signalzeitenoptimierung (über Detektion von Stau und Räumzeiten), sowie seit neuestem Sicherheitsanalysen. Letztere beruhen auf Trajektorien aus den Videos: Safety Studies erkennt automatisch Konflikte und berechnet Surrogat-Metriken (TTC, PET) mit dem Versprechen, Unfallwahrscheinlichkeiten zu prognostizieren. Miovision bietet eine Web-Plattform, auf der die Ergebnisse visualisiert und exportiert werden können. Das Lizenzmodell ist proprietär, oft in Form von Service-Verträgen oder Datenauswertungs-Paketen; Hardware (Scout) wird verkauft oder vermietet. Für Entwickler gibt es eingeschränkte API-Zugänge (Miovision Datalink), aber die Plattform ist nicht offen. Miovision ist besonders in Nordamerika verbreitet, viele Kommunen nutzen es für temporäre Zählungen an Knoten. In Deutschland ist Miovision ebenfalls stark im Markt vertreten. Die Flexibilität der Datenauswertung (z. B. eigene Analysen oder Rohdatenzugriff) ist gering – man erhält vordefinierte Berichte. Datenschutz ist insofern adressiert, als Miovision die Videos nach Auswertung wieder löscht und nur aggregierte Daten bereitstellt, aber es bleibt eine Auslands-Datenverarbeitung.

4.1.1.3 *Viscando (Schweden)*

Viscando bietet eine etwas andere Lösung – ihr Produkt (Viscando Insight) beruht auf stereoskopischen Kamerasensoren, die oberhalb von Knotenpunkten installiert werden [3]. Durch Stereovision kann das System 3D-Trajektorien in einem Bereich (z. B. Kreuzung oder Straßenabschnitt) erfassen, ohne externe Kalibrierung. Das System ist aber stationär und deckt einen begrenzten Bereich ab (ca. 35x50m Radius). Im Vergleich mit OpenTrafficCam_live ist Viscando ein spezialisierter Sensor, während OTC ein Framework ist. Viscando ist ideal, wenn eine Stadt wenige kritische Knoten hat, an denen man dauerhafte, präzise Überwachung möchte, aber für eine breite Einsetzbarkeit an vielen Orten wäre es kostenintensiv.

4.1.1.4 *DataCollect (Argos) (Deutschland)*

DataCollect Traffic Systems ist ein deutscher Anbieter von Verkehrszähltechnik, der mit ARGOS seit kurzem ein KI-basiertes Videosystem im Portfolio hat [4]. ARGOS mobile ist eine mobile Video-Erfassungseinheit mit KI-Auswertung direkt vor Ort („Edge AI“). Damit ähnelt es dem Miovision-Scout, wirbt aber mit „Made in Germany“ und DSGVO-Konformität. Der Funktionsumfang umfasst multimodale Verkehrszählungen und -klassifizierungen, Erkennung von Geschwindigkeitsverteilungen und auch das Berechnen von Abbiegevorgängen (laut Produktinfo). DataCollect integriert die Ergebnisse in die hauseigene Plattform myTrafficData, wo Kunden die Daten aufbereitet abrufen können. Das Lizenzmodell ist proprietär, Argos kann gemietet oder gekauft werden; Auswertungssoftware als Service. Eine API ist nicht öffentlich dokumentiert, jedoch können Ergebnisse als Standard-Reports exportiert werden (Excel, PDF etc.). Da DataCollect aus der klassischen Zählbranche kommt, liegt der Fokus eher auf der Auswertung von Zählwerten als auf Trajektorien – eine offene Bereitstellung der gemessenen Trajektorien ist nicht vorgesehen, wodurch tiefergehende Analysen (Sicherheitsindikatoren) nicht direkt vom Nutzer durchgeführt werden können. Eine Stärke von DataCollect ist die Integration mit anderen eigenen Produkten (z. B. stationäre Dauerzählstellen, Radar), was ein komplettes Paket ergeben kann. Für OpenTrafficCam_live stellt DataCollect einen Wettbewerber im heimischen Markt dar, der mit DSGVO-Konformität und einfacher Nutzbarkeit punktet, aber als proprietäres System weniger flexibel ist.

4.1.1.5 *OpenDataCam (Open-Source, global)*

OpenDataCam ist eine frei verfügbare Software (MIT-Lizenz) zur Echtzeit-Verkehrszählung mittels Objekterkennung. Sie richtet sich an Maker, Smart-City-Enthusiasten und Forscher, die kostengünstig eigene Videoanalysen durchführen wollen. Der Funktionsumfang bietet die Detektion und

Tracking von Objekten mit vortrainierten YOLO-Modellen, Zählung über definierbare Linien/Zonen, Ausgabe von Trajektorien und Zählungen über eine Web-Oberfläche oder API. OpenDataCam unterstützt mehrere Objektklassen (Standard: Fahrzeug, Fahrrad, Person, etc.) out-of-the-box. Es kann sowohl auf Live-Kamera-Streams als auch auf Videodateien angewendet werden. Die Software läuft auf Edge-Geräten wie NVIDIA Jetson oder auch auf PCs mit CUDA-GPU. Das Lizenzmodell ist Open Source (Quellcode auf GitHub, kostenlos). Es gibt eine REST-API, über die z. B. der Start/Stop, das Abrufen der Zählerstände oder das Auslesen der Trajektorien möglich ist. Das letzte offizielle Release ist aus dem Jahr 2021. Seit 2022 wurden nur noch sehr wenige Änderungen am Code vorgenommen. Ein großer Pluspunkt ist die Datenhoheit: Alle Daten bleiben auf dem Gerät des Nutzers, es findet keine Cloud-Weiterleitung statt, was datenschutztechnisch vorteilhaft ist.

4.1.1.6 *Traffic Intelligence / ICTCT Tools (Open-Source, akademisch)*

Unter dem Namen Traffic Intelligence existiert ein Open-Source-Projekt aus Kanada (Polytechnique Montréal), das eine Bibliothek für Verkehrsvideoanalyse bietet. Es umfasst Feature-basierte Tracking-Algorithmen, Klassifizierung von Verkehrsteilnehmern und die Ausgabe von Trajektorien zur weiteren Analyse (insb. für Sicherheitsstudien). Darüber hinaus gibt es Tools wie T-Analyst (Lund University) für halbautomatische Konfliktauswertung und STRUDL (aus dem EU-Projekt In-DeV) für Deep-Learning-basiertes Tracking mit Web-GUI. Diese akademischen Werkzeuge sind meist frei verfügbar (Traffic Intelligence z.B. unter GPL) und erlauben Forschern, eigene Skripte auf die Daten anzusetzen. Allerdings erfordern sie oft intensiven manuellen Eingriff oder spezielles Know-how, um sie zu bedienen. Beispielsweise ist Traffic Intelligence eher eine Code-Bibliothek als eine schlüsselfertige Software. Die Lizenz ist Open Source (diverse Lizenzen, meist GPL/BSD). Die Datenqualität variiert je nach Tool – T-Analyst und Co. setzen auf manuelle Klassifikation oder klassische Bildverarbeitung, was nicht die Genauigkeit moderner KI erreicht, aber für Sicherheitsanalysen (die sehr hohe Positionsgenauigkeit erfordern) nützlich sein kann. Eine Marktlücke, die OpenTrafficCam_live füllt, ist gerade die Überführung solcher fortgeschrittenen Forschungsansätze in ein benutzerfreundliches, praxisnahes Werkzeug für Verkehrsingenieure.

4.1.1.7 *Weitere Lösungen*

Neben den genannten gibt es noch kommerzielle Angebote wie GoodVision (UK, Cloud-Auswertung von hochgeladenen Videos mit Fokus auf schnelle Turnaround-Zeit und einfacher Benutzeroberfläche), IntuVision Traffic (USA, lizenzierbare Software für eigene Kameranetzwerke), oder Briefcam (eigentlich Videoüberwachungsanalyse, aber teils auch für Verkehrsflüsse genutzt). Diese verfolgen ähnliche Ansätze – proprietär, mit KI-Erkennung und übersichtlichen Dashboards.

4.1.2 Markttrends

Ein klarer Trend ist die Echtzeit-Analyse und Integration in Smart-City-Systeme. Städte wollen Verkehrsflüsse live überwachen und steuern – videobasierte Detektion wird zum Bestandteil von adaptiven Verkehrssteuerungen oder Informationssystemen. Dafür werden edge-basierte Lösungen favorisiert, da sie Latenz und Datenschutz verbessern (KI direkt in der Kamera oder am Knoten). DataFromSky's Edge Produkte oder Miovision's SmartView folgen diesem Trend. Ein zweiter Trend ist multimodale Sicherheitsanalyse: Aufgrund von Vision Zero Initiativen steigt die Nachfrage nach Konflikt- und Gefahrenerkennung, insbesondere um die Sicherheit für Fußgänger und Radfahrer zu erhöhen. Systeme, die automatisch Rotläufer an Lichtsignalanlagen, Beinahe-Unfälle oder Geschwindigkeitsübertretungen erkennen, gewinnen an Interesse. Das erfordert jedoch eine sehr zuverlässige Erkennung und Tracking – ein Feld in dem weiterhin Entwicklung nötig ist, was

eine Marktlücke darstellt: Nicht alle Anbieter können valide Sicherheitsdaten liefern, hier ist weiteres Potenzial (OpenTrafficCam_live könnte sich profilieren, wenn es z.B. valide TTC-Messungen open-source ermöglicht).

Datenschutz & Akzeptanz sind ebenfalls marktprägend in Europa. Lösungen, die Privacy by Design umsetzen (z. B. lokale Anonymisierung, Verpixelung von Gesichtern/Nr.-Schildern, keine Personendaten speichernd), haben einen Vorteil. OpenTrafficCam_live adressiert dies ausdrücklich. Viele kommerzielle Systeme argumentieren zwar, dass sie keine personenbezogenen Daten ausgeben – aber die Rohvideos existieren oft dennoch. Hier könnte ein Open-Source-Projekt vertrauensbildender wirken, da die Methoden transparent sind.

Zusammenfassend lassen sich einige Lücken identifizieren, die OpenTrafficCam_live füllen will: (1) Open-Source-Plattform für Verkehrsanalysen – bisher gab es keine frei verfügbare, umfassende Software, nur Teil-Tools (OpenDataCam, TrafficIntelligence) oder kommerzielle Produkte. Eine solche Plattform ermöglicht Behörden und Forschung unabhängiger zu agieren und maßgeschneiderte Lösungen zu entwickeln. (2) Kostengünstige Multi-Kamera-Lösung – aktuell gibt es kaum Produkte, die eine Multi-Cam-Trajektorienfusion offen anbieten; vorhandene Systeme sind entweder Single-Cam (die meisten) oder proprietär (z. B. Flir/Bosch haben multi-sensor Lösungen, aber geschlossen). (3) Lokale Datenspeicherung und -verarbeitung als Feature – in Zeiten der Cloud ist dies eine Nische: OpenTrafficCam könnte On-Premise in Kommunen laufen, die Cloud nur optional. (4) Anpassbarkeit und Community – viele Verkehrsanwender haben Spezialfragen (etwa Parkdauererhebung auf Straßenabschnitten). Derzeit müssen sie nutzen, was kommerzielle Software bietet, oder selbst programmieren. Ein offenes Framework erlaubt Plugins oder Erweiterungen für solche Use-Cases. So sind z.B. auch Parkraumanalysen (Belegungsrate, Standzeiten) als Auswertung einfach möglich, was in der breiten Masse der Konkurrenz noch kein Standard ist.

Es existieren ausgereifte proprietäre Lösungen, die jedoch meist teuer und wenig transparent sind. Open-Source-Angebote waren bisher entweder nicht anwenderfreundlich oder funktional begrenzt. OpenTrafficCam_live positioniert sich dazwischen – als offenes, aber professionell entwickeltes Framework, speziell zugeschnitten auf die Bedürfnisse der Verkehrsplanung in Deutschland. Langfristig könnte durch die offene Natur zudem eine breite Nutzer-Community entstehen, die das System stetig verbessert und an neue Trends anpasst – ein Vorteil, den geschlossene Lösungen nicht ohne weiteres bieten können.

4.2 Videoerfassung

Die Videoerfassung bildet die Grundlage der Verkehrsanalyse. Zum Einsatz kommen entweder vorhandene Verkehrsüberwachungskameras, speziell entwickelte Sensorsysteme oder mobile Aufnahmeeinheiten. OpenTrafficCam_live setzt auf ein eigenes Kamerasystem (*OTCamera*), das auf handelsüblichen Low-Cost-Komponenten basiert und für einfache Nachbaubarkeit ausgelegt ist. Die Hardware wurde gegenüber der Vorstudie weiterentwickelt, inklusive einer neuen Elektronikplatine und robustem Gehäuse, sodass alle Teile online beschaffbar sind. Die Kamera ist modular erweiterbar, um z. B. Weitwinkel-Optiken einsetzen zu können. Externe Fachleute unterstützen bei notwendigen Prüfungen und Optimierungen der Hardware, damit ein zuverlässiger Dauerbetrieb möglich ist.

Ein zentrales Thema beim Einsatz mehrerer Kameras ist die Synchronisation. Das bedeutet, die Uhren der Kameras müssen abgeglichen und ihre Positionen/Orientierungen im Raum kalibriert werden, sodass Ereignisse im Video zeitgleich und perspektivisch korrekt zusammengeführt werden können. Multi-Kamera-Systeme können insbesondere bei komplexen Verkehrssituationen

(z. B. viele verdeckte Sichtbereiche) die Datengüte steigern, indem sie Lücken in Trajektorien füllen und 3D-Informationen liefern. Entsprechend wird in OpenTrafficCam_live eine Fusion der Objekttrajektorien aus verschiedenen Kameras implementiert.

Neben stationären Kameras werden in der Praxis auch mobile Perspektiven genutzt. Beispielsweise erlauben Drohnenvideos eine Vogelperspektive, die besonders für die Erfassung von Trajektorien auf Strecken und Knoten geeignet ist. Solche Luftaufnahmen liefern sehr detaillierte Trajektorien Datensätze. Eine Herausforderung bei Drohnen- und schrägen Kameraaufnahmen sind jedoch Verdeckungen und variable Perspektiven, die robust ausgeglichen werden müssen (siehe 4.3.3) sowie die mögliche Aufnahmedauer.

Schließlich spielt die Bildqualität eine Rolle: Hohe Auflösung und Bildraten ermöglichen präzisere Analysen, erhöhen aber Datenmengen und Rechenaufwand. In der Praxis genügen oft 20–30 FPS und 720p-Auflösung (in der Regel auch deutlich weniger), um Geschwindigkeiten und Abstände ausreichend genau zu bestimmen. Für Nachtaufnahmen können Infrarot-beleuchtete Kameras oder Thermalbildkameras eingesetzt werden, um auch bei Dunkelheit Fahrzeuge und Personen zu erkennen – dies erfordert jedoch geeignete KI-Modelle. Insgesamt muss das Erfassungssystem so ausgelegt sein, dass es den Anforderungen der Verkehrsdatenerhebung (Wetter, Vibrationen, Dauerbetrieb) standhält und gleichzeitig datensparsam arbeitet, um Datenschutz und Übertragungsbandbreite zu berücksichtigen.

4.3 Computer Vision

4.3.1 Objekterkennung

Die Objekterkennung identifiziert relevante Verkehrsteilnehmende in den Videobildern und markiert sie typischerweise durch begrenzende Rechtecke (*Bounding Boxes*) samt Klassifikation (z. B. als Pkw, Lkw, Fahrrad oder Fußgänger). Seit 2020 haben sich hier vor allem Deep-Learning-Verfahren durchgesetzt, die auf neuronalen Netzen basieren. Ein großer Vorteil ist, dass vortrainierte Modelle – etwa auf dem COCO-Datensatz – frei verfügbar sind und für spezifische Verkehrsdomänen adaptiert werden können. Die OpenTrafficCam-Vorstudie nutzte bereits ein vortrainiertes YOLOv5-Modell für erste Detektionen und im aktuellen Projekt werden eigene Datensätze aufgebaut, um Modelle gezielt auf deutsche Verkehrsbedingungen nachzutrainieren.

Technisch unterscheidet man grob Two-Stage-Detektoren und One-Stage-Detektoren. Zwei-Stufen-Modelle wie *Faster R-CNN* [5] zerlegen die Aufgabe in Regionenvorschläge und anschließende Klassifikation, was sehr präzise Ergebnisse liefert, aber mit höherer Latenz. Einfachere *Single-Shot*-Modelle wie *SSD* [6] oder die YOLO (You Only Look Once)-Familie erledigen Erkennung und Lokalisation in einem Durchlauf und erreichen dadurch hohe Geschwindigkeiten. Insbesondere die YOLO-Modelle haben seit 2016 die Echtzeit-Objekterkennung revolutioniert. Aktuelle Iterationen wie YOLOv5 [7] und YOLOv7 [8] kombinieren die Echtzeitfähigkeit mit einer Genauigkeit, die früher nur Two-Stage-Ansätze erreichten. So wurde berichtet, dass YOLO-basierte Systeme auch unter schwierigen Bedingungen (z. B. Nacht, Regen) über 97 % mAP (mittlere durchschnittliche Präzision) für Fahrzeugklassen erzielen können [8]. YOLOv7 gilt 2023 als eines der leistungsfähigsten Modelle und wurde speziell auch für kleinere Objekte und komplexe Szenen optimiert [9]. Im Januar 2023 erschien YOLOv8, das weitere Verbesserungen in Architektur und Trainingsoptimierung bietet [10] und den Einsatz noch anwenderfreundlicher gestaltet.

Für die Verkehrsanalyse ist neben der reinen Detektionsrate auch die Klassifizierung der Objekte in korrekte Kategorien wichtig, etwa um PKW von LKW oder Fahrräder von Motorrädern zu unter-

scheiden. Klassische Netze werden oft auf COCO- oder Cityscapes-Daten trainiert, die grobe Kategorien enthalten. OpenTrafficCam_live adressiert dies, indem eigene Bilder unter Berücksichtigung der FGSV-Klassifizierung (deutsche Standard-Fahrzeugklassen) annotiert und verwendet werden. Dadurch sollen z. B. Lieferwagen vs. LKW oder E-Scooter vs. Fahrrad erkannt werden können, was mit COCO-Standardmodellen nicht zuverlässig möglich ist. Erste Transfer-Learning-Experimente haben gezeigt, dass das Nachtrainieren (Fine-Tuning) eines YOLOv5-Modells auf solche spezialisierten Klassen die Erkennungsgenauigkeit deutlich steigern kann (vgl. Jakubec u. a. 2025 für ein ähnliches Vorgehen an städtischen Knoten). Darüber hinaus werden schwierige Szenarien adressiert: Teilverdeckte Fußgänger oder Radfahrer an unübersichtlichen Stellen. Hierzu werden z. B. gezielt Trainingsbilder mit Verdeckungen eingesetzt. Forschungsseitig gibt es Ansätze mit ergänzenden Modulen wie z. B. YOLOv4-CSP in Kombination mit Infrarot für Nacht [12] oder YOLOX mit speziellen Datenaugmentierungen für Regen und Schnee [13], um die Robustheit weiter zu erhöhen. Insgesamt ist der Stand der Technik bei der Objekterkennung so weit fortgeschritten, dass unter Laborbedingungen sehr hohe Erkennungsraten möglich sind – die Herausforderung besteht darin, diese Performance in heterogenen Realweltszenarien (verschiedene Kamerawinkel, Wetter, Stadt vs. Landstraße) aufrechtzuerhalten. Hier kommen dann die im Projekt vorgesehenen Validierungsmaßnahmen ins Spiel, z. B. automatische Qualitätsprüfungen der erkannten Trajektorien und (teil-)automatisierte Anpassungen von Detektionsparametern. Dies soll sicherstellen, dass die Erkennung im Feld die gewünschte Genauigkeit erreicht und Schwächen früh erkannt werden.

4.3.2 Objektverfolgung

Auf die Erkennung folgt die Objektverfolgung (*Multi-Target Tracking*), welche die einzelnen Detektionen über die Zeit zu kohärenten Trajektorien verknüpft. Jede erfasste Bewegung eines Verkehrsteilnehmers soll als Trajektorie (d. h. zeitlicher Pfad mit Positionen pro Frame) abgebildet werden, um daraus z. B. Fahrtstrecken, Verweildauern oder Konflikte abzuleiten. Technisch handelt es sich meist um *Tracking-by-Detection*-Verfahren: In jedem neuen Videobild werden Objekte detektiert (siehe 4.3.1) und anschließend einem bestehenden Track zugeordnet oder als neuer Track initialisiert. Dabei stellen Probleme wie Fehldetektionen, kurze Verdeckungen oder dicht gedrängte Objekte eine Herausforderung dar. Der Stand der Technik hat hier mehrere Lösungsansätze hervorgebracht:

- IOU Tracker [14]: IOU-Tracker gelten durch ihren simplen Aufbau als besonders schnell in der Ausführung, allerdings auch als fehleranfällig. Je größer die Genauigkeit der Detektion und je höher die Framerate des zugrunde liegenden Videomaterials, desto besser schneiden IOU-Tracker ab. Das Funktionsprinzip eines IOU-Trackers ist folgendermaßen: Für zwei aufeinander folgende Frames werden die Bounding Boxes übereinander gelegt und der Quotient aus ihrer Schnittmenge und Vereinigungsmenge gebildet. Liegt der IOU-Wert oberhalb eines Schwellwerts, wird die Trajektorie erweitert.
- SORT (Simple Online Real-time Tracking) [15]: Ein minimalistischer Ansatz, der pro Objekt einen Kalman-Filter zur Bewegungsschätzung und die Zuweisung neuer Detektionen anhand der Bounding-Box-Überschneidung (IoU) vornimmt. SORT erreicht mit sehr hoher Geschwindigkeit akzeptable Ergebnisse, versagt aber wenn Objekte einander verdecken oder das Erscheinen/Verschwinden nicht gut modelliert wird.
- DeepSORT [16]: Erweiterung von SORT um ein visuelles Wiedererkennungsmodell (ReID). Zusätzlich zur Positionsprognose wird hier für jedes Objekt ein tiefes Merkmal (Feature-Vector) berechnet, um es über mehrere Frames wiederzuerkennen. Damit können auch

bei kurzzeitigem Verschwinden (z. B. wenn ein Auto hinter einem LKW durchfährt) die Identitäten besser erhalten bleiben. DeepSORT gilt seit ca. 2017 als bewährter Tracker und wird auch in OpenDataCam eingesetzt. Allerdings kann das ReID-Modul die Laufzeit erhöhen und ist empfindlich gegenüber stark wechselnden Blickwinkeln.

- Tracktor [17]: Dieser Ansatz verfolgt Objekte, indem er direkt das Bounding-Box-Regressionmodul eines Objektdetektors "zweckentfremdet". Statt eigene Motion-Modelle zu nutzen, lässt Tracktor den Detektor vorhersagen, wo bereits erkannte Objekte im nächsten Frame liegen müssten. Dadurch wird das Tracking stark vereinfacht – solange der Detektor robust ist. Tracktor zeigte, dass mit einem guten Detektor (wie Faster R-CNN) auf Tracking-Benchmarks neue Bestleistungen erreicht werden können, ohne speziell für Tracking zu trainieren. Allerdings gerät Tracktor an Grenzen, wenn mehrere ähnlich aussehende Objekte nahe beieinander sind, da dann Verwechslungen auftreten können (kein explizites ID-Merkmal außer Position).
- Neuere Tracker: In den letzten Jahren sind noch weitere State-of-the-Art-Tracker erschienen. Zu nennen ist etwa FairMOT [18], der in einem gemeinsamen Netzwerk sowohl Detektion als auch ReID-Features lernt, um ein faires Gleichgewicht zwischen beidem zu erreichen. Sehr erfolgreich war auch ByteTrack, der einen simplen aber effektiven Trick nutzt: Er berücksichtigt bei der Assoziation *alle* Detektionen, auch solche mit niedrigem Score, anstatt – wie üblich – schwache Detektionen zu verwerfen. ByteTrack konnte damit die MOT-Challenge-Benchmarks dominieren und wird inzwischen in vielen Wettbewerbsbeiträgen verwendet [19]. Im OpenTrafficCam_live-Kontext ist ByteTrack interessant, da es robust gegenüber schwankenden Detektionsscores ist – etwas, das bei unterschiedlichen Verkehrsszenen (Innenstadt vs. Landstraße) auftreten kann.

OpenTrafficCam_live plant, mehrere Tracking-Frameworks modular einzubinden. Für einfache Zählungen kann z. B. ein ressourcensparender IOU-Tracker laufen, während für detaillierte Trajektorienanalysen DeepSORT oder ByteTrack genutzt werden kann. Zudem können nach erfolgter Objektverfolgung Verfahren eingesetzt werden, unterbrochene Trajektorien zu identifizieren und zu vervollständigen. Dies bedeutet, dass wenn ein Objekt zeitweise nicht erkannt wurde (z. B. Verdeckung oder Detektionslücke), das System versucht, die Lücke zu füllen – etwa durch Interpolation oder das Zusammenführen von zwei Fragmenten, die wahrscheinlich dasselbe Objekt darstellen. Solche Verfahren erhöhen die Vollständigkeit der Daten und basieren teils auf Heuristiken (plausible Geschwindigkeiten, durchgängige Richtungen) oder KI-Modellen, die Klassifikationen ganzer Trajektorien vornehmen.

Ein Spezialfall der Objektverfolgung ist die Multi-Kamera-Verfolgung. Wenn ein Verkehrsteilnehmer den Blickbereich einer Kamera verlässt und in einer anderen erscheint, soll idealerweise seine Trajektorie dennoch durchgängig erfasst werden. Dafür müssen die Tracks zwischen Kameras verknüpft werden. Trajektorien aus verschiedenen Kameras werden hierzu in einem gemeinsamen Weltkoordinatensystem zusammengeführt. Voraussetzung ist eine exakte zeitliche Synchronisation (siehe 4.2) und geometrische Kalibrierung (siehe 4.3.3). Ist diese gegeben, kann z. B. ein Fahrzeug, das aus Kamera A herausfährt, in Kamera B nahtlos weiterverfolgt werden, solange die Weltposition konsistent ist. Ansätze aus der Literatur [20], [21], [22], [23], [24] nutzen hierbei häufig die *ReID*-Merkmale über Kameras hinweg (personenübergreifende Wiedererkennung in Kameranetzwerken) – im Verkehrskontext könnte man z. B. Farb- und Typ-Merkmale eines Autos über Kameras vergleichen. OpenTrafficCam_live verfolgt den *geometrischen* Ansatz: Durch Fusion aller Kamera-Messungen auf gemeinsame Koordinaten übernimmt quasi das *Mapping* die Verknüpfung. Dadurch reduziert sich das Problem auf räumliche Nähe und zeitliche Übereinstimmung im

Weltmodell, was robuste Matches erlaubt, ohne direkt visuelle IDs vergleichen zu müssen. Dieser Bereich ist Gegenstand aktueller Forschung, da Multi-Kamera-MOT in offenen Szenarien (ohne kontrollierte Umgebung) nach wie vor schwierig ist.

4.3.3 Korrektur von Verzeichnung und Verzerrung

Für metrisch korrekte Analysen müssen geometrische Bildfehler korrigiert werden. Als *Verzeichnung* bezeichnet man i. Allg. die durch die Kameralinse verursachten Abbildungsfehler (v.a. radial/tangential), während *Verzerrung* hier im Sinne der perspektivischen Verzerrung verstanden werden kann. Beide Effekte bewirken, dass Längen- und Winkelmessungen im Bild nicht ohne Weiteres der Realität entsprechen. Im Verkehrsbereich ist dies kritisch, wenn z. B. aus Pixelverschiebungen Geschwindigkeiten in km/h berechnet oder Abstände in Metern geschätzt werden sollen. OpenTrafficCam_live entwickelt daher Methoden zur Kamerakalibrierung und Entzerrung, die integraler Bestandteil des Systems sind.

4.3.3.1 Verzeichnung

Unter Verzeichnung versteht man optische Abbildungsfehler der Kameralinse, insbesondere tonnenförmige oder kissenförmige Verzeichnung, wie sie bei Weitwinkel- oder Fischaugenobjektiven auftreten. Gerade Linien im Realraum erscheinen dabei gekrümmt im Bild. Für genaue räumliche Berechnungen muss dieser Fehler korrigiert werden. Üblicherweise wird dazu eine Kamerakalibrierung nach dem Modell von Brown [25] durchgeführt: Man filmt ein Kalibrieremuster (z. B. Schachbrett) und berechnet Linsenparameter (radiale und tangentielle Distortion-Koeffizienten). Mit diesen kann jedes Video frameweise *entzerrt* werden, sodass die Pixelkoordinaten wieder einer idealen pinhole-Kamera entsprechen. Da OpenTrafficCam modulare Kameras vorsieht (unterschiedliche Aufsatzlinsen, Fischaugen für 360°-Blick etc.), muss die Kalibrierung für mehrere Module bereitgestellt werden. Im Ergebnis soll der Nutzer später für ein gegebenes Kamera-Modell entweder vordefinierte Kalibrationsparameter laden oder eine einfache Kalibrierprozedur (z. B. mit einem Schachbrett am Aufstellort) durchführen können. Damit werden *Verzeichnungsartefakte* wie gebogene Straßenränder im Bild behoben.

Für sehr große Weitwinkel (180°) ist eine perfekte Entzerrung allerdings oft nicht möglich, ohne Bereiche am Bildrand zu verlieren oder Unschärfen zu erhalten. Hier kann man stattdessen auch spezielle Objektivmodelle in die späteren Auswertelgorithmen integrieren – z. B. wurden in Forschungsprojekten bereits angepasste geometrische Modelle für Low-Cost-Fisheye-Kameras entwickelt [26]. Die Notwendigkeit dieser Korrekturen zeigt sich vor allem, wenn genaue Trajektorien in Weltkoordinaten gebraucht werden: Eine nicht entzerrte Optik könnte sonst zu systematischen Positionsfehlern führen (etwa Fahrzeuge, die am Bildrand fahren, würden näher zur Kamera scheinbar beschleunigt). Bereits in der Vorstudie wurde eine grundlegende Verzeichnungs-Korrektur im Prototyp implementiert; nun soll diese auf alle Kameravarianten erweitert und automatisiert werden.

4.3.3.2 Verzerrung

Verzerrung beschreibt die perspektivische Verzerrung bzw. Maßstabsverzerrung, die dadurch entsteht, dass die Kamera in der Regel in schräger Position zum Verkehrsraum ausgerichtet ist. Objekte erscheinen je nach Entfernung zur Kamera kleiner, und Entfernungen im Raum projizieren sich verkürzt ins Bild. Für die Verkehrsanalyse muss dies korrigiert bzw. berücksichtigt werden, um beispielsweise aus Pixelverschiebungen pro Frame eine reale Geschwindigkeit zu berechnen. Die klassische Lösung ist eine Homographie-Transformation: Ist die Kamera intern kalibriert und man kennt die Lage einer ebenen Straßenoberfläche relativ zur Kamera, so kann man das Videobild auf die Vogelperspektive dieser Ebene umrechnen (*Bildentzerrung*). Dann lassen sich Distanzen direkt

in Metern abgreifen. Viele Open-Source-Tools (z. B. *TrafficIntelligence*; „Traffic Intelligence (Website)“ 2025) verlangen vom Nutzer eine manuelle Kalibrierung durch Markieren einiger Punkte im Bild, die eine bekannte reale Distanz haben, um so die perspektivische Transformation zu bestimmen. STRUDL, ein EU-Forschungsframework für Verkehrssicherheitsanalyse, betont ebenfalls die Wichtigkeit, Trajektorien in Weltkoordinaten zu konvertieren, um aussagekräftige Kennwerte berechnen zu können [28]. OpenDataCam bietet eine einfachere Variante an, bei der man die Kamerahöhe und Neigung annimmt und dann z. B. einen virtuellen *Meterstab* einblendet, den man an die realen Abmessungen anpasst – für grobe Geschwindigkeitsabschätzungen ausreichend, aber nicht hochpräzise [29].

