

Robotic Vision Projekt

Lars Brinkmann
Matrikelnummer: 7106769
Fachbereich Informationstechnik
Fachhochschule Dortmund
Dortmund, Deutschland
lars.brinkmann003@stud.fh-dortmund.de

Abstract—Stereovisuelle Pfadrekonstruktion ist ein wesentliches Verfahren zur präzisen und sicheren Navigation autonomer mobiler Roboter in komplexen Umgebungen. Dieser Bericht dokumentiert die Validierung eines stereovisuellen Datensatzes im Rahmen des Robotic-Vision-Projekts. Es werden essentielle Voraussetzungen für eine Pfadrekonstruktion, wie eine präzise Kamerakalibrierung und eine ausreichende Anzahl an Orientierungspunkten, verifiziert. Der Fokus liegt hierbei auf der Ermittlung geeigneter Feature Matches zur 3D-Triangulation von Landmarks. Außerdem wird die Posenbestimmung über Apriltags analysiert. Die Ergebnisse zeigen, dass der Datensatz zur Pfadrekonstruktion geeignet ist, wobei zukünftig eine höhere Auflösung in Betracht gezogen werden kann.

I. EINLEITUNG

Für das Robotic-Vision-Projekt wurde ein dedizierter Datensatz mit einer Stereokamera auf einem mobilen Roboter aufgenommen. Die verwendete Hardware sowie die jeweils zur Ansteuerung verwendete Software sind in Abbildung 1 dargestellt. Für die Pfadrekonstruktion wurde ein Kurs aus Boxen und anderen Objekten aufgebaut. Beim Aufbau des Szenarios wurden Apriltags an einigen Hindernissen angebracht und Wert auf eine große Texturvielfalt der verbauten Elemente gelegt. Zur späteren Verifikation der stereovisuellen Pfadrekonstruktion wurde der Roboter außerdem mit reflektierenden Kugeln ausgestattet, sodass die Position zusätzlich über ein Vicon Motion-Tracking-System aufgezeichnet werden konnte. Der Roboter wurde anschließend mittels eines ESP32 angesteuert und über Tastaturkommandos durch den Kurs geleitet. Die aufgezeichneten Stereobilder wurden in einem ROS-Bag gespeichert und bilden den hier thematisierten Datensatz.

Dieser Bericht befasst sich mit der Validierung des Datensatzes. Es gilt zu prüfen, ob der Datensatz eine präzise Pfadrekonstruktion auf Basis der Stereobilder zulässt. Zuerst werden im Folgenden die Kameraparameter verifiziert. Anschließend folgt die Betrachtung der Apriltags als separate Orientierungspunkte. Es wird gezeigt, dass keine stereovisuelle Triangulation notwendig ist, um die Position der Apriltags zu ermitteln. Danach wird die Keypoint-Detektion und das Feature Matching untersucht. Es wird eine mögliche Definition eines guten Matches formuliert und überprüft, wie viele dieser Matches innerhalb einzelner Frames zu finden sind. Abschließend folgt eine Pfadrekonstruktion anhand der gefundenen Matches sowie ein Vergleich mit den Ground-Truth-Daten des Vicon-Systems.

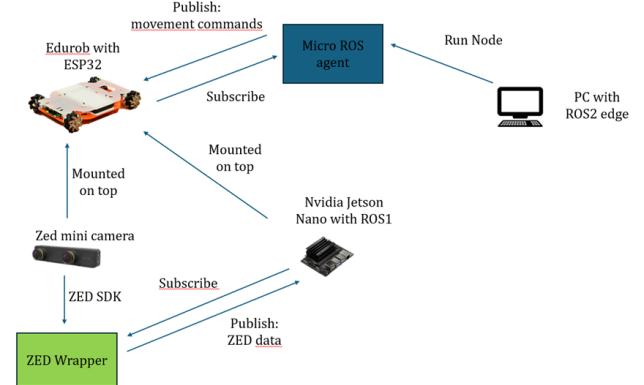


Fig. 1: Versuchsaufbau und Techstack

II. METHODISCHE GRUNDLAGEN

Dieses Kapitel gibt einen groben Umriss über die für diese Arbeit relevanten Methoden der stereovisuellen Odometrie.

A. Stereo-Triangulation

In einem digitalen Bild werden 3D-Weltpunkte \mathbf{X} über eine Projektionsmatrix \mathbf{P} auf 2D-Bildpunkte \mathbf{x} projiziert:

$$\mathbf{x} = \mathbf{P}\mathbf{X}. \quad (1)$$

Hierbei handelt es sich um homogenisierte Koordinaten $\mathbf{X} = [X, Y, Z, 1]^\top$ und $\mathbf{x} = [x, y, w]^\top$. Bei einem Stereo-Setup liegen zwei Bildpunkte \mathbf{x} und \mathbf{x}' vor und es gilt analog:

$$\mathbf{x}' = \mathbf{P}'\mathbf{X}. \quad (2)$$

Im Kern der Stereo-Triangulation steht das Ermitteln eines 3D-Weltpunktes aus zwei (oder mehr) zugehörigen 2D-Bildpunkten. Im Rahmen dieser Arbeit erfolgt die Triangulation mittels des Direct Linear Transformation (DLT) Algorithmus [1].

