

Institut für
Photogrammetrie und Geoinformation

Automatische Bestimmung der Pose einer Kugel zu deren Nutzung als Eingabegerät

(Automatic determination of the pose of a sphere
for using the sphere as an input device)

Masterarbeit

Studiengang: Navigation und Umweltrobotik

Bearbeiter: Dennis Wittich (Matrikelnr. 2793280)
denniswittich@hotmail.de

Erstprüfer: apl. Prof. Dr. techn. Franz Rottensteiner
(Institut für Photogrammetrie und GeoInformation)

Zweitprüfer: Dr.-Ing. Manfred Wiggenhagen
(Institut für Photogrammetrie und GeoInformation)

Ausgabedatum: 10.04.2017

Abgabedatum: 21.09.2017

I Kurzfassung

In der vorliegenden Masterarbeit wird ein Verfahren vorgestellt, mit dem die Rotation einer Kugel gegenüber einem Rahmen, in welchem sie frei drehbar gelagert ist, automatisch durch den Einsatz einer Kamera bestimmt werden kann. Der Hintergrund dieses Verfahrens ist die Nutzung dieser Kugel als Eingabegerät für einen Computer an Stelle einer Maus. Die Definition des Objektkoordinatensystems erfolgt mit Hilfe von am Rahmen angebrachten Passpunkten, die mit Hilfe von farbigen, konzentrischen Kreisen markiert sind. Auf der Kugel werden kreisförmige, schwarze Marker angebracht, deren Bewegung durch das Verfahren erfasst wird. Das Verfahren benötigt zur Initialisierung die Kalibrierung der Kamera sowie eine Systemkalibrierung, in welcher der Mittelpunkt und Radius der Kugel und die Koordinaten der Passpunkte auf dem Rahmen in einem gemeinsamen Koordinatensystem durch Ausgleichung ermittelt werden. Es werden auch Methoden entwickelt, mit denen die Zentren beider Arten von markierten Punkten in den Bildern detektiert werden können, sodass die Messung sowohl bei der Initialisierung als auch bei der eigentlichen Nutzung automatisch erfolgt. Bei der eigentlichen Nutzung der Kugel als Eingabegerät wird diese von der gegenüber dem Objektkoordinatensystem fix montierten Kamera erfasst. Die ersten Bilder der dabei aufgenommenen Sequenz werden dazu genutzt, um die Orientierung der Kamera in Bezug auf das Objektkoordinatensystem aus den Passpunkten durch räumlichen Rückwärtsschnitt zu ermitteln. Anschließend werden in jedem neu erfassten Bild der Sequenz die markierten Punkte auf der Kugel detektiert und deren Objektkoordinaten durch Schnitt der entsprechenden Bildstrahlen mit der Kugel berechnet. Diese gehen als Beobachtungen in einen Extended Kalman Filter ein, mit dessen Hilfe die Verdrehung der Kugel in Echtzeit bestimmt wird. Die Zuordnung von neu detektierten Punkten zu solchen, die bereits in früheren Bildern detektiert worden sind, erfolgt anhand eines einfachen geometrischen Kriteriums. Das physikalische Bewegungsmodell der Kugel beruht auf der Annahme einer konstanten Drehgeschwindigkeit. Für den Filter werden zwei Modelle entwickelt, die sich im Aufbau des Zustandsvektors unterscheiden. Aus den durch den Filter geschätzten Lagewinkeln und Drehraten der Kugel wird ein 2D Bewegungsvektor abgeleitet, der die Bewegung der Maus zur Steuerung von Computerprogrammen ersetzen kann. Das Verfahren wird in umfangreichen Experimenten auf Grundlage eines realen Datensatzes und mehrerer synthetischer Videosequenzen getestet. Dabei werden die Filtermodelle verglichen und der Einfluss der Belichtung, des Bildrauschens, der Kamerakonfiguration und der Anzahl der Marker auf der Kugel auf die Qualität der Ergebnisse untersucht. Die Ergebnisse zeigen, dass mit dem entwickelten Verfahren die Pose der Kugel in Echtzeit mit ausreichender Genauigkeit für ihren Einsatz als Eingabegerät bestimmt werden kann.

Abstract

This M.Sc. thesis presents an approach to automatically determine the rotation of a sphere with respect to a frame by using a camera. This work is motivated by the idea to use the sphere as an input device for a computer instead of a mouse. The object coordinate system is defined by control points which are marked by targets consisting of coloured concentric circles. Circular black targets are attached to the sphere. The movement of these targets is to be determined automatically. The initialization of the procedure includes the calibration of the camera and the calibration of the entire system to determine the coordinates of the control points as well as the centre and the radius of the sphere in a common coordinate system. The thesis introduces methods to determine the centre of each visible target in the image, so that all image-based measurements are made automatically. When using the sphere as an input device, the camera is fixed with respect to the frame. The exterior orientation of the camera is determined based on the image and object coordinates of the control points, which are detected in the first few frames, by spatial resection. Afterwards, for each new image delivered by the camera, the targets on the sphere are detected and the object coordinates of the target points are computed by intersecting the corresponding image rays with the sphere. These object coordinates are used as observations in an extended Kalman filter. This filter is used to determine the rotation of the sphere with respect to the fixed object coordinate system in real time. The association of the detected points and those already detected in earlier frames is achieved using a simple geometrical criterion. The physical model required for the prediction step in the filter is based on the assumption of constant angular velocity. Two filter models are developed which differ in the structure of the state vector. The calculated position angles and angular rates of the sphere are used to compute a 2D movement vector that can be used instead of a mouse to control computer programs. The procedure is evaluated in multiple experiments based on a real data set and several synthetic video sequences. In the experiments, the filter models are compared and the influence of the scene illumination, the image noise, the camera configuration and the number of targets on the sphere on the results of the procedure are investigated. The experiments show that the procedure is able to determine the pose of the sphere in real time with a sufficient accuracy to use the sphere as an input device.

II Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die Arbeit selbständig verfasst habe und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht sind und die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen hat.

Hannover, den 21.09.2017

(Dennis Wittich)

Inhaltsverzeichnis

I	Kurzfassung	I
II	Selbstständigkeitserklärung	III
III	Abbildungsverzeichnis	VI
IV	Tabellenverzeichnis	VIII
1	Einleitung	1
1.1	Stand der Technik	3
1.1.1	Marker und Markerdetektion	3
1.1.2	3D-Rekonstruktion	4
1.1.3	Tracking	4
1.1.4	Diskussion	5
2	Grundlagen	6
2.1	Bildverarbeitung	6
2.1.1	Farbräume	6
2.1.2	Bildrauschen und Reduktion des Rauschens	8
2.1.3	Binäre morphologische Operationen	9
2.2	Rotationsmatrizen	12
2.3	Parameterschätzung	13
2.3.1	Gauß-Helmert-Modell	17
2.3.2	Extended Kalman Filter	19
2.4	Zentralprojektion	21
2.4.1	Geometrie und Kollinearitätsgleichungen	22
2.4.2	Verzeichnung	23
2.4.3	Räumlicher Rückwärtsschnitt	26
2.4.4	Bündelblockausgleichung	27
2.4.5	Kamerakalibrierung	29
3	Lösungsansatz	31
3.1	Geometrie des Eingabegerätes	31
3.2	Ablauf	32
3.3	Kalibrierung	33
3.3.1	Korrespondenzen der detektierten Marker	35
3.4	Initialisierung	36

3.5	Nutzung	37
3.6	Marker und deren Detektion	38
3.6.1	Marker auf der Kugel	39
3.6.2	Marker auf dem Rahmen	42
3.7	Bestimmen der Kugelparameter	45
3.8	Schnitt eines Bildstrahles mit der Kugel	46
3.8.1	Berechnung des Schnittpunktes	47
3.8.2	Fehlerfortpflanzung	48
3.9	Extended Kalman Filter	50
3.9.1	Modell 1: Punkte als Konstante gespeichert	50
3.9.2	Assoziation der Beobachtungen	54
3.9.3	Modell 2: Punkte im Zustandsvektors gespeichert	55
3.9.4	Umrechnung in Mausbewegung	59
4	Experimente	60
4.1	Datensätze	60
4.1.1	Reale Aufnahme	60
4.1.2	Synthetische Datensätze	64
4.2	Wahl der Programmparameter	67
4.3	Kalibrierung des untersuchten Eingabegerätes	68
4.4	Innere Genauigkeit der Blob-Marker	73
4.5	Parameteranalyse	75
4.6	Vergleich der Filtermodelle	77
4.7	Einfluss der Markeranzahl	80
4.8	Einfluss der Kamerakonfiguration	81
4.9	Einfluss des Bildrauschens und der Belichtung	84
4.10	Diskussion	87
5	Zusammenfassung und Ausblick	88
6	Quellenverzeichnis	90
	Anhang	IX