Zusätzlich zu dieser *präventiven* Verzerrungskorrektur durch Kalibrierung gibt es noch digitale Bildstabilisierung und -entzerrung als nachgelagerte Schritte. Dies erhöht die Tracking-Genauigkeit, da Bewegungen dann fast ausschließlich echte Objektbewegungen sind. Auch Rolling-Shutter-Verzerrungen (bei CMOS-Sensoren) könnten darüber reduziert werden, falls relevant.

Insgesamt entspricht der Stand der Technik hier einem photogrammetrischen Ansatz: Durch sorgfältige Kalibrierung (sowohl innere Kameraparameter zur Verzeichnungsentfernung als auch äußere Orientierung zur Verzerrungs-/Perspektivkorrektur) lassen sich Videoaufnahmen in metrische Daten umwandeln. Diese Verfahren sind rechen- und aufwandsintensiv, aber notwendig für belastbare Ergebnisse. OpenTrafficCam_live integriert sie weitgehend automatisiert im Hintergrund, um den Nutzern die Komplexität abzunehmen. Damit hebt es sich von einfacheren Zähl-Lösungen ab, die oft nur pixelbasiert arbeiten und z. B. Geschwindigkeiten in „Pixel pro Frame“ angeben. Hier wird stattdessen angestrebt, dass Ausgaben wie Geschwindigkeit in km/h oder Abstände in Metern direkt, plausibel und geeicht sind (ggf. Abgleich mit Referenzmessungen im Testfeld).

4.4 Verkehrsanalyse

Auf Basis der erkannten und verfolgten Objekte können vielfältige verkehrliche Analysen durchgeführt werden. Hier fließen die Daten der Computer-Vision-Stufe in verkehrstechnische Kenngrößen und Auswertungen ein. OpenTrafficCam_live bündelt solche Funktionen im Modul OTAnalytics. Dabei werden gängige Anforderungen aus der Verkehrspraxis adressiert, von einfachen Zählungen bis hin zu komplexeren Parametern wie Zeitlücken oder Qualitätsstufen nach HBS. Im Folgenden werden zwei Grundkategorien hervorgehoben: die Verkehrszählung und die Ableitung von Geschwindigkeiten/Beschleunigungen. Darauf aufbauende Analysen (z. B. Kapazitätsabschätzungen, Sicherheitsindikatoren) werden teils ebenfalls kurz erwähnt, da sie aktuelle Trends darstellen.

4.4.1 Verkehrszählung

Die Verkehrszählung ist eine der unmittelbarsten Anwendungen der Videoanalyse: Aus den detektierten Objekten werden Zählwerte abgeleitet – etwa Fahrzeugintensitäten pro Zeiteinheit, Verkehrsstärken auf Fahrstreifen oder an Knotenpunkten, sowie modalspezifische Anteile (Kfz vs. Rad vs. Fuß). Traditionell wurden hierzu manuelle Strichlisten oder pneumatische Schlauchzähler eingesetzt. KI-gestützte Videoanalytik ermöglicht nun automatisierte Zählungen mit hoher Genauigkeit und in hoher zeitlicher Auflösung. Studien zeigen, dass automatisierte Auswertungen mit modernen Erkennungsalgorithmen sehr nah an manuelle Referenzen herankommen – Miovision z. B. wirbt mit rund 95 % Genauigkeit seiner Videozählungen [2].

Für eine Zählung müssen die Trajektorien der Objekte bestimmten Zählbereichen zugeordnet werden. In der Regel werden hierfür virtuelle Zähllinien oder -zonen definiert. Überschreitet ein Track

eine solche Linie in definierter Richtung, wird ein Zählimpuls ausgelöst, der mit Zeitstempel, Objektklasse und Richtung gespeichert wird. So lassen sich etwa Knotenstromzählungen erstellen: Man legt pro Zufahrt drei Linien (Ein, Geradeaus, Rechtsabbieger) und ordnet die Durchfahrten entsprechend zu.

Neben Knotenstromzahlen sind Querschnittszählungen (z. B. an Straßenabschnitten) und Bereichszählungen (z. B. wie viele Fahrzeuge befinden sich in einer definierten Zone) möglich. Durch die kontinuierliche Videoanalyse können auch dynamische Größen wie Stauaufbau erfasst werden – z. B. die Anzahl wartender Fahrzeuge an einer Ampel über die Rotphase hinweg. Hierfür kombiniert man Zählung mit Tracking: Fahrzeuge, die sich anstellen, werden gezählt und in ihrer Position relativ zur Haltelinie verfolgt, bis sie wieder abfließen. Solche Anwendungen gehen über einfache Zähler hinaus und nutzen die Trajektorien direkt.

Die Datenqualität der Zählungen hängt von der Erkennungs- und Verfolgungsleistung ab. Verpasste oder fälschlich getrackte Objekte führen zu Zählfehlern. Moderne Systeme erreichen jedoch in vielen Settings bereits eine Güte, die für verkehrsplanerische Zwecke ausreichend ist (oft Abweichungen $< 5\%$). Wichtig ist eine Validierung: Möglichkeiten bieten sich hier durch den Abgleich mit Kurzzeit-Manuellzählungen, gezielter Nachvalidierung der Detektionsklasse bestimmter Tracks und ggf. einer automatische Parameternachsteuerung. So kann z. B. die Sensitivität des Detektors erhöht werden, falls ein Abgleich zeigt, dass Fahrzeuge bei hohem Verkehrsaufkommen untererfasst wurden.

Ein Vorteil der Videoanalyse ist die Multimodalität: Während Induktionsschleifen oder Radar häufig nur Kfz zählen, können KI-Systeme alle Verkehrsteilnehmer differenziert zählen (Autos, Lkw, Busse, Motorräder, Fahrräder, Fußgänger etc.). Dies ist für moderne Planungen wichtig (Stichwort „geschützte Radstreifen“ oder Fußgängerpotenziale). OpenTrafficCam_live legt hier besonderen Fokus drauf – es werden separate Fußgänger- und Radverkehrszählungen implementiert. Die Genauigkeit bei diesen weichen Verkehrsteilnehmern war bisher in vielen kommerziellen Lösungen niedriger, u.a. wegen ihrer geringeren Größe und dem unregelmäßigen Bewegungsmuster. Durch spezialisierte Trainingsdaten (Radfahrer in Gruppen, Fußgänger in Zebrastreifen-Situationen etc.) soll dieses Manko behoben werden.

Zusätzlich zu reinen Zählwerten können auch direkt verarbeitete Kennzahlen ausgegeben werden: z. B. Kreuzungsbelastungsgrade, Level of Service (LOS) nach HBS für Knoten. Hier fließen neben den Zähldaten auch Wartezeiten und vielleicht Lückenzeitmessungen ein. Durch die vollautomatische Erfassung können diese Angaben objektiv und zeitlich hoch aufgelöst gewonnen werden, was ein großer Fortschritt gegenüber punktuellen manuellen Erhebungen ist.

4.4.2 Ableitung von Geschwindigkeiten

Die Geschwindigkeit der Verkehrsteilnehmer ist eine zentrale Größe für Verkehrsfluss- und Sicherheitsanalysen. Aus Video-Trajektorien lässt sie sich ableiten, indem man die zurückgelegte Wegstrecke pro Zeitintervall bestimmt. Dafür müssen (wie in 4.3.3 beschrieben) die Trajektorien in metrischen Koordinaten vorliegen oder gemessene Abstände zwischen den beiden Sections vorliegen. Wenn die Kameras kalibriert sind (bekannte Maßstäbe), kann aus der Bildverschiebung zwischen zwei Frames und der Framezeit (z. B. $1/30$ s) die Geschwindigkeit berechnet werden. Moderne Systeme erzielen hier recht gute Genauigkeiten: In Tests mit Vergleich zu Radarmessungen lagen die Abweichungen oft unter ± 5 km/h. Im Forschungsprojekt InDeV wurde berichtet, dass mit Video-Tracking ermittelte Geschwindigkeiten in $>90\%$ der Fälle innerhalb ± 2 km/h zum Referenz-

Lidar lagen [30]. Wichtig ist eine exakte Synchronisation der Zeitbasis; daher werden Kameras ggf. mit GPS-Zeitstempeln versehen, um z. B. 0,1 s Genauigkeit zu erreichen.

OpenTrafficCam_live integriert Geschwindigkeitsberechnungen in OTAnalytics. So können Durchschnittsgeschwindigkeiten pro Abschnitt, Geschwindigkeitsverteilungen (V85, V50) oder auch Verläufe über den Tagesgang generiert werden. Zudem lassen sich Geschwindigkeitsprofile eines einzelnen Fahrzeugs entlang seiner Trajektorie erstellen – hilfreich etwa zur Bewertung von Beschleunigungstreifen oder Kreisverkehren. Die Ableitung der Beschleunigung erfolgt als zeitliche Änderung der Geschwindigkeit (Differenzquotient). Rauschen in den Trajektorien kann hier allerdings große Auswirkungen haben, weshalb oft geglättete Trajektorien oder längere Intervalle genutzt werden, um Beschleunigungen abzuschätzen. Trotzdem kann man z. B. starkes Abbremsen erkennen (negative Beschleunigungsspitzen), was auf kritische Situationen hindeutet.

Ist eine Umrechnung der Trajektorien in metrische Koordinaten nicht möglich, kann vereinfachend auch mit dem gemessenen Abstand zwischen den beiden Sections eines Flows die Geschwindigkeit bestimmt werden. Diese Möglichkeit wurde ebenfalls in OpenTrafficCam_live berücksichtigt.

Eine weitere Anwendung ist die Verkehrsleistungsfähigkeit. In Kombination mit den Zählraten kann etwa die mittlere Reisegeschwindigkeit auf einem Streckenabschnitt ermittelt werden, um Stau oder freie Fahrt zu klassifizieren. Auch die Beschleunigung kann verkehrstechnisch interpretiert werden: Viele Beschleunigungsvorgänge hintereinander deuten auf stop-&-go hin, starke Verzögerungen können auf Kapazitätsengpässe oder Gefahrensituationen hindeuten. Durch Videodaten ließe sich z. B. der Anteil „harte Brems- und Anfahrvorgänge“ quantifizieren.

Nicht zuletzt ist die Geschwindigkeitserfassung aus Video auch für die Überwachung interessant (Section Control). Allerdings stehen hier datenschutzrechtliche Aspekte im Vordergrund, da eine Verfolgung konkreter Kfz über längere Distanzen personenbezogene Daten (Kennzeichen) involvieren kann.

4.4.3 Weitere Verkehrsanalysen

In der Verkehrssicherheitsforschung gewinnen solche Surrogat-Maße an Bedeutung: Anstatt auf Unfälle zu warten, analysiert man Beinahe-Unfälle mittels Indikatoren wie Time-to-Collision (TTC) oder Post-Encroachment-Time (PET). Diese lassen sich aus Trajektorien, Geschwindigkeiten und Beschleunigungen berechnen. Beispielsweise ergibt sich der TTC zwischen zwei sich kreuzenden Verkehrsteilnehmern aus ihrem Abstand und relativer Geschwindigkeit. Aktuelle Markttrends zeigen eine Entwicklung in diese Richtung: Miovision etwa bietet eine Safety Analysis, die aus Videodaten Konflikte identifiziert und daraus Unfallwahrscheinlichkeiten prognostiziert – angeblich mit 94 % Vorhersagegenauigkeit für Unfallschwere [2]. Voraussetzung sind jedoch hochgenaue Trajektorien in Weltkoordinaten, damit Abstände und relative Geschwindigkeiten stimmen. Hierfür ist die in 4.3.3 beschriebene aufwendige Kalibrierung wichtig.

5 Zusammenarbeit mit Dritten

5.1 Workshop mit Verkehrsfachleuten zum Thema Datenschutz

Ziel des Workshops „Datenschutz bei videobasierten Verkehrserhebungen“ war die Präsentation der Erkenntnisse aus AP1 für eine breite Fachöffentlichkeit. Die Veranstaltung fand online statt und dauert insgesamt drei Stunden. Insgesamt nahmen an dem Workshop 77 Teilnehmende aus Kommunen, Forschung und Ingenieurbüros teil. Es fand ein reger Austausch statt.

Der Workshop gliederte sich in eine Einführung zu grundlegenden Begriffen, eine detaillierte rechtliche Einordnung und eine abschließende Diskussion der wichtigsten Folgerungen. Zu Beginn wurde erläutert, weshalb Datenschutz in diesem Kontext bedeutsam ist: Er dient dem Schutz der Privatsphäre und zugleich wirtschaftlichen Interessen, erfordert aber stets eine sorgfältige Abwägung. Als zentrale Rechtsgrundlagen wurden die Datenschutz-Grundverordnung, das Bundes- und die Landesdatenschutzgesetze sowie einschlägige Spezialgesetze vorgestellt.

Der Begriff der „Verarbeitung“ wurde breit gefasst und umfasst jeden Umgang mit personenbezogenen Daten, von der Erhebung bis zur Löschung. Dabei wurde herausgestellt, dass Personen schon dann identifizierbar sind, wenn eine Zuordnung mittelbar möglich ist; folglich gelten die Datenschutzregeln auch für vermeintlich anonyme Videoaufnahmen. Eine echte Anonymisierung ist laut Workshop nur schwer zu erreichen; deshalb wurden verschiedene Verfahren – Aggregation, Zufalls- und Synthesemethoden – sowie deren Grenzen erläutert.

Im Mittelpunkt stand der Grundsatz der Erforderlichkeit: Videotechnik darf nur eingesetzt werden, wenn kein gleich geeignetes, weniger eingriffsintensives Mittel zur Verfügung steht und die Verarbeitung auf einer tragfähigen Rechtsgrundlage beruht. Der Workshop verglich hierzu Alternativen wie manuelle Zählungen, Wärmebild- oder LiDAR-Sensorik und zeigte, warum diese entweder technisch unzureichend oder sogar stärker eingriffsintensiv sein können.

Die datenschutzrechtlichen Grundsätze des Art. 5 DSGVO – Rechtmäßigkeit, Zweckbindung, Datenminimierung, Richtigkeit, Speicherbegrenzung sowie Integrität und Vertraulichkeit – wurden als Leitplanken für jede Videoerhebung hervorgehoben. Verantwortlich bleibt stets die Stelle, die Zweck und Mittel der Verarbeitung bestimmt; sie muss zudem Transparenzpflichten erfüllen und geeignete technische sowie organisatorische Maßnahmen implementieren.

Konkrete Risikoszenarien wurden ebenfalls diskutiert: Bußgelder, mögliche Straftatbestände, Schadenersatz- und Reputationsrisiken. Zur Risikominderung empfahl der Workshop praxisnahe Maßnahmen wie eine begrenzte Pixeldichte und absichtliche Defokussierung, feste Mindesthöhen und Abstände bei der Kameramontage, Zugangskontrollen zu Aufnahmen, verschlüsselte Speicherung, frühzeitige Löschung nicht mehr benötigter Daten sowie regelmäßige Überprüfungen der Technik und Schulungen der Beteiligten.

Im Fazit wurde betont, dass videobasierte Verkehrserhebungen bei geeigneter Rechtsgrundlage und strikter Beachtung der Datenschutzprinzipien eine effiziente Methode darstellen und dass die Verarbeitung personenbezogener Daten durch gezielte Anonymisierungs- und Minimierungsstrategien weitgehend vermeidbar ist.

5.2 Austausch mit fachlich ähnlich gelagerten mFUND-Projekten

Der Austausch mit mFUND-Projekten, die OpenTrafficCam in eigenen Fragestellungen einsetzen, hat den Wissenstransfer und die Verwertbarkeit der im Vorhaben entwickelten Methoden maßgeblich beschleunigt. So stellte das Konsortium im Projekt „Ready for Robots (R4R)“ seine Trajektorien-Workflows vor; die dortigen Partner nutzen die hochaufgelösten Bewegungsdaten seither, um realitätsnahe Eingangsdaten für Mikromobil-Simulationen zu erzeugen und so die Einsatzfähigkeit autonomer Liefer- und Sharing-Roboter zu evaluieren.

Im Projekt „EffiDat“ wurden die Erfahrungen mit OTCamera-Hardware, Datenschutz-Konzept und Qualitätskontrolle in zwei gemeinsamen Workshops aufgegriffen, um ein kombiniertes Messdesign aus stationären OTC-Knoten und mit Kameras ausgerüsteten Straßenbahnen aufzubauen, das fahrstreifen- und richtungsgenaue Verkehrsstärken für die flächendeckende Datenerhebung

in Zwickau liefert ([bmv.de](https://www.bmv.de)). Die dabei gewonnenen Erkenntnisse zu Montagehöhen, Fokus-Einstellungen und automatisierter Kalibrierung flossen wiederum in die Optimierung von OTCamera-Seriengeräten ein, während die Projektpartner von den offen dokumentierten Schnittstellen und der Möglichkeit profitierten, OTC-Trajektorien direkt in ihre Analyse- und Visualisierungstools einzubinden.

Bei der Erhebung im Testfeld wurden Synergien genutzt, die sich mit dem mFUND-Projekt SivaS ergaben. Das Projekt SivaS hatte die Zielstellung Erhebungs- und Simulationsmethoden für Verkehrsdaten weiterzuentwickeln und wurde von der Professur für Kraftfahrzeugtechnik (LKT), der Professur für Mobilitätssystemplanung (MSP), der FSD Fahrzeugsystemdaten GmbH (FSD) und der Stadt Hoyerswerda bearbeitet. So diente das OTC-Gesamtsystem als stationäre Erhebungsmethode für das Projekt SivaS. Im Gegenzug lieferte das Messfahrzeug der FSD genaue GPS-Bewegungsdaten auf dem Testfeld in Dresden, die als Ground Truth zur Genauigkeitsbewertung des OTC-Gesamtsystems und der Trajektorienfusion in OT-Multicamera dienten.

5.3 Teilnahme an mFUND-Formaten

Während der Projektlaufzeit wurde das Projekt OpenTrafficCam_live sowie der jeweils aktuelle Produktstand von OpenTrafficCam in folgenden mFUND-Formaten präsentiert bzw. haben Mitglieder des Projektteams an folgenden mFUND-Formaten teilgenommen.

- mFUND Konferenz
 - 2022
 - 2023 (Leitung Workshop „Open Source Geschäftsmodelle“)
 - 2024
- mFUND Checkout am 21.05.2025
- mFUND Workshop Geschäftsmodellentwicklung 2024
- verschiedene mFUND Auftakt- und Abschlussveranstaltungen

5.4 Öffentlichkeitsarbeit

Während der Projektlaufzeit wurde das Projekt OpenTrafficCam_live sowie der jeweils aktuelle Produktstand von OpenTrafficCam auf folgenden Kanälen präsentiert.

- LinkedIn
 - Eigener Auftritt
 - Horizon
- GitHub
 - Code-Veröffentlichung
 - Öffentlicher Issue-Tracker
- Messebesuche:
 - polisMobility 2024
 - SmartCountry Convention 2024
 - FGSV-Jahrestagung 2024
 - Heureka 2024 in Stuttgart
- Workshop Datenschutz (s.o.)
- Startup-Pitches
 - polisMobility 2024
 - Future Mobility Award 2024 (2. Platz)
 - mFUND Konferenz 2024
- Auftaktveranstaltung

- Abschlussveranstaltung
- Webinar Invenium
- BAI.CON 2023 Ingolstadt

6 Verwendung der Zuwendung und erzielttes Ergebnis

6.1 AP 1 Datenschutz (platomo)

Im Verlauf des Vorhabens wurde der Themenkomplex Datenschutz systematisch aufgearbeitet und schrittweise in das laufende Entwicklungsgeschehen integriert. Zentraler Punkt war eine gemeinsam mit einer auf IT- und Datenschutzrecht spezialisierten Kanzlei erstellte Risikoanalyse. In mehreren Iterationen wurden dabei sämtliche Verarbeitungsschritte – von der Kamerainstallation über die Bildaufzeichnung bis hin zur daten-getriebenen Auswertung – im Hinblick auf die einschlägigen Vorgaben der DSGVO untersucht. Die Analyse liefert einen strukturierten Überblick über potenzielle Risiken sowie konkrete Handlungsempfehlungen und bildete damit die Grundlage für viele weitere Entscheidungen (z. B. zusätzliche Lagesensoren im Rahmen der weiteren Entwicklung der OTCamera).

Um die Ergebnisse der Risikoanalyse auch außerhalb des Projektteams nachvollziehbar zu machen, ist im Anschluss ein kompakt gehaltenes Handout entstanden. Dieses Dokument fasst das technische und organisatorische Datenschutzkonzept in verständlicher Sprache zusammen, erläutert die vorgesehenen Schutzmaßnahmen und dient zugleich als interne Arbeits- und externe Informationsgrundlage (z. B. für kommunale Genehmigungsstellen).

Die erarbeiteten Inhalte wurden anschließend in einem ganztägigen Datenschutzworkshop mit allen Praxis- und Entwicklungspartnern diskutiert. Der Workshop diente dazu, offene Fragen zum rechtlichen Rahmen, zu praxisnahen Schutzmaßnahmen sowie zu verbleibenden Restrisiken zu klären.

Auf Basis der Risikoanalyse sind bereits mehrere Maßnahmen umgesetzt oder in Auftrag gegeben worden. Erstens wurde die Beschaffung eines Bewegungssensors angestoßen, der das Kamerasystem nur bei tatsächlicher Verkehrspräsenz aktiviert; die Finanzierung erfolgt aus Drittmitteln und wird in der Verwertungsplanung abgebildet. Zweitens wurden verbindliche Grenzwerte für erforderliche Pixeldichten sowie Mindestmontagehöhen definiert. Diese stellen sicher, dass Personen grundsätzlich nicht identifizierbar sind und gleichzeitig die für die Verkehrsanalyse notwendige Bildqualität gewährleistet bleibt. Die festgelegten Parameter sind inzwischen in die Installationsrichtlinien und die projektspezifischen Standard-Operating-Procedures aufgenommen.

6.2 AP 2 OTCamera (platomo)

Im Rahmen des Arbeitspakets 2 wurden umfangreiche Maßnahmen zur Optimierung und Weiterentwicklung der Hardware- und Softwarekomponenten des Kamera -Systems durchgeführt. Dieses Arbeitspaket zielte darauf ab, die Zuverlässigkeit, Benutzerfreundlichkeit und Effizienz des Systems zu verbessern, neue Funktionen zu implementieren und es für den breiten Einsatz in der Praxis vorzubereiten.

6.2.1 Wiederkehrende Optimierung Performance und Zuverlässigkeit Soft- und Hardware (platomo)

Die wiederkehrende Optimierung der Performance und Zuverlässigkeit der Soft- und Hardware wurde von platomo erfolgreich umgesetzt. Die OTCamera wurde für den wiederkehrenden Einsatz im Feld optimiert und konnte bereits an über 200 Messstellen in mehr als 20 Städten in ganz

Deutschland und Dänemark eingesetzt werden. Dabei hat die OTCamera bereits über 10.000 Stunden Videomaterial aufgezeichnet und verarbeitet, was die hohe Zuverlässigkeit des Systems unter Beweis stellt. Die kontinuierliche Verbesserung basierte auf Feedback aus realen Einsatzszenarien und führte zu einer deutlichen Steigerung der Systemstabilität auch unter widrigen Umweltbedingungen. In der ersten Projektphase wurden grundlegende Optimierungen der Hardware vorgenommen, die die Basis für die spätere Skalierung bildeten. Im weiteren Projektverlauf konnte eine signifikante Steigerung der Einsatzzahlen erreicht werden, die durch die verbesserte Zuverlässigkeit und einfachere Handhabung ermöglicht wurde. Konkret wurden vor allem die Stromversorgung und der Stromverbrauch im Einsatz verbessert, sowie die Konnektivität außerhalb vom Feld zur jetzt teilautomatisierten Wartung und Aktualisierung. Die Aktualisierung der Gehäusekomponenten hat außerdem zu einer besseren Witterungsbeständigkeit und kürzerer Montagezeit geführt.

6.2.2 Erweiterung der Platine als Fertigbauteil inkl. Montage und Elektrotechnische Prüfung (platomo mit UA)

Die Erweiterung der Platine als Fertigbauteil inklusive Montage und elektrotechnischer Prüfung wurde von platomo in Zusammenarbeit mit einem Unterauftragnehmer realisiert. In einem ersten Schritt hat platomo den Versionsstand so angepasst, dass dieser automatisiert fertigbar wurde. Die Fertigung wurde dann bei Aisler durchgeführt und es wurden mehrere Prototypen gefertigt, bis das System den Anforderungen entsprach. Die bestehende Platine kann nun beim Auftragsfertiger Aisler bestellt werden, was eine effiziente Produktion ermöglicht und die Montagekosten deutlich gesenkt hat. Nach der Mittelfreigabe kurz vor Ende 2024 wurde zudem eine Weiterentwicklung der Platine beauftragt, um weitere Verbesserungen zu implementieren. Im Rahmen dieser Weiterentwicklung wurde ein vollständig überarbeitetes Platinenlayout entwickelt, das eine verbesserte Anordnung der Komponenten, optimierte Signalwege und erhöhte Störfestigkeit bietet. Das neue Layout ermöglicht zudem die Integration zusätzlicher Sensoren und Schnittstellen für zukünftige Erweiterungen. Die Entwicklung begann mit ersten Prototypen der Platine, bei denen verschiedene Verbesserungspotenziale identifiziert wurden. Im weiteren Projektverlauf erfolgte die erfolgreiche Überführung in ein Fertigbauteil, was zu einer deutlichen Qualitätssteigerung und Vereinfachung des Produktionsprozesses führte. Es wurden außerdem Komponenten zur Online-Kommunikation (v.a. LTE-Modul) integriert und eine neue Stromversorgung implementiert, um in Zukunft mit eigenen Akkupacks arbeiten zu können. Das ermöglicht einerseits die Überwachung der Kamerafunktionalität im Feld und andererseits eine längere Akkulaufzeit.

6.2.3 Automatisierte Installation, Modularisierung (z.B. Weitwinkel) und Online-Fähigkeit (platomo)

Im Bereich der automatisierten Installation, Modularisierung und Online-Fähigkeit hat platomo Fortschritte erzielt. Die OTCamera wurde so weiterentwickelt, dass sie sowohl mit Standard- als auch mit Weitwinkelobjektiven kompatibel ist. Weitwinkelobjektive wurden bereits erfolgreich in Dänemark zur Erfassung aus großer Höhe eingesetzt. Die gesamte Hardware (Platine, Pufferakku, Kameramodul) sind in ein komplett neu entwickeltes 3D-gedrucktes Gehäuse implementiert worden. Durch den Einsatz von 3D-Druck Verfahren ist es möglich, Anpassungen schnell und einfach vorzunehmen, ohne den Produktionsprozess an sich ändern zu müssen. So lassen sich nun innerhalb weniger Stunden Anpassungen vornehmen, um ein völlig anderes Kamera-Modul aufzunehmen. Ein so angepasstes Gehäuse ist dann direkt herstellbar.

Zu Beginn des Projekts wurde die manuelle Installation der Software-Pakete und des zentralen Betriebssystems als zeitaufwändiger Prozess identifiziert, der die Skalierbarkeit des Systems einschränkte. Als Lösung wurde die Installation durch mehrere ansible-Playbooks automatisiert und modularisiert, wodurch bestellte OTCameras effizient mit diesem Skript vollautomatisch installiert werden konnten. Diese Automatisierungslösung erwies sich als wesentliche Verbesserung für den Produktionsprozess und ermöglichte eine deutlich schnellere und fehlerfreiere Inbetriebnahme neuer Geräte. Die Onlinefähigkeit wurde im Rahmen der Weiterentwicklung der Platine nach der Mittelfreigabe kurz vor Ende 2024 beauftragt und implementiert.

6.2.4 Mini-OTCamera (MSP)

Ziel der Mini-OTCamera war eine Version der OTCamera ohne eigene Platine zu erstellen. Dadurch sollte der Open-Hardware Gedanke der OTCamera weiter ausgebaut werden. Eine Person hätte ohne spezielle Hardware von platomo ein OTCamera bauen und betreiben können. Bei einer Markt- und Technologiesondierung wurde zum einen festgestellt, dass ein dediziertes Energiemanagement der Kamera notwendig ist. Hierfür gab es keine für den Anwendungsfall passenden Komponenten. Zum anderen wurde während dem Projekt das Innengehäuse der Kamera nach der Entwicklung der Platine zu einem Fertigbauteil optimiert und verkleinert. Diese Optimierung ermöglicht die Verwendung eines kompakteren Outdoor-Gehäuses, wodurch die OTCamera insgesamt verkleinert werden konnte. OTCamera konnte in eine insgesamt kleinere Version überführt werden. Dies verringert die Notwendigkeit einer separaten Mini-OTCamera.

Basierend auf dem neuen Platinenlayout wurde ein komplett neues 3D-Modell des Innengehäuses entworfen. Dieses Modell berücksichtigt optimierte Luftströme für eine verbesserte Kühlung der Elektronik, bietet besseren Schutz gegen Umwelteinflüsse und ermöglicht eine einfachere Montage und Wartung. Durch den Einsatz moderner CAD-Software und 3D-Drucktechnologie konnte das Gehäuse präzise an die Anforderungen angepasst und in kleinen Serien kostengünstig produziert werden. Die Entwicklung des Innengehäuses durchlief mehrere Iterationen, beginnend mit ersten Prototypen, bei denen Herausforderungen bezüglich Wärmemanagement und Montagefreundlichkeit auftraten. In der weiteren Entwicklung konnten diese Probleme durch das neue 3D-Modell erfolgreich gelöst werden, was zu einem deutlich verbesserten Gehäusedesign führte.

6.2.5 Erweiterung Korrektur der Objektivverzeichnung auf verschiedene Kameramodule (PHOTO)

Im Projekt OpenTrafficCam_live wurde die perspektivische Verzerrung umfassend durch eine 3D-Rekonstruktion des Szenarios gelöst. Das heißt, man beschränkt sich nicht nur auf eine planare Homographie, sondern bestimmt die volle Projektionsgeometrie der Kamera. Es wurde dafür ein geometrisches Modell des Kamerasystems entwickelt. Anschließend muss ein Bündelblockausgleichung (*Bundle Adjustment*) durchgeführt werden, um alle Kameraposen und gegebenenfalls 3D-Punkte in einer Szene zu optimieren. Dafür können natürliche Punkte (Markierungen auf der Fahrbahn, Ecken von Gebäuden) oder künstliche Marker genutzt werden. Letztlich resultiert daraus für jede Kamera eine Transformationsmatrix, mit der man Bildkoordinaten in Weltkoordinaten (z. B. Meter im Straßennetz) umrechnen kann.

Drauf aufbauend wurden im Rahmen von OTMulticamera verschiedene Korrekturen der Objektivverzeichnung getestet. Dazu zählt eine Korrektur mittels offener Algorithmen in OpenCV sowie einer proprietären Lösung von Aicon. Die proprietäre Lösung von Aicon ist hierbei genauer, aber auch mit höheren Kosten verbunden. Die Entwicklung in diesem Bereich begann mit ersten grundlegenden Versuchen zur Korrektur der Objektivverzeichnung, die wichtige Erkenntnisse für die

weitere Arbeit lieferten. Im weiteren Projektverlauf erfolgte eine umfassende vergleichende Analyse der verschiedenen Lösungsansätze, die deren Auswirkungen auf die Genauigkeit der Verkehrsdatenerfassung detailliert untersuchte und Empfehlungen für den optimalen Einsatz der Technologien in verschiedenen Szenarien lieferte.