Über die obigen Gleichungen lässt sich aufgrund des homogenen Skalierungsfaktors w in den Bildpunkten \mathbf{x} und \mathbf{x}' nicht direkt die Position eines Weltpunktes bestimmen. Um die Abhängigkeit von diesem Faktor zu eliminieren, wird das Kreuzprodukt gebildet:

$$\mathbf{x} \times \mathbf{P}\mathbf{X} = 0 \quad ; \quad \mathbf{x}' \times \mathbf{P}'\mathbf{X} = 0. \quad (3)$$

Sei \mathbf{K} die Kameramatrix nach dem Pinhole-Modell:

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (4)$$

und seien \mathbf{R} die Rotationsmatrix und \mathbf{t} der Translationsvektor, welche die Pose der Kamera relativ zur Welt beschreiben:

$$[\mathbf{R} \mid \mathbf{t}] = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}. \quad (5)$$

Dann fasst die Projektionsmatrix \mathbf{P} die intrinsischen und extrinsischen Kameraparameter zusammen:

$$\mathbf{P} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}] = \begin{bmatrix} \mathbf{p}_1^\top \\ \mathbf{p}_2^\top \\ \mathbf{p}_3^\top \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}. \quad (6)$$

Damit gilt:

$$\mathbf{P}\mathbf{X} = \begin{bmatrix} \mathbf{p}_1^\top \mathbf{X} \\ \mathbf{p}_2^\top \mathbf{X} \\ \mathbf{p}_3^\top \mathbf{X} \end{bmatrix}. \quad (7)$$

Für das Kreuzprodukt ergibt sich:

$$\mathbf{x} \times (\mathbf{P}\mathbf{X}) = \begin{bmatrix} y(p_3^\top \mathbf{X}) - w(p_2^\top \mathbf{X}) \\ w(p_1^\top \mathbf{X}) - x(p_3^\top \mathbf{X}) \\ x(p_2^\top \mathbf{X}) - y(p_1^\top \mathbf{X}) \end{bmatrix} = 0. \quad (8)$$

Von diesen drei skalaren Gleichungen sind genau zwei linear unabhängig. Per Konvention werden hier die ersten beiden Gleichungen verwendet:

$$(1) \quad y(p_3^\top \mathbf{X}) - w(p_2^\top \mathbf{X}) = 0, \quad (9)$$

$$(2) \quad w(p_1^\top \mathbf{X}) - x(p_3^\top \mathbf{X}) = 0. \quad (10)$$

Für \mathbf{x}' ergeben sich analog ebenfalls zwei linear unabhängige Gleichungen:

$$(3) \quad y'(p'_3 \mathbf{X}) - w'(p'_2 \mathbf{X}) = 0, \quad (11)$$

$$(4) \quad w'(p'_1 \mathbf{X}) - x'(p'_3 \mathbf{X}) = 0. \quad (12)$$

Fasst man die Koeffizienten der Gleichungen in einer Matrix \mathbf{A} zusammen, folgt:

$$\mathbf{AX} = 0. \quad (13)$$

Anschließend wird die Singulärwertzerlegung (SVD) von \mathbf{A} durchgeführt. Die Lösung für \mathbf{X} ist der rechte Singulärvektor zum kleinsten Singulärwert. Da es sich hierbei noch um homogene Koordinaten handelt, wird \mathbf{X} abschließend durch Normierung der vierten Komponente in inhomogene Weltkoordinaten überführt.

B. Perspective-n-Points (PnP) Problem

Das Perspective-n-Points (PnP) Problem befasst sich mit dem Ermitteln der Kamerapose über eine Menge von 3D-Weltpunkten und deren zugehörigen 2D-Bildpunkten. Gemäß obiger Gleichungen gilt:

$$\mathbf{x} = \mathbf{PX} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]\mathbf{X}. \quad (14)$$

Hier seien \mathbf{x} und \mathbf{X} sowie \mathbf{K} bekannt, und es wird für \mathbf{R} und \mathbf{t} gelöst. Für jedes 3D–2D-Punktpaar gilt:

$$\mathbf{x}_i \approx \mathbf{P}\mathbf{X}_i. \quad (15)$$

In einem realen Szenario wird dies zu einem Optimierungsproblem, da die Projektionen aufgrund von Ungenauigkeiten nicht exakt sind. Im Rahmen dieser Arbeit werden zwei Ansätze zum Lösen dieses Optimierungsproblems verwendet.

1. Levenberg–Marquardt-Optimierung [2] [3]:

Die Levenberg–Marquardt-Optimierung basiert auf dem Least-Squares-Prinzip. Gesucht wird jene Projektionsmatrix θ (bzw. \mathbf{R} und \mathbf{t}), welche die Kostenfunktion minimiert:

$$\min_{\theta} \sum_{i=1}^N \|(\mathbf{x}_i, y_i) - \text{Proj}(\theta, \mathbf{X}_i)\|^2. \quad (16)$$

Die Optimierungsschritte sind eine Kombination aus Gauß-Newton- und Gradientenabstiegsverfahren.