III Abbildungsverzeichnis

1	Eingabegerät	1
2	Darstellung der Farbräume eines Bildes	7
3	Beispiel der Anwendung eines Medianfilters	8
4	Ergebnis der Filterung durch Median- und Gaußfilter versch. Größe	9
5	Mögliche Strukturelemente für morphologische Operationen	10
6	Beispiel für Dilatation	10
7	Beispiel für Erosion	11
8	Beispiel für Opening (Entfernen kleiner Bereiche)	11
9	Beispiel für Opening (Entfernen eines Stegs)	12
10	Geometrie der Zentralprojektion	22
11	Bild vor (links) und nach (rechts) Korrektur der Verzerrung	26
12	Beispielhafte Geometrie einer Bündelblockausgleichung	28
13	Aufbau des Eingabegerätes und Koordinatensysteme	32
14	Ablauf der Kamerakalibrierung	33
15	Ablauf der Systemkalibrierung	34
16	Ermitteln der Kugelparameter und Passpunktkoordinaten im \mathbf{KS}_0	35
17	Ablauf der Initialisierungsphase	37
18	Ablauf der Nutzungsphase	38
19	Marker auf dem Rahmen und der Vorderseite der Kugel	38
20	Bereiche der Masken für die Detektion von schwarzen Blobs	40
21	Ablauf der Detektion von Blobs und der Bestimmung ihrer Zentren anhand mehrerer Masken und Restriktionen	42
22	Marker auf dem Rahmen für den RRWS	43
23	Ablauf der Detektion von farbkodierten Markern und der Bestimmung der Markerzentren	44
24	Projektion eines detektierten Markers auf die Kugel	47
25	Assoziationen zwischen Beobachtungen und gespeicherten Punkten über mehrere Epochen und Fälle der Datenassoziation	55
26	Eingabegerät mit Markern	61
27	Bild der laufenden Auswertung des Datensatzes R_1	62
28	Darstellung der zeitlichen Verschiebung und der Skalierung zwischen Refe- renz und berechneten Ergebnissen	63
29	Bild der laufenden Auswertung des synthetischen Datensatzes S_2	65
30	Vorgegebene Lagewinkel der synthetischen Datensätze	66
31	Kamerapositionen und Passpunkte (Kamerakalibrierung)	68
32	Aufnahmen des Eingabegerätes für die 3D-Rekonstruktion	70
33	Positionen der Kamera und der Markerzentren	71
34	Detektierte Marker auf der Kugel	74
35	Vergleich der Mausgeschwindigkeiten beider Filtermodelle für den Daten- satz R_1	78
36	Vergleich der Mausgeschwindigkeiten und Geschwindigkeitsdifferenzen bei- der Filtermodelle für den Datensatz S_1 (X-Achse)	79
37	Vergleich der Mausgeschwindigkeiten und Geschwindigkeitsdifferenzen bei- der Filtermodelle für den Datensatz S_1 (Y-Achse)	80

38	Bilder der laufenden Auswertung mit unterschiedlichen Kamerakonfigurationen. Konfiguration nah (links) und fern (rechts)	81
39	Sichtbare Fläche der Kugel in Abh. der Entfernung des Projektionszentrums zum Kugelmittelpunkt	83
40	Bild der Auswertung des verrauschten und überbelichteten Datensatzes S_1	85
41	Lagewinkel und Differenzen zu Referenzwerten für verrauschten und überbelichteten Datensatz	86
42	Kamerapositionen und Passpunkte bei der Kamerakalibrierung und Darstellung der Aufnahmen (Auflösung der Bilder: 640×480 Pixel)	IX

IV Tabellenverzeichnis

1	Distanzen zwischen Markern (Realer Versuch)	61
2	Vorgegebene innere und äußere Orientierung der Kamera im Modell	64
3	Eigenschaften der synthetischen Datensätze	65
4	Wahl der Programmparameter	67
5	Parameter der inneren Orientierung und der Verzeichnung	69
6	Mittlere Rückprojektionsfehler der Passpunkte nach der Kamerakalibrierung	69
7	Objektkoordinaten der Passpunkte für die Bestimmung der Punktnummern	69
8	Objektkoordinaten der Passpunkte für die BBA	71
9	RMSS der Objektkoordinaten der Markerzentren (BBA)	72
10	RMSS der Objektkoordinaten der Markerzentren (BBA)	72
11	Korrelationen, Standardabweichungen und Zahlenwerte der ermittelten Kugelparameter	72
12	Korrelationen und Standardabweichungen der äußeren Orientierung	73
13	Korrelationen und Standardabweichungen der Objektkoordinaten der Markerzentren	74
14	Korrelationskoef. für verschiedene Parameterkombinationen (Datensatz R_1)	75
15	Korrelationskoef. für verschiedene Parameterkombinationen (Datensatz S_1)	75
16	Kombinierter RMSE der Lagewinkel (Datensatz S_1)	76
17	Kombinierter RMSE der Drehraten (Datensatz S_1)	76
18	Kombinierter RMSE der Mausgeschwindigkeiten (Datensatz S_1)	76
19	Vergleich der Filtermodelle	77
20	Ergebnisse für unterschiedliche Anzahl von Markern	81
21	Kor. und Std.-Abw. der äußeren Orientierung für Datensatz S_2 (fern)	82
22	Kor. und Std.-Abw. der äußeren Orientierung für Datensatz S_4 (nah)	82
23	Ergebnisse für unterschiedliche Kamerakonfigurationen	83
24	Ergebnisse für verschieden starkes Bildrauschen und Helligkeiten	85

1 Einleitung

In dieser Arbeit wird ein bereits existierendes Eingabegerät untersucht, das speziell für ein Computerspiel entwickelt wurde. In diesem rollt die Spielfigur eine Kugel auf den Gipfel eines Berges. Gesteuert wird das Spiel, indem der Benutzer eine in einem Rahmen drehbar auf Kugellagern gelagerte Kugel mit einem Durchmesser von ca. 120 cm auf der Stelle dreht. Die Drehbewegung der realen Kugel bewegt einen Trackball auf der Unterseite der Kugel, welcher die Drehbewegung auf die Bewegung der virtuellen Kugel im Spiel überträgt. Mit dem entwickelten Programm soll die Aufgabe des Trackballs durch eine Kamera erfüllt werden. Während der derzeitige genutzte Trackball lediglich Drehbewegungen um zwei Achsen aufnehmen kann, soll es durch den Einsatz der Kamera möglich sein, Drehbewegungen um alle drei Achsen sowie die Lagewinkel der Kugel zu bestimmen. Diese zusätzlichen Informationen können von der Zielsoftware genutzt werden, um eine komplexere und realistischere Steuerung zu ermöglichen. Abbildung 1 zeigt das beispielhaft untersuchte Eingabegerät.