6.2.6 Sondierung und Test alternativer Hardwarekomponenten (platomo)

Schließlich wurden durch platomo alternative Hardwarekomponenten sondiert und getestet. Zu Beginn des Projekts wurde die Kommunikation über LoRaWAN getestet, wobei verschiedene Limitierungen bezüglich Datenrate, Reichweite und Netzabdeckung identifiziert wurden, die für den angestrebten Einsatzzweck berücksichtigt werden mussten. In einer späteren Projektphase wurde die Stromversorgung über ein Solarmodul prototypisch getestet, was zu erfolgreichen Ergebnissen führte. Aus diesen Tests wurden detaillierte Anforderungen für einen autarken Betrieb abgeleitet und für die weitere Entwicklung dokumentiert. Diese Erkenntnisse bilden eine wichtige Grundlage für zukünftige Weiterentwicklungen des Systems mit Fokus auf Energieautarkie und drahtlose Kommunikation. Außerdem wurden als Basis für die Kamera verschiedene Raspberry Pi Modelle getestet. Letztendlich ist weiterhin der Raspberry Pi Zero W und dessen Nachfolger Raspberry Pi Zero W 2 die Hardware, welche eingesetzt wird. Vor allem der geringe Stromverbrauch und die geringen Kosten führten zu dieser Entscheidung. Platomo hat allerdings die Erkenntnis erlangt, dass die größere Raspberry Pi Modelle in Zukunft vsl. für die Live-Systeme eingesetzt werden können. Diese wurden auch mit PoE+ Adaptern getestet, um die gemeinsame Stromversorgung und Datenübertragung über ein einzelnes Kabel zu ermöglichen. Vermutlich werden in weiteren Entwicklungen auch die für industrielle Zwecke gedachten Modelle Raspberry Pi Compute Module eingesetzt, da diese eine höhere Integrationstiefe zulassen.

Um eine gleichbleibend hohe Qualität der OTCamera zu gewährleisten, wurde ein standardisierter Produktionsprozess entwickelt und dokumentiert. Dieser Prozess umfasst detaillierte Arbeitsanweisungen für die Montage der Hardware, die Installation der Software sowie Qualitätskontrollen in verschiedenen Produktionsstadien. Die Standardisierung ermöglicht eine effiziente Fertigung und minimiert Fehlerquellen während der Produktion. Die Einführung dieses standardisierten Prozesses erwies sich als entscheidender Faktor für die erfolgreiche Skalierung der Produktion und führte zu einer signifikanten Reduzierung von Produktionsfehlern und notwendigen Nacharbeiten. Durch die klare Dokumentation und Schulung der Mitarbeitenden konnte zudem die Einarbeitungszeit neuer Teammitglieder deutlich verkürzt und eine konsistente Produktqualität sichergestellt werden.

Ein wesentlicher Bestandteil der Qualitätssicherung ist der eingeführte Prozess zur Kalibrierung für jede OTCamera vor der Auslieferung. Dieser Prozess umfasst die Einstellung der Kameraoptik auf den Nahbereich, um die erforderliche Unschärfe ab einer gewissen Montagehöhe sichern zu können. Die Kalibrierungsdaten werden für jede Kamera dokumentiert und können bei Bedarf für spätere Referenzzwecke herangezogen werden. Die Entwicklung dieses Kalibrierungsprozesses stellte einen wichtigen Meilenstein für die Qualitätssicherung dar und hatte nachweislich positive Auswirkungen auf die Zuverlässigkeit und Genauigkeit der Datenerfassung im Feld. Durch die systematische Kalibrierung konnte die Fehlerrate bei der Verkehrsdatenerfassung signifikant reduziert werden, was zu einer höheren Datenqualität und Kundenzufriedenheit führte. Der standardisierte Kalibrierungsprozess ist inzwischen fester Bestandteil der Produktionskette und trägt maßgeblich zur Qualitätssicherung des Gesamtsystems bei.

6.2.7 Zusammenfassung aktueller Stand

OTCamera ist das erste System in der Gesamtverarbeitungskette von OpenTrafficCam. OTCamera besteht elektronisch im Kern aktuell aus

- einem Raspberry Pi Zero W,
- einem Raspberry Pi Camera Modul v2 NoIR,
- einem Stromversorgungsmodul (Adafruit Powerboost 1000C),
- einem internen Akku,
- einer Realtime Clock (Adafruit PiRTC Precision),
- 4 Schaltern zur Steuerung,
- 3 LEDs zur Informationsausgabe und
- einer von platomo entwickelten Platine zur Integration der genannten Komponenten.

Die Komponenten sind in ein von uns entwickeltes Gehäuse integriert und dieses wiederum in ein wetterfestes, abschließbares Gehäuse. Das Gehäuse lässt sich mittels Bandschelle und einem Kugelkopf an einem fast beliebigen Masten im Straßenraum montieren. Auch die Montage an Hauswänden und anderen Gegenständen der Straßeneinrichtung ist schon in verschiedenen Städten erfolgt. Der aktuell verwendete Kugelkopf ist gut geeignet, um eine OTCamera häufig an neuen Stellen schnell und einfach auszurichten und dabei eine möglichst hohe Stabilität sicherzustellen. Die Komponenten von OTCamera sind vielfach erprobt und kommen regelmäßig zum Einsatz. Wir selbst haben als platomo GmbH etwa 30 Kameras im Einsatz und weitere Exemplare an Dritte verkauft. Die Auswahl der Komponenten beruht bislang auf der Annahme, dass das System mit einem Akku betrieben wird und die Laufzeit möglichst hoch sein soll.

Mit OTCamera werden Videos des Verkehrsgeschehens lokal aufgezeichnet. Die Videos haben eine fest definierte Aufnahmedauer und werden mit einem Zeitstempel versehen. Die aktuelle Uhrzeit wird durch die Real-Time-Clock gegeben und ist in der Regel auf wenige Sekunden genau. Die Uhrzeit wird aktuell vor jeder Messung mittels WLAN über das Internet mit Zeitservern synchronisiert. OTCamera ist im Moment hauptsächlich darauf ausgelegt, autark über wenige Stunden bis einige Tage an einem wechselnden Ort montiert zu werden (mobile Kurzzeitzählungen). In der Community von OpenTrafficCam gibt es Anwendende, die mehrere OTCameras über Monate hinweg mit einem USB-Netzteil dauerhaft aufzeichnen lassen und die Videos periodisch herunterladen. Eine fest eingebaute Möglichkeit zur kontinuierlichen Stromversorgung bietet OTCamera aktuell nicht.

Auf dem Raspberry Pi in OTCamera kommt eine von uns entwickelte Software zum Einsatz. Diese nimmt die Bedienungseingaben über die Schalter entgegen, gibt entsprechende Informationen per LEDs zurück, steuert das eingebaute WLAN und stellt eine lokale Website mit Statusinformationen und der Vorschau des Videobildes bereit. Darüber hinaus steuert die Software vor allem das Kameramodul und verarbeitet den Bild-Stream der Kamera mittels quelloffener Bibliotheken, sodass diese auf der SD-Karte des Raspberry Pi gespeichert werden können. Die Videos werden dabei von der integrierten Encoder-Einheit des Raspberry Pi mit dem h264 Codec verarbeitet und als Streams gespeichert. Jeweils zur vollen Viertelstunde wird eine neue Videodatei begonnen.

Der Datenschutz ist bei der Aufzeichnung von Videos im öffentlichen Raum von zentraler Bedeutung. Die Rechte von einzelnen Personen und Unternehmen sind mit allen zur Verfügung stehenden Mitteln einzuhalten. Zurzeit wird durch die Defokussierung der Kamera-Linse sichergestellt. Das Camera Module V2 beinhaltet eine einfache Linse, welche es erlaubt den Fokuspunkt festzulegen. Gleichzeitig wird das Videobild in geringer Auflösung aufgezeichnet. Durch diese Maßnah-

men werden personenbezogene und -beziehbare Daten (wie Gesichter und Kennzeichen) zu keinem Zeitpunkt erfasst oder verarbeitet. Sofern eine Kamera dauerhaft an einer Stelle montiert wird, können weitere Anforderungen an den Datenschutz entstehen. Diese müssen bei der Weiterentwicklung von OTCamera berücksichtigt werden. Gleiches gilt für die Sicherung gegen unbefugten Zugriff auf die Hardware und den Speicher der OTCamera. Diese Themen wurden im Rahmen des Datenschutz-Workshops adressiert, um sie in weiteren Entwicklungen in Zukunft einfließen zu lassen.

6.3 AP 3 (PHOTO)

Bei Verwendung einer einzelnen Kamera entstehen insbesondere hinter großen, nah an der Kamera befindlichen Fahrzeugen Sichtschatten, welche dahinterliegende Verkehrsteilnehmende verdecken und deren gemessene Trajektorien unterbrechen können. Um diese Informationslücken zu schließen wurde der Einsatz von mehreren Kameras ermöglicht, welche zeitgleich aus verschiedenen Perspektiven einen Verkehrsknotenpunkt beobachten. Ziel war es, durch die verschiedenen Blickwinkel und die redundante Messung von Trajektorien genauere und vollständigere Ergebnisse zu erhalten. Zu diesem Zweck wurden Testdatensätze mit Mehrkamerasystemen erhoben, entsprechende Auswertemethoden entwickelt und getestet und diese im Softwaremodul OTMulticamera implementiert. Das Modul fügt sich in die OpenTrafficCam-Pipeline ein, indem die aus OTVision erhaltenen bildbasierten Trajektorien für die einzelnen Kameras auf geeignete Weise mit OTMulticamera fusioniert und in ein 3D-Weltkoordinatensystem übertragen werden. Die resultierenden gemittelten 3D-Trajektorien können dann an OTVision zurückgegeben und weiterverarbeitet werden.

6.3.1 Testdatenaquise

Als Basis für die Entwicklungsarbeiten in OTMulticamera wurden Testdatensätze mit Mehrkamerasystemen erhoben, sowie alle notwendigen Messungen durchgeführt, die zu einer exakten Positionierung und Orientierung der Kameras im Raum und zur Georeferenzierung notwendig sind. Zunächst entstanden Aufnahmen mit handelsüblichen Spiegelreflexkameras in höherer Auflösung von einem eigenen Testfahrzeug, auf deren Basis eine erste grobe Auswertemethodik entwickelt wurde. Des Weiteren wurden Testdaten mit vier OTCamera-Systemen mit datenschutzkonformer Auflösung an einer Kreuzung im laufenden Verkehr erhoben, anhand derer die Methodik schließlich auf reale Gegebenheiten übertragen wurde. Die Messungen für die Georeferenzierung beinhalteten die Aufnahme von Bildern für die Kamerakalibrierung, die GNSS-Messung von Passpunkten sowie die Aufnahme von Laserscans, um eine exakte geometrische Repräsentation der Fahrbahnoberfläche zu erhalten (z.B. um Neigungswechsel der Fahrbahn zu erfassen).

6.3.2 Modul OTMulticamera

Als Grundlage für die Entwicklungs- und Programmierarbeiten zu OTMulticamera wurde eine grafische Benutzeroberfläche erstellt, welche eine geeignete Visualisierung von Punktwolken und Bilddaten, Nutzerinteraktionen und die grafische Darstellung von Ergebnissen ermöglicht. Sie stellt das Grundgerüst dar, in welches die entwickelten Methoden sukzessive eingearbeitet wurden. Die einzelnen Schritte des unter (Absatz 6.3.3) beschriebenen Workflows für die Multi-Kamera-Auswertung sind dabei jeweils als Menüpunkte integriert und individuell ausführbar (vgl. Abbildung 1)

Für das Einlesen initialer Parameter und Inputdaten in die OTMulticamera-Software sowie die Zwischenspeicherung und den Wiederaufruf von Datenvisualisierung, Bearbeitungszwischenständen und Zwischenergebnissen wurde ein Projektmanagement erarbeitet und in die Software integriert. Über eine Projektdatei (*.otcam) werden die entsprechenden Daten verwaltet.

Um die in OTVision erzeugten Resultate des Trackings der Verkehrsobjekte in den Videos eines Multi-Kamerasystems in die OTMulticamera-Auswertung integrieren zu können, erfolgte die Implementation einer entsprechenden Schnittstelle.

Zur Verwaltung von Videobilddaten und geometrischen Parametern mehrerer Kameras als auch der zugehörigen detektierten Trajektorien wurden entsprechende Klassen entwickelt, die z.B. Elementzugriff, Suche oder Sortierung in den umfangreichen Daten erleichtern.

Für eine Verwendung von Resultaten aus dem Auswerteprozess außerhalb der OTMulticamera - Software wurden entsprechende Exportformate entwickelt und Exportoptionen implementiert. So können die aus OTVision eingelesenen 2D-Trajektorien mit 3D-Weltkoordinaten angereichert wieder ausgegeben werden, als auch die Liste der fusionierten und gemittelten 3D-Trajektorien, welche Verweise auf die jeweiligen Ursprungstrajektorien und deren Kamerazugehörigkeit enthält.

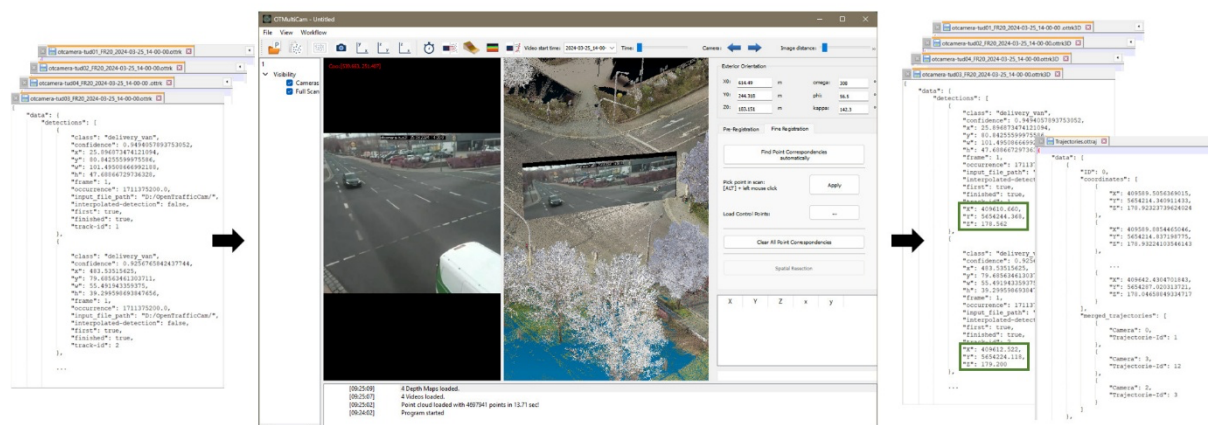


Abbildung 1: Modul OTMulticamera

6.3.3 Methodik

Um bildbasierte Trajektorien aus verschiedenen Kameras fusionieren zu können sind die in Abbildung 2 dargestellten Schritte zu erfüllen. Die methodischen Arbeiten für OTMulticamera unterteilen sich somit in 4 wesentliche Hauptaufgaben, deren Umsetzung im Folgenden näher beschrieben wird:

- **Kamerasynchronisierung:** Schaffung eines gemeinsamen Zeitbezugs für die verschiedenen Kameras
- **Georeferenzierung:** Schaffung eines genauen räumlichen Bezuges zwischen den Aufnahmen der verschiedenen Kameras und der Möglichkeit der Transformation von 2D-Bildtrajektorien in 3D-Trajektorien
- **Trajektorienzuordnung:** Identifizierung von Trajektorien aus verschiedenen Kameras, die zum selben Objekt gehören

- *Zielwerte ableiten*: Fusion der Trajektorien eines Objektes zu einer vollständigeren und gemittelten Trajektorie sowie Untersuchungen zur Nutzbarkeit der erhaltenen 3D-Informationen zur Bestimmung der Fahrzeuggrundfläche bzw. Fahrzeugflächenwahl

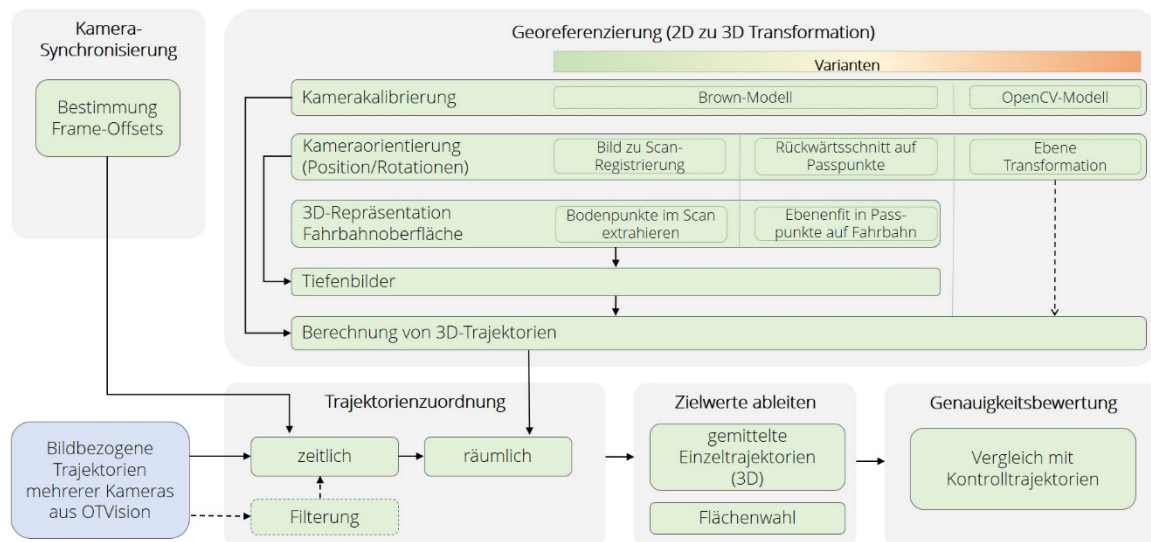


Abbildung 2: Workflow

6.3.3.1 Kamerasynchronisierung

Für die Auswertung von Multikamerasystemen bedarf es eines gemeinsamen Zeitbezugs für die Videos aller Kameras. Kann dieser Zeitbezug nicht bereits durch hardwareseitige Synchronisierung bereitgestellt werden, bietet OTMultikamera die Möglichkeit zeitlich zusammengehörende Videos von mehreren Kameras mittels optischer Signale framegenau zu synchronisieren. Dazu sind zeit-identische Frames in den Videos verschiedener Kameras visuell zu identifizieren. Es wurde ein Doppelbildviewer implementiert, mit dessen Hilfe die jeweiligen Frames, welche das optische Signal enthalten, in den Videos zweier Kameras detektiert und ihre Frame-Differenzen berechnet werden können (vgl. Abbildung 3). Schrittweise können so alle Kameras zueinander synchronisiert und mittels kombinatorischer Auswertung die jeweiligen Frameoffsets zur Masterkamera des Multikamerasystems bestimmt werden.

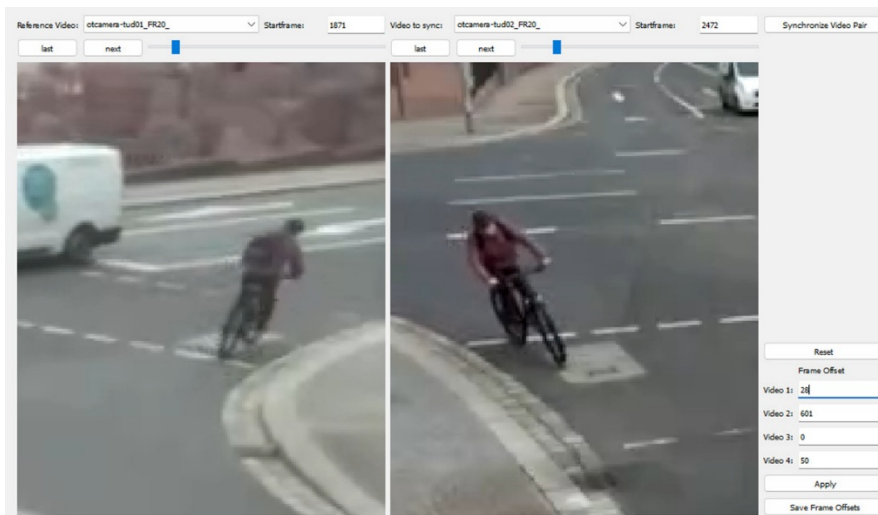


Abbildung 3: Kamerasynchronisierung

6.3.3.2 Georeferenzierung

Unter dem Arbeitsschritt der „Georeferenzierung“ sind alle Teilschritte zusammengefasst, die notwendig sind, um im Bildkoordinatensystem vorliegende 2D-Punkte in ein 3D-Weltkoordinatensystem zu übertragen (vgl. Abbildung 2):

- *Kamerakalibrierung*: Bestimmung der inneren Geometrie der Kameras
- *Kameraorientierung*: Bestimmung der Position und der Orientierungswinkel der Kameras im Weltkoordinatensystem
- *3D-Fahrbahnoberfläche*: Bereitstellung von Messdaten, welche die Fahrbahnoberfläche im Weltkoordinatensystem repräsentieren
- *Transformation in 3D-Weltkoordinaten*: Berechnung von Tiefenbildern und Anwendung des mathematischen Kameramodells zur Transformation von 2D-Bild- in 3D-Weltkoordinaten

Für den Workflow der Georeferenzierung wurden verschiedene Ansätze entwickelt, die je nach Anwendungsziel ausgewählt werden können. Dabei hängt die Wahl der Methode davon ab, ob der Fokus auf einer hohen Zuverlässigkeit und Genauigkeit der Ergebnisse oder auf der praktischen Anwendbarkeit der Methodik liegt. Anwendungen, bei denen eine hohe Zuverlässigkeit und Genauigkeit erforderlich sind, erfordern in der Regel einen höheren Messaufwand. Dagegen ist bei Anwendungen, die eine einfache praktische Handhabung priorisieren, ein geringeres Potenzial für Genauigkeit in Kauf zu nehmen. Im Folgenden werden die einzelnen Arbeitsschritte der Georeferenzierungsvariante beschrieben, die auf eine möglichst hohe Genauigkeit und Zuverlässigkeit ausgelegt ist:

Kamerakalibrierung: Für die OTMulticamera-Software wurde das mathematische Modell zur Transformation zwischen Bild- und Objektraum nach (Luhmann, 2010) in Kombination mit dem Verzerrungspolynom nach (Brown, 1971) implementiert. Zur Kalibrierung wird ein mobiles Testfeld (Abbildung 4 links) verwendet, mit welchem vor Ort, zeitnah zur Installation der Kamera Kalibrieraufnahmen gemacht werden können, indem das Testfeld vor den Kameras bzw. die Kamera vor dem Testfeld bewegt wird. Für eine genaue Bestimmung der inneren Kalibrierparameter werden alle Kameras anhand der Testfeldbilder extern kalibriert und die erhaltenen Kalibrierparameter ins Programm eingelesen (bzw. über die Projektdatei importiert). Für die zur Testdatengewinnung verwendeten OT-Kamerasysteme (RasPiCam3 mit kleinbildäquivalenter Brennweite 30 mm)

konnten Kalibrierungenauigkeiten zwischen 0,1 - 0,2 Pixel erreicht werden. Innerhalb OTMulticamera werden die Bilder um die ermittelten Verzeichnungen (Abbildung 4 rechts) rechnerisch korrigiert.

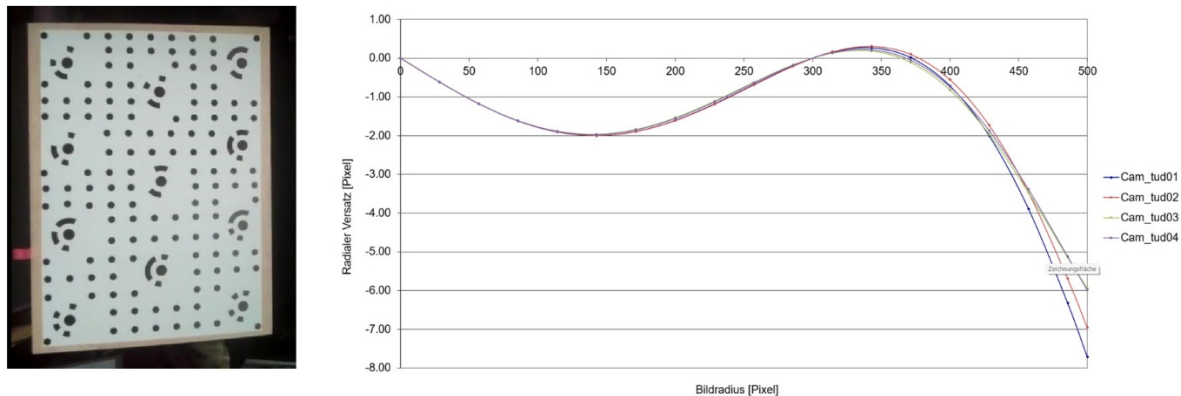


Abbildung 4: Kalibriertafel(links) und Verzeichnungskurven(rechts)

Kameraorientierung: Grundlage für die Herstellung des Raumbezugs von mehreren Kameras zueinander und zur 3D-Repräsentation der Fahrbahn (z.B. Scan-Punktwolke) ist die Bestimmung von Position und Orientierungswinkeln welche die Lage und Ausrichtung der einzelnen Kamera im Raum beschreiben (vgl. Abbildung 5 c). Für diesen Zweck wurde ein Modul in die OTMulticamera-Software integriert, für welches eine Kombination aus manuellen und automatischen Ansätzen entwickelt bzw. implementiert wurden, mit deren Hilfe die entsprechenden Kameraparameter generiert werden können:

Neben der Möglichkeit Näherungswerte für die Position und Orientierungswinkel über die Projektdatei einzulesen, können diese auch manuell durch Klicken der Kameraposition und Rotieren der Kameras in der Punktwolke beschafft werden.

Die exakte Bestimmung der äußeren Orientierung der Kamera ist möglich, wenn zu einer ausreichenden Zahl an Bildpunkten im Kamerabild (vgl. Abbildung 5a) die korrespondierenden 3D-Weltkoordinaten (vgl. Abbildung 5b) zugeordnet werden können. Für die Realisierung dieser Zuordnung wurden ein automatischer Ansatz sowie ein manueller Ansatz implementiert.

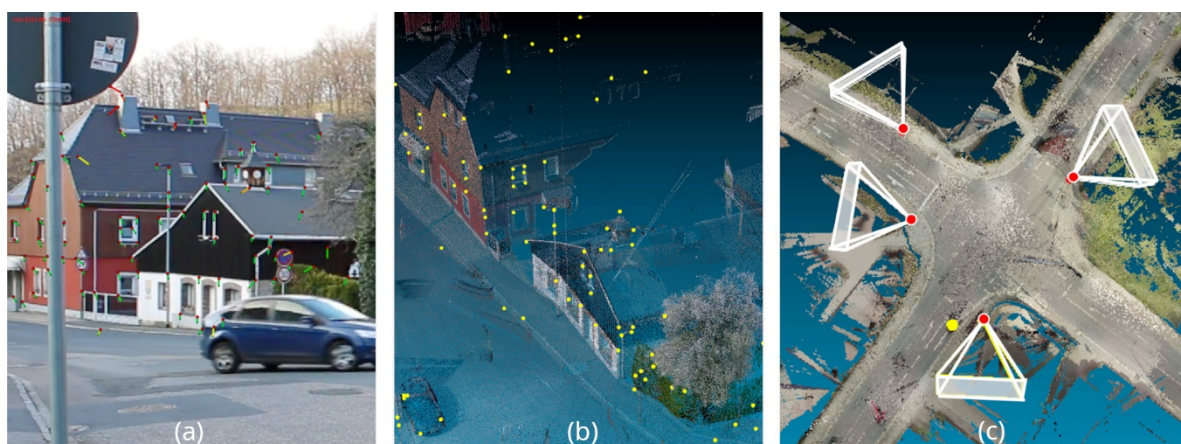


Abbildung 5: Kameraorientierung

Beim automatischen Ansatz wird für Näherungsorientierung der Kameras jeweils ein virtuelles Bild aus der Laserscanpunktvolke erzeugt. Mit dem KI-basierten Punktzuordnungsalgorithmus „SuperGlue“ (Sarlin et al., 2020) werden automatisch identische Punkte im virtuellen Punktwolkenbild und im reellen Kamerabild gesucht. Da dem Punktwolkenbild direkt 3D-Koordinaten zugrunde liegen können den Bildpunkten im Kamerabild auf diese Weise ihre entsprechenden 3D-Koordinaten zugeordnet werden.

3D-Repräsentation der Fahrbahnoberfläche: Da die stark konvergente Ausrichtung der Kameras im Mehrkamerasystem eine klassische Stereoauswertung nicht zulässt, erfolgt die Auswertung der Kameras unabhängig voneinander. Um dennoch eine Transformation von 2D- zu 3D-Koordinaten für eine Einzelkamera durchführen zu können, sind Modellannahmen oder zusätzliche Messungen erforderlich, um Informationen über die dritte Dimension zu gewinnen.

Hierzu müssen dem Verfahren externe Messdaten bereitgestellt werden, die die Fahrbahnoberfläche dreidimensional im Weltkoordinatensystem abbilden. In einfachen Fällen kann dies durch wenige Passpunkte erfolgen, in die eine ausgleichende Ebene eingepasst wird. Lässt sich die Fahrbahn jedoch nicht ausreichend genau durch eine Ebene approximieren, besteht die Möglichkeit, die Oberfläche mittels Laserscanning zu erfassen. Diese Methode liefert eine exakte dreidimensionale Repräsentation der Fahrbahn, welche ein hohes Genauigkeitspotential bei der Berechnung der 3D-Trajektorien ermöglicht. Dadurch wird zugleich die Zuverlässigkeit bei der korrekten Zuordnung zusammengehöriger Trajektorien erhöht.

Transformation von Trajektorien in 3D-Weltkoordinaten: Ist für jeden mit Bildkoordinaten vorliegenden Punkt der Trajektorie die Entfernung zum korrespondierenden 3D Punkt auf der Fahrbahn bekannt, so kann eine effiziente Umrechnung von in den Videobildern getrackten Trajektorien zu 3D-Trajektorien erfolgen. Für diesen Zweck wurden Funktionen entwickelt und implementiert, mit denen aus einer Scan-Punktwolke für jede einzelne Kamera ein Tiefenbild generiert werden kann. Um Verdeckungen durch Vegetation oder Objekte im Messbereich zu vermeiden, wurde ein automatischer Ansatz in den Workflow integriert, mit welchem aus der Scan-Punktwolke Bodenpunkte extrahiert werden können (vgl. Abbildung 6a). Diese werden genutzt, um für jedes Pixel des jeweiligen Kamerabildes, den Entfernungswert zu interpolieren und so ein entsprechendes Tiefenbild zu erzeugen (vgl. Abbildung 6b). Schließlich wurden Funktionen implementiert, die unter Verwendung von Kalibrierwerten, Orientierungsparametern und der Tiefenkarte einer Kamera, die Umrechnung und Ausgabe der Trajektorien von Bild- in Weltkoordinaten ermöglichen (vgl. Abbildung 6c).