2. RANSAC-Prinzip [4]:

In der Praxis ist nicht ausgeschlossen, dass einige 3D–2D-Zuordnungen inkorrekt sind. Solche Ausreißer (Outlier) können starke Auswirkungen auf klassische Least-Squares-Optimierungsverfahren haben. Das RANSAC (Random Sample Consensus) Prinzip adressiert dieses Problem, indem explizit nach einer Lösung θ gesucht wird, welche möglichst viele konsistente Zuordnungen (Inlier) beinhaltet. Als Inlier gelten hierbei Punkte, deren Reprojektionsfehler unter einem definierten ε liegen:

$$\text{Inlier: } \|(\mathbf{x}_i, y_i) - \text{Proj}(\theta, \mathbf{X}_i)\| < \varepsilon. \quad (17)$$

Hierfür werden Stichproben aus den 3D–2D-Korrespondenzen gezogen, und diejenige Hypothese für θ gewählt, die die maximale Anzahl an Inliern liefert. Die resultierende Lösung sowie die zugehörigen Inlier werden anschließend mittels der Levenberg–Marquardt-Methode weiter verfeinert.

C. Oriented FAST and Rotated BRIEF (ORB)

Für diverse Computer-Vision-Aufgaben ist es nützlich, Keypoints in einem Bild zu identifizieren und zu klassifizieren, insbesondere bei Stereo-Anwendungen, bei denen es keine vorab definierten Orientierungspunkte gibt. Hierfür wurde der 2011 vorgestellte ORB-Algorithmus [5] entwickelt. Dieser beruht auf drei Kernideen.

1. FAST (Features from Accelerated Segment Test) [6] [7]: FAST ist ein Verfahren zum Identifizieren von Ecken in einem Bild. Ecken sind grundsätzlich dadurch definiert, dass sich die Intensität eines Pixels maßgeblich von jener in der direkten Umgebung unterscheidet. Bei FAST wird um jeden Pixel p ein Kreis gelegt. Anschließend werden n Punkte auf diesem Kreis ausgewählt:

$$C = \{p_1, p_2, \dots, p_n\}. \quad (18)$$

Es wird überprüft, ob alle $p_i \in C$ heller oder dunkler sind als p :

$$\text{FAST}(p) = \begin{cases} 1, & \forall p_i \in C : I(p_i) > I(p) + t \text{ oder} \\ & \forall p_i \in C : I(p_i) < I(p) - t, \\ 0, & \text{sonst.} \end{cases} \quad (19)$$

2. Auswahl der besten Keypoints über den Harris-Score [8]:

Um die Güte der über FAST ermittelten Keypoints zu quantifizieren, wird der Harris-Score verwendet. Dieser gibt Auskunft über die Güte eines gefundenen Keypoints. Der Harris-Score für einen Bildpunkt (x, y) ist definiert als:

$$R(x, y) = \det(\mathbf{M}) - \kappa (\text{trace}(\mathbf{M}))^2, \quad (20)$$

wobei

$$\mathbf{M} = \begin{bmatrix} \sum I_x^2 & \sum I_x I_y \\ \sum I_x I_y & \sum I_y^2 \end{bmatrix} \quad (21)$$

die Strukturmatrix basierend auf den Gradienten in der Umgebung eines Pixels darstellt. κ ist hierbei ein empirischer Parameter und I_x, I_y die partiellen Ableitungen der Intensität in x - und y -Richtung. Es werden jene Ecken behalten, die den höchsten Harris-Score aufweisen.

3. BRIEF (Binary Robust Independent Elementary Features) [9]:

Um die gefundenen Keypoints auch in anderen Bildern oder Frames wiederzufinden, müssen diese klassifiziert werden. Bei BRIEF wird hierzu ein binärer Deskriptor verwendet, der durch Vergleichen benachbarter Pixelpositionen um einen Keypoint entsteht. Für einen einfachen BRIEF-Deskriptor mit Länge n werden n zufällige Punktpaare (p_i, q_i) aus der direkten Umgebung (Patch) des Keypoints gewählt. Für jedes Paar wird ein Bit gemäß

$$\text{BRIEF}_i(x) = \begin{cases} 1, & I(x + p_i) < I(x + q_i), \\ 0, & \text{sonst.} \end{cases} \quad (22)$$

gesetzt. Das Resultat ist ein n -dimensionaler Binärvektor. Um den Deskriptor robust gegen Rotationen zu machen, erhält jeder Keypoint eine Orientierung Θ , welche über das Intensity-Centroid-Verfahren berechnet wird [5]. Entsprechend werden die Punktpaare zuvor mit einer Rotationsmatrix $\mathbf{R}(\Theta)$ transformiert:

$$\mathbf{p}'_i = \mathbf{R}(\Theta) \mathbf{p}_i, \quad \mathbf{q}'_i = \mathbf{R}(\Theta) \mathbf{q}_i, \quad (23)$$

mit

$$\mathbf{R}(\Theta) = \begin{bmatrix} \cos \Theta & -\sin \Theta \\ \sin \Theta & \cos \Theta \end{bmatrix}. \quad (24)$$

Damit ergibt sich der Oriented-BRIEF-Deskriptor gemäß:

$$\text{ORB-Deskriptor}_i(x) = \begin{cases} 1, & I(x + p'_i) < I(x + q'_i), \\ 0, & \text{sonst.} \end{cases} \quad (25)$$

Für den Vergleich zwischen zwei Bildern wird ein Brute-Force-Matcher [10] verwendet. Dabei wird für jeden Deskriptor aus dem ersten Bild der Abstand zu allen Deskriptoren des zweiten Bildes über die Hamming-Distanz [11] berechnet.