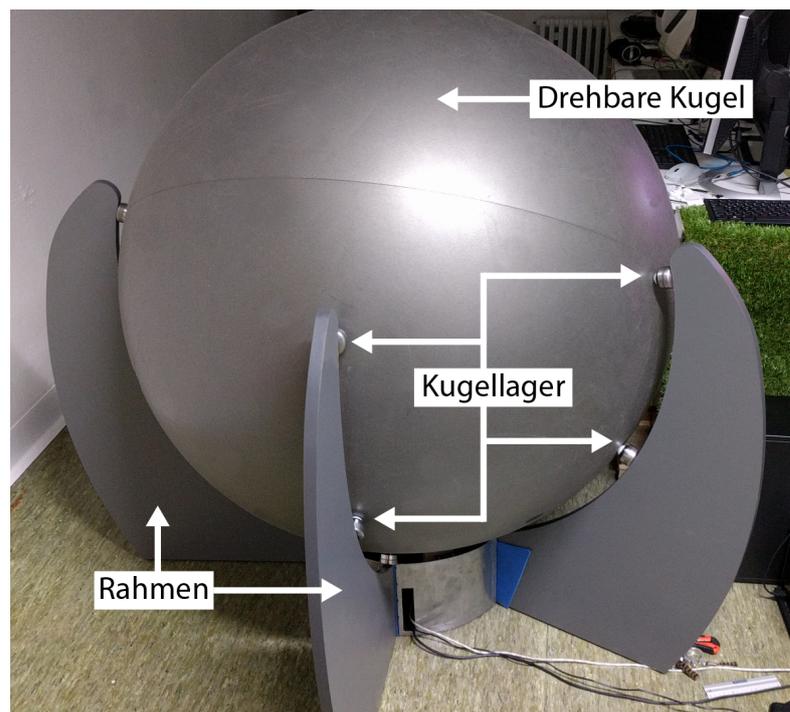


Abbildung 1: Eingabegerät

Das aus der Zielstellung folgende Problem besteht in der fortlaufenden Bestimmung der Verdrehung der Kugel gegenüber dem Rahmen mit Hilfe einer Kamera, deren Pose relativ zum Rahmen während der Nutzung unverändert, jedoch im Vorhinein unbekannt ist. Die Pose der Kamera relativ zum Rahmen zu bestimmen ist folglich Bestandteil der Problematik.

Geometrisch betrachtet ist es das Ziel, die Verdrehung eines an die Kugel gebundenen Koordinatensystems gegenüber einem zweiten, übergeordneten Koordinatensystem, welches an den Rahmen gebunden ist, zu bestimmen. Die Definition jedes Koordinatensystems (Position des Ursprungs und Ausrichtung der Achsen) erfordert Objektpunkte, deren Objektkoordinaten im jeweiligen Koordinatensystem bekannt sind oder berechnet werden können. Wird eine Kamera eingesetzt, um die Position und Orientierung der Koordinatensysteme automatisch zu identifizieren, so müssen diese Objektpunkte im Bild automatisch detektierbar sein. Da weder der Rahmen, noch die Kugel markante Punkte aufweisen, die eine automatische Detektion solcher Punkte ermöglichen, werden künstliche Marker auf dem Rahmen und auf der Kugel angebracht. Durch den Einsatz der Marker ist es möglich, die Pose der Kamera, sowie die Parameter der Kugel in einem übergeordneten Koordinatensystem, das an den Rahmen gebunden ist, zu bestimmen. Mit diesen Informationen werden die Objektkoordinaten der Marker auf der Kugel berechnet. Diese Objektkoordinaten sind an das Koordinatensystem der Kugel gebunden. Die Bewegungen der Marker auf der Kugel bezogen auf das übergeordnete Koordinatensystem werden genutzt, um die Verdrehung der Kugel gegenüber dem Rahmen zu bestimmen.