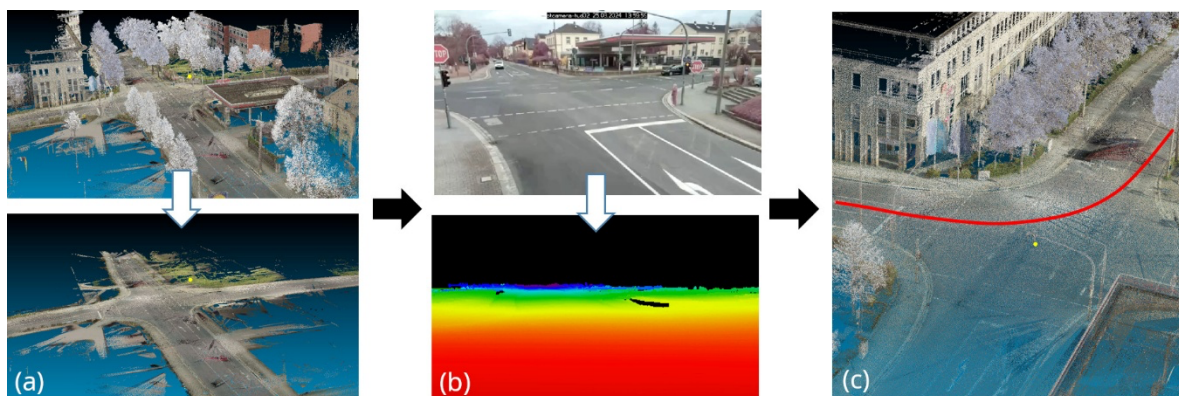


Abbildung 6: Schritte zur Transformation von 2D Bildkoordinaten in 3D Weltkoordinaten

6.3.3.3 Trajektorienzuordnung

Bevor eine Fusion redundanter Trajektorien erfolgen kann, müssen zunächst dem gleichen Objekt zugehörige Trajektorien, die von unterschiedlichen Kameras stammen, identifiziert und einander zugeordnet werden. Diese Zuordnung erfolgt zeitlich und räumlich.

Da durch verrauschte oder fehlerbehaftete Trajektorien Fehlzusammenordnungen verursacht werden, wurde dem Verfahren ein Ausreißertest sowie ein einfacher Glättungsalgorithmus vorgeschaltet, bei dem ein Gleitender-Mittelwert-Filter jeweils auf die X- und Y- Koordinaten der Trajektorien angewandt wird, wodurch vorhandenes Rauschen in den Trajektorien abgedämpft werden kann.

Zeitliche Zuordnung: Aus den detektierten bildbasierten 2D-Trajektorien werden 3D-Trajektorien berechnet und deren Zeitstempel mit Hilfe der ermittelten Frameoffsets korrigiert, so dass Trajektorien in Weltkoordinaten mit einheitlichem Zeitbezug vorliegen. Die Trajektorien einer Kamera werden frameweise untersucht. Für jede Trajektorienposition werden die zeitidentischen Frames der anderen Kameras mit ihren Objektdetektionen identifiziert (Abbildung 8a). Befinden sich zu dem jeweiligen Zeitpunkt mehrere Verkehrsteilnehmende im Sichtbereich der Kameras, ist aus diesen nun der jeweils passende Kandidat auszuwählen.

Räumliche Zuordnung: Um die bestehenden Mehrdeutigkeiten nach der zeitlichen Zuordnung aufzulösen werden die 3D-Weltkoordinaten der gefundenen Kandidaten betrachtet. Da bei Objekten größerer Ausdehnung der Mittelpunkt der Grundfläche, auf welchen sich die jeweilige Trajektorienkoordinate beziehen soll, nicht direkt beobachtet und gemessen werden kann, muss dieser auf geeignete Weise approximiert werden. Der Mittelpunkt der unteren Linie der Boundingbox des detektierten Objektes stellt eine erste grobe Näherung dar. Da mehrere Kameras das Objekt von verschiedenen Seiten beobachten gruppieren sich in diesem Fall die resultierenden Punkte im Weltkoordinatensystem in der Größenordnung der Ausdehnung des Objektes um das Objekt herum (Abbildung 8b). Dies erschwert die eindeutige Zuordnung der Punkte und damit der Trajektorien zum zugehörigen Objekt.

Eine Approximation des Fahrzeugmittelpunktes erfolgt über den jeweiligen vertikalen Blickwinkel α der Kamera auf das Fahrzeug (Abbildung 7). Die beiden Extremfälle für den möglichen Blickwinkel stellen zum einen der Frontalblick mit $\alpha=0^\circ$ und die Draufsicht auf das Objekt mit $\alpha=90^\circ$ dar. Im ersten Fall entspricht der Fahrzeugmittelpunkt der Mitte der unteren Boundingboxlinie. Im zweiten Fall entspricht er der Boundingboxmitte. Entsprechend des berechneten Blickwinkels für jeden Trajektorienpunkt wird der die Fahrzeugmitte repräsentierende Punkt innerhalb der Box linear zwischen diese beiden Grenzfälle interpoliert.

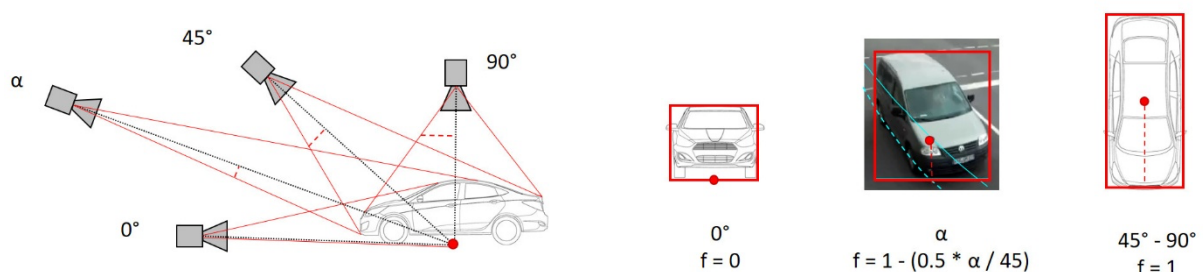


Abbildung 7: Approximieren des Fahrzeugmittelpunktes

Nun können die Punkte über ihre räumliche Nähe zueinander als zum gleichen Objekt zugehörig identifiziert werden und über Abstandskriterien besser von Punktgruppen abgegrenzt werden, welche ein anderes Objekt repräsentieren (Abbildung 8c).

6.3.3.4 Trajektorienfusion

Für die Fusion der Trajektorien werden die jeweils korrespondierenden Positionen entlang der einzelnen Trajektorien gemittelt (Abbildung 8d). Da die Positionsgenauigkeit der verschiedenen Kamerasysteme variiert – abhängig von der Entfernung zum Objekt sowie vom Einfallswinkel der Kamera – erfolgt die Mittelung gewichtet. Dabei fließen sowohl die Kameradistanz als auch der Einfallswinkel als gewichtende Faktoren in die Berechnung ein.

Vor der Mittelung ist jedoch sicherzustellen, dass die betrachteten Trajektorienpunkte tatsächlich demselben Objekt zugeordnet werden können. Da sich temporäre Punktansammlungen auch zwischen verschiedenen Fahrzeugen bilden können, die sich kurzzeitig nahe beieinander befinden, ist eine entsprechende Prüfung notwendig. Hierzu wird im Fall mehrerer potenzieller Übereinstimmungen innerhalb einer Kamera der jeweils nächstgelegene Punkt ausgewählt. Zusätzlich wird überprüft, ob über die gesamte Länge der Trajektorien hinweg eine ausreichende Anzahl gemeinsamer Punkte vorhanden ist, sodass eine tatsächliche Übereinstimmung vorliegt und nicht lediglich eine kurzzeitige Annäherung („Berührung“) zweier Trajektorien stattgefunden hat.

Die gemittelten Positionen entlang der zusammengehörenden Trajektorien ergeben schließlich die resultierende mittlere Gesamttrajektorie (Abbildung 8d - rote Linie).

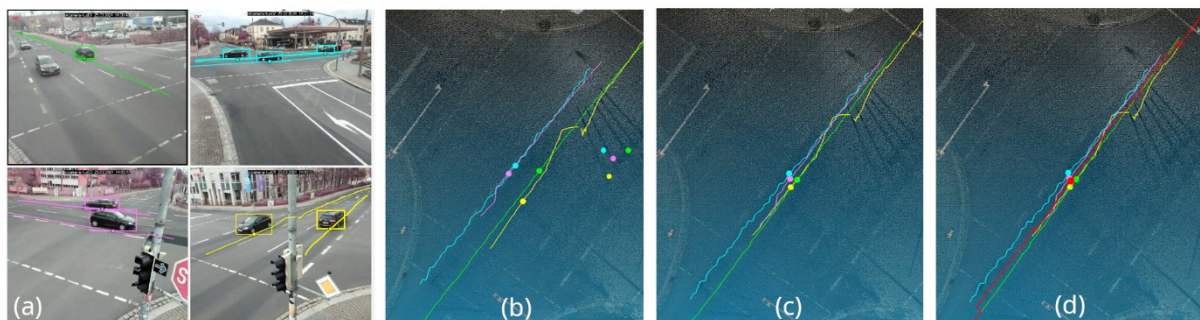


Abbildung 8: Trajektorienzuordnung und Trajektorienfusion

6.3.3.5 Flächenwahl

Die in Abbildung 8b dargestellten Positionen aus den einzelnen Kameras repräsentieren im Idealfall jeweils die Außenbegrenzung des beobachteten Fahrzeugs. Um zu untersuchen, ob sich aus diesen Positionen Rückschlüsse auf die Grundfläche und -form des Fahrzeugs ableiten lassen und damit auch auf die Flächenwahl des Fahrzeuges, wurde ein Ansatz entwickelt, bei dem für jede Trajektorienposition die jeweilige Boundingbox in das Weltkoordinatensystem transformiert wird.

Sind ausreichend viele Kameras mit guter Sichtabdeckung auf das Fahrzeug vorhanden, überlagern sich die projizierten Bereiche der transformierten Boundingboxen. Aus diesen Überlagerungen ergibt sich eine Fläche, innerhalb derer sich die Grundfläche des Fahrzeugs befinden sollte. Die Eckpunkte der Fahrzeuggrundfläche liegen dabei auf den transformierten Unterkanten der Boundingboxen (vgl. Abbildung 9a).

Eine einzelne Überlagerungsfläche für eine bestimmte Trajektorienposition erlaubt jedoch noch keine belastbaren Aussagen über Form und Größe der Fahrzeuggrundfläche. Da sich die Kameraperspektiven auf das Fahrzeug jedoch ändern, während es sich entlang der Trajektorie weiterbewegt, ändern sich auch die durch die Boundingboxen aufgespannten Flächen (vgl. Abbildung 9b).

Werden diese Überlagerungsflächen für jede Position entlang der Trajektorie berechnet und anschließend – unter Berücksichtigung der Bewegungsrichtung – übereinandergelegt, entstehen überlagerte Bereiche mit unterschiedlicher Häufigkeit der Flächenbedeckung. Daraus lässt sich eine Art „Heatmap“ ableiten (vgl. Abbildung 9c), in die ein Rechteck eingepasst wird. Die Maße dieses Rechtecks liefern Rückschlüsse auf die Dimensionen der Fahrzeuggrundfläche.

Für den Erfolg dieser Methode sind für die transformierten Boundingboxen Lagegenauigkeiten im Zentimeterbereich erforderlich. Solche Genauigkeiten sind prinzipiell mit kamerabasierten Messsystemen erreichbar. Im Kontext verkehrsanalytischer Anwendungen im öffentlichen Raum sind jedoch aus datenschutzrechtlichen Gründen nur begrenzte Bildauflösungen zulässig. Diese ermöglichen typischerweise nur Genauigkeiten im Bereich von einigen Dezimetern (vgl. Absatz 6.3.4). Damit ist das Verfahren für den praktischen Anwendungsfall im öffentlichen Raum nicht geeignet.

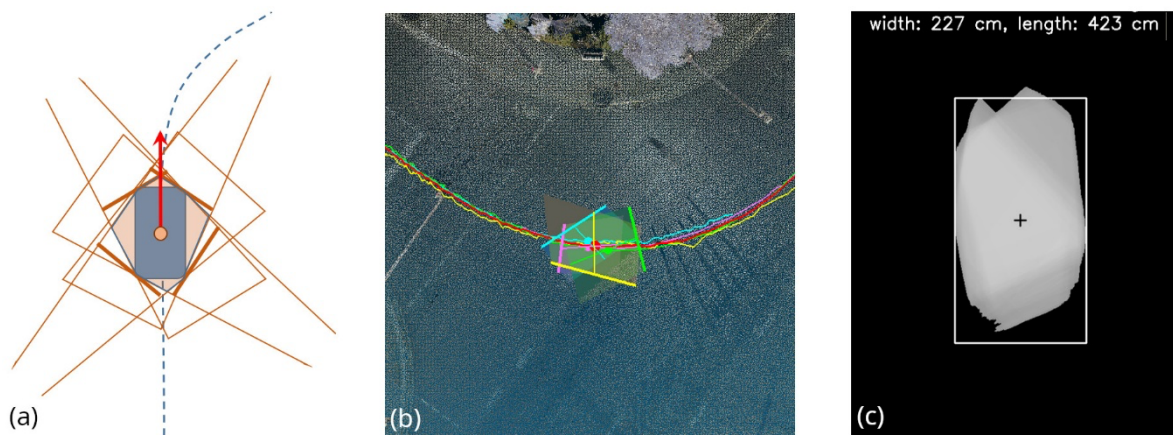


Abbildung 9: Methodik zur Bestimmung der Fahrzeuggrundfläche

6.3.4 Genauigkeitsaspekte

Zur Einordnung der Genauigkeit, der aus dem Mehrkamerasystem resultierenden, gemittelten Trajektorien erfolgte ein Vergleich mit Kontrolltrajektorien, welche mittels INS/GNSS durch ein Messfahrzeug erhoben wurden. Vom Messfahrzeug wurde eine Testkreuzung in verschiedene Richtungen befahren während gleichzeitig mit einem aus 4 Kameras bestehenden Mehrkamerasystem Videos aufgenommen wurden. In Abbildung 10 sind die Orthogonalabweichungen der mehrkamerabasiert gemessenen Trajektorien von den direkt gemessenen INS/GNSS-Trajektorien des Messfahrzeugs dargestellt. Die Kontrolltrajektorien besitzen eine Genauigkeit von wenigen Zentimetern bis ca. 1 Dezimeter. Etwa die Hälfte der gemittelten Trajektorienpunkte fällt in diesen Genauigkeitsbereich. Insgesamt ergeben sich Abweichungen im Bereich von wenigen Dezimetern. Diese können systematisch die gesamte Trajektorie betreffen oder nur vereinzelte Punkte bzw. kleinere Trajektorienabschnitte. Während für erstere eher Fehlereinflüsse ursächlich sind, welche aus der Georeferenzierung herrühren und auf die gesamte Trajektorie wirken, resultieren letztere

eher aus Fehlereinflüssen, die bei der bildbasierten Objektdetektion und beim Tracking lokal begrenzt innerhalb einer Trajektorie auftreten können.

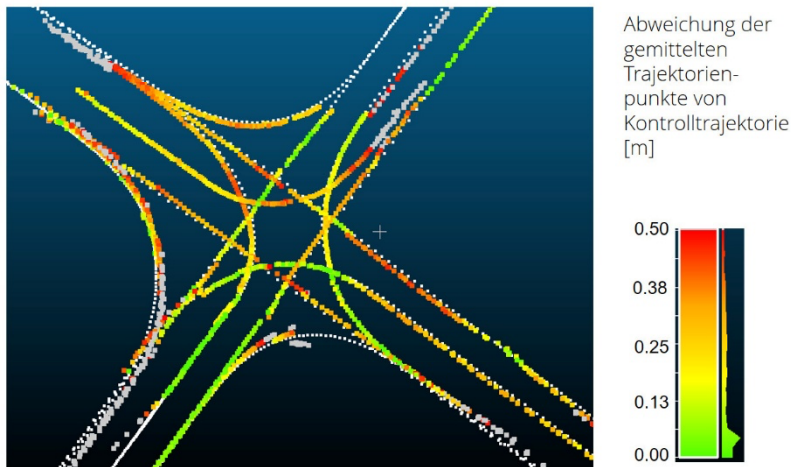


Abbildung 10: Lagegenauigkeit der gemittelten Trajektorien

6.3.5 Weiterverwendung

Werden Mehrkamerasysteme an einem Verkehrsknotenpunkt eingesetzt, kann OT-Multicamera optional in den Auswerteprozess eingebunden werden, um fusionierte Trajektorien zu erzeugen, welche wiederum in der OpenTrafficCam-Auswertepipeline weiterverwendet werden können. Die fusionierten Trajektorien besitzen folgenden Mehrwert gegenüber Trajektorien aus Einzelkameraauswertungen:

Vermeidung von Trajektorienabbrüchen infolge von Verdeckungen: Wird ein Objekt aufgrund von Verdeckungen im Sichtfeld einer Kamera zeitweise nicht erkannt, kann dies zu einem Abbruch der zugehörigen Trajektorie führen. Alternativ entstehen zwei separate Trajektorienfragmente für dasselbe Objekt, die jedoch nicht miteinander verknüpft sind und somit nicht als zusammengehörig erkannt werden.

Wie in Abbildung 11a dargestellt, können in solchen Fällen die Trajektorien aus den übrigen Kameraperspektiven diese Lücken schließen. Sie ermöglichen die Identifikation von Trajektorienfragmenten und deren korrekte Zuordnung zum entsprechenden Objekt, wodurch eine durchgängige Rekonstruktion der Objektbewegung gewährleistet wird.

Erweiterung des Sichtbereichs für ein Objekt: Mit einer Einzelkamera lassen sich die verschiedenen Abbiegespuren und Bewegungsrichtungen der Verkehrsteilnehmenden nicht gleichwertig erfassen. Trajektorien, die in unmittelbarer Nähe zur Kamera entstehen, können in der Regel mit höherer Genauigkeit lokalisiert werden, sind jedoch häufig kürzer. Für weiter entfernte Trajektorien steht zwar ein größerer Sichtbereich zur Verfügung, jedoch erfolgt deren Erfassung meist unter einem flacheren Blickwinkel, was die Positionsgenauigkeit negativ beeinflussen kann.

Diese jeweils spezifischen Nachteile werden durch die Auswertung mehrerer Kameras und die gewichtete Mittelung der daraus abgeleiteten Trajektorien weitgehend kompensiert. Darüber hinaus ist der Beobachtungsbereich einer fusionierten Trajektorie stets größer als jener der einzelnen Kameras. Der Einsatz von Mehrkamerasystemen ermöglicht somit eine signifikante und flexible Erweiterung des Messbereichs für einzelne Trajektorien (vgl. Abbildung 11b).

Identifizierung und Eliminierung von Ausreißern: Je mehr Kameras zur Berechnung einer gemittelten Trajektorienposition beitragen, desto besser ist diese Position kontrolliert. Durch den Vergleich der Einzelbeobachtungen können Ausreißer zuverlässig identifiziert und aus der Berechnung ausgeschlossen werden (vgl. Abbildung 11c). Dies trägt wesentlich zur Erhöhung der Gesamtgenauigkeit der rekonstruierten Trajektorien bei.

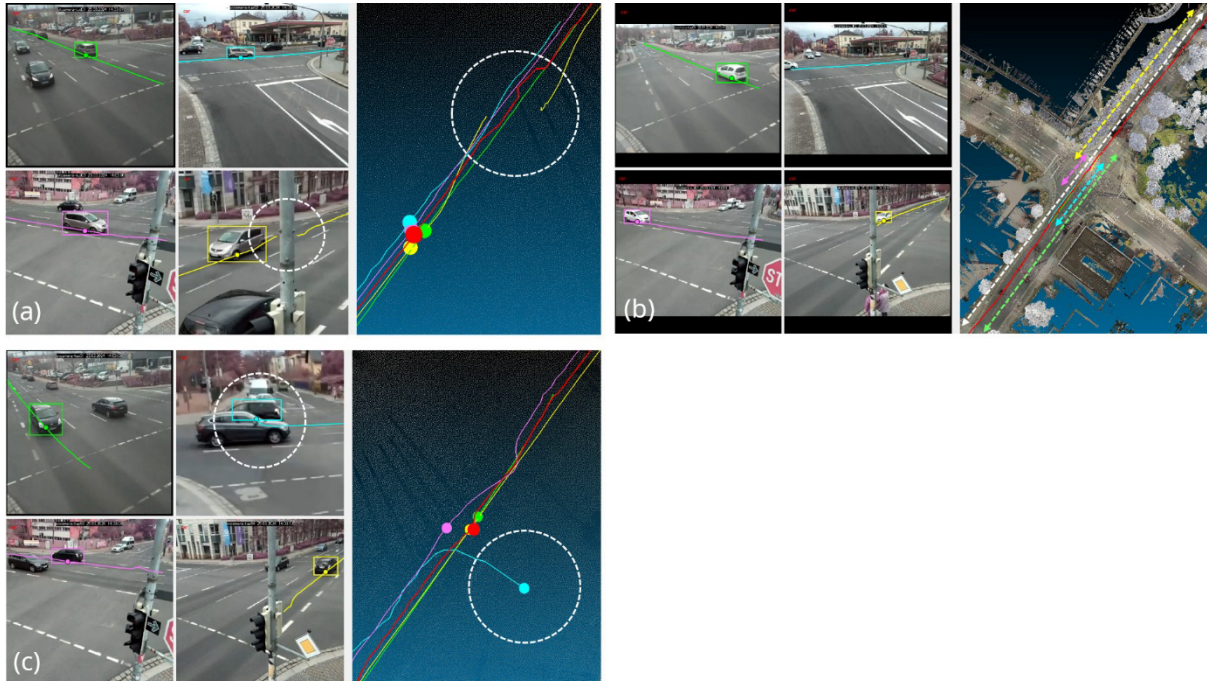


Abbildung 11: Vorteile der Trajektorienbestimmung mit Mehrkamerasystemen

6.4 AP 4 OTLabels (platomo)

OTLabels bildet ein zentrales Modul innerhalb von OpenTrafficCam. OTLabels fokussiert sich auf die Bereitstellung annotierter Bilddaten von Verkehrsteilnehmern sowie Algorithmen zum Training von Objekterkennungsmodellen auf individuellen Datensätzen.

Das Modul stellt einen systematischen Workflow zur Nachschulung von KI-Modellen zur Verfügung, der auf maßgeschneiderten Videoframes und bewährten Open-Source-Softwarepaketen basiert. Diese Spezialisierung ist notwendig, da standardmäßige, auf internationalen Datensätzen trainierte Modelle oft nicht die erforderliche Präzision für die spezifischen Anforderungen deutscher Verkehrssituationen und -teilnehmer erreichen.

Das System nutzt bewährte Python-Bibliotheken wie NumPy, pandas, OpenCV und ultralytics. Eine leistungsstarke GPU mit CUDA-Unterstützung für effizientes Training wird dringend empfohlen. Zentraler Baustein der Architektur ist das FiftyOne-Framework, das als Datenmanagement-Hub fungiert und die nahtlose Integration zwischen verschiedenen Workflow-Komponenten ermöglicht.

6.4.1 Aufbau einer Bilddatenbank mit mind. 10.000 Bildern (MSP)

Der systematische Aufbau der Bilddatenbank erfolgte in einem mehrstufigen, iterativen Verfahren. Zunächst wurde das vorhandene Videomaterial aus rund fünfzig unterschiedlichen Kamerapositionen systematisch gesichtet. Unter Berücksichtigung wesentlicher Einflussgrößen – Witterung und Beleuchtung, Blickwinkel, Auflösung sowie verschiedene Kameramodule – wurden in drei aufeinanderfolgenden Iterationen zunächst 5.000 Einzelbilder extrahiert, anschließend nach

einer ersten Gap-Analyse weitere 5.000 und schließlich noch einmal 2.000 Bilder ergänzt. Damit lagen nach Abschluss der projektspezifischen Extraktion rund 12.000 Bilder vor.

Um die Szenenvielfalt weiter zu erhöhen, flossen zusätzlich rund 5.000 Einzelbilder aus einem thematisch verwandten Forschungsprojekt in den Bestand ein. Diese Aufnahmen wurden nach denselben Auswahl- und Qualitätskriterien geprüft, sodass sie sich nahtlos in den bestehenden Korpus einfügen konnten.

Alle Bilder wurden in einer klaren Ordnerstruktur abgelegt, mit zentralen Metadaten versehen und anschließend in eine Bilddatenbank auf Basis des Open-Source-Frameworks FiftyOne überführt. FiftyOne ermöglichte dabei das Einlesen großer Bildmengen, das Anreichern mit frei definierbaren Attributen (beispielsweise Szenentyp, Witterungskategorie oder Kameratyp) sowie das komfortable Filtern und Versionieren des Bestands.

Durch die Kombination aus sorgfältiger Videosichtung, diversitätsorientierter und iterativer Bildextraktion sowie der datenbankgestützten Ablage mittels FiftyOne steht nun ein homogener, strukturierter und skalierbarer Bildkorpus von insgesamt rund 17.000 Einzelbildern zur Verfügung. Diese grundlegende Kategorisierung nach Aufnahmestandorten, Kameratypen und zeitlichen Parametern ermöglichte spätere komplexe Filteroperationen und gewährleistete die Nachvollziehbarkeit der Datenherkunft.

6.4.2 Erweiterung des Workflows der Vorstudie zum teilautomatisierten Annotieren (platomo)

Die Weiterentwicklung des Annotationsworkflows basierte auf einer vollständigen Automatisierung der Datenaufbereitung und des CVAT-Uploads. Ein wesentlicher Effizienzgewinn wurde durch die automatisierte Pre-Annotation erreicht. Diese Komponente nutzt ein Standardmodell oder das aktuell beste YOLO-Modell zur Vorannotation der Bildframes. Obwohl die Ergebnisse zunächst nicht die gewünschte Qualität erreichten, sparte die Pre-Annotation erhebliche Zeit, da nicht alle Labels von Grund auf neu gezeichnet werden mussten.

Das intelligente Filtersystem von FiftyOne ermöglichte komplexe Auswahlkriterien:

- Statusbasierte Filterung: Nur Samples mit Status »imported« oder »pre-annotated« wurden für neue Annotationsrunden berücksichtigt.
- Klassenbasierte Filterung: Spezifische Verkehrsteilnehmerklassen konnten ein- oder ausgeschlossen werden.
- Standortbasierte Auswahl: Geografische oder kameratypspezifische Einschränkungen.
- Temporale Filterung: Zeitraumbasierte Datenauswahl für saisonale oder ereignisspezifische Modelle.

Die Integration mit CVAT erfolgte über eine dedizierte OpenTrafficCam-Instanz. Das System wurde als Docker-Container implementiert und bot teambasierte Annotationsmöglichkeiten mit Benutzerregistrierung und E-Mail-Verifikation. Der automatisierte Workflow für Aufbereitung und Upload in CVAT eliminierte manuelle Zwischenschritte und gewährleistete eine konsistente Datenübertragung.

6.4.3 Pretest mit Annotation (MSP) und Modelltraining mit anschließender Optimierung des Workflows (platomo)

Die Validierung des entwickelten Workflows erfolgte durch einen Pretest, der sowohl die Annotationsqualität als auch die Trainingseffizienz evaluierte. Dabei wurde eine repräsentative Teilmenge

der Bilddatenbank durch das CVAT-System prozessiert und die resultierenden Annotationen für ein erstes Modelltraining genutzt.

Der Export zu CVAT wurde durch die CVAT-Klasse orchestriert, die gefilterte Datensätze in Segmente von typischerweise 100–500 Bildern unterteilt und verschiedenen Annotatoren zuwies. Die Annotationsqualität wurde durch vordefinierte Klassenschemata (17 Klassen) sichergestellt, die spezifisch für deutsche Verkehrssituationen entwickelt wurden.

Nach Abschluss der manuellen Annotation erfolgte der Datenimport von CVAT zurück in das Fifty-One-System. Das System konvertierte dabei die CVAT-Annotationen in das COCO-Format, das von YOLO benötigt wird. Nicht in der Konfigurationsdatei definierte Labels wurden automatisch gelöscht, um Konsistenz zu gewährleisten.

Die Erkenntnisse aus dem Pretest führten zu einer kontinuierlichen Optimierung des Workflows, insbesondere bei der Segmentierung der Annotationsaufgaben und der Qualitätskontrolle der resultierenden Labels.

6.4.4 Annotation der gesamten Bilddatenbank MSP)

Nach erfolgreicher Validierung und Optimierung wurde die vollständige Bilddatenbank von 17.000 Einzelbildern systematisch annotiert. Dieser umfangreiche Annotationsprozess resultierte in einem detaillierten Datensatz mit insgesamt 208.890 individuell markierten Objekten, die sich auf siebzehn spezialisierte Verkehrsteilnehmerklassen verteilen:

Tabelle 1: Übersicht der Anzahl annotierter Objekte

KLASSE	ANZAHL LABELS
FUßGÄNGER (INKL. ESCOOTER)	64.740
RADFAHRENDE (INKL. ANHÄNGER & LASTENRAD)	20.133
MOTORRADFAHRER	2.081
PKW (INKL. ANHÄNGER)	86.968
PRIVATVAN (INKL. ANHÄNGER)	4.541
BUS	1.816
LIEFERWAGEN (INKL. ANHÄNGER)	15.039
LASTWAGEN	6.200
LASTWAGEN MIT (SATTEL-)ANHÄNGER	6.762
ZUG	610

Hinweis: Die in der Tabelle dargestellten zehn Hauptkategorien fassen die siebzehn feineren Klassen zusammen, die für das Training des vollständigen 17-Klassen-Modells maßgeblich sind.

Das einheitliche OTC-Klassenschema in der Version 1.2 bildet die Grundlage für alle nachgelagerten Verarbeitungsschritte. Es definiert siebzehn Kategorien. Gegenüber früheren Iterationen wurden insbesondere die Van-Klassen aufgesplittet, eine eigenständige Kategorie für Cargo-Bikes geschaffen und die Anhängerbeziehung bei Lkw standardisiert.

6.4.5 Aufbau und Training Detektions-Modelle (Fuß/Rad, 3+0, 8+1) für OTCamera und Drohnen (platomo)

Basierend auf dem umfangreichen annotierten Datensatz wurden verschiedene spezialisierte Detektionsmodelle entwickelt. Das Training erfolgte über die Ultralytics YOLO v8-Implementierung,

wobei die aufbereiteten Daten als Input dienen. Das System erkannte automatisch verfügbare CUDA-Hardware und optimierte die Trainingsparameter entsprechend.

Ein entscheidender Vorteil des OTLabels-Workflows liegt in der Möglichkeit iterativer Modellverbesserung. Nach jeder Trainingsrunde können neue Videoframes mit dem aktualisierten Modell pre-annotiert werden, wodurch sich die Qualität der automatischen Vorhersagen kontinuierlich verbessert und der manuelle Annotationsaufwand reduziert.

Es wurde ein vollständiges 17-Klassen-Modell entwickelt, das alle annotierten Verkehrsteilnehmer-typen unterscheiden kann und beliebige Aggregationen für verschiedene Analysezwecke ermöglicht. Die Modelle wurden sowohl für stationäre OTCamera-Systeme als auch für Drohnenaufnahmen optimiert, wobei die unterschiedlichen Perspektiven und Auflösungen berücksichtigt wurden. Zusätzlich wurde ein dezidiertes Modell mit 8+1 Klassen nach TLS 2012 [31] trainiert.

6.4.6 Veröffentlichung und Integration des Modells in OTVision (platomo)

Die finalen Modelle durchliefen eine umfassende Evaluierung hinsichtlich ihrer Veröffentlichungsfähigkeit, da die Modelle Teil des Geschäftsmodells von platomo darstellen. Die durch OTLabels trainierten Modelle werden direkt in OTVision integriert, um präzisere Verkehrsteilnehmerdetektion zu ermöglichen. Dies führt zu detaillierteren und individuelleren Verkehrsdaten, von klassischen Zähldaten über Geschwindigkeitsmessungen bis hin zu komplexen Trajektorienanalysen.

Die Intention für die Veröffentlichung eines einfachen Modells (beispielsweise das 3+0-Modell) ist vorhanden und wird als Beitrag zur Open-Source-Community von OpenTrafficCam realisiert. Dies ermöglicht es Forschungseinrichtungen und Verkehrsplanern, grundlegende Verkehrsanalysen durchzuführen, während spezialisierte Modelle für kommerzielle Anwendungen verfügbar bleiben.