Diese beschreibt die Anzahl unterschiedlicher Bits zwischen zwei Binärvektoren. Eine geringe Distanz weist auf ein gutes Match hin.

III. VALIDIERUNG DES STEREO SETUPS

Eine präzise 3D-Triangulation setzt ein kalibriertes Stereo Setup voraus. Im Rahmen dieser Arbeit werden die Kameraparameter der ZED SDK [12] verwendet. Basierend auf diesen wird die Kameramatrix \mathbf{K} (Formel 4), sowie die Rotationsmatrix \mathbf{R} und der Translationsvektor \mathbf{t} gemäß Formel 5 gebildet. Zusätzlich wird der Verzerrungsvektor \mathbf{d} gebildet, welcher die radialen und tangentialen Verzerrungen beschreibt.

$$\mathbf{d} = [k_1, k_2, p_1, p_2, k_3]$$

Nun werden die Bildpaare rektifiziert und überprüft, ob die Epipolarlinien horizontal und deckungsgleich verlaufen. Sind die Kameraparameter valide, dann sollte ein Objekt im linken Bild nach dem Rektifizieren dieselbe Höhe im rechten Bild aufweisen. Die Abbildung 2 zeigt das rektifizierte Bildpaar des ersten Frames.

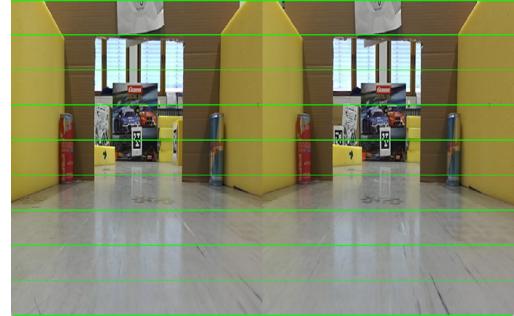


Fig. 2: Rektifiziertes Bildpaar mit Orientierungslinien

Eine detailliertere Einsicht liefert der Vergleich von Keypoint Matches. Hierfür wird im linken Bild via ORB nach Features gesucht und ein Brute-Force-Matcher verwendet, um die selben Features im rechten Bild zu finden. Die Rektifizierung lässt sich dann validieren indem die y -Koordinatendifferenz der Matches betrachtet wird. Für das erste Frame werden im linken Bild 3721 Keypoints detektiert (vgl. 3a). Mit einer y -Koordinatendifferenz ≤ 1 bleiben 2264 Matches über (3b). Dies weist auf eine valide Kalibrierung des Stereo Setups hin.

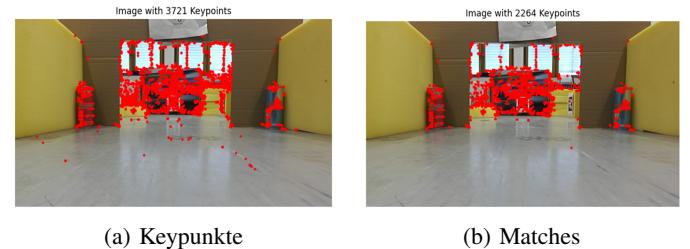


Fig. 3: a) Detektierte Keypunkte im linken Bild und b) übrige Matches nach y -Thresholding

IV. APRILTAGS ALS ORIENTIERUNGSHILFE

Als zusätzliche Orientierungspunkte wurden insgesamt 20 Apriltags im Kurs angebracht. Apriltags lassen sich eindeutig und robust über ihre ID identifizieren. Außerdem ist die Geometrie der Tags (hier $16,2 \times 16,2$ cm) bekannt, weshalb die Pose der Tags präzise berechnet werden kann. Um die Apriltags ideal zu verwenden, sollten in den meisten Frames Tags zu sehen sein. Abbildung 4 zeigt, welche Apriltags in jedem Frame zu erkennen sind. Es ist zu erkennen, dass nahezu kontinuierlich Tags im Bild erscheinen. Die einzige längere Sequenz ohne Detektion erstreckt sich von Frame 76 bis 95, da hier das Apriltag mit der ID 2 nur teilweise sichtbar ist.