Methodisch lässt sich die Aufgabe in drei Teilprobleme aufteilen. Das erste Teilproblem ist die Kalibrierung der Kamera sowie die Systemkalibrierung des Eingabegerätes, wobei die Kalibrierung der Kamera zuerst erfolgt. Die Systemkalibrierung besteht aus der automatischen 3D-Rekonstruktion der Kugel und des Rahmens, um die Position und den Radius der Kugel sowie die Objektkoordinaten der Marker auf dem Rahmen in einem gemeinsamen Koordinatensystem zu bestimmen. Zu diesem Zweck werden Marker auf der Kugel und dem Rahmen angebracht, mehrere Bilder der Konstruktion aufgenommen, in denen die Zentren der sichtbaren Marker automatisch bestimmt werden. Die Objektkoordinaten der Marker auf dem Rahmen und der sichtbaren Marker auf der Kugel werden in einer Bündelblockausgleichung (vgl. [Kraus, 2004]) bestimmt. Anschließend wird der Kugelradius und der Mittelpunkt der Kugel durch Ausgleichung ermittelt. Es werden kreisförmige Marker auf der Kugel eingesetzt. Der Rahmen wird mit speziell für diese Anwendung entwickelten Markern, die aus farbigen Ringen bestehen, markiert. Die Detektion aller sichtbaren Marker im Bild erfolgt durch eine Maskierung des Bildes, gefolgt von einer Konturensuche. Die Farbkodierung der Marker auf dem Rahmen dient der Identifikation der selben Marker in mehreren Aufnahmen. Die Detektion der Marker und das Bestimmen der Markerzentren ist das zweite Teilproblem. Das letzte Teilproblem liegt in der Bestimmung der Verdrehung der Kugel gegenüber dem Rahmen. Dies wird durch den Einsatz eines Extended Kalman Filters (vgl. [Thrun et al., 2006]) gelöst, in welchen die Objektkoordinaten der Marker auf der Kugel im übergeordneten Koordinatensystem als Beobachtungen eingehen. Diese Objektkoordinaten werden jeweils durch einen Schnitt der

Bildstrahlen der detektierten Markerzentren auf der Kugel und der modellierten Kugel berechnet.

Da es möglich sein soll, die Zielsoftware in Echtzeit zu bedienen, werden ausschließlich echtzeitfähige Verfahren eingesetzt. Weiterhin werden alle relevanten Berechnungen stochastisch durchgeführt.

1.1 Stand der Technik

Im folgenden wird der Stand der Technik im Hinblick auf die in dieser Arbeit auftretenden Teilprobleme dargelegt und die existierenden Methoden im Bezug auf die vorliegende Arbeit bewertet.

1.1.1 Marker und Markerdetektion

Künstliche, visuelle Marker sind künstlich erzeugte Muster, die so ausgelegt sind, dass sie automatisch und robust in einem Bild detektiert werden können. Darüber hinaus sind viele solcher Markertypen durch ihr Muster kodiert, was eine eindeutige Identifikation ermöglicht. Bekannte industrielle Anwendungen sind die **Quick Response Codes**. Mark Fiala entwickelt in [Fiala, 2005] das sog. **ARTag**, das eine digitale Kodierung verwendet. Die Kodierung erfolgt hierbei durch die Anordnung von schwarzen und weißen Quadraten in einem 6×6 Muster innerhalb eines schwarzen oder weißen Rahmens. Hieraus ergeben sich 2002 mögliche Kodierungen.

In vielen Anwendungen werden auch künstliche Marker eingesetzt, die nicht kodiert sind. Hierunter fallen die einfarbigen kreisförmigen Marker. Ein Verfahren zur Detektion solcher Marker wird von Garcia et. al. in [Garcia et al., 2011] vorgestellt. In dem entwickelten Verfahren wird zunächst ein binäres Schwellwertbild erstellt, in welchem eine Konturensuche mit dem von Chang et. al. in [Chang et al., 2004] entwickelten Verfahren durchgeführt wird. Die Näherungen der gesuchten Markerzentren entsprechen den Schwerpunkten der von den detektierten Konturen umschlossenen Segmente. Es folgt eine Suche der maximalen Gradienten auf Linien im originalen Bild, die von den genäherten Bildkoordinaten der Markerzentren in verschiedene Richtungen ausgehen. Hierdurch können die Randpunkte der Segmente mit Subpixelauflösung bestimmt werden. Für jeden Marker folgt ein **Least-Squares-Fitting** einer Ellipse, um das Verfahren robust gegen mögliche Verdeckungen zu gestalten. Das **Fitting** basiert auf den Methoden aus [Gander et al., 1994] und [Halir und Flusser, 1998]. Die resultierende Ellipse beschreibt den detektierten Marker.

1.1.2 3D-Rekonstruktion

In dem hier entwickelten Ansatz wird die 3D-Rekonstruktion genutzt, um die Objektkoordinaten der Marker und die Kugelparameter in einem gemeinsamen Koordinatensystem zu bestimmen. Die möglichst genaue Rekonstruktion einer Szene anhand mehrerer Aufnahmen ist Aufgabenfeld der Photogrammetrie und der **Computer Vision**, wobei verschiedene Ansätze existieren, um diese Problematik zu lösen. Die genaueste Lösung liefert die Bündelblockausgleichung (vgl. [Szeliski, 2011]). Die Bündelblockausgleichung ist ein Verfahren zur simultanen Bestimmung der Orientierung aller Bilder eines Bockverbandes und der Objektkoordinaten aller unbekanntenen Objektpunkte in einem Guss. Das Verfahren beruht auf der Methode der kleinsten Quadrate. Eine alternative Möglichkeit zur Rekonstruktion einer Szene liefern Ansätze der visuellen Odometrie. Forster et. al. stellen beispielsweise in [Forster et al., 2014] ein Verfahren vor, mit dem die Pose einer Kamera anhand der aufgenommenen Videosequenz berechnet wird. Das Verfahren basiert auf dem Tracking natürlicher Features, deren Objektkoordinaten berechnet werden können, da sich die Kamera bewegt und Objektpunkte aus mehreren Perspektiven betrachtet werden. Auch wenn das Verfahren in erster Linie entwickelt wurde, um die Bewegung der Kamera zu verfolgen, liefert es ein 3D-Modell der von der Kamera aufgenommenen Szene.