6.5 AP 5 OTVision (platomo)

Das Modul OTVision zur Detektion und zum Tracking der Verkehrsteilnehmenden in den Videos wurde in verschiedener Weise weiterentwickelt. Im hier beschriebenen Teilvorhaben wurden ein Refactoring der Softwarearchitektur nach den Prinzipien der Clean Architecture geplant und umgesetzt. Während dieser Entwicklung wurde auch bereits der Betrieb auf dem Hochleistungsrechner der TU Dresden gemeinsam mit der TU Dresden integriert. Es wurde für OTVision eine einfache, erste Release-Strategie entwickelt, welche regelmäßige und halbautomatisierte Software-Releases ermöglicht. Im Jahr 2023 wurden die Datenformate für die Ausgabedateien finalisiert. Das Format ist in Form von Quellcode beschrieben.

6.5.1 Wiederkehrende Optimierung Performance und Zuverlässigkeit der Software (platomo)

Während der gesamten Laufzeit des Projekts wurde kontinuierlich an der Zuverlässigkeit und Performance von OTVision gearbeitet. Durch den intensiven Einsatz in praktischen Anwendungen bei der platomo GmbH konnten zahlreiche Fehler identifiziert und behoben werden. Zudem wurden durch die Anforderungen großer Messkampagnen gezielt Performance-Verbesserungen vorgenommen. Die aktuelle Version wurde bereits in mehreren BAST-Projekten, mehreren Kundenprojekten und einem EU-Projekt erfolgreich von der platomo GmbH eingesetzt.

6.5.2 Modularisierung der Detektions-Algorithmen (platomo)

Ein wesentlicher Entwicklungsschritt war die Modularisierung von OTVision, um insbesondere den Detektionsalgorithmus flexibel austauschbar zu gestalten. Diese Kapselung ermöglicht es nun, unterschiedliche Detektionsmodelle nach Bedarf einzusetzen. Unterstützt werden derzeit mehrere Modelle der YOLO-Familie oder daraus abgeleitete Modelle, konkret YOLOv5, YOLOv8, YOLOv11 sowie YOLOv12. OTVision kann dadurch flexibel in unterschiedlichen Anwendungsszenarien eingesetzt werden.

6.5.3 Implementierung Key-Feature Algorithmen zur Detektion Fuß- und Radverkehr (MSP)

Neben der Detektion durch das Erkennen des Gesamtobjektes wie beispielsweise YOLO existieren auch Algorithmen die auf der Erkennung von Key-Features setzen. Dadurch werden z.B. die Beine, der Torso oder die Arme der zu Fuß Gehenden und der Radfahrenden als eigenes Objekt erkannt. Dies kann Vorteile wie eine Erkennung auch bei einer Verdeckung des Großteils des zu erkennen- den Objektes sowie Informationen über das Verhalten der Verkehrsteilnehmer mit sich bringen.

Es wurden die open-source Key-Feature Algorithmen AlphaPose [32] und OpenPose [33] auf durch OTCamera aufgenommene Videos von zu Fuß Gehenden und Radfahrenden angewendet. Dabei stellte sich heraus, dass diese Algorithmen auf die in der Verkehrsbeobachtung typischen Entfernung weder die zu Fuß Gehenden noch die Radfahrenden ausreichend gut erkannten. Die mit den Key-Feature Algorithmen verbundenen Vorteile konnten damit auch nicht genutzt werden. Eine Implementierung von open-source Key-Feature Algorithmen wurde somit nicht weiterverfolgt.

6.5.4 Optimierung Detektionsalgorithmen für das GPU-Cluster der TU Dresden (MSP)

Ein wichtiger Meilenstein war die erfolgreiche Installation und Optimierung von OTVision auf den HPC-Systemen der TU Dresden. Versuche, OTVision in Container-Formaten für einen vereinfachten Betrieb bereitzustellen, erwiesen sich allerdings als nicht erfolgreich, da die Containerformate inkompatibel waren und keine Laufzeitvorteile boten. Dagegen brachte die direkte Konvertierung der Modelle auf die Hardware-Architektur des GPU-Clusters erhebliche Performance-Gewinne. Zusätzlich wurde OTVision erweitert, um native Modellformate sowohl auf CUDA-GPUs als auch auf macOS-Systemen effizient verarbeiten zu können.

6.5.5 Modularisierung und Implementierung weiterer Tracking-Algorithmen (MSP)

Das Tracking-Modul wurde ebenfalls umfassende modularisiert, wodurch Tracking-Algorithmen leicht austauschbar wurden. Neben eigenen Implementierungen sind nun standardisierte Verfahren wie deepSORT, centroid tracker mit Kalman-Filter, VIOU und BoT-SORT verwendbar.

Zusätzlich wurde im Rahmen einer studentischen Abschlussarbeit [34] eine eigene Erweiterung des IOU-Trackers entwickelt und getestet. Diese nimmt die im OpenTrafficCam-Framework erzeugten Trajektorien als Ausgangspunkt, um typische Fehlerquellen im detektionsbasierten Tracking zu identifizieren und zu beheben. Es wird eingangs gezeigt, dass insbesondere kurzfristige Objektverdeckungen durch andere Fahrzeuge oder statische Hindernisse im Verkehrsraum zu fragmentierten Trajektorien führen, während bei Fußgängergruppen ungenaue Detektionen die Hauptursache für Lücken und Verwechslungen sind. Um diesen Effekten entgegenzuwirken, wird der in OTVision implementierten IOU-Tracker um eine Heuristik erweitert, die unter Annahme kon-

tinuierlicher Bewegungsverläufe fehlende Objektpositionen extrapoliert und bei Wiederauftauchen erneut mit demselben Track verknüpft. Für die Bewertung dieser Erweiterung wird ein eigener, aussagekräftiger Ground-Truth-Zähl Datensatz erstellt und umfangreiche Experimente durchgeführt. Die Ergebnisse belegen, dass der Extrapolated IOU-Tracker (EIOU) die Anzahl unvollständiger Trajektorien je nach Fahrzeugklasse um 9,7 % bis 44,1 % reduziert und die korrekte Verfolgung von Fußgängern um 6,5 % erhöht, während der Zeitaufwand durch die zusätzlichen Berechnungen auf rund 150 % der ursprünglichen Tracking-Geschwindigkeit ansteigt. Damit demonstriert die Arbeit, dass einfache, domänenbasierte Heuristiken das detektionsbasierte Tracking im Verkehrsbereich substantiell verbessern können, ohne die Gesamtperformanz unakzeptabel zu beeinträchtigen. Dieser Beitrag liefert eine fundierte methodische Grundlage für die Integration einer Trajektorienextrapolation in OTVision und unterstützt das Gesamtprojekt OpenTraffic-Cam_live maßgeblich bei der Steigerung der Messqualität und Reduzierung des Validierungsaufwands.

6.5.6 Identifikation und Vervollständigung korrupter Trajektorien (MSP, PHOTO)

Um zu verstehen, an welchen Stellen Trajektorien im verarbeiteten Datenmaterial abbrechen, wurde eine Trajektorienanalyse durchgeführt. Dabei zeigte sich, dass Abbrüche von Objekttrajektorien vor allem durch Verdeckungen verursacht werden. Bereits das kurzzeitige Verdecken eines Objekts – beispielsweise durch einen Mast oder einen anderen Verkehrsteilnehmenden – kann ausreichen, um den Sichtkontakt zu unterbrechen und somit die Trajektorie abzureißen, siehe Beispiele in Abbildung 12, Abbildung 13, Abbildung 14 und Abbildung 15.



Abbildung 12: Verdeckung durch die Umgebung



Abbildung 13: Verdeckung durch einen anderen Verkehrsteilnehmenden



Abbildung 14: Fehlende Detektion



Abbildung 15: Falsche Detektion und verschobene Bounding Box

Um derartige Lücken in den Trajektorien zu schließen, wurden manuelle Strategien zur Vervollständigung und Verknüpfung fragmentierter Trajektorien entwickelt. Eine dieser Strategien basiert auf vordefinierten Bereichsabschnitten (sogenannten *AreaSections*), innerhalb derer gezielt nach einer Fortsetzung der Trajektorie gesucht wird. Durch die manuelle Definition solcher Bereiche und die fokussierte Suche innerhalb dieser *Sections* konnten zusammengehörige Trajektoriensegmente identifiziert und anschließend miteinander verknüpft werden. Die manuelle Definition des Suchbereichs schränkt den Suchraum erheblich ein. Gleichzeitig wird bei der Verknüpfung von Trajektorien verkehrliches Wissen des Anwenders integriert, wodurch falsche Verknüpfungen verringert werden können.

Parallel zu den manuellen Verfahren wurden verschiedene Ansätze für eine automatische Verbindung getrennter Trajektorien erprobt. Zu den getesteten Methoden gehörten unter anderem:

- **Constraint-basierte Verknüpfung:** Ein regelbasierter Ansatz, der vordefinierte räumliche und zeitliche Bedingungen nutzt, um passende Trajektoriensegmente einander zuzuordnen.
- **Empirische Mustererkennung:** Ein Ansatz, der typische Bewegungsmuster aus den Daten ableitet und verwendet, um Fortsetzungen von Trajektorien anhand dieser Muster zu identifizieren.
- **Bewegungsvorhersage:** Ein prognosebasierter Ansatz, der die zukünftige Bewegung eines Objekts extrapoliert, um eine Lücke bis zum nächsten detektierten Auftauchen zu überbrücken.

Die Auswertung der verschiedenen Verknüpfungsmethoden zeigte klare Unterschiede in ihrer Wirksamkeit. Das manuelle Zusammenführen von Tracks mittels einer speziell konfigurierten *Merge-Section* erzielte dabei insgesamt die zuverlässigsten Ergebnisse. Diese Vorgehensweise erfordert lediglich einen geringen zusätzlichen Konfigurationsaufwand und führte in den Tests dazu, dass zusammengehörige Trajektorienabschnitte korrekt miteinander verbunden wurden.

Demgegenüber erwies sich das automatische Verbinden der Trajektorien mit den verfügbaren Daten als nur eingeschränkt erfolgreich. Keiner der getesteten automatischen Ansätze konnte in allen Fällen die unterbrochenen Trajektorien lückenlos rekonstruieren, insbesondere bei kurzzeitigen Teilverdeckungen der Objekte. In solchen Fällen stoßen die Algorithmen an ihre Grenzen, da die Unsicherheiten bei verdeckten Objekten häufig zu Fehlverknüpfungen oder verlorenen Tracks führen.

Aus den Erkenntnissen dieser Untersuchungen lässt sich ableiten, dass eine verbesserte Objekterkennung bei teilweiser Verdeckung von großer Bedeutung ist. Durch robustere Erkennungsverfahren, die Objekte auch bei teilweise verdeckter Sicht zuverlässig erkennen, ließen sich Trajektorienabbrüche künftig weiter verringern oder deren automatische Verknüpfung deutlich verbessern. Somit stellt die Weiterentwicklung derartiger Verfahren einen wichtigen Ansatzpunkt für die zukünftige Optimierung der Trajektorienverfolgung dar.

6.5.7 Ground Truth (MSP) & Annotations-Tools (platomo) zur Optimierung von Detektion & Tracking

Es wurde in diesem Projekt ein durchgängiger Werkzeugstapel aufgebaut, um robuste Trainings- und Validierungsdaten für die Objektdetektion sowie eine verlässliche Qualitätssicherung der resultierenden Trajektorien bereitzustellen. Herzstücke dieses Stapels sind OTGroundTruther zur Erstellung von Ground-Truth-Datensätzen, ein auf CVAT aufsetzender Annotation-Workflow (Modul *OTLabels*) und das OTC-Validation-Tool für die abschließende Plausibilitätsprüfung der automatisch erkannten Fahr- und Gehbewegungen.

OTGroundTruther entstand im zweiten Projektjahr und wurde als eigenständiges Open-Source-Werkzeug fertiggestellt. Die Anwendung dient dazu, auf Basis der von *OTAnalytics* erzeugten Ereignisse manuell verifizierte Referenztrajektorien anzulegen und damit sowohl die Parametrisierung als auch die spätere Gütebewertung der Detektionsalgorithmen zu unterstützen. Über die GitHub-Plattform wird der Quellcode öffentlich gepflegt und kann von Dritten weiterentwickelt oder im Rahmen eigener Studien genutzt werden.

Für das effiziente Labeln der Trainingsbilder wurde der in *OTLabels* (siehe 6.4) dargestellte CVAT-Workflow ausgebaut. Eine automatische Vor-Annotation mit jeweils aktuellen YOLO-Modellen reduziert den manuellen Aufwand erheblich; anschließende Korrekturen, Versionierung und Status-Verfolgung erfolgen in einer eigens entwickelten Pipeline. Die eigentliche Detailannotation wird auf einer selbst gehosteten CVAT-Instanz vorgenommen, wodurch die rund 12.000 Bilder umfassende Datenbank zeitnah fertiggestellt und für das kontinuierliche Re-Training in *OTLabels* bereitgestellt werden konnte.

Die resultierenden Trajektorien werden abschließend mit dem OTC-Validation-Tool geprüft. Neben einer interaktiven Sichtkontrolle bietet das Tool eine automatisierte Bewertung der Detektionssicherheit. Dazu wurde eine im Projekt einheitlich eingesetzte Metrik definiert, die die über alle Frames aufsummierten Klassenwahrscheinlichkeiten eines Tracks auswertet. Hierfür wird die intrinsische Wahrscheinlichkeit der verkehrlichen Klasse eines erkannten Objekts bestimmt. Das

Neuronale Netz exportiert für jede Detektion innerhalb eines Frames die Wahrscheinlichkeit, dass die detektierte Klasse korrekt bestimmt wurde. Werden mehrere Detektionen zu einer Fahrzeugbewegung (Trajektorie) zusammengefügt, liegen unterschiedliche Wahrscheinlichkeiten und evtl. sogar unterschiedliche detektierte Klassen vor. Eine Trajektorie erhält hierbei die Klasse der höchsten summierten Wahrscheinlichkeit aller detektierten Klassen. Zur Bewertung, ob dies korrekt ist, werden folgende Annahmen getroffen:

1. Je häufiger z.B. ein Fahrzeug innerhalb des Tracks der Track-Klasse zugeordnet wurde, desto wahrscheinlicher ist es, dass das Fahrzeug zu dieser Klasse gehört.
2. Wird die Track-Klasse in mehreren Frames mit sehr hoher Wahrscheinlichkeit zugeordnet, ist es wahrscheinlich, dass das Fahrzeug tatsächlich zu dieser Klasse gehört.
3. Dabei muss das Fahrzeug jedoch nicht in einem Großteil der Frames dieser Klasse zugeordnet worden sein.
4. Allerdings darf es auch nicht nur in einem einzelnen Frame zugeordnet worden sein (sofern die Trajektorie ausreichend lang ist).
5. Wird ein Fahrzeug in mehreren Frames als eine andere Klasse erkannt und ist die Wahrscheinlichkeit in diesen Frames hoch, sinkt die Wahrscheinlichkeit, dass das Fahrzeug zur Track-Klasse gehört.
6. Es ist wahrscheinlicher, dass ein Fahrzeug zur Track-Klasse gehört, wenn die anderen Detektionen verschiedenen Klassen zugeordnet worden sind und nicht alle derselben.

Punkt 1 kann adressiert werden, indem der Anteil der korrekt erkannten Detektionen ins Verhältnis zu allen Detektionen gesetzt wird. Die Punkt 2, 3 und 4 können durch die mittlere und die maximale Einzelwahrscheinlichkeit beschrieben werden. Durch das 90% Perzentil werden alle Punkte gleichzeitig adressiert. Punkt 5 wird berücksichtigt, indem für jede Klasse das 90% Perzentil der Einzelwahrscheinlichkeiten mit der Anzahl der Frames mit dieser Klasse multipliziert wird. Das gewichtete 90% Perzentil der Track-Klasse wird ins Verhältnis mit der Summe der gewichteten 90%-Perzentile aller Klassen gesetzt. Punkt 6 ist implizit durch die Bildung des Verhältnisses erfüllt, da bei vielen anderen Klassen die einzelnen 90% Perzentile niedriger sein werden als bei einer einzelnen anderen Klasse.

Die Wahrscheinlichkeit für der richtigen Klassenzuordnung eines Tracks definieren wir nachfolgender Formel:

$$P(trackclass) = Q_{90,trackclass} \cdot \frac{Q_{90,trackclass} \cdot N_{trackclass}}{\sum_{class}^n Q_{90,class} \cdot N_{class}}$$

mit

- $P(trackclass)$: Wahrscheinlichkeit der richtigen Klassenzuordnung eines Tracks
- Q_{90} : 90. Perzentil
- N : Anzahl der Frames in denen die Klasse erkannt wurde
- n : Anzahl der unterschiedlichen Klassen innerhalb einer Trajektorie

Die Punkte 2 und 3 adressieren hierbei Fahrzeuge, die nicht komplett im Bild sichtbar sind und dadurch wechselnde detektierte Klassen zugewiesen bekommen. Z.B. ein LKW mit Anhänger: zu-

nächst ist nur der LKW sichtbar, anschließend der LKW und der Anhänger in Teilen und abschließend nur der Anhänger. Solange der LKW allein im Bild ist, kann weder ein Mensch noch das neuronale Netz wissen, dass es sich um einen LKW mit Anhänger handelt. Sobald der Anhänger hinzukommt, ist eine korrekte Klassifizierung wiederum möglich. Je nach Aufnahmewinkel kann dabei jedoch bereits ein Teil des LKW wieder aus dem Bild verschwunden sein. Auf diese Weise befinden sich mehr Frames mit der Detektion „LKW“ in der Trajektorie als mit der Detektion „LKW mit Anhänger“. Die Punkte 2 und 3 sollen dieses Problem adressieren, um geeignete Lösungen dafür zu finden.

Diese Metrik wird auch bei der bundesweiten Straßenverkehrszählung 2024/2025 verwendet und ermöglicht so eine konsistente Qualitätsaussage über unterschiedliche Datensätze hinweg. Die manuelle Endkontrolle verbleibt dennoch als letzter Schritt, um etwaige Spezialfälle oder verdeckte Objekte zu prüfen.

Durch die Kombination aus sorgfältig annotierten Trainingsdaten, einem reproduzierbaren CVAT-Workflow und der beschriebenen Validierungsmethodik lässt sich die Erkennungs- und Klassifikationsgüte transparent nachweisen und – wo erforderlich – gezielt verbessern.

6.5.8 Implementierung offener Algorithmen zur mehrdimensionalen Bildstabilisierung (PHOTO)

Im Arbeitspaket 3 wurde zur Reduktion unterschiedlicher Störeinflüsse – von leichten Kameravibrationen über Luftflimmern bis hin zu detektionsbedingtem Pixelrauschen – eine mehrdimensionale Bildstabilisierung eingeführt. Herzstück ist ein gleitender Mittelwertfilter, der separat auf die X- und Y-Koordinaten jeder Trajektorie angewandt wird und so hochfrequentes Rauschen dämpft, bevor die Daten in nachgelagerten Schritten wie der Trajektorienfusion oder der Beschleunigungsberechnung genutzt werden. Dadurch lassen sich Bewegungen im Videobild weitgehend auf reale Objektbewegungen zurückführen und Scheinspitzen, die allein aus Messrauschen stammen, zuverlässig unterdrücken.

6.5.9 Darstellung der Trajektorien in Weltkoordinaten auf digitalen Orthophotos aus VGeoBund (PHOTO)

Die aus den Kamerabildern gewonnenen Trajektorien werden anschließend in ein metrisches Weltkoordinatensystem überführt und sowohl auf hochauflösende Orthophotos des VGeoBund als auch auf ein mittels Laserscanning erzeugtes 3D-Modell des untersuchten Knotenpunkts projiziert. Die Punktwolke liefert eine präzise dreidimensionale Repräsentation der Fahrbahnoberfläche; daraus abgeleitete Tiefenbilder ermöglichen, für jedes Trajektorienpixel den exakten Abstand zur Straße zu bestimmen und die 2D-Bildkoordinaten verlässlich in 3D-Weltkoordinaten umzusetzen. Auf diese Weise stehen Lage- und Bewegungsinformationen der Verkehrsteilnehmenden unmittelbar in georeferenzierten Plan- oder Simulationsumgebungen zur Verfügung und unterstützen sowohl klassische Orthofotoanalysen als auch weiterführende 3D-Auswertungen.

6.5.10 Zusammenfassung aktueller Stand

Die Software OTVision wird aktuell zur Auswertung von Videos aus Kurzzeiterhebungen eingesetzt. Sie ist eine Konsolenanwendung und besteht aus den drei Kernkomponenten *convert*, *detect* und *track*. *convert* konvertiert ein Video aus dem h264 (Aufnahmeformat OTCamera) in das mp4 Format, indem es die Metadaten zur Videolänge und zur Framerate ergänzt. *detect* detektiert in jedem Bild eines Videos Verkehrsteilnehmende. Hierzu werden das Video und das verwendete Detektionsmodell angegeben. *detect* verwendet PyTorch bzw. mit PyTorch erstellte Modelle zur Detektion.

Die Modelle lassen sich hierbei auf die Ausführungsumgebung optimieren, um eine höhere Performance zu erzielen. Modelle für CoreML (macOS) und TensorRT (Nvidia-GPU) wurden bereits erfolgreich getestet. Mittels TensorRT können bis zu 180 Bilder pro Sekunde aus OTCamera Videos auf einer Nvidia RTX 3090 verarbeitet werden.

track ermittelt aus den erzeugten Verkehrsteilnehmenden je Bild zusammenhängende Detektionsfolgen (Trajektorien bzw. Tracks). Dabei wird jeder Detektion eine Track-ID zugewiesen. Alle Detektionen mit derselben Track-ID gehören zum selben Verkehrsteilnehmenden. Für das Zusammenfügen von Detektionen zu einem Track wird eine von den OpenTrafficCam-Entwickelnden erweiterte Version des IOU-Trackers nach Bochinski et al. 2017 verwendet. Dieser lädt alle Detektionen eines Videos in den Speicher, verbindet diese zu Tracks und exportiert die Detektionen des Videos inklusive Track-Id. Alle Tracks, die am Ende des Videos noch nicht beendet sind, werden für das nächste zu verarbeitende Video vorgehalten.

6.6 AP 6 OTAnalytics (platomo)

OTAnalytics wurde im Rahmen des Projekts OpenTrafficCam_live grundlegend neu entwickelt, um aus den Trajektorien erkannter Verkehrsteilnehmender verkehrsrelevante Kennwerte zu gewinnen. Die Neuentwicklung begann Anfang 2023 mit dem Ziel, den prototypischen Ansatz aus der Vorstudie in ein marktreifes Minimalprodukt (MVP) zu überführen. Hierbei wurde großer Wert auf eine modulare Softwarearchitektur nach dem Clean-Architecture-Prinzip gelegt und eine geeignete UI-Technologie ausgewählt.

Die neue OTAnalytics-Software liest die mit dem Modul OTVision erzeugten Track-Dateien ein und ermöglicht darauf aufbauend die Definition konfigurierbarer Zählquerschnitte (Sections) und Verkehrsströme (Flows) als zusätzliche Eingabeparameter für die Analyse. Über die grafische Benutzeroberfläche können Anwender festlegen, an welchen virtuellen Querschnitten (Sections) die Trajektorien geschnitten werden und welche Abfolgen von Querschnitten als Verkehrsströme (Flows) definiert werden, siehe Abbildung 16. Dabei lassen sich verschiedene Eigenschaften konfigurieren, wie beispielsweise der Referenzpunkt eines Fahrzeugs, der für die Trajektorienbildung herangezogen wird. Die gesamte Analysekonfiguration kann vom Nutzer gespeichert und später wieder geladen werden, was auch eine iterative Verfeinerung der Einstellungen ermöglicht.

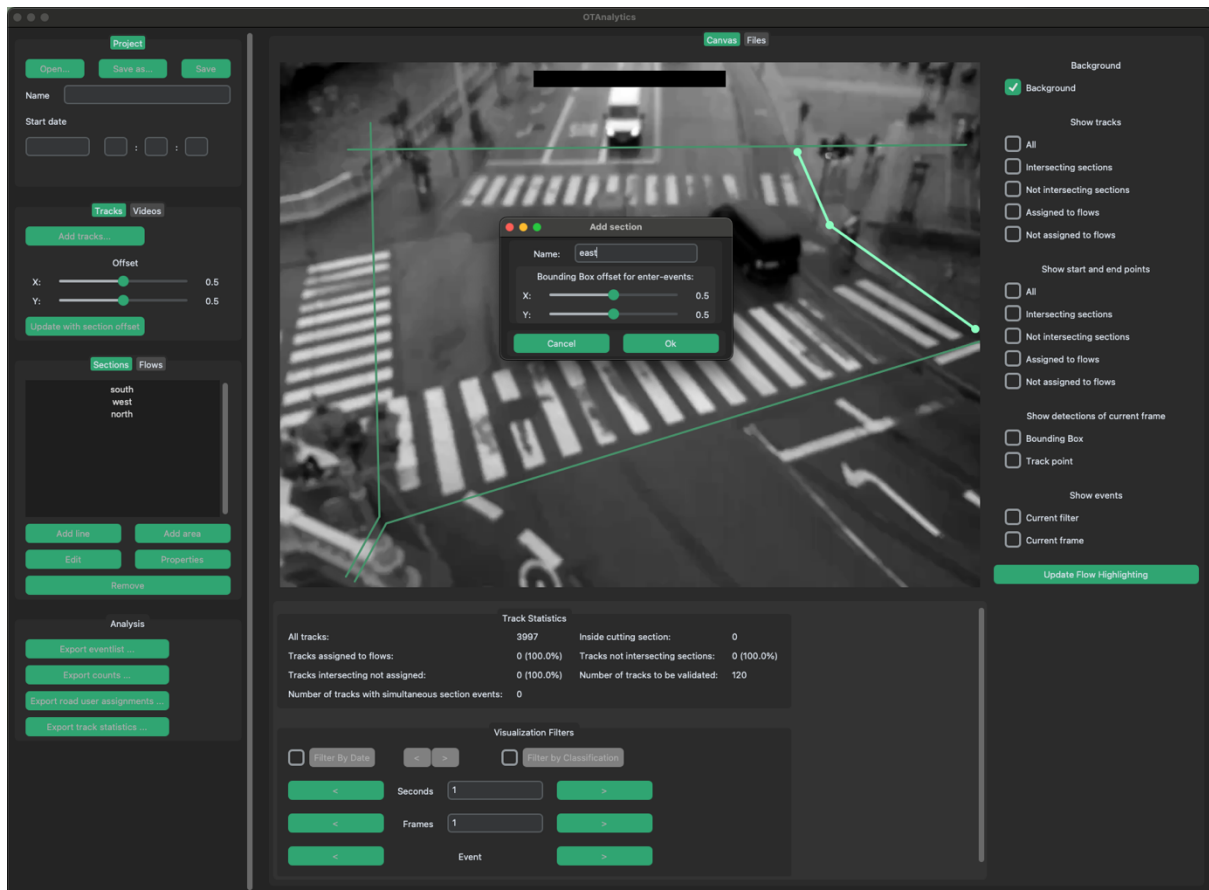


Abbildung 16: Konfiguration einer Section in OTAnalytics

Herzstück der Anwendung ist der neu implementierte Event-Algorithmus. Dieser berechnet die Schnittpunkte zwischen den Trajektorien der Verkehrsteilnehmenden und den definierten Sections. Jeder solcher Schnittpunkt wird als Event mit Zeitstempel, Position sowie den IDs des Tracks und der Section protokolliert. Anhand der chronologischen Abfolge dieser Events können die einzelnen Tracks anschließend bestimmten Verkehrsströmen (Flows) zugeordnet werden, sofern sie die beiden definierten Querschnitte in der vorgegebenen Reihenfolge passieren. Aus dieser Zuordnung werden wiederum Verkehrsstärken für alle definierten Relationen ermittelt und in verschiedenen Aggregationsstufen ausgewertet. Die generierten Ereignisdaten und Auswertungen lassen sich in diversen Formaten (u.a. CSV und kundenspezifische Formate) exportieren, wobei alle implementierten Exportfunktionen umfassend getestet und die Ergebnisse verifiziert wurden.

Ein wesentliches Merkmal von OTAnalytics ist die interaktive Visualisierung. In der Benutzeroberfläche lassen sich die Analysen konfigurieren, die erfassten Daten visualisieren und die Ergebnisse effizient exportieren. Die Benutzeroberfläche gliedert sich in vier Hauptbereiche, siehe Abbildung 17: eine links angeordnete Konfigurationsleiste mit den Komponenten *Projekt*, *Tracks/Videos*, *Abschnitte/Flüsse* und *Analyse*; den zentralen Arbeitsbereich (Canvas) zur Ansicht der Videoaufnahmen und Trajektorien; rechts daneben ein Panel für Visualisierungsebenen; sowie darunter einen Bereich für Visualisierungsfiler.

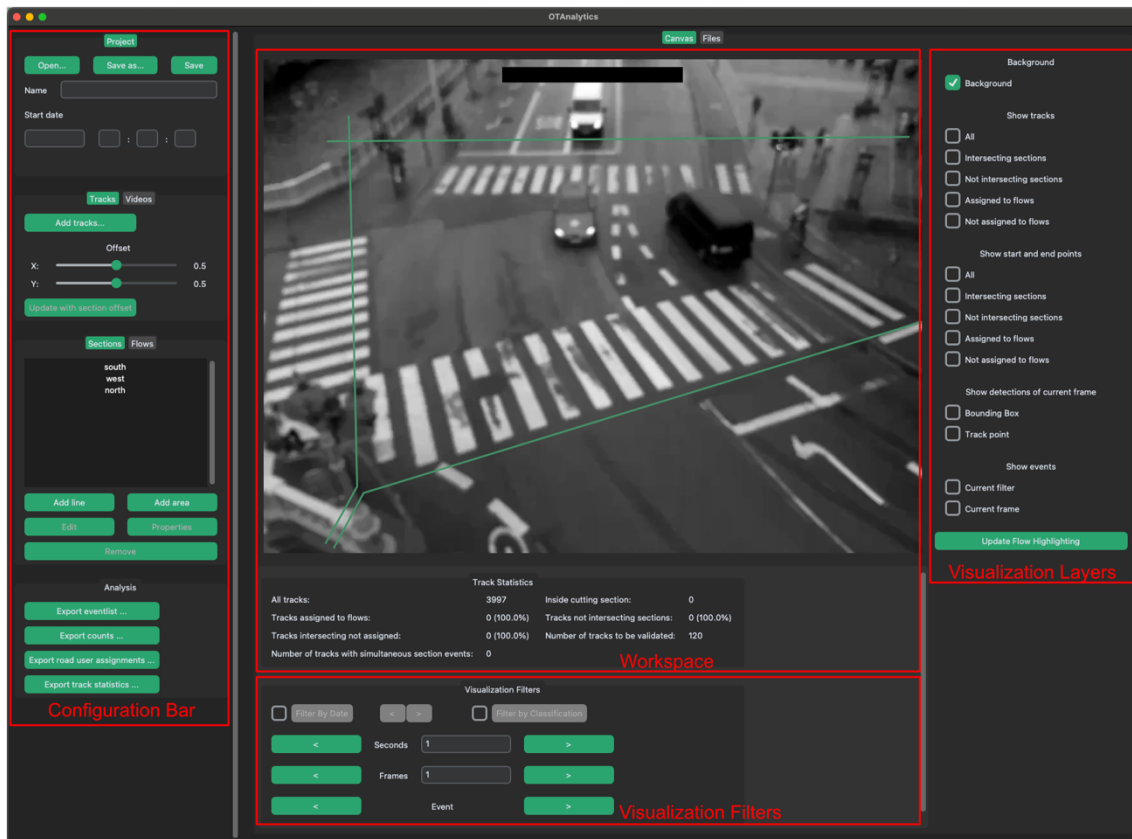


Abbildung 17: Aufbau der OTAnalytics Benutzeroberfläche

In der Konfigurationsleiste werden u. a. Projektdaten verwaltet (Name, Zeitbasis der Aufnahmen, Laden der Trajektorien- und Videodateien), Sections im Videobild erstellt und zu *Flows* verknüpft, sowie abschließend die Ergebnis-Exporte angestoßen. Die vollständige Konfiguration eines Projekts lässt sich in einer Datei speichern (*.otconfig*) und bei Bedarf wieder laden. Alternativ kann die Section-/Flow-Definition auch separat als *.otflow*-Datei gesichert und bei anderen Projekten am selben Standort wiederverwendet werden.