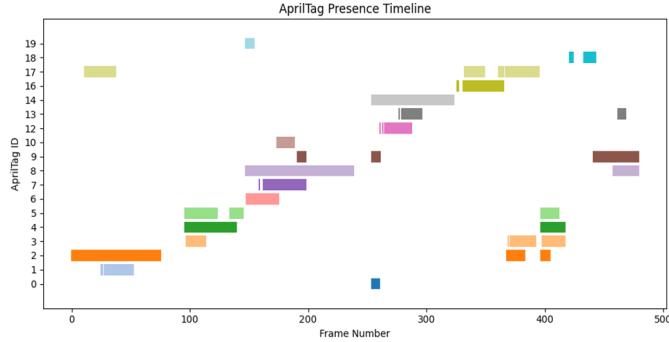


Fig. 4: Sichtbare Apriltags nach Frames

Nach der Detektion kann die Pose der Apriltags als PnP-Problem berechnet werden. Die Vorhersage der Pose der ersten 3 sichtbaren Tags ist in Abbildung 5 dargestellt. Verglichen mit den Vicon-Daten sind die Positions berechnungen bis auf ca. 10cm präzise.

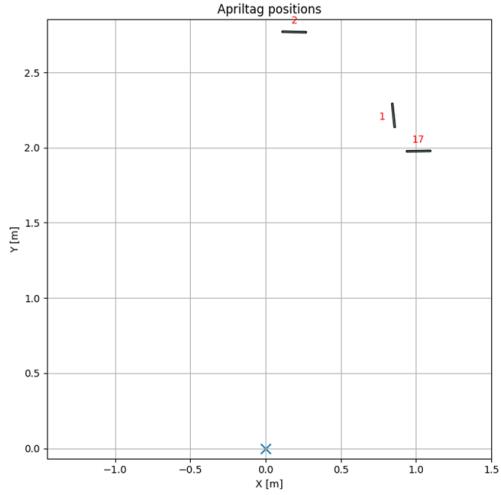


Fig. 5: Posenberechnung der ersten 3 Apriltags

V. QUALITATIVE KEYPOINTS UND MATCHES

Ein essentielles Element der stereovisuellen Pfadrekonstruktion ist das Ermitteln von Keypoints. Es gilt einzigartige und robuste Punkte zu identifizieren, anhand derer die Navigation erfolgt. Ein Lösungsansatz ist Keypoint basiertes SLAM

(Simultaneous Localization and Mapping). Hierbei wird eine globale Karte gepflegt, in der markante Keypoints über ihre triangulierte 3D-Position und den zugehörigen Deskriptor als sogenannte Landmarks gespeichert werden. Um die Karte mit Landmarks zu füllen müssen ausreichend hochwertige Keypoint Matches in jedem Frame verfügbar sein. Damit ein Match als hochwertig deklariert wird, soll für die y -Koordinaten im linken und rechten Bild gelten:

$$|y_l - y_r| < 1. \quad (26)$$

Da es sich um rektifizierte Bilder handelt, ist zu erwarten, dass ein Match die selbe y -Koordinate hat. Abweichungen im Subpixelbereich werden hier toleriert, da diese durch die ORB-Keypointberechnung auftreten können. Außerdem soll das Match einzigartig sein um Uneindeutigkeiten bei der Zuordnung zu vermeiden. Hierfür sei m das beste Match und n das zweitbeste für einen Keypunkt k . Dann soll für die Hamming-Distanz gelten:

$$\text{dist}_{k,m} < 0,7 \times \text{dist}_{k,n} \quad (27)$$

Die Abbildung 6 zeigt die Anzahl gefundener Keypoints pro Frame. Im Durchschnitt werden 2411 Keypoints pro Frame gefunden, wobei eine große Varianz zwischen einzelnen Passagen besteht. Aus den Keypoints können durchschnittlich 846 hochwertige Matches nach obiger Definition ermittelt werden.

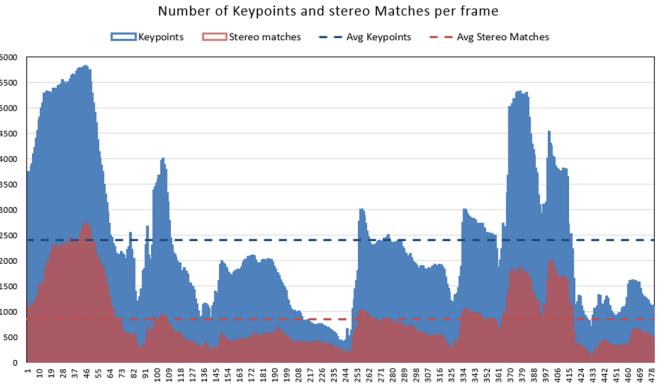


Fig. 6: Keypoints (blau) und gute Matches (rot) per Frame

Frames mit einer hohen Texturvielfalt liefern besonders viele Keypunkte und Matches. In Abbildung 7 sind zwei besonders feature-reiche Frames inklusive der gefundenen Matches dargestellt. Die meisten Detektionen liefert Frame 45 mit 5826 Keypoints und 2718 Matches. Die Matches befinden sich primär an den texturreichen Kartons und den Apriltags. Frame 378 stellt dieselbe Szene aus einer anderen Perspektive dar. Hier werden außerdem Matches im Hintergrund erkannt.

Einige Abschnitte des Kurses bieten schwierige Bedingungen für die Keypoint Erkennung. Zwei Frames mit niedrigen Detektionsraten sind in Abbildung 8 dargestellt. Frame 240 (8a) liefert mit 400 erkannten Keypunkten die niedrigste Detektionsrate innerhalb des Datensatzes. Einerseits ist die Kamera zu nah vor einem Hindernis um Features im Hintergrund zu nutzen und andererseits ist die Texturvielfalt gering.