1.1.3 Tracking

Auch das letzte Teilproblem, das Tracking eines Objektes bzw. das Verfolgen seines Zustandes über die Zeit mit Hilfe einer Kamera ist Teil aktueller Forschungsarbeiten. Ein spezifischerer Ansatz wird von Hajimolahoseini et al. in [Hajimolahoseini et al., 2014] präsentiert. Ein Extended Kalman Filter [Thrun et al., 2006] wird dort eingesetzt, um die sechs Freiheitsgrade (Rotation und Translation) eines Objektes fortlaufend zu bestimmen. Hierzu werden in jedem Bild der Videosequenz vier natürliche Features detektiert, deren Koordinaten im Objektraum relativ zum Objekt, welches getrackt werden soll, konstant und bekannt sind. Ein alternativer Ansatz wird in [Azad et al., 2011] vorgestellt. Das Verfahren basiert auf einem 3D-Modell des zu trackenden Objektes. Das Modell beinhaltet Informationen über die Form und Textur des Objektes. Mit diesem Modell können Bilder mit verschiedenen Posen des Objektes gerendert werden. Die angenommene Pose des Objektes ergibt sich aus dem gerenderten Bild, dass die größte Übereinstimmung mit dem aufgenommenen Bild erzielt. Hierbei werden verschiedene Features wie Linien oder Eckpunkte abgeglichen. Die Auswertung erfolgt durch einen Partikelfilter [Thrun et al., 2006].

1.1.4 Diskussion

Die Literaturrecherche zeigt, dass folgende Probleme ungelöst sind: Zwar gibt es bereits Verfahren zur Detektion von kreisförmigen Markern, jedoch sind diese für die hier untersuchte Anwendung unzureichend. Das von Garcia et. al. in [Garcia et al., 2011] entwickelte Verfahren verwendet nur ein Schwellwertbild zur Konturensuche. In dem hier vorgestellten Verfahren werden mehrere Masken gebildet, mit jeweils unterschiedlichen Grenzen für die Grauwerte der Pixel. Dies ist notwendig, da die Grauwerte der abgebildeten Marker auf der Kugel sich teilweise nur minimal von dem umliegenden Hintergrund unterscheiden und die Helligkeit der Marker und des Hintergrundes stark variiert. Gleichzeitig wird angenommen, dass eine Detektion der Randpunkte der Marker im Subpixelbereich nicht notwendig ist, um das Ziel dieser Arbeit zu erreichen. Es existieren auch Verfahren zur Detektion und Identifikation von digital kodierten Markern [Fiala, 2005] jedoch sind die dort eingesetzten Marker dafür ausgelegt, mehrere Tausend Marker zu unterscheiden. Für die hier untersuchte Anwendung ist der Einsatz von simplen, farbkodierten Markern zweckmäßiger.

Die vorgestellten Verfahren zur 3D-Rekonstruktion einer Szene beschränken sich auf das Bestimmen der Koordinaten der Objektpunkte und der Pose der Kamera. Für die hier entwickelte Anwendung ist es zusätzlich notwendig, die Kugel zu parametrisieren, was Bestandteil der entwickelten Systemkalibrierung ist.

Auch existieren bereits Verfahren zum Tracking der Pose eines Objektes im Raum. Ein modellbasierter Ansatz wie in [Azad et al., 2011] ist hier nicht umsetzbar, da eine Kugel getrackt werden soll, deren Kontur sich nicht durch eine Rotation verändert und die über keine ausreichende Textur verfügt. Der Ansatz aus [Hajimolahoseini et al., 2014] ähnelt dem hier entwickelten Verfahren, jedoch ist hier eine spezifische Initialisierung notwendig, in der die Pose der Kamera bestimmt wird. Weiterhin werden ein physikalisches Bewegungsmodell der Kugel und ein Verfahren zur Datenassoziation entwickelt, die speziell auf die untersuchte Anwendung angepasst sind.

2 Grundlagen

In diesem Kapitel werden die Grundlagen dargelegt, die in der vorliegenden Arbeit Anwendung finden. In Abschnitt 2.1 werden Methoden und Definitionen der Bildverarbeitung vorgestellt, die relevant für die Detektion der Marker sind. Darauf folgt in Abschnitt 2.2 die Definition von Rotationsmatrizen, welche genutzt werden um die relative Verdrehung der Kugel sowie der Kamera gegenüber dem Rahmen zu beschreiben. Weiterhin werden in Abschnitt 2.3 Algorithmen und Modelle der Parameterschätzung präsentiert. Nach einem Überblick über die Grundlagen der Methode der kleinsten Quadrate anhand des Gauß-Markov-Modelles wird das Gauß-Hermet-Modell sowie das Extended Kalman Filter beschrieben. Zuletzt wird in Abschnitt 2.4 das Modell der Zentralprojektion sowie auf diesem Modell basierende Methoden aufgeführt.