Der zentrale Arbeitsbereich zeigt das Videobild mit den darin erkannten Trajektorien und dient als Interaktionsfläche für die Konfiguration. Hier können beispielsweise per Maus neue Sections direkt im Bild eingezeichnet, mit einem Namen versehen und damit als Detektionsbereiche definiert werden, siehe Abbildung 16. Anschließend lassen sich jeweils zwei Abschnitte als Start- und Endpunkt zu einem Flow kombinieren. Bei Bedarf können auch automatisiert alle möglichen Abschnittskombinationen als Flows erzeugt werden, um den Konfigurationsaufwand zu verringern. OTAnalytics ordnet in der Folge jede Trajektorie basierend auf den durchfahrenen Sections dem entsprechenden Flow zu, sodass z. B. die Anzahl der Bewegungen der einzelnen Flows automatisch ermittelt wird.

Zur Unterstützung der Konfiguration und Validierung bietet OTAnalytics vielfältige Visualisierungsoptionen. Rechts neben dem Arbeitsbereich können über an- und abwählbare Visualisierungsebenen unterschiedliche Informationen eingeblendet werden. So lässt sich etwa der Videohintergrund als Referenz aktivieren, und es können alle oder nur bestimmte Trajektorien hervorgehoben werden – beispielsweise beschränkt auf solche, die ausgewählte Sections passieren oder definierten Flows zugeordnet sind, siehe Abbildung 18.

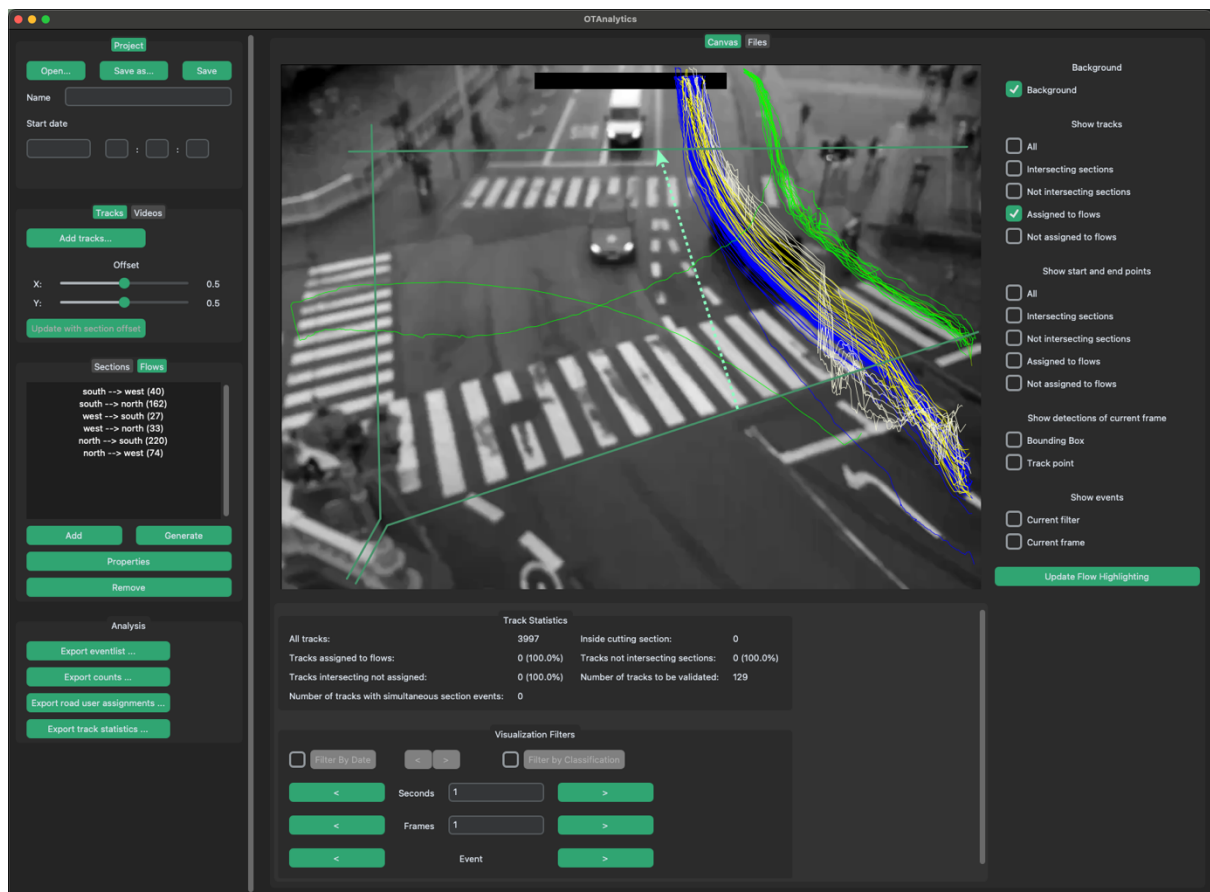


Abbildung 18: Filterung der angezeigten Trajektorien auf alle, die einem Flow zugewiesen wurden

Ebenso können die Start- und Endpunkte der erfassten Trajektorien angezeigt werden, um Abbruchstellen in den Bewegungen zu erkennen, oder die Positionen aller registrierten Events visualisiert werden, um räumliche Häufungen zu identifizieren, siehe Abbildung 19.

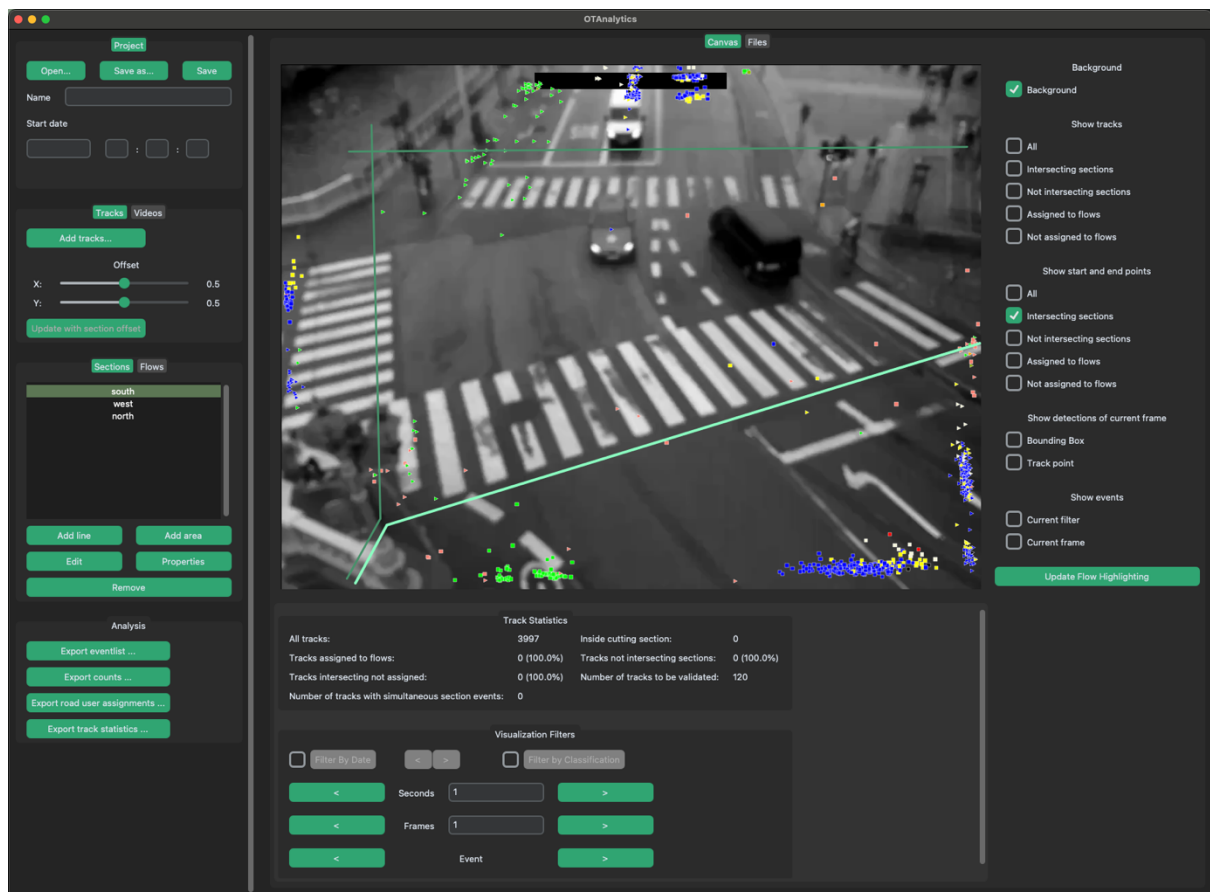


Abbildung 19: Darstellung aller Start- und Endpunkte der Trajektorien, die eine Section schneiden

Darüber hinaus besteht die Möglichkeit, die Einzeldetektionen eines ausgewählten Videoframes einzublenden – wahlweise als Begrenzungsrahmen (Bounding Boxes) der Objekte oder als repräsentative Trajektorienpunkte – um die Qualität der Track-Erkennung im Detail zu prüfen, siehe Abbildung 20.

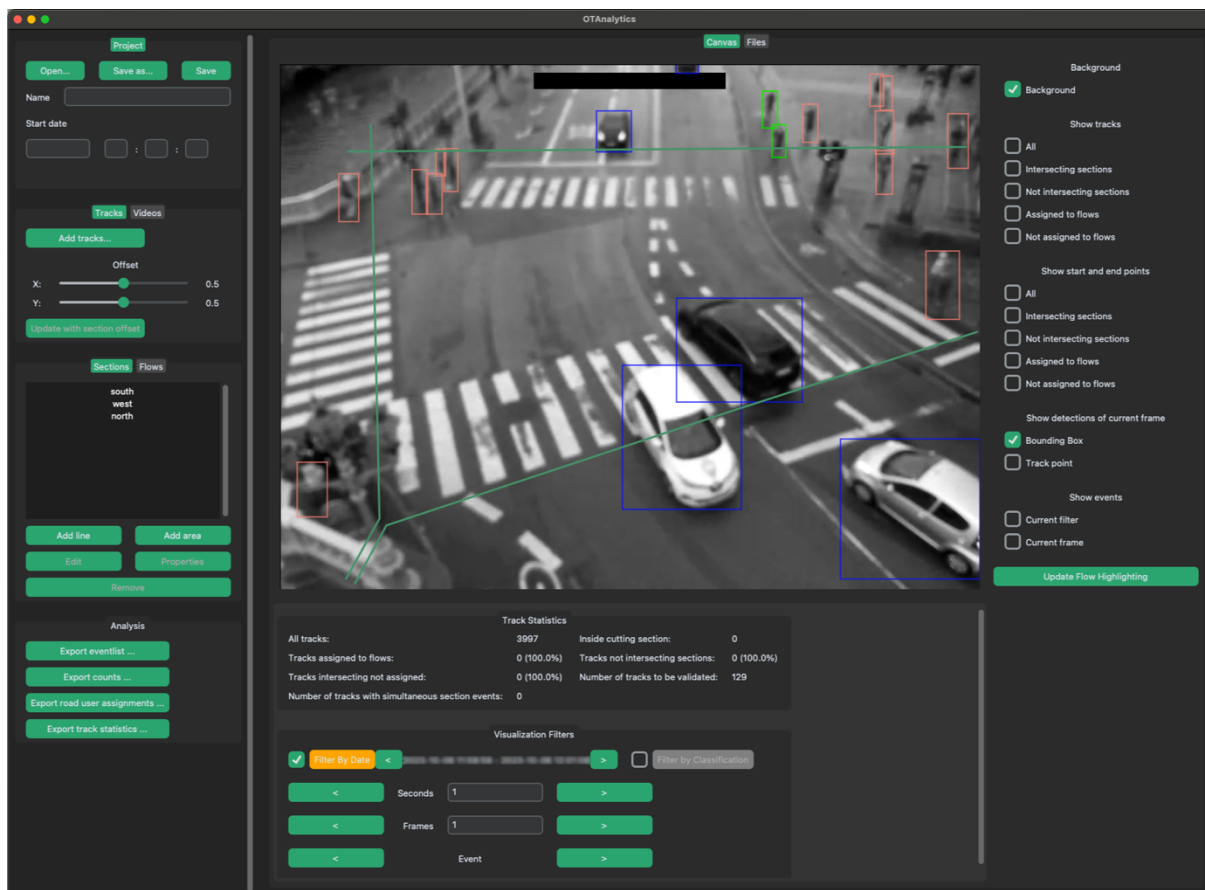


Abbildung 20: Darstellung der Bounding Boxen aller erkannten Objekte in einem Bild

Unterhalb des Arbeitsbereichs ermöglichen schließlich Zeit- und Objektfilter eine gezielte Eingrenzung der Darstellung. So kann beispielsweise ein bestimmter Analysezeitraum oder nur eine spezifische Fahrzeug- bzw. Verkehrsteilnehmerklasse ausgewählt werden, um die Visualisierung auf diese Daten zu beschränken.

Nach Abschluss der Konfiguration und Überprüfung der Einstellungen können im *Analyse*-Bereich verschiedene Auswertungsergebnisse als CSV-Dateien exportiert werden. Dazu zählen unter anderem aggregierte Verkehrszählungen über einstellbare Zeitintervalle, tabellarische Eventlisten sowie detaillierte Statistiken zu einzelnen Trajektorien (z. B. der zugeordneten Flow je Verkehrsteilnehmer).

Für die Verarbeitung sehr umfangreicher Datensätze wurde ergänzend zur GUI eine Kommandozeilenanwendung (CLI) implementiert. Mit der CLI können Analysen auf leistungsfähiger Hardware ausgeführt werden, ohne die lokale Arbeitsumgebung des Nutzers zu belasten. Dabei lassen sich nach Bedarf sowohl Batch-Verarbeitungen (gleichzeitige Verarbeitung großer Datenblöcke) als auch Stream-Verarbeitungen (fortlaufende sequentielle Verarbeitung einzelner Dateien zur unbegrenzten Skalierung) durchführen. Durch diese zweigleisige Auswertungsstrategie und die Auslagerung rechenintensiver Prozesse auf Server konnte die Software auch in Richtung einer kontinuierlichen Verarbeitung (Echtzeitfähigkeit) weiterentwickelt werden.

Neben OTAnalytics entstand im Projekt auch das Tool OTConfig, das eine vereinfachte Erstellung von Auswertungskonfigurationen ermöglicht. OTConfig bietet eine reduzierte Oberfläche, mit der Kunden der platomo GmbH eigenständig Sections und Flows für ihre Videos definieren können.

Das Tool exportiert eine Konfigurationsdatei, die sämtliche Angaben – einschließlich der Referenzen zu den Videodateien – enthält. Diese Datei kann anschließend durch das Team von platomo geladen und verarbeitet werden, woraufhin die Kunden die Ergebnisse in dem gewünschten Format erhalten. Für solche Dienstleistungsprozesse wurden spezifische Exportformate (u.a. projekt-spezifische Kundenformate) implementiert. Sowohl OTAnalytics als auch OTConfig werden im Rahmen der platomo-Release-Strategie kontinuierlich weiterentwickelt und veröffentlicht.

Die Leistungsfähigkeit der neuen OTAnalytics-Version wird fortlaufend überwacht und optimiert. Da insbesondere die Schnittpunktermittlung zwischen Trajektorien und Sections rechenaufwändig ist, wurden automatisierte Benchmarks definiert, die täglich ablaufen und die Performance messen. Diese Messergebnisse werden auf einer Weboberfläche visualisiert, um potenzielle Engpässe frühzeitig zu erkennen und Optimierungen abzuleiten, siehe Abbildung 21. Die so entwickelte MVP-Version von OTAnalytics (sowie OTConfig) hat bereits den Praxiseinsatz erreicht und wird erfolgreich in ersten Kundenprojekten – unter anderem in Zusammenarbeit mit der BAST – eingesetzt.

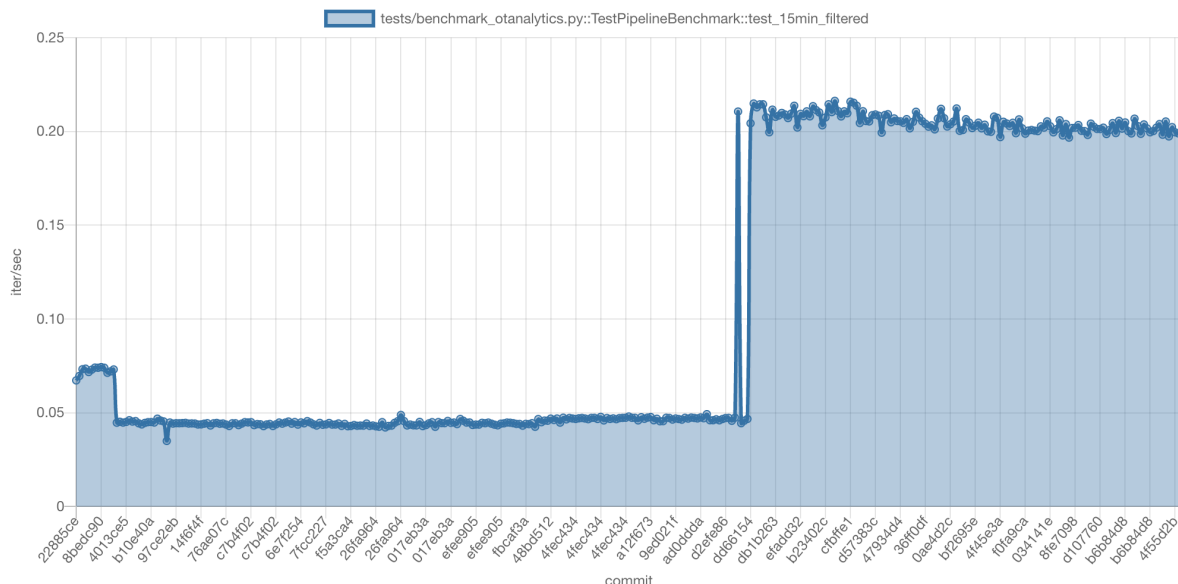


Abbildung 21: Ergebnisse des automatisierten Benchmarks auf github

6.6.1 Wiederkehrende Optimierung Performance und Zuverlässigkeit der Software (platomo)

Für OTAnalytics wurden wiederholt Optimierungen der Performance und Zuverlässigkeit durchgeführt. Insbesondere wurden die Schnittalgorithmen zwischen Trajektorien und Sections in mehreren Iterationen verbessert, um die Effizienz zu steigern. Darüber hinaus wurde die Berechnung der Events hinsichtlich des Speicherbedarfs optimiert, sodass auch große Trajektorien Datensätze ohne Überlastung des Arbeitsspeichers verarbeitet werden können. Zur effizienten Analyse der Trajektorien wurden sowohl eine Batch- als auch eine Stream-Verarbeitung implementiert. Im Batch-Modus werden alle Daten auf einmal verarbeitet, was insbesondere für die Darstellung der Trajektorien und die Konfiguration der Sections in der Benutzeroberfläche nützlich ist. Im Gegensatz dazu verarbeitet der Stream-Modus die Daten sequentiell, wodurch beliebig große Datenmengen analysiert werden können, ohne alle Detektionen gleichzeitig in den Speicher laden zu

müssen. Diese duale Verarbeitungsstrategie bietet Flexibilität in der Datenhandhabung und ermöglicht sowohl eine benutzerfreundliche Konfiguration bei kleineren Datenmengen als auch die Skalierbarkeit für sehr große Datenvolumina.

Des Weiteren wurde eine Kommandozeilen-Anwendung (CLI) von OTAnalytics entwickelt, um die Verarbeitung auf leistungsstarker Server-Hardware zu ermöglichen. Diese CLI-Variante gestattet es, umfangreiche Datenmengen auf einem Server auszuwerten und entlastet dadurch den Rechner des Nutzers. Die Analyse kann in der grafischen Oberfläche vorbereitet und anschließend über die CLI auf dem Server ausgeführt werden. Damit verbindet OTAnalytics die einfache Bedienung der GUI bei der Einrichtung mit der hohen Performance der serverbasierten Verarbeitung. Insgesamt stellen diese Optimierungen und Erweiterungen sicher, dass OTAnalytics Verkehrsdaten zuverlässig, effizient und in großem Umfang verarbeiten kann.

6.6.2 Implementierung Fuß- und Rad-Zählung, Zeitlückenbestimmung, Sicherheitskennwerte (MSP)

Im Rahmen dieses Arbeitspakets wurde OTAnalytics um weitere Analysefunktionen für spezifische Anwendungsfälle erweitert. So wurde beispielsweise eine gezielte Filterung nach Verkehrsteilnehmerklassen implementiert, um Fußgänger und Radfahrende separat zählen zu können. Diese Funktion kam etwa in einem Projekt der BAST zum Einsatz, um an 112 Messstellen in ganz Deutschland Fußgängerfrequenzen automatisiert zu ermitteln. Durch die Fokussierung der Analyse auf ausgewählte Klassen können Ergebnisse gezielt und effizient gewonnen werden.

Ein weiteres umgesetztes Feature in der Nachverarbeitung von OTAnalytics ist die Bestimmung von Zeitlücken (zeitlichen Abständen) zwischen aufeinanderfolgenden Verkehrsteilnehmern, was exemplarisch auf einem größeren Datensatz getestet wurde.

Parallel dazu wurden verschiedene Sicherheitskennwerte untersucht. Mit Hilfe der aus OTAnalytics exportierten Daten, können nun beispielsweise die Abstände zwischen Trajektorien bestimmt werden, nachdem diese in Weltkoordinaten überführt wurden. Damit lässt sich analysieren, wie nahe sich zwei Verkehrsteilnehmende räumlich und zeitlich kommen. Hierfür werden die Trajektorien georeferenziert und sowohl der zeitliche als auch der räumliche Abstand zwischen zwei Verkehrsteilnehmenden in einem Prototyp berechnet. Diese Fähigkeit wurde perspektivisch genutzt, um potenziell gefährliche Situationen im Verkehr zu identifizieren. So kam die Funktion in einem EU-Forschungsprojekt zum Einsatz, in dem automatisiert kritische Annäherungen erkannt und anschließend von Probanden in einer Umfrage hinsichtlich ihrer Gefährlichkeit bewertet wurden.

Allerdings hat sich gezeigt, dass die Genauigkeit solcher Sicherheitskennwerte stark von der Qualität der Transformation in Weltkoordinaten abhängt. Mit einfachen bzw. frei verfügbaren Algorithmen zur Georeferenzierung (wie sie beispielsweise im Modul OTMulticamera eingesetzt werden) war die Präzision nicht ausreichend, um Sicherheitsindikatoren im Forschungskontext verlässlich zu berechnen. Somit kann das System zwar potenzielle Konfliktsituationen aufzeigen, doch für eine exakte Bestimmung von Sicherheitskennwerten (z.B. Zeit-zu-Kollision oder geringster Abstand) sind präzisere Kalibrierungsmethoden erforderlich. Insgesamt erweitern die Implementierungen für Fußgänger- und Radverkehrs-zählung, Zeitlückenmessung und erste Sicherheitsanalysen die Auswertungsmöglichkeiten von OTAnalytics erheblich, auch wenn in puncto Genauigkeit weiterer Entwicklungsbedarf besteht.

6.6.3 Implementierung Geschwindigkeits-, Wartezeitmessung, HBS-Bewertung für Knotenpunkte (platomo)

Nachgelagert zu OTAnalytics wurden Funktionen zur Geschwindigkeits- und Wartezeitanalyse umgesetzt. So wurde prototypisch eine Geschwindigkeitsanalyse anhand bekannter Distanzen zwischen definierten Sections implementiert. Dabei wird gemessen, wie viel Zeit ein Fahrzeug benötigt, um zwei Zählquerschnitte zu durchfahren, und aus der hinterlegten Distanz zwischen diesen Querschnitten die Durchschnittsgeschwindigkeit errechnet. Diese Funktion wurde bereits mit einem Kunden erprobt. Eine präzisere Geschwindigkeitsberechnung auf Basis von Weltkoordinaten (tatsächlichen Entfernungen) wäre zwar wünschenswert, ist für Kurzzeiterhebungen jedoch aufgrund des Aufwands der Georeferenzierung meist nicht praktikabel. Zudem wird eine detaillierte Geschwindigkeitsauswertung von Kundenseite nur selten angefragt.

Des Weiteren können nachgelagert zu OTAnalytics Aufenthaltsdauern bzw. Wartezeiten von Verkehrsteilnehmenden ermittelt werden, indem sogenannte AreaSections verwendet werden. Eine AreaSection erzeugt – anders als ein linienbasierter Querschnitt – zwei Events pro Track: eines beim Einfahren in den definierten Bereich und ein weiteres beim Verlassen. Hierüber lässt sich die Verweildauer eines Fahrzeugs in einem bestimmten Gebiet bestimmen. Mit dieser Methode wurde beispielsweise die Standzeit von Fahrzeugen analysiert, um deren Parkdauer zu ermitteln. Dadurch zeigt sich, dass OTAnalytics neben reinen Zähldaten auch zeitbezogene Kenngrößen der Verkehrsabläufe erfassen kann.

Schließlich wurde mit dem "HBS-Buddy" ein Konzept prototypisch umgesetzt, das eine Anbindung an das HBS-Verfahren für Knotenpunkte ermöglicht. Die Idee dahinter war, ausgewählte Kenngrößen aus OTAnalytics (z.B. Verkehrsstärken, Geschwindigkeiten) zusammen mit zusätzlich konfigurierbaren Parametern in Berechnungen nach HBS-Richtlinien einfließen zu lassen, um den Aufwand bzw. die Leistungsfähigkeit eines Knotenpunktes abschätzen zu können. Es zeigte sich, dass eine solche Integration technisch machbar ist – OTAnalytics liefert alle nötigen Kenngrößen und die erforderlichen Parameter können bereitgestellt werden. Allerdings wäre die vollständige Umsetzung eines HBS-Auswertungsmoduls im Rahmen dieses Projekts zu aufwändig gewesen, sodass es vorerst bei dem Konzeptnachweis in Prototypform blieb.

6.6.4 Implementierung automatische und interaktive Berichte und Diagramme (platomo)

Ein weiterer Schwerpunkt war die nutzergerechte Aufbereitung der Ergebnisse. Hierzu wurde eine Markterkundung durchgeführt, um die Anforderungen der Anwender hinsichtlich Auswertungsberichten und Grafiken zu ermitteln. Dabei zeigte sich, dass interaktive Berichte und Diagramme von den meisten Kunden nicht nachgefragt werden. Stattdessen ist die Integration der Ergebnisse in bestehende Formate und Systeme der Anwender von größerer Bedeutung. Folglich verlagerte sich der Entwicklungsfokus darauf, dass OTAnalytics die Analyseergebnisse in unterschiedlichen gängigen Formaten (z.B. CSV, Excel oder spezifische Kundenformate) exportieren kann. Öffentliche Formate wie CSV und Excel können direkt aus OTAnalytics im offenen OpenTrafficCam-Format exportiert werden. Für kundenspezifische Formate wurden eigene Exportskripte bei platomo entwickelt. Dadurch können Anwender die gewonnenen Verkehrsdaten nahtlos in ihre eigenen Reporting-Prozesse und Softwareumgebungen übernehmen, was sich als wichtiger herausgestellt hat als die Bereitstellung einer eigenen interaktiven Berichtsoberfläche.

6.6.5 Implementierung Datenexport zu anderen Systemen (MSP)

Um die Nutzung der Analyseergebnisse in externen Systemen zu erleichtern, wurden in OTAnalytics flexible Schnittstellen für den Datenexport geschaffen. So können die erzeugten Verkehrskennzahlen in verschiedenen kunden- und branchenspezifischen Formaten ausgegeben werden. Beispielsweise unterstützt OTAnalytics den Export der Zählraten im Format der bundeseinheitlichen Straßenverkehrszählung (SVZ) 2025 sowie weitere gängige Austauschformate. Durch die Bereitstellung mehrerer Exportoptionen können Anwender die Ergebnisse ohne zusätzlichen Konvertierungsaufwand in ihre vorhandenen Systeme und Auswertungsumgebungen integrieren.

Zudem war die platomo GmbH bei Projektende in Abstimmung mit der Firma Softwarekontor Helmer, um eine Importschnittstelle der Standardexportformate der OpenTrafficCam in die gängige Software VZAweb zu integrieren.

6.6.6 Zusammenfassung aktueller Stand

Die Software OTAnalytics ist auf die Verarbeitung von Tracks aus mobilen Kurzzeiterhebungen ausgelegt. Hierzu werden von OTVision erzeugte Track-Dateien (ottrk) eingelesen. OTAnalytics besteht aus einer Konsolenanwendung zur Verarbeitung großer Datenmengen und einer Benutzeroberfläche zur Konfiguration der Auswertung.

Die Konsolenanwendung von OTAnalytics ermöglicht die Auswertung beliebig großer Datenmengen. Hierzu werden alle konfigurierten Track-Dateien in den Speicher geladen. Anschließend werden alle Tracks mit den konfigurierten *cutting*-Sections getrennt. *cutting*-Sections definieren die Region-of-Interest, d.h. den Bildbereich, indem die für die Auswertung relevanten Detektionen und Track-Teile liegen. Vor allem weit entfernte, nicht zum Untersuchungsbereich gehörende Track-Teile bleiben bei der Verkehrsanalyse unberücksichtigt. Anschließend werden alle Tracks mit Zählquerschnitten (Sections) geschnitten. Aus jedem Schnittpunkt wird ein Event mit Zeit, Ort, Track- und Section-ID erzeugt. Ferner können Flows (Verkehrsströme) angelegt werden, die durch eine bestimmte Reihenfolge zweier Sections definiert sind. Das Anlegen kann auch automatisiert für alle Kombinationen von Sections erfolgen. Basierend auf den Events werden anschließend die Tracks den Flows zugewiesen und als Counts (Zählraten) gezählt. Die Events, deren Zuweisung zu den Flows und die Counts werden anschließend als einzelne Dateien exportiert. Bisher unterstützt OTAnalytics den Export der Zählung der Fahrzeuge in unterschiedlichen Aggregationen und Formaten, die Zuweisung zu Flows sowie den Export der Events.