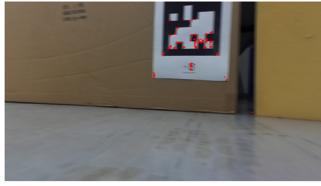


(a) Frame 45



(b) Frame 378

Fig. 7: Frames mit hoher Detektionsrate



(a) Frame 240



(b) Frame 431

Fig. 8: Frames mit niedriger Detektionsrate

Ein ähnliches Bild ergibt sich für Frame 431 (8b). Hier können allerdings Features im Hintergrund genutzt werden. Für beide Bilder liefern die Apriltags einen Großteil der Keypoints.

Um hochwertige Matches in Landmarks für die globale Karte zu konvertieren, werden die 3D-Koordinaten \mathbf{X} zu den bekannten Bildpunkten \mathbf{x}, \mathbf{x}' trianguliert. Anschließend wird überprüft, ob die Z -Koordinate der berechneten 3D-Koordinaten positiv ist. Eine negative Z -Koordinate würde einen Punkt hinter der Kamera beschreiben und ist fehlerhaft. Final wird der Reprojektionsfehler berechnet. Hierfür werden die 3D-Koordinaten in homogener Form \mathbf{X}^h und die bekannte Projektionsmatrix der linken Seite \mathbf{P}_{left} verwendet, um den homogenen Bildpunkt $\tilde{\mathbf{x}}$ (linksseitig) zu berechnen:

$$\tilde{\mathbf{x}} = \mathbf{P}_{\text{left}} \mathbf{X}^h = \begin{bmatrix} \tilde{x} \\ \tilde{y} \\ \tilde{w} \end{bmatrix}. \quad (28)$$

Die Normalisierung über \tilde{w} liefert den projizierten 2D-Bildpunkt \mathbf{x}_{proj} :

$$\mathbf{x}_{\text{proj}} = \pi(\tilde{\mathbf{x}}) = \begin{bmatrix} \tilde{x}/\tilde{w} \\ \tilde{y}/\tilde{w} \end{bmatrix}. \quad (29)$$

Der Reprojektionsfehler e ergibt sich dann über die euklidische Distanz zwischen \mathbf{x}_{proj} und dem bekannten linken Bildpunkt \mathbf{x}_{left} :

$$e = \|\mathbf{x}_{\text{proj}} - \mathbf{x}_{\text{left}}\|_2. \quad (30)$$

Für ein robustes Landmark soll $e < 1$ Pixel gelten. Für jedes Landmark werden die 3D-Koordinaten sowie der zugehörige Deskriptor in der globalen Karte hinterlegt. Abbildung 9 visualisiert die Anzahl neuer Landmarks, sowie die Summe aller Landmarks in der Karte. Insgesamt werden 253.687 einzigerartige Landmarks detektiert. Pro Frame werden im Durchschnitt 529 neue Landmarks in die Karte eingepflegt.

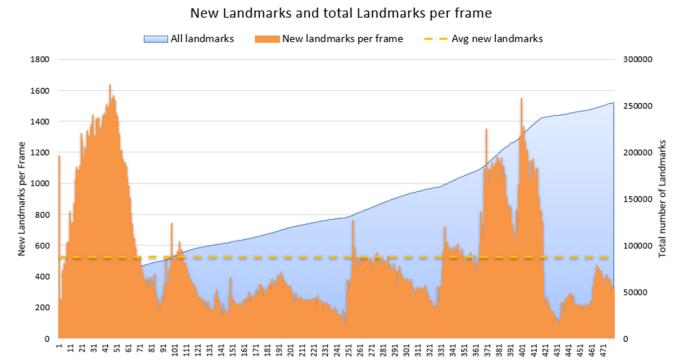


Fig. 9: Anzahl detekтирter neuer Landmarks pro Frame (orange) und Summe bekannter Landmarks (blau)

Zur Orientierung gilt es nun in jedem Frame (außer dem ersten) Matches zwischen aktuell sichtbaren Keypoints und gespeicherten Landmarks zu finden. Es werden Keypoints im linken Bild ermittelt und die Deskriptoren mit der globalen Karte verglichen. Wie zuvor wird ein Distanzcheck gemäß Formel 27 implementiert. Anschließend folgt eine Inlier-Suche mittels PnP-RANSAC über 1000 Iterationsschritte. Die Anzahl an Matches wird in Abbildung 10 in grün dargestellt. Im Durchschnitt werden 406 bekannte Landmarks pro Frame gefunden. Diese Inlier-Matches werden zur Posenbestimmung in einem iterativen PnP-Problem mit Levenberg–Marquardt verwendet.