2.1 Bildverarbeitung

Im Folgenden werden einige Grundlagen der Bildverarbeitung betrachtet. Die mathematische Darstellung von digitalen Bildern in verschiedenen Farbräumen wird dargelegt. Weiterhin werden Verfahren zur Reduktion des Bildrauschens, sowie binäre morphologische Operationen erläutert.

2.1.1 Farbräume

Ein Bild kann in verschiedenen Farbräumen dargestellt werden. Die in dieser Arbeit verwendeten Farbräume sind RGB, HSV und Grauwertbilder, wobei die Definitionen aus [Szeliski, 2011] genutzt werden. Ein Bild, das im RGB Farbraum repräsentiert wird, besteht aus drei Kanälen die den jeweiligen Rot-, Grün- und Blauwert jedes Pixels beinhalten. Ein Bild mit der Breite w und Höhe h kann folglich durch drei Matrizen dargestellt werden, wobei jede Matrix die Dimension $h \times w$ aufweist. Alternativ können die Informationen eines Bildes auch im HSV Raum dargestellt werden. Dieser setzt sich aus den Kanälen Hue, Saturation und Value zusammen. Hue beschreibt den Farbton eines Pixels, Saturation dessen Sättigung und Value dessen Helligkeit. Die Darstellung eines Bildes im HSV Raum eignet sich dazu, Bildpunkte nach ihrem Farbton zu identifizieren, unabhängig von der jeweiligen Sättigung und Helligkeit. Um ein Bild, das im RGB Raum vorliegt, in den HSV Raum zu konvertieren, müssen die folgenden Gleichungen für jeden Pixel mit den RGB Werten r, g, b angewendet werden, um die HSV Werte h, s, v zu erhalten.

$$v = \max(r, g, b) \quad m = \min(r, g, b) \quad c = v - m \quad (1)$$

$$h = \begin{cases} \text{undefiniert} & , \text{ falls } c = 0 \\ 60^\circ \cdot \left(\frac{g-b}{c} \bmod 6\right) & , \text{ falls } v = r \\ 60^\circ \cdot \left(\frac{b-r}{c} + 2\right) & , \text{ falls } v = g \\ 60^\circ \cdot \left(\frac{r-g}{c} + 4\right) & , \text{ falls } v = b \end{cases} \quad s = \begin{cases} 0 & , \text{ falls } v = 0 \\ \frac{c}{v} & , \text{ falls } v \neq 0 \end{cases} . \quad (2)$$

Zuletzt findet in dieser Arbeit die Darstellung eines Bildes als Grauwertbild Anwendung. Der Grauwert p_{grau} eines Pixels mit den RGB Werten r, g, b berechnet sich zu

$$p_{\text{grau}} = \frac{r + g + b}{3}. \quad (3)$$

Wird ein farbiges Bild in ein Grauwertbild konvertiert, geht die Farbinformation verloren, da alle Informationen auf einen Kanal reduziert werden. Viele Algorithmen der Bildverarbeitung basieren auf Grauwertbildern, da diese meist noch genug Informationen enthalten. Abbildung 2 zeigt die unterschiedlichen Kanäle eines Bildes. Hierbei gilt, je heller ein Bildpunkt dargestellt ist, desto höher der Wert im entsprechenden Kanal. Das Eingangsbild (oben links) wird in die Kanäle Rot, Grün und Blau (oben rechts) bzw. Hue, Saturation und Value (unten links) aufgeteilt. Das entsprechende Grauwertbild ist unten rechts abgebildet. Für eine detaillierte Übersicht der Farbräume und die beschriebenen Zusammenhänge zwischen diesen wird auf [Szeliski, 2011], Kapitel 2.3.2 verwiesen.