Der limitierende Faktor bei der Menge der zu verarbeitenden Daten ist aktuell der verfügbare Arbeitsspeicher, da alle Detektionen in den Arbeitsspeicher geladen werden. Je länger der Auswertungszeitraum und die damit einhergehende Anzahl an Detektionen ist, desto mehr Zeit nimmt die Verarbeitung in Anspruch. Dies ist der ursprüngliche Grund für die zusätzliche Entwicklung Konsolenanwendung (Command Line Interface bzw. CLI). Die Analyse kann auf einem Server mit genügend Arbeitsspeicher gestartet werden, ohne den Computer des Nutzers zu belasten. Eine effiziente Parallelisierung auf dem Server war ursprünglich nicht möglich. Daher wurde eine Erweiterung für die Konsolenanwendung entwickelt, bei der ähnlich zu OTVision die Track-Dateien einzeln verarbeitet werden. Der Speicherverbrauch beschränkt sich hierbei auf die Anzahl Tracks einer Datei und bleibt in der realen Anwendung nahezu konstant. Gleichzeitig bietet diese Erweiterung eine gute Grundlage für die kontinuierliche Verarbeitung von Tracks werden, wie sie in einer Echtzeitanwendung gefordert wird. Die Auswertung der Konsolenanwendung kann dabei über Parameter beim Aufruf gesteuert oder in der Benutzeroberfläche konfiguriert werden.

In der Benutzeroberfläche werden die auszuwertenden Track-Dateien, die Sections, Flows und die Region-of-Interest konfiguriert. Alle eingelesenen Tracks können auf dem Videobild visualisiert werden. Es können die gesamten Trajektorien sowie die Bounding-Boxen der einzelnen Detektionen angezeigt werden. Über verschiedene Visualisierungsebenen lassen sich einzelne Tracks hervorheben, z.B. Tracks, die eine oder mehrere Sections überqueren bzw. schneiden, einem Flow zugewiesen sind oder die Start- und Endpunkte der Tracks. Zusätzlich lassen sich die angezeigten Tracks nach Zeit und Klasse filtern. Die angezeigten Informationen bzw. der Filterbereich lässt sich beliebig verschieben (um Einzelbilder, Sekunden oder Zeitbereiche) oder es kann zum nächsten Event gesprungen werden. Über die verschiedenen Filter und Visualisierungen können die Sections optimal an die Auswertung angepasst werden. Nachdem die Konfiguration der Analyse abgeschlossen ist, können die Ergebnisse direkt aus der Benutzeroberfläche exportiert werden. Je größer die zu analysierenden Datenmengen, desto leistungsfähiger muss die verwendete Maschine sein für geringe bzw. vertretbare Ladezeiten bei der Bedienung der Benutzeroberfläche. Statt großer Datenmengen (z.B. Tracks aus mehrtägigem Videomaterial) kann auch nur ein Teil der Daten (z.B. Tracks aus einer oder wenigen Stunden Videomaterial) in der Benutzeroberfläche geladen werden, um die Auswertung zu konfigurieren. Das Antwortverhalten der Benutzeroberfläche ist dadurch signifikant besser und die Anzahl der Tracks ist dennoch groß genug, um auch mögliche Besonderheiten in den Track-Verläufen erkennen und die Sections passgenau anlegen zu können. Anschließend kann die gesamte Datenmenge effizient über das CLI prozessiert werden.

6.7 AP 7 OTCloud (platomo)

OTCloud dient als zentrale Prozessierungsplattform für die Videoanalyse innerhalb der platomo GmbH. Im Rahmen des Projekts OpenTrafficCam_live wurde OTCloud von Grund auf neu entwickelt. Die Implementierung erfolgte nach den Prinzipien der Clean Architecture, sodass eine modulare Softwarearchitektur entstand, die den einfachen Austausch einzelner Komponenten oder Technologien ermöglicht. Zu Projektbeginn wurden verschiedene Technologien für die Benutzeroberfläche evaluiert; zunächst kam eine einfache Web-Technologie zum Einsatz, die aufgrund der Modularität der Architektur bei Bedarf künftig ausgetauscht oder erweitert werden kann.

OTCloud übernimmt die automatische Verarbeitung eingehender Videoaufnahmen mithilfe des Moduls OTVision. Zu diesem Zweck werden die Videos gemeinsam mit einer zugehörigen Konfigurationsdatei auf dem System abgelegt. OTCloud erkennt neue Videos anhand der vorliegenden Konfiguration und startet daraufhin selbstständig die Verarbeitung mit den erforderlichen Komponenten von OTVision. Während des Verarbeitungsprozesses wird der aktuelle Bearbeitungsstatus in Echtzeit über die Benutzeroberfläche visualisiert.

Zum Abschluss des Projekts wurde OTCloud innerhalb der platomo GmbH in Betrieb genommen und dient dort als Kernplattform für die automatisierte Videoverarbeitung. Das System hat sich dabei bewährt, um begrenzte Ressourcen – insbesondere GPU-Rechenleistung – effizient zwischen mehreren Nutzenden zu koordinieren. OTCloud wird auch nach Projektende kontinuierlich weiterentwickelt; für das Jahr 2024 sind weitere Funktionserweiterungen und Optimierungen geplant, um die Plattform noch leistungsfähiger und benutzerfreundlicher zu gestalten.

6.7.1 Konzept und Evaluierung der erwarteten User Experience (platomo mit UA)

In Zusammenarbeit mit dem Verbundpartner wurden das Nutzungskonzept und die erwartete User Experience für OTCloud erarbeitet. Es wurden Interviews mit potenziellen Anwendern durch-

geführt, um deren Erwartungen an die automatisierte Verarbeitung von Verkehrsvideos zu erheben. Dabei wurden insbesondere folgende Fragen adressiert: *Welche Erwartungen haben die Nutzenden an die Prozessierung von Verkehrsvideos zur automatischen Verkehrszählung?* Welche Teile des Aufwands möchten sie im eigenen Unternehmen leisten und welche Aufgaben würden sie lieber an externe Dienstleister vergeben? Die Ergebnisse dieser Befragung bildeten die Grundlage für die Gestaltung der OTCloud-Benutzeroberfläche. Aus den Interviews kristallisierten sich zwei typische Kundengruppen bzw. Anwendungsszenarien heraus: (1) Vorhaben, in denen die Videoerhebung und -auswertung komplett an ein externes Unternehmen (z. B. platomo) ausgelagert werden – in solchen Fällen ist die Benutzeroberfläche hauptsächlich für den internen Gebrauch beim Dienstleister relevant. (2) Vorhaben, in denen die Datenerhebung und -auswertung im Unternehmen des Kunden selbst erfolgt – hier nutzt der Kunde eigenständig die Software, weshalb dessen Anforderungen und Rückmeldungen direkt in die Benutzeroberfläche einfließen müssen.

Für die erste Phase der Umsetzung zeigte sich, dass viele Interessenten zunächst eine Verarbeitung der Videos über die Infrastruktur von platomo bevorzugten, um das System zu erproben und die Ergebnisse mit ihren bisherigen Verfahren zu vergleichen. Entsprechend wurde der Fokus im Projekt zunächst auf die Optimierung der User Experience für die interne Nutzung von OTCloud bei platomo gelegt. Platomo-interne Anwender dienten als Haupt-Fokusgruppe für die iterativen Verbesserungen der Nutzeroberfläche. Um Anpassungen zügig umzusetzen, wurde entgegen dem ursprünglichen Plan, diese Tätigkeiten extern zu vergeben, ein Webentwickler mit UI-Expertise in das Team aufgenommen, der neue Designentwürfe und Funktionserweiterungen entsprechend schnell implementieren konnte. Im Projektverlauf fanden mehrere Diskussionsrunden zwischen den Nutzenden und den Entwickelnden statt, um Feedback auszutauschen und die Bedienbarkeit weiter zu steigern. Die internen Anwender zeigten sich mit der im Projekt erreichten Benutzererfahrung sehr zufrieden.

Nach Projektabschluss ist vorgesehen, weitere Anwendungsfälle umzusetzen und dabei verstärkt externe Nutzende in die Evaluierung der Benutzeroberfläche einzubeziehen. Die Rückmeldungen dieser externen Testpersonen sollen als Grundlage für ein überarbeitetes UI-Konzept dienen, um die Benutzeroberfläche für einen breiteren Nutzerkreis weiter zu optimieren.

6.7.2 Integration aller Teilmodule des Projekts und Fusion der Teil-Daten (platomo)

OTCloud wird vorrangig als Plattform zur Ausführung von OTVision genutzt. Ein Hauptgrund hierfür ist, dass die benötigte GPU-Rechenleistung ein knappes Gut darstellt, das zwischen mehreren Anwendern aufgeteilt werden muss. Durch die zentrale Verarbeitung in OTCloud kann die verfügbare GPU effizient gemeinsam genutzt und die Abarbeitung der Videos in einer geordneten Warteschlange sichergestellt werden.

Das Modul OTAnalytics wird derzeit als Desktop-Anwendung betrieben und ist nicht direkt in OTCloud integriert. Stattdessen können die von OTVision erzeugten Ausgabedaten in OTAnalytics importiert werden, um weiterführende verkehrsanalytische Auswertungen durchzuführen. Die Ergebnisse aus OTAnalytics fließen wiederum in verschiedene interne Weiterverarbeitungsskripte bei platomo ein. Auf diese Weise ist sichergestellt, dass die in OTCloud verarbeiteten Videodaten trotz der Aufteilung auf separate Anwendungen nahtlos in der gesamten Auswertungskette genutzt werden können.

6.7.3 Implementierung Prozessierung der Live-Daten, Ausfallschutz Backend (platomo)

Als weiterer Schritt wurde prototypisch die Verarbeitung von Live-Daten erprobt. Hierfür entstand ein Demonstrator eines Live-Systems. Es handelte sich dabei nicht um ein marktreifes Produkt, sondern um einen Versuchsaufbau, der primär der Anforderungsanalyse nach dem Stil der agilen Entwicklung diente. Mit diesem Prototyp konnte gezeigt werden, dass eine Echtzeit-Verarbeitungskette grundsätzlich funktioniert – beginnend mit einem Live-Videostream der OTCamera über die Objekterkennung in OTVision bis hin zur laufenden Datenanalyse in OTAnalytics. Die dabei gewonnenen Erkenntnisse wurden genutzt, um Anforderungen für die zukünftige Entwicklung eines ausgereiften Live-Systems abzuleiten.

Zur Steigerung der Betriebssicherheit von OTCloud wurde zudem ein grundlegender Ausfallschutz im Backend implementiert. Dieser Mechanismus stellt sicher, dass nach einem Neustart des Systems bereits laufende Verarbeitungsaufträge automatisch wiederaufgenommen und fortgeführt werden. Dadurch gehen auch bei einem unerwarteten Systemausfall keine langen Verarbeitungsläufe verloren, was die Robustheit der Plattform erhöht.

6.7.4 Containerisierung, Einrichten Testumgebung und Einrichtung von Kundenworkspaces (platomo)

Zu Projektbeginn wurde untersucht, inwieweit OTVision in Containern ausgeführt werden kann, um Deployment und Skalierung zu erleichtern. Hierzu wurden verschiedene Container-Images getestet. Allerdings erwies sich die Performance der containerisierten Umgebung im ersten Anlauf als unzureichend und die entstehenden Container-Images waren sehr groß. Vor diesem Hintergrund wurde im Projekt zunächst auf den produktiven Einsatz von Containern für OTVision verzichtet. Inzwischen sind jedoch deutlich effizientere Basis-Images für GPU-Anwendungen verfügbar, sodass der Einsatz von Container-Technologie als Laufzeitumgebung in zukünftigen Vorhaben erneut geprüft wird.

Nichtsdestotrotz kam Containerisierung im Projekt punktuell erfolgreich zum Einsatz. So wurde beispielsweise ein Prototyp zur HBS-Bewertung (nach dem *Handbuch für die Bemessung von Straßenverkehrsanlagen*) innerhalb eines Containers entwickelt und betrieben. Die Erstellung und Bereitstellung des entsprechenden Container-Images konnten mit geringem Aufwand realisiert werden, und auch der Betrieb dieses Containers – einschließlich Durchführung von Updates – verlief erfolgreich. Dieses Beispiel zeigte, dass Containerisierung für bestimmte Module oder Anwendungsfälle klare Vorteile bei der Bereitstellung und Wartung bietet.

Für OTCloud wurde eine zweistufige Betriebsumgebung eingerichtet, bestehend aus einer Testumgebung (Staging) und einer Produktionsumgebung. Neue Funktionen werden zunächst in der Staging-Umgebung mit realen Daten getestet, um eventuelle Fehler frühzeitig zu erkennen. Erst nach erfolgreicher Erprobung werden die Änderungen in die Produktionsumgebung übernommen. Dieses Vorgehen stellt sicher, dass neue Releases der Software stabil laufen und die Gefahr von Ausfällen im Produktivbetrieb minimiert wird.

Zusätzlich wurden bei platomo getrennte Kunden-Arbeitsbereiche (Workspaces) eingerichtet, um die Daten verschiedener Auftraggeber sicher und vertraulich zu verwalten. In aller Regel unterliegen die von Kunden bereitgestellten Verkehrsvideos einer Vertraulichkeitsvereinbarung (NDA) und dürfen ausschließlich für den jeweiligen Auftrag genutzt werden. Jeder Kunde erhält daher bei platomo ein eigenes geschütztes Verzeichnis zum Upload seiner Videodaten und zum Download der Ergebnisse. Die Verarbeitung dieser Daten erfolgt intern auf einer zentralen OTCloud-Instanz

von platomo, wobei eine strikte Trennung der Aufträge nach Kunden gewährleistet ist. Auf Wunsch kann für einen Kunden auch eine dedizierte OTCloud-Instanz betrieben werden, um eine vollständige physische Separierung der Daten und Prozesse zu erreichen.

6.7.5 Implementierung einheitliches Frontend für alle Teilmodule (platomo)

Für alle Hauptmodule von OpenTrafficCam steht eine Kommandozeilenschnittstelle (CLI) zur Verfügung. Die Bedienparameter dieser CLIs sind nach einem einheitlichen Namensschema benannt, was eine konsistente Handhabung der verschiedenen Module ermöglicht, und die Einarbeitung erleichtert.

Darüber hinaus verfügt OTAnalytics über eine grafische Benutzeroberfläche (GUI) zur komfortableren Durchführung von Analysen (siehe Kapitel *OTAnalytics* für Details). Diese GUI wurde unter Verwendung der Python-Bibliothek Tkinter entwickelt und knüpft gestalterisch an die bisherigen Oberflächen von OTVision und OTAnalytics aus der Vorstudie an. Benutzer, die bereits mit den vorherigen Versionen gearbeitet haben, finden sich dank der vertrauten Gestaltung auch in der neuen Oberfläche schnell zurecht.

Für OTCloud wurde eine webbasierte Benutzeroberfläche umgesetzt, die sowohl der Überwachung der laufenden Verarbeitungsschritte als auch dem Starten neuer Verarbeitungsaufträge dient. Bei der Implementierung dieser Web-UI kommt ein moderner Technologie-Stack zum Einsatz, der Frontend- und Backend-Komponenten kombiniert:

- **nicegui** – ein Python-basiertes Webframework zur einfachen Definition und Bereitstellung von Benutzeroberflächen.
- **Uvicorn** – ein leistungsfähiger Webserver (WSGI-Server) zur Auslieferung der Web-Anwendung.
- **FastAPI** – ein Framework zur Definition und Implementierung der HTTP-Schnittstellen (API) im Backend.
- **Matplotlib** – eine Plotting-Bibliothek zur grafischen Darstellung von Daten (hier z. B. zur Visualisierung von Trajektorien innerhalb der UI).
- **Requests** – eine HTTP-Client-Bibliothek, die verwendet wird, um externe Web-Services aus dem Backend heraus anzusprechen.
- **Tailwind CSS** – ein CSS-Framework, das ein modernes responsives Design der Weboberfläche ermöglicht.
- **Vue.js** – ein JavaScript-Framework zur Erstellung reaktiver Frontend-Komponenten, das interaktive Funktionen und dynamische Inhalte in der UI ermöglicht.
- **Starlette** – ein ASGI-Framework für asynchrone Serverkommunikation, das Echtzeit-Updates und WebSockets zwischen Frontend und Backend unterstützt.
- **Quasar Framework** – ein Framework, das die dynamische Bindung der UI-Komponenten an die Datenmodelle des Backends erleichtert und die Entwicklung komplexer Webinterfaces beschleunigt.

Die OTCloud-Weboberfläche zeigt den aktuellen Betriebszustand des Systems übersichtlich an, siehe Abbildung 22. So wird beispielsweise eine Liste aller Verarbeitungsaufträge inklusive ihres jeweiligen Fortschritts und Status angezeigt (z. B. Schritt im Prozess oder Warteschlangenposition).

Außerdem können über die Web-UI neue Videoverarbeitungs-Jobs gestartet werden. Hierfür steht ein Konfigurationsdialog bereit, in dem alle relevanten OTVision-Parameter eingestellt werden können, ohne dass der Nutzer manuell eine Konfigurationsdatei erstellen muss.

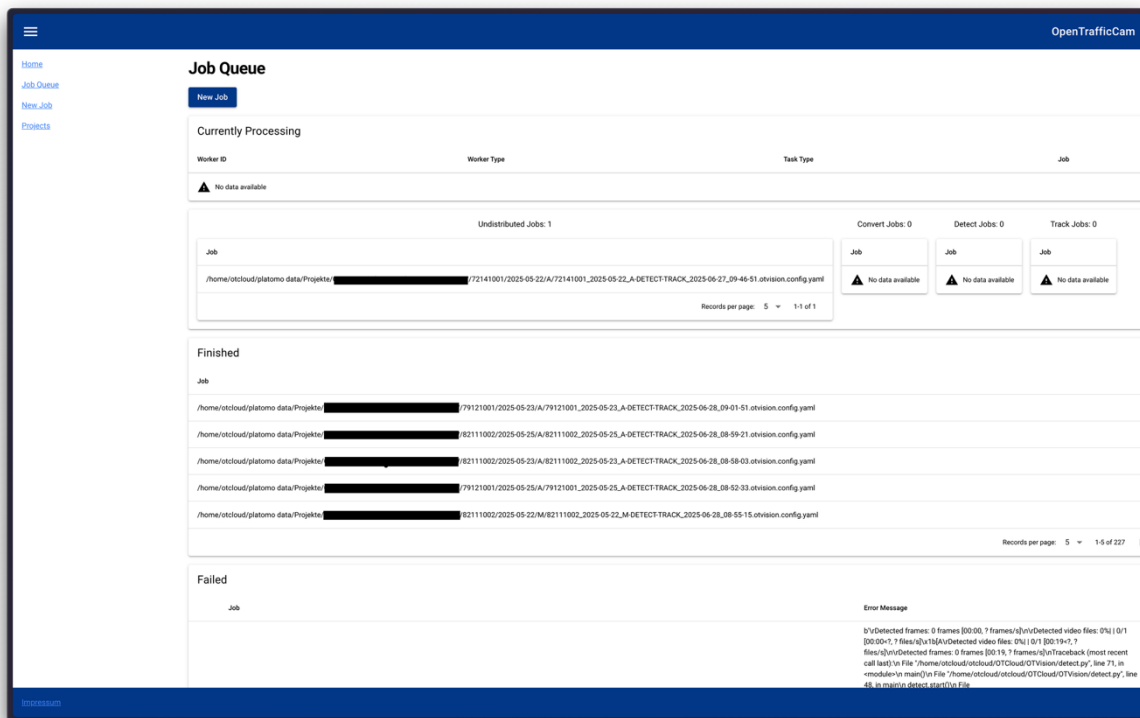


Abbildung 22: OTCloud Benutzeroberfläche zur Anzeige des aktuellen Verarbeitungszustandes

6.7.6 Konzept und Implementierung Continuous Integration und Deployment (platomo)

Im Projekt wurde ein umfassendes Continuous-Integration/Continuous-Deployment-(CI/CD)-Konzept umgesetzt. Alle relevanten Software-Repositorys nutzen eine GitHub-Actions-basierte Pipeline, die für sämtliche Teilprojekte einheitlich gestaltet ist. Diese automatisierten Pipelines führen bei jedem Code-Änderungsstand verschiedene Qualitätssicherungs-Schritte durch.

Zu den zentralen Bestandteilen der CI/CD-Pipelines gehören:

- **Automatisierte Unittests:** Bei jedem neuen Softwarestand werden sämtliche Modultests ausgeführt, um sicherzustellen, dass bestehende Funktionen weiterhin korrekt arbeiten und um Regressionen sofort zu erkennen.
- **Statische Code-Prüfungen:** Die Einhaltung definierter Code-Formatierungsrichtlinien wird überprüft, damit der Code konsistent und wartbar bleibt. Zudem wird mittels statischer Code-Analyse die Einhaltung des Typsystems kontrolliert.
- **Sicherheitsüberprüfung:** Spezialisierte Tools untersuchen den Quellcode auf potenzielle Sicherheitslücken oder Schwachstellen.
- **Performance- und Regressionstests:** Insbesondere für OTAnalytics werden zusätzliche Performance-Benchmarks und Regressionstests auf größeren Datenmengen durchgeführt, um sicherzustellen, dass neue Softwareversionen keine Leistungseinbußen aufweisen und identische Ergebnisse wie vorherige Versionen liefern.

- **Automatisierte Veröffentlichung:** Nach erfolgreichem Durchlauf aller Tests werden neue Versionen von OTAnalytics und OTVision als Python-Pakete auf *PyPI* bereitgestellt. Dadurch können externe Anwender diese Module einfach über Paketmanager installieren und nutzen.

Durch diese CI/CD-Maßnahmen wird ein hoher Qualitätsstandard der Software gewährleistet. Fehler werden frühzeitig erkannt, und es wird sichergestellt, dass alle Komponenten des Systems stabil und in gleichbleibender Qualität ausgeliefert werden. Die einheitliche Gestaltung der Pipelines für alle Projektteile reduziert zudem den Aufwand für Wartung und Weiterentwicklung.

6.7.7 Sondierung Technologiepartner und Evaluierung geeigneter Server (platomo)

Im Verlauf des Projekts wurden verschiedene Technologiepartner und Server-Infrastrukturen sondiert, um einen geeigneten Betrieb der OTCloud-Plattform sicherzustellen. Zunächst erfolgte eine Anforderungsanalyse für die geplante Datenverarbeitung. Wesentliche Anforderungen waren dabei:

- **Leistungsstarke GPU-Ressourcen:** Für die Objektdetektion in Videos mittels OTVision wird eine Grafikkarte mit hoher Rechenleistung benötigt.
- **Nachvollziehbares, planbares Abrechnungsmodell:** Die Kosten für die Nutzung von Cloud- oder Serverdiensten sollten transparent und gut kalkulierbar sein, um die Wirtschaftlichkeit des Betriebs abschätzen zu können.
- **DSGVO-konforme Datenverarbeitung:** Die Verarbeitung der Videodaten muss den Vorgaben der Datenschutz-Grundverordnung entsprechen, da mitunter Personen und Fahrzeuge erkannt werden und sensible Informationen geschützt werden müssen.

Vor diesem Hintergrund wurden mehrere potenzielle Anbieter und Hosting-Lösungen evaluiert. Unter anderem wurden Testläufe bei Ionos (Cloud-Dienstleister) und mit Shadow-PC (virtuelle Hochleistungsrechner als Service) durchgeführt, um deren Eignung für OTCloud zu prüfen. Darüber hinaus wurde – beschränkt auf die vom Partner gelieferten Datensätze – das Hochleistungsrechencluster der TU Dresden genutzt, um die Videoverarbeitung unter realistischen Bedingungen zu testen. Dieses Cluster kann allerdings nur für Videos der TU Dresden zum Einsatz kommen und nicht für externe Daten, da das Rechenzentrum Forschungsvorhaben der TU-Dresden und anderen Forschungseinrichtungen vorbehalten ist. Des Weiteren wurden erste Gespräche mit A1 Digital als möglichem IT-Infrastrukturpartner geführt. Die Ergebnisse dieser Sondierungen flossen in die Entscheidung ein, welche Serverumgebung während des Projekts genutzt wurde, und bilden eine wichtige Grundlage für die Auswahl geeigneter Hosting-Strategien in zukünftigen Projekten.

6.7.8 Wiederkehrende Optimierung Performance und Zuverlässigkeit der Hard- und Software (platomo)

Im Projektverlauf wurden kontinuierlich Maßnahmen zur Optimierung der Performance sowie der Zuverlässigkeit von Hard- und Software umgesetzt. So wurden gezielt Performance-Verbesserungen – etwa bei dem testweisen Einsatz von Containern – durchgeführt, um die Effizienz der Videoverarbeitung zu steigern.

OTCloud wurde im Frühjahr 2024 bei platomo in den Produktivbetrieb überführt. Anschließend wurde die Plattform fortlaufend weiterentwickelt und den praktischen Erfordernissen angepasst.

Zur Qualitätssicherung kamen sogenannte *Quality Gates* zum Einsatz, die an verschiedenen Punkten im Verarbeitungsprozess Prüfungen durchführen:

- **Validierung vor der Prozessierung:** Bereits vor dem Start eines Auftrags werden die Eingangsdaten (Video und Konfiguration) auf Plausibilität und Vollständigkeit überprüft.
- **Validierung nach der Prozessierung:** Nach Abschluss der Verarbeitung wird das Ergebnis geprüft, um sicherzustellen, dass alle vorgesehenen Ausgabedateien erzeugt wurden und keine Fehler aufgetreten sind.
- **Plausibilitätsprüfung der erkannten Objekte:** Die erfassten Verkehrsteilnehmer (detektierte Klassen) werden hinsichtlich realistischer Mengen und Verteilungen validiert, um etwaige Fehlzählungen oder -klassifizierungen zu identifizieren.

Darüber hinaus wurde das System so erweitert, dass sich auch umfangreiche Messkampagnen komfortabel verarbeiten lassen. So können beispielsweise groß angelegte Verkehrszählungen – wie die bundesweite **Straßenverkehrszählung (SVZ)** – mit OTCloud effizient ausgewertet und die Ergebnisse validiert werden. Die Plattform ist in der Lage, große Datenmengen und eine hohe Zahl paralleler Aufträge zu bewältigen, was die Analyse solcher komplexen Datensätze erheblich vereinfacht.

6.8 AP 8 Testfeld (MSP)

Zur Belegung der Praxistauglichkeit sowie der Entwicklung an OTMulticamera wurde das Gesamtsystem auf einem realen Testfeld – dem innerstädtischen vierarmigen Knotenpunkt Chemnitzer Straße/Würzburger Straße in Dresden – erprobt. Das Testfeld umfasste dabei vier an Laternenmasten auf etwa 4 Metern Höhe befestigte und auf den Knotenpunktbereich ausgerichtete OTCameras. Der jeweilige Aufnahmebereich der vier stationären Kameras ist in Abbildung 23 zu sehen. Vom Testfeld wurde des Weiteren ein umfangreicher 3D Scan angefertigt, der primär für die Entwicklung von OTCamera diente (siehe dazu Kapitel 6.3).

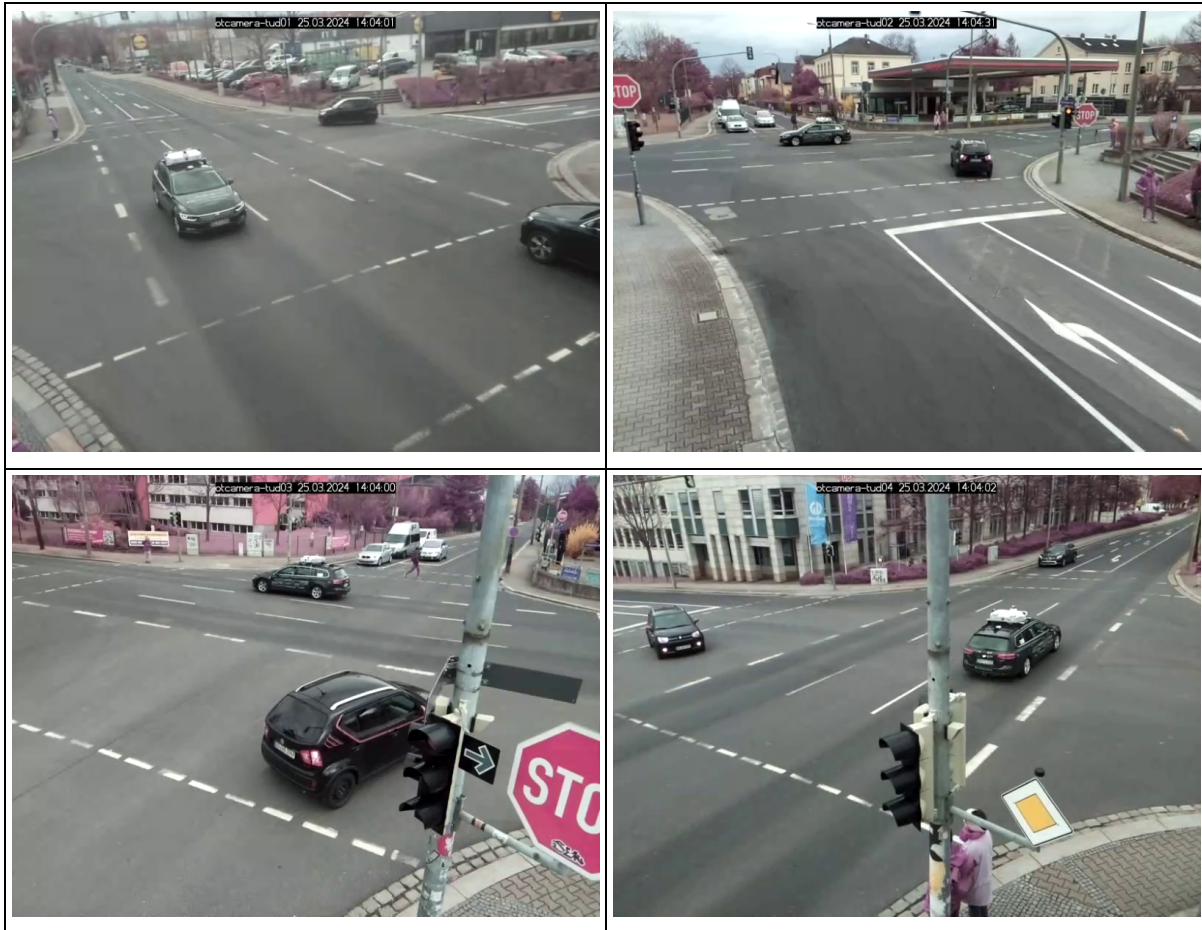


Abbildung 2324: Screenshots mit Messfahrzeug aus den Videos der vier Kameras des Testfelds Knotenpunkt Chemnitzer Straße/Würzburger Straße in Dresden

Zur Beurteilung der vom Gesamtsystems ausgegebenen stromfeinen Verkehrszählwerte wurde die automatisierte Zählung einer OTCamera mit den real vorliegenden stromfeinen Verkehrszählwerten (Ground Truth) verglichen. Zur Erstellung dieser Ground Truth wurden die Videos der OT-Cameras und der für die manuelle Zählung im Projekt entwickelte OTGroundTruther verwendet. Der OTGroundTruther ist eine open-source Software und ermöglicht die effiziente und genaue stromfeine manuelle Zählung des Verkehrs mit Hilfe von Videos. Er dient damit sowohl zur Erhebung der Ground Truth, als Zähltool für Zählaufgaben, an denen die automatisierte Auswertung an ihre Grenzen kommt, und zur visuellen Darstellung der Zählung des OTC-Gesamtsystems.

Das Videomaterial der OTCamera mit Blick Richtung Süden (in Abbildung 23 oben links) wurde mit der entwickelten OTC-Software verarbeitet und ausgewertet, sodass ein Ergebnis für die stromfeinen Kfz-Verkehrsströme für eine Stunde vorliegen. Der gleiche Zeitraum wurde mit Hilfe des OT-GroundTruthers und des Videomaterials – zur Gewährung der Genauigkeit - manuell gezählt, sodass auch eine stromfeine Ground Truth als Vergleich vorlag. In dieser Stunde fuhren 762 Kfz über den Knotenpunkt. Das OTC-Gesamtsystem zählte hingegen 731 Kfz und ordnete 726 davon dem korrekten Verkehrsstrom zu. Somit ergibt sich eine stromfeine Zählgenauigkeit für Kfz von 95%. 36 Kfz wurden hingegen nicht gezählt, indem sie gar nicht erkannt oder dem falschen Strom zugeordnet wurden. 5 Kfz wurden vom Gesamtsystem nicht dem korrekten Strom zugeordnet. In Tabelle 2 ist die zugehörige Konfusionsmatrix dargestellt.