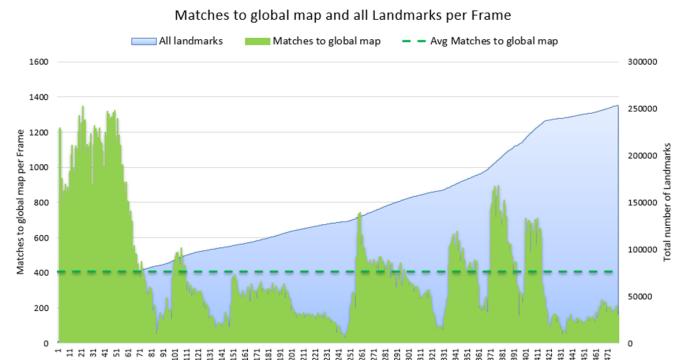


Fig. 10: Landmark zu Keypunkt matches (grün) und Summe bekannter Landmarks (blau)

VI. PFADREKONSTRUKTION UND GROUND TRUTH VERGLEICH

Neben der Stereokamera wurde der Roboter zusätzlich mit 3 reflektierenden Kugeln ausgestattet, um eine Bewegungsverfolgung über das Vicon-System zu ermöglichen. Diese Aufnahme soll als Ground Truth dienen, um die stereovisuelle Vorhersage zu validieren. Der aufgezeichnete Roboterpfad ist in Abbildung 11 dargestellt. Die Sequenz ist über eine Colormap kodiert, wobei der Startpunkt in dunkelblau und der Endpunkt in gelb dargestellt ist. Der Pfad weist zwei kurze Unterbrechungen auf, da es an diesen Stellen zu Verdeckungen der Kugeln gekommen ist.

Top-Down Object Trajectory

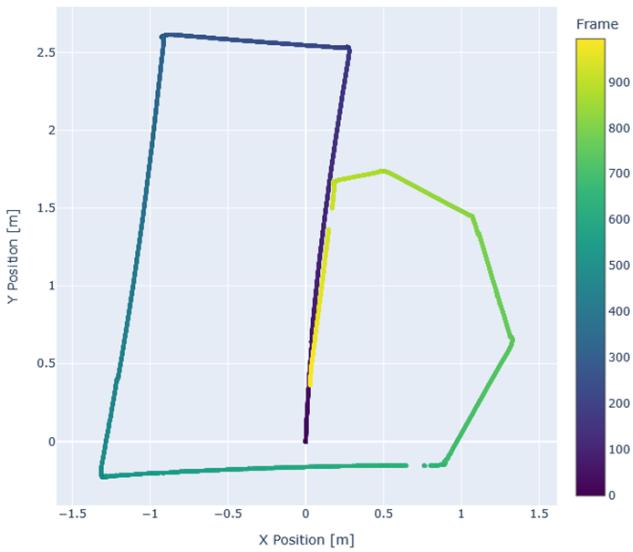


Fig. 11: Roboterpfad gemäß Vicon-Aufnahmen

Die stereovisuelle Pfadrekonstruktion liefert den Pfad in Abbildung 12. Diese Berechnung basiert rein auf den Keypoint Matches ohne die bekannte Geometrie der Apriltags zu nutzen. Es wurden keine der bei SLAM gängigen Optimierungsmethoden verwendet. Es ist zu erkennen, dass der Pfad weitestgehend mit dem aus Abbildung 11 übereinstimmt. Auch die Distanzwerte passen gut zueinander.

Top-Down Object Trajectory

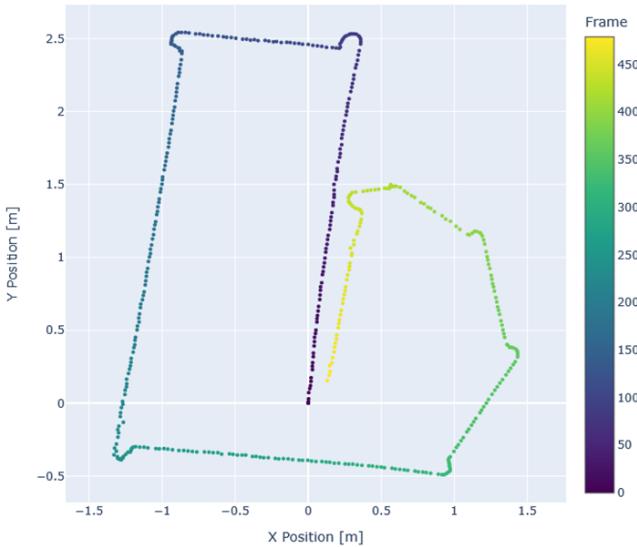


Fig. 12: Stereovisuelle Pfadrekonstruktion

Auffällig sind die Unterschiede in den Kurven. Bei den Vicon-Daten sind die Kurven abrupt während der rekonstruierte Pfad weit geschwungene Kurven zeigt. Dies liegt an dem unterschiedlichen Messpunkt der beiden Ansätze und wird in Abbildung 13 am Beispiel einer Linkskurve verdeutlicht. Das Vicon-System erkennt den Roboter als Dreieck, wobei die Eckpunkte die 3 reflektierenden Kugeln sind. Der Mittelpunkt dieses Dreiecks liefert die aktuelle Position des Roboters. Damit trackt das Vicon-System annähernd den Mittelpunkt des Roboters. Die ZED-Kamera sitzt hingegen an der Front des Roboters. Der verwendete EduRob Roboter rotiert bei einer Kurve um den eigenen Mittelpunkt, indem der linke und rechte Antrieb entgegengesetzt betrieben werden. Dadurch wirken Kurven in der Vicon-Aufnahme eckig und sofortig, während die Kameraposition aufgrund des Aufbaus an der Front einen größeren Drehwinkel durchläuft.