Tabelle 2: Konfusionsmatrix der stromfeinen Verkehrszählung am Testfeld

	Kfz vorhanden	Kein Kfz vorhanden	Summe
Kfz erkannt	726	5	731
Kein Kfz erkannt	36	-	-
Summe	762	-	

Es konnte somit gezeigt werden, dass das Gesamtsystem den Kfz-Verkehr auf einem hohen Genauigkeitsniveau stromfein zählen kann. Dabei ist jedoch zu berücksichtigen, dass die Genauigkeit stark von dem Verkehrsraum, der Verkehrszusammenstellung, den Lichtverhältnissen, der Kameraposition und der Kameraausrichtung abhängt. So sind beispielsweise viele der nicht erkannten Kfz das Resultat von Verdeckungen durch andere, meist große Kfz. Eine hohe Kameraposition reduziert das Auftreten von Verdeckungen und wirkt Sicht somit positiv auf die Zählgenauigkeit aus.

6.9 AP 9 Projektmanagement und Öffentlichkeitsarbeit (platomo)

6.9.1 Gesamtprojektleitung (Issue-Management, Code-Reviews, Zeitplanmanagement und Kommunikation mit Dritten) (platomo)

Für das strukturelle Projektmanagement wurde OpenProject eingeführt. Neue Feature-Anfragen und Fehler werden als Arbeitspakete erfasst, mit den entsprechenden GitHub-Pull-Requests verknüpft und gemeinsam mit veröffentlichten Versionen als Milestones abgebildet, was eine transparente Nachverfolgung des Fortschritts ermöglicht.

Ein verbindlicher Code-Review-Prozess wurde implementiert. Dabei wurden klare Prüfungskriterien definiert – beispielsweise Code-Struktur, Einhaltung von Coding-Richtlinien, Testabdeckung und Dokumentationsgrad – um eine konstante Codequalität sicherzustellen.

Für das Zeitplanmanagement wird Clockify eingesetzt. Alle Teammitglieder erfassen täglich ihre Entwicklungszeiten. Diese Daten werden regelmäßig ausgewertet, um die Ressourcennutzung im Projektverlauf nachvollziehbar und optimierbar zu machen.

Die zentrale Anlaufstelle für externe Kontaktaufnahmen ist die E-Mail-Adresse info@opentrafficcam.org. Zusätzlich wurde die Projektwebsite mit umfassenden Dokumentationen bereitgestellt, um Interessierte jederzeit mit aktuellen Informationen zu versorgen.

6.9.2 Projektdokumentation (öffentliche Auftakt- und Abschlusstreffen, Zwischen- bzw. Endberichte, drei Statusberichte) (Alle Partner)

Die Auftakt- und Abschlusstreffen fanden als Online-Veranstaltung über Microsoft Teams statt und lockten jeweils mehr als 50 Teilnehmende aus dem gesamten Bundesgebiet an. Die hohe Beteiligung zeigte sich in zahlreichen Fragen, die direkt in den Meetings beantwortet wurden; offene Punkte wurden im Nachgang geklärt. Die Präsentationsmaterialien wurden jeweils mit den Teilnehmenden geteilt. Darüber hinaus wurden von allen Projektpartnern die erforderlichen Statusberichte erstellt und eingereicht.

6.9.3 Dokumentation des Quellcodes und Erstellung Anwendungsdokumentation (Alle Partner)

Alle Module des Projekts sind detailliert auf opentrafficcam.org dokumentiert:

6.9.3.1 OTCamera

Die Schritt-für-Schritt-Anleitung erläutert den Aufbau (Raspberry Pi Zero W, Kamera-Modul, PCB, Li-Ion-Zelle, USB-Battery sowie wetterfestes Gehäuse und Ballhead-Befestigung) sowie das sichere Montieren vor Ort – unter Beachtung von Sicherheitshinweisen wie Absturzschutz, Diebstahlsicherung, Genehmigungen und Datenschutzanforderungen. Die Bedienung erfolgt über vier Schalter – Reset, Boot, Wi-Fi, 24/7 – sowie drei LEDs, die den Systemstatus (z. B. Stromversorgung, WLAN-Timer oder Aufnahmeleitung) anzeigen. Es werden verschiedene Blinkmuster erklärt, z. B. kontinuierlich leuchtendes grünes LED für Boot oder fünfsekündiges Blinken der Aufnahmelampe

6.9.3.2 OTVision

Die Dokumentation beschreibt ausführlich die CLI-Nutzung mit den drei Pipeline-Schritten convert, detect und track. Mit convert werden .h264-Videos in .mp4 konvertiert, die Frame-Rate kann aus dem Dateinamen abgeleitet werden (via _FR<fps>_) oder explizit angegeben werden. Bei detect erfolgt die Objekterkennung via YOLOv8 unter Angabe des Video-Dauerparameters sowie eines bestimmten Benennungsformats (YYYY-MM-DD_HH-MM-SS), um lückenlose Verknüpfung von Detections über Videogrößen hinweg sicherzustellen. Im Schritt track werden Detektionen aus .otdet-Dateien mit konfigurierbaren Parametern (z. B. sigma_l, sigma_h, sigma_iou, t_min, t_miss_max) zu Trajektorien über Sequenzen hinweg verbunden. Die Parameter für alle Module können per CLI oder YAML-Konfigurationsdatei angepasst werden.

6.9.3.3 OTAnalytics

Die GUI ermöglicht das Anlegen von Projekten mit definierbarem Namen und Startzeitpunkt (ISO-Format oder deutsches Datumsformat). Trajektorien werden per .ottrk-Datei geladen, Videos optional zugeordnet, und über definierte Sektionen und Verkehrsflüsse analysiert. Das Interface bietet Konfigurationsoptionen für Trajektorien- und Filterdarstellung sowie umfangreiche Exportfunktionen (CSV/Excel) für Verkehrszählungen, Fahrzeugklassifikationen und mehr.

6.9.3.4 OTLabels

Der Trainingsworkflow umfasst die Annotation im CVAT, Aufbereitung der Daten (z. B. cvat_to_yolo), Retraining des Modells sowie die Validierung mit OTValidate (Leistungsmetriken, Modellvergleich, Objektklassifikation & Tracking).

6.9.4 Erstellung von Walkthroughs / FAQ (MSP, platomo)

Die im Rahmen der Vorstudie erzeugten FAQ wurden in diesem Projekt verfeinert. Zusammen mit der Dokumentation wurden sie auf www.opentrafficcam.org veröffentlicht.

6.9.5 Erstellung von Blog- oder Social Media Beiträgen und Pressemitteilungen (MSP, platomo)

Alle Messeauftritte sowie Vortragstermine wurden über LinkedIn bekanntgeben und begleitet. Beteiligungen an mFUND-Formaten wurden zusätzlich über mFUND-Kanäle beworben. Darüber hinaus wurden Fortschritte der Entwicklung mit zunehmender Reichweite teilweise ebenfalls auf LinkedIn veröffentlicht.

6.9.6 Organisation und Durchführung eigener mFUND-Formate (Workshop, Fachaustausch und Vorträge) (Alle Partner)

In Kooperation mit der Open Source Business Alliance wurde ein Workshop zur Geschäftsmodellentwicklung im Open-Source-Kontext durchgeführt. Dabei wurde zunächst das Projekt OpenTraf-

ficCam_live vorgestellt. Anschließend wurden mögliche Geschäftsfelder für die platomo GmbH diskutiert. Zum Schluss wurde zusammen mit den Teilnehmenden eine Anforderungsanalyse an ein Geschäftsmodell aus Kunden- und Unternehmenssicht erarbeitet.

Zudem wurde ein Workshop zum Thema „Datenschutz bei videobasierten Verkehrserhebungen“ organisiert und mit über 70 Teilnehmenden durchgeführt.

6.9.7 Aktive Teilnahme an anderen mFUND-Formaten und Veröffentlichungen (Alle Partner)

Die Projektpartner haben an folgenden mFUND-Formaten aktiv teilgenommen.

- Abschlussveranstaltung mFUND-Projekt „KI-Cam“
- mFUND Konferenz
 - 2022
 - 2023 (Leitung Workshop „Open Source Geschäftsmodelle“)
 - 2024
- mFUND Checkout am 21.05.2025

7 Zahlenmäßiger Nachweis

Die platomo GmbH konnte das Projekt nahezu im vorab kalkulierten Kostenrahmen abschließen. Die insgesamt entstandenen Selbstkosten lagen mit 822.008,85 € um 6.198,45 € (0,76%) über den vorab kalkulierten Kosten.

Projektpartner	Geplante Kosten	Tatsächliche Kosten
platomo GmbH	815.810,40 €	822.008,85 €
TUD (MSP und PHOTO)	390.220,00 €	361.107,37 €

7.1 platomo GmbH

Position	Beschreibung	Tatsächliche Kosten
0837	Personalkosten	764.908,21 €
0838	Reisekosten	7.510,20 €
0847	Vorhabensspezifische Abschreibungen	2.792,60 €
0850	Sonstige unmittelbare Leistungen	46.797,84 €

Die Kosten für die Entwicklung der Soft- und Hardware bestanden im Wesentlichen aus den Personalkosten der Mitarbeitenden der platomo GmbH. Personalkosten unter Position 0837 sind Aufwände für das direkt bei der platomo GmbH angestellte Personal. Die Reisekosten sind im Wesentlichen durch die beschriebenen Messe- und Kongressbesuche entstanden und entsprechen unter einem Prozent der gesamten Kosten. Unter vorhabensspezifischen Abschreibungen wurde neue Hardware verbucht, die für das Training neuer Modelle und die Datenspeicherung notwendig war. Sonstige unmittelbare Leistungen enthalten Unteraufträge zur datenschutzrechtlichen Einordnung videobasierter Erhebungen, der externen Prüfung unserer Hardwarekomponenten, der Produktion von Hardwareprototypen und der Beschaffung von Werbematerial für OpenTrafficCam auf verschiedenen Messen.

7.2 TUD (MSP und PHOTO)

Position	Beschreibung	Tatsächliche Kosten
0837	Personalkosten	336.815,53 €
0838	Reisekosten	5.367,22 €
0847	Vorhabensspezifische Abschreibungen	8.155,12 €
0850	Sonstige unmittelbare Leistungen	10.769,50 €

Die Kosten für die Entwicklung der Software bestanden aus den Personalkosten der Mitarbeitenden der Professuren. Die Reisekosten ergaben sich durch Abstimmungstermine mit der platomo GmbH und der Teilnahme an Kongressbesuchen. Vorhabensspezifische Abschreibungen umfassen angeschaffte Hardware zum Aufbauen verschiedener Kameras zu Testzwecken, zur Anfertigung von Videomaterial und für Performancetests. Sonstige unmittelbare Leistungen enthalten Unteraufträge wie der Prüfung von Hardwarekomponenten und die Durchführung eines Online-Workshop mit Verkehrsfachleuten zum Thema Datenschutz beim Filmen des öffentlichen Verkehrsraums.

8 Notwendigkeit und Angemessenheit der geleisteten Arbeit

Die in den vergangenen drei Projektjahren erbrachten Arbeiten waren in ihrem Umfang und ihrer zeitlichen Tiefe notwendig, um die in der Vorhabenbeschreibung definierten Entwicklungsziele zu erreichen. Alle acht Arbeitspakete (AP Datenschutz, OTCamera, OTMultiCamera, OTLabels, OTVision, OTCloud, OTAnalytics / Testfeld sowie Projektleitung / Öffentlichkeitsarbeit) adressieren unmittelbar die wesentlichen Bausteine einer marktreifen, datenschutzkonformen Videotechnik für multimodale Verkehrserhebungen. Ihre Inhalte – von der Hardware-Weiterentwicklung über den Trainings-Workflow bis zur Bereitstellung einer Cloud-Plattform – bilden eine logisch aufeinander aufbauende Prozesskette, sodass der Erfolg des Gesamtvorhabens den vollständigen Durchlauf aller AP voraussetzt. Eine Reduktion einzelner Schritte hätte zwangsläufig zu funktionalen Lücken geführt und war daher nicht vertretbar.

Zu Projektbeginn war vorgesehen, dass drei Geschäftsführer-Entwickler den Großteil der Entwicklungstätigkeiten in Vollzeit übernehmen. Im Verlauf des Vorhabens wurden diese ursprünglich eingeplanten Positionen teilweise durch angestellte Entwickler ersetzt. Da ein Teil der neuen Teammitglieder noch vergleichsweise wenig Berufserfahrung besaß, konnte ihr Bruttolohn unterhalb des für Geschäftsführer üblichen Niveaus angesetzt werden; im Gegenzug benötigten sie für einzelne Aufgaben mehr Zeit, um die angestrebte fachliche Tiefe zu erreichen. Durch diese Balance aus niedrigerem Stundensatz und höherem Zeitaufwand blieben die Gesamtkosten für das Personal annähernd auf dem ursprünglich kalkulierten Niveau. Ergänzend wurden weiterhin studentische Hilfskräfte flexibel eingesetzt, um Lastspitzen bei Annotationen, CI/CD-Automatisierung und Cloud-Tests kosteneffizient abzufedern. Dieses angepasste Staffing-Modell erwies sich als tragfähig, sicherte alle kritischen Entwicklungsschritte intern ab und verhinderte Know-how-Verluste.

Der überwiegende Teil der Mittel floss wie geplant in Personalaufwendungen, während Sach- und Reisekosten bewusst niedrig gehalten wurden; externe F&E-Dienstleistungen wurden nicht beauftragt. Investitionen in Hardware beschränkten sich auf Entwicklungs- und Trainingsinfrastruktur, die für die Modellierung der KI-Verfahren zwingend erforderlich war. Damit wurde das Budget konsequent dort eingesetzt, wo es unmittelbar zur Wertschöpfung beitrug, und blieb insgesamt im beantragten Rahmen.

Die Umsetzung orientierte sich am im Antrag vorgelegten Balkenplan; substanzielle Planabweichungen oder zusätzliche Iterationsschleifen traten nicht auf. Einzelne Teilaufgaben – etwa die

modularisierte Objektdetektion in OTVision oder die Abstimmung des Datenschutzkonzepts – erforderten phasenweise mehr Entwicklungszyklen, konnten jedoch innerhalb des vorgesehenen Gesamtbudgets aufgefangen werden. Die parallel geführte Projektleitung sicherte eine kontinuierliche Abstimmung mit Praxispartnern, minimierte Reibungsverluste und verhinderte Doppelarbeiten.

In Summe belegen Umfang, Qualität und Ergebnisse der Arbeitspakete, dass sowohl der geleistete Zeitaufwand als auch die eingesetzten Ressourcen exakt dem tatsächlichen Entwicklungsbedarf entsprachen und damit als angemessen zu bewerten sind.

9 Voraussichtlicher Nutzen und Verwertbarkeit

Aufbauend auf den Ergebnissen konnte die platomo GmbH bereits mehrere Projekte mit einem Gesamtvolumen von ca. 1,3 Mio. € akquirieren, in dem die OpenTrafficCam eingesetzt wird. Zahlreiche weitere Anfragen laufen.

- Verkauf OTCamera (Kommunen, Büros): ca. 36.000 € Umsatz
- SVZ 2025: ca. 110.000 € Umsatz
- Erhebungen für Planungsprojekte: ca. 250.000€ Umsatz
- Videoauswertungen: ca. 10.000€ Umsatz
- Parkraumerhebungen: ca. 40.000€ Umsatz
- Live-System (TUD): ca. 900.000€ Umsatz

Auch die Professur für Mobilitätssystemplanung der TU Dresden plant den Einsatz der Technik in anderen bereits bewilligten Forschungsprojekten.

Darüber hinaus haben auch bereits vereinzelt externe Programmierende in geringem Umfang Quellcode zur Entwicklung von OpenTrafficCam beigetragen.

10 Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen

Seit Beginn des Projekts OpenTrafficCam_live im Jahr 2022 haben sich die technischen Möglichkeiten der Edge-Prozessierung erheblich weiterentwickelt. KI-basierte Kamerasysteme, welche Video Daten unmittelbar am Ort der Erfassung verarbeiten, sind heute keine Vision der Zukunft mehr, sondern bereits in zahlreichen Pilotprojekten und kommerziellen Anwendungen etabliert. Diese Kameras erkennen, klassifizieren und zählen Verkehrsteilnehmer direkt im Gerät und ermöglichen somit Echtzeitauswertungen vor Ort. Führende Hersteller setzen hierfür auf spezialisierte Deep-Learning-Chips, welche unterschiedliche Verkehrsteilnehmertypen zuverlässig unterscheiden können, darunter Fußgänger, PKW, LKW und Fahrräder [2], [3], [4].

Vor diesem Hintergrund erscheint die Entscheidung von OpenTrafficCam_live, weiterhin auf nachgelagertes Post-Processing zu setzen, zunächst widersprüchlich. Im Gegensatz zu Live-Analysen verfolgt OpenTrafficCam_live jedoch ein klares Ziel: die datenschutzkonforme, qualitativ hochwertige Datenerfassung für Forschungs- und Planungszwecke. Statt der unmittelbaren Steuerung des Verkehrs steht hier die umfassende, präzise und rechtskonforme Erhebung von Verkehrsdaten im Vordergrund. Die aufwändigen Verfahren zur Detektion und Verfolgung von Verkehrsteilnehmern sowie komplexe Klassifizierungen nach deutschen Richtlinien (wie z. B. [31]) erfordern auch heute noch eine Rechenleistung, die auf kompakten Edge-Geräten nicht ohne weiteres zur Verfügung steht. Daher erlaubt das nachgelagerte Post-Processing eine flexiblere und qualitativ hochwertigere Datenanalyse, die zudem jederzeit nachvollziehbar bleibt und eine Validierung der Ergebnisse ermöglicht.

Auch die Einhaltung des Datenschutzes spielt bei diesem Ansatz eine entscheidende Rolle. Im Sinne eines „Privacy by Design“-Prinzips wurde die Pipeline von OpenTrafficCam so entwickelt, dass erst gar keine personenbezogenen Daten erhoben werden, wohingegen die Prozessierung auf der Kamera mit leistungsschwächeren Modellen meist eine bessere Bildqualität benötigt, um vergleichbare Ergebnisse zu liefern. Die Vorgehensweise von OpenTrafficCam ermöglicht jedoch eine bessere Kontrolle und Anpassung an rechtliche Vorgaben und ist in dieser Hinsicht gegenüber autonom arbeitenden KI-Kameras vorteilhaft.

Ein weiterer Grund für das Post-Processing ist auch wirtschaftlicher und praktischer Natur. Edge-Geräte mit ausreichender Rechenkapazität, wie GPU-Module, sind kostenintensiv und wartungsanfällig. Demgegenüber setzt OpenTrafficCam auf günstige Hardware wie Raspberry Pi, die modular erweiterbar ist und deren Datenanalyse zentralisiert durchgeführt wird. Dieser Ansatz ist insbesondere für kurzfristige und gezielte Verkehrsuntersuchungen sinnvoll und ermöglicht eine effiziente Bündelung und Auswertung der Daten. Kommunen und Ingenieurbüros haben durch diesen modularen Aufbau die Möglichkeit, jetzt günstige Hardware zu beschaffen und die kostenintensive Komponente (GPU) auf ihre Bedürfnisse hin anzupassen. Sollen später neue Funktionen ergänzt werden, müssen nicht alle Kameras im Straßenraum, sondern lediglich die GPU im Serverzentrum und die Auswertesoftware erweitert werden.

Ein zentrales technisches Element des Projekts ist die YOLO-Architektur („You Only Look Once“), die sich seit 2015 als Standard für die Echtzeit-Objektdetektion etabliert hat. Die rasanten Weiterentwicklungen der YOLO-Modelle von v5 bis aktuell v12 brachten signifikante Verbesserungen hinsichtlich Geschwindigkeit und Genauigkeit. YOLOv7 stellte 2022 einen neuen Maßstab dar, insbesondere bei der Erkennung kleinerer Objekte und Objekt-Tracking in komplexen Verkehrsszenen [8]. YOLOv8, erschienen 2023, fokussierte auf praktische Verbesserungen der Benutzerfreundlichkeit und Anpassbarkeit [10]. Die Version YOLOv9 integrierte innovative Konzepte wie Programmable Gradient Information (PGI) und Generalized Efficient Layer Aggregation Network (GELAN), welche konvolutionale Detektoren deutlich effizienter und genauer machen [35].

In der Verkehrsanalyse hat YOLO daher einen hohen Stellenwert erlangt. Insbesondere YOLOv7 zeigte sich in ersten Studien als besonders robust und geeignet für den Verkehrseinsatz, mit Verbesserungen durch zusätzliche Aufmerksamkeitsmodule, ist allerdings aufgrund des verwendeten Frameworks nicht ohne weiteres in die Python-Umgebung von OpenTrafficCam zu integrieren. YOLOv8 erweiterte die YOLO-Funktionalität weiter und machte diese leichter zugänglich und anpassbar, was für OpenTrafficCam_live besonders relevant ist, da projektspezifische Anpassungen an Verkehrsdaten vorgesehen sind. YOLOv9 eröffnet perspektivisch sogar die Möglichkeit, kleinere und dennoch präzise Modelle einzusetzen, was wiederum die Implementierung von Edge-basierten Lösungen erleichtern könnte [35].

Zusammenfassend bleibt OpenTrafficCam_live mit seinem Post-Processing-Ansatz bewusst auf einem methodisch gut begründeten Weg. Während die Technologie des Edge-Computing rasante Fortschritte macht, priorisiert das Projekt derzeit Qualität, Flexibilität und Datenschutz. Gleichzeitig werden die Entwicklungen im Edge-Computing genau verfolgt, um zukünftig bei entsprechender technischer Reife eine Integration von Echtzeitauswertungselementen vorzunehmen. Damit positioniert sich OpenTrafficCam_live strategisch zwischen den aktuellen Anforderungen der Verkehrsforschung und den zukunftsweisenden technischen Innovationen.

11 Veröffentlichungen

Alle implementierten Funktionen sind als Open Source Code unter der freien Lizenz GPL3 auf GitHub veröffentlicht. Die Website mit weiterführenden Informationen ist online.

Bisher sind keine wissenschaftlichen Veröffentlichungen erfolgt.

Es ist, nach Laufzeitende, eine gemeinsame Veröffentlichung der Ergebnisse in der Fachzeitschrift „Straßenverkehrstechnik“ des Kirchbaumverlags geplant. Es ist außerdem geplant die Ergebnisse in folgenden Arbeitskreisen der Forschungsgesellschaft für Straßen- und Verkehrswesen vorzustellen, in denen Mitarbeitende der Partner Mitglied sind:

- AA 2.13 Verkehrssicherheitsbewertung von Straßen
- AK 3.10.1 – Empfehlungen zur mikroskopischen Verkehrsflusssimulation
- AK 3.13 BG Radverkehr
- AK 3.9.8 Merkblatt zur Örtlichen Unfalluntersuchung in Unfallkommissionen

Zudem arbeiten die Projektpartner im FGSV AA 1.2 Erhebung und Prognose des Verkehrs und diversen untergeordneten Arbeitskreisen (1.2.5 Empfehlungen für die Erhebungen von Verkehr, 1.2.X Empfehlungen für Kurzzeitzählungen (in Gründung)) mit.

Literaturverzeichnis

- [1] „Data from Sky (Website)“. 30. Juni 2025. Zugriffen: 30. Juni 2025. [Online]. Verfügbar unter: <https://datafromsky.com>
- [2] „mioVision (Website)“. 30. Juni 2025. Zugriffen: 30. Juni 2025. [Online]. Verfügbar unter: <https://datafromsky.com>
- [3] „Viscando (Website)“. 30. Juni 2025. Zugriffen: 30. Juni 2025. [Online]. Verfügbar unter: <https://datafromsky.com>
- [4] „DataCollect (Website)“. 30. Juni 2025. Zugriffen: 30. Juni 2025. [Online]. Verfügbar unter: <https://www.datacollect.com>
- [5] S. Ren, K. He, R. Girshick, und J. Sun, „Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.“, *Advances in Neural Information Processing Systems*, Bd. 28, 2015.
- [6] W. Liu u. a., „SSD: Single Shot MultiBox Detector“, in *Computer Vision – ECCV 2016*, Bd. 9905, B. Leibe, J. Matas, N. Sebe, und M. Welling, Hrsg., in Lecture Notes in Computer Science, vol. 9905. , Cham: Springer International Publishing, 2016, S. 21–37. doi: 10.1007/978-3-319-46448-0_2.
- [7] G. Jocher u. a., *ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation*. (22. November 2022). Zenodo. doi: 10.5281/ZENODO.3908559.
- [8] C.-Y. Wang, A. Bochkovskiy, und H.-Y. M. Liao, „YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors“, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, S. 7464–7475.
- [9] Y. Zhang, Y. Sun, Z. Wang, und Y. Jiang, „YOLOv7-RAR for urban vehicle detection“, *Sensors*, Bd. 23, Nr. 4, S. 1801, 2023.
- [10] YOLOv8. (10. Januar 2023). Ultralytics. Zugriffen: 30. Juni 2025. [Online]. Verfügbar unter: <https://docs.ultralytics.com/models/yolov8/>
- [11] M. Jakubec, M. Cingel, E. Lieskovská, und M. Drliciac, „Integrating Neural Networks for Automated Video Analysis of Traffic Flow Routing and Composition at Intersections“, *Sustainability*, Bd. 17, Nr. 5, S. 2150, März 2025, doi: 10.3390/su17052150.
- [12] J. Tao, J. Shi, Y. Chen, und J. Wang, „Improved YOLOv4 Infrared Image Pedestrian Detection Algorithm“, in *Machine Learning and Intelligent Communications*, Bd. 438, X. Jiang, Hrsg., in Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 438. , Cham: Springer International Publishing, 2022, S. 226–233. doi: 10.1007/978-3-031-04409-0_21.
- [13] I. Ashraf, S. Hur, G. Kim, und Y. Park, „Analyzing Performance of YOLOx for Detecting Vehicles in Bad Weather Conditions“, *Sensors*, Bd. 24, Nr. 2, S. 522, Jan. 2024, doi: 10.3390/s24020522.
- [14] E. Bochinski, V. Eiselein, und T. Sikora, „High-Speed tracking-by-detection without using image information“, in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Lecce, Italy: IEEE, Aug. 2017, S. 1–6. doi: 10.1109/AVSS.2017.8078516.
- [15] A. Bewley, Z. Ge, L. Ott, F. Ramos, und B. Upcroft, „Simple Online and Realtime Tracking“, in *2016 IEEE International Conference on Image Processing (ICIP)*, Sep. 2016, S. 3464–3468. doi: 10.1109/ICIP.2016.7533003.
- [16] N. Wojke, A. Bewley, und D. Paulus, „Simple Online and Realtime Tracking with a Deep Association Metric“, 21. März 2017, *arXiv*: arXiv:1703.07402. doi: 10.48550/arXiv.1703.07402.
- [17] P. Bergmann, T. Meinhardt, und L. Leal-Taixe, „Tracking without bells and whistles“, in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Okt. 2019, S. 941–951. doi: 10.1109/ICCV.2019.00103.
- [18] Y. Zhang, C. Wang, X. Wang, W. Zeng, und W. Liu, „FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking“, *Int J Comput Vis*, Bd. 129, Nr. 11, S. 3069–3087, Nov. 2021, doi: 10.1007/s11263-021-01513-4.

- [19] Y. Zhang u. a., „ByteTrack: Multi-Object Tracking by Associating Every Detection Box“, 7. April 2022, *arXiv*: arXiv:2110.06864. doi: 10.48550/arXiv.2110.06864.
- [20] H.-M. Hsu, J. Cai, Y. Wang, J.-N. Hwang, und K.-J. Kim, „Multi-Target Multi-Camera Tracking of Vehicles Using Metadata-Aided Re-ID and Trajectory-Based Camera Link Model“, *IEEE Trans. on Image Process.*, Bd. 30, S. 5198–5210, 2021, doi: 10.1109/TIP.2021.3078124.
- [21] P. Khorramshahi, V. Shenoy, M. Pack, und R. Chellappa, „Scalable and Real-time Multi-Camera Vehicle Detection, Re-Identification, and Tracking“, 15. April 2022, *arXiv*: arXiv:2204.07442. doi: 10.48550/arXiv.2204.07442.
- [22] C. Zhang, C. Yang, D. Wu, H. Dong, und B. Deng, „Cross-view vehicle re-identification based on graph matching“, *Appl Intell*, Bd. 52, Nr. 13, S. 14799–14810, Okt. 2022, doi: 10.1007/s10489-022-03349-y.
- [23] Z.-Y. Yang, C.-W. Liu, C.-H. Chen, C.-W. Chen, und M.-C. Hsyu, „Score-based matching for city-scale multi-target multi-camera vehicle tracking“, *Multimed Tools Appl*, Bd. 84, Nr. 10, S. 7647–7663, Apr. 2024, doi: 10.1007/s11042-024-19108-9.
- [24] Y. Hou, Z. Wang, S. Wang, und L. Zheng, „Adaptive Affinity for Associations in Multi-Target Multi-Camera Tracking“, *IEEE Trans. on Image Process.*, Bd. 31, S. 612–622, 2022, doi: 10.1109/TIP.2021.3131936.
- [25] D. Brown, „Close-Range Camera Calibration“, *Photogramm. Eng.*, Bd. 37, Dez. 2002.
- [26] E. Schwalbe, „Geometric modelling and calibration of fisheye lens camera systems“, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Bd. 36, Mai 2012.
- [27] „Traffic Intelligence (Website)“. 30. Juni 2025. Zugriffen: 30. Juni 2025. [Online]. Verfügbar unter: <https://sourceforge.net/projects/traffic-intelligence/>
- [28] M. Bornø Jensen u. a., „A Framework for Automated Traffic Safety Analysis from Video Using Modern Computer Vision“, in *Transportation Research Board Annual Meeting 2019*, Jan. 2019.
- [29] „OpenDataCam (Dokumentation)“. 30. Juni 2025. Zugriffen: 30. Juni 2025. [Online]. Verfügbar unter: <https://opendata.cam/docs/>
- [30] J. Sochor u. a., „Comprehensive Data Set for Automatic Single Camera Visual Speed Measurement“, *IEEE Trans. Intell. Transport. Syst.*, Bd. 20, Nr. 5, S. 1633–1643, Mai 2019, doi: 10.1109/TITS.2018.2825609.
- [31] *Technische Lieferbedingungen für Streckenstationen, Ausgabe 2012*, Bergisch Gladbach., 2013. [Online]. Verfügbar unter: https://www.bast.de/DE/Publikationen/Regelwerke/Verkehrstechnik/Unterseiten/V5-tls-2012.pdf?__blob=publicationFile&v=1
- [32] H.-S. Fang u. a., „AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time“, *IEEE Trans. Pattern Anal. Mach. Intell.*, Bd. 45, Nr. 6, S. 7157–7173, Juni 2023, doi: 10.1109/TPAMI.2022.3222784.
- [33] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, und Y. Sheikh, „OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields“, 30. Mai 2019, *arXiv*: arXiv:1812.08008. doi: 10.48550/arXiv.1812.08008.
- [34] J. Bremer, „Optimierung von Tracking-Algorithmen für die Anwendung im Verkehrsbereich“, Masterarbeit, KIT - Institut für Verkehrswesen, Karlsruhe, 2024.
- [35] C.-Y. Wang, I.-H. Yeh, und H.-Y. M. Liao, „YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information“, 29. Februar 2024, *arXiv*: arXiv:2402.13616. doi: 10.48550/arXiv.2402.13616.