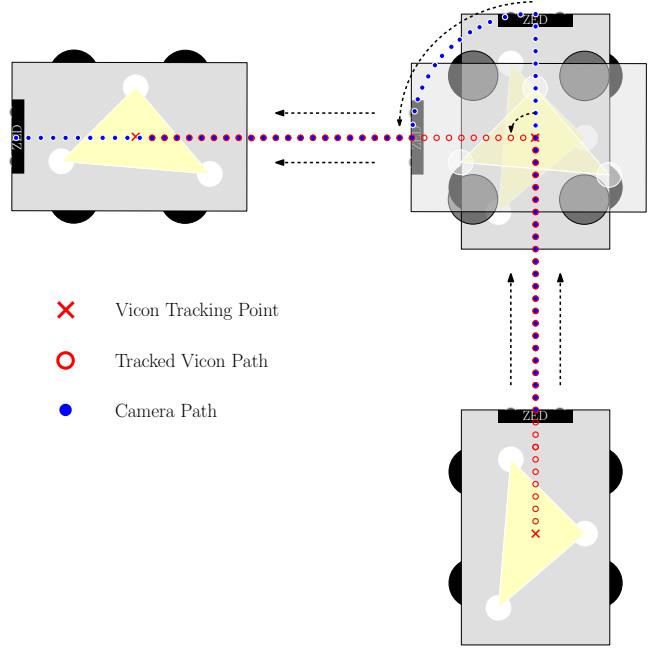


Fig. 13: Unterschiede im Pfad zwischen Vicon und Kamera am Beispiel einer Linkskurve

Außerdem zeigt sich eine Abweichung in der Kurve ab Frame 250. Hier entsteht ein zu großer Winkel weshalb die zweite Hälfte des Pfades ein leicht verzogen ist. Dieser Fehler ließe sich allerdings über Pose Graph Optimierung mit dem bekannten Loop-Closure korrigieren.

VII. FAZIT UND VERBESSERUNGSMÖGLICHKEITEN

In dieser Arbeit wurde gezeigt, dass der aufgenommene Datensatz für eine stereovisuelle Pfadrekonstruktion zu Lehrzwecken geeignet ist. Die Kamerakalibrierung erwies sich als hinreichend genau, um nach der Rektifizierung Epipolarlinien auf derselben Höhe zu erhalten. Jedes Frame verfügte über genügend qualitativ hochwertige Keypoint Matches, um präzise Odometriemessungen durchzuführen. Hier zahlte

sich insbesondere die Auswahl texturreicher Hindernisse aus. Zusätzlich enthielten die meisten Frames sichtbare Apriltags, die als Orientierungspunkte genutzt werden können. Die abschließende Pfadrekonstruktion lieferte bereits ohne gängige Optimierungsmethoden eine gute Näherung des tatsächlichen Roboterpfades.

Im Verlauf der Arbeit traten mehrere Verbesserungspotenziale zutage. Die Verwendung eines ROS1-ZED-Wrappers führte dazu, dass die Videoauflösung von 720p auf 360p reduziert wurde, da der Wrapper standardmäßig die Auflösung der publizierten Bilder halbiert. In künftigen Projekten sollte diese Einstellung deaktiviert werden. Eine höhere Videoauflösung bietet mehr Bilddetails und damit zusätzliche potenzielle Keypoints. Zudem verbessert sich die Lokalisierung der Keypoints, wodurch Subpixelfehler reduziert werden. Trotz der höheren Rechenanforderungen ist daher eine gesteigerte Auflösung für Projekte dieser Art empfehlenswert.

REFERENCES

- [1] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.
- [2] K. Levenberg, “A method for the solution of certain problems in least squares,” *Quarterly of Applied Mathematics*, vol. 2, pp. 164–168, 1944.
- [3] D. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters,” *SIAM Journal on Applied Mathematics*, vol. 11, pp. 431–441, 1963.
- [4] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [5] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “Orb: an efficient alternative to sift or surf,” 11 2011, pp. 2564–2571.
- [6] E. Rosten and T. Drummond, “Machine learning for high-speed corner detection,” vol. 3951, 07 2006.
- [7] E. Rosten, R. Porter, and T. Drummond, “Faster and better: A machine learning approach to corner detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105–119, 2010.
- [8] C. Harris and M. Stephens, “A combined corner and edge detector,” in *Proceedings of the Alvey Vision Conference*, vol. 15, 1988, pp. 147–151.
- [9] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features,” in *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2010, pp. 778–792.
- [10] OpenCV Contributors, “Feature matching — opencv documentation,” 2024, accessed: 2025-02-24. [Online]. Available: https://docs.opencv.org/4.x/dc/dc3/tutorial_py_matcher.html
- [11] R. W. Hamming, “Error-detecting and error-correcting codes,” *Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.
- [12] Stereolabs, “ZED SDK – Spatial Perception Framework,” <https://www.stereolabs.com/developers>, accessed: 2025-08-09.