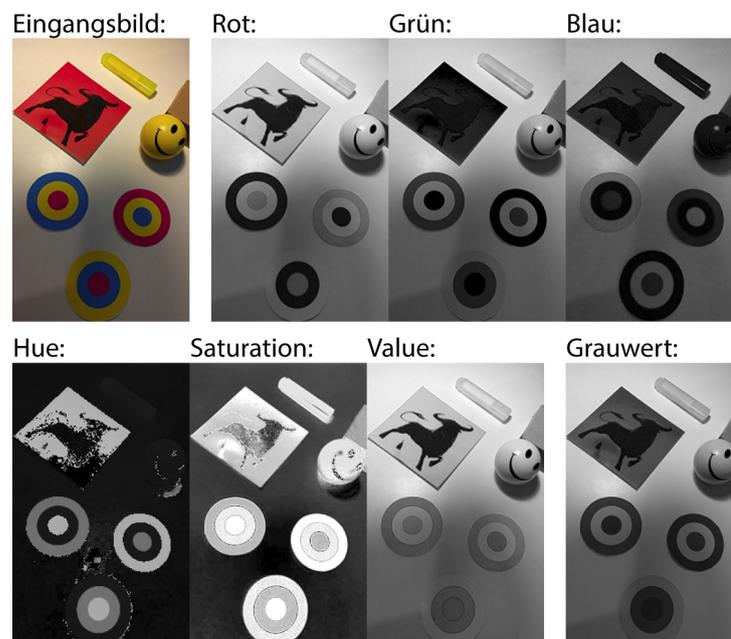


Abbildung 2: Darstellung der Farbräume eines Bildes

2.1.2 Bildrauschen und Reduktion des Rauschens

Das Bildrauschen bezeichnet zufällige Fehler der Farbwerte einzelner Pixel. Bei der Analyse eines Bildes können diese Fehler einen negativen Einfluss auf die Qualität der Ergebnisse haben. Aus diesem Grund werden in der Bildverarbeitung diverse Methoden eingesetzt um das Rauschen eines Bildes zu reduzieren, bevor weitere Algorithmen angewendet werden (vgl. [Szeliski, 2011], Seite 89). Eine Methode ist die lineare Filterung mit einer Filtermatrix \mathbf{K} . Hierbei wird durch Faltung der Mittelpunkt von \mathbf{K} auf jeden Bildpunkt p eines Grauwertbildes \mathbf{B} gelegt und die übereinanderliegenden Einträge von \mathbf{K} und \mathbf{B} multipliziert. Die Summe dieser Produkte ergeben den neuen Wert des Pixels p . Der Aufbau der Filtermatrix (Größe und Wahl der Einträge) bestimmt den Effekt der Filterung. Um das Bildrauschen zu reduzieren werden beispielsweise Gauss- oder Blockfilter genutzt. Diese haben jedoch den Effekt, Konturen von abgebildeten Objekten zu verwaschen, was von Nachteil ist, wenn diese Konturen möglichst präzise bestimmt werden sollen. Neben den linearen Filtern gibt es auch nicht lineare Filter. Zu dieser Gruppe gehört der Medianfilter, der in dieser Arbeit genutzt wird. Bei diesem wird ebenfalls eine Filtermatrix auf jeden Bildpunkt gelegt. Der neue Wert des Bildpunktes entspricht dem Median aller Pixel die sich unterhalb der Filtermatrix befinden. Der Medianfilter wird häufig eingesetzt, wenn das eingehende Bild für eine Konturensuche vorbereitet werden soll [Patidar et al., 2010]. Insbesondere eignet er sich, falls ein Salt-and-pepper Rauschen vorliegt, wohingegen lineare Filter wie der Gaussfilter bessere Ergebnisse bei einem Normal- oder Poissonverteilten Rauschen erzielen [Patidar et al., 2010]. Abbildung 3 zeigt die Anwendung eines Medianfilters mit einer Filtermatrix der Größe 3×3 auf ein verrauschtes Eingangsbild. Das Eingangsbild zeigt einen weißen Punkt auf schwarzem Hintergrund wobei einige Punkte grau gefärbt sind, um das Bildrauschen zu simulieren. Zu sehen ist, dass der Filter das Bild bis auf drei Pixel wiederherstellen kann. In der Mitte der Abbildung sind beispielhaft drei Positionen der Filtermatrix und die Ermittlung des Median dargestellt.

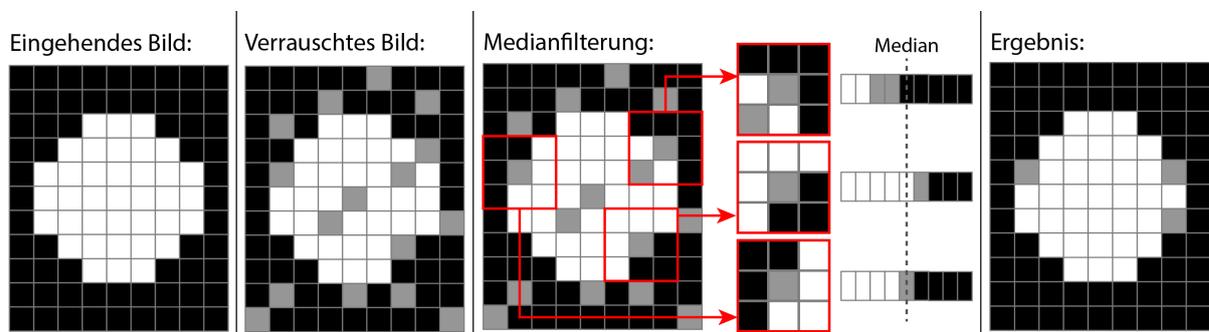


Abbildung 3: Beispiel der Anwendung eines Medianfilters

Der direkte Vergleich eines Gauss- und Medianfilters ist in Abbildung 4 zu sehen. Das eingehende Grauwertbild (links) mit einer Abmessung von 196×316 Pixeln wurde mit einem künstlichen weißen Rauschen ($\sigma = 11$) versehen (zweites Bild von links). Anschließend wurden Gauss- und Medianfilter verschiedener Größe angewendet. Zu sehen ist, dass beide Filtertypen mit zunehmender Größe der Filtermatrix das Rauschen besser entfernen. Während der Gaussfilter die Konturen der abgebildeten Objekte innerhalb des Bildes verschwimmen lässt, tendiert der Medianfilter dazu, Details zu entfernen. Für weitere Informationen zur Verarbeitung von Bildern durch Filter wird auf Kapitel 3.2 und 3.3 in [Szeliski, 2011] verwiesen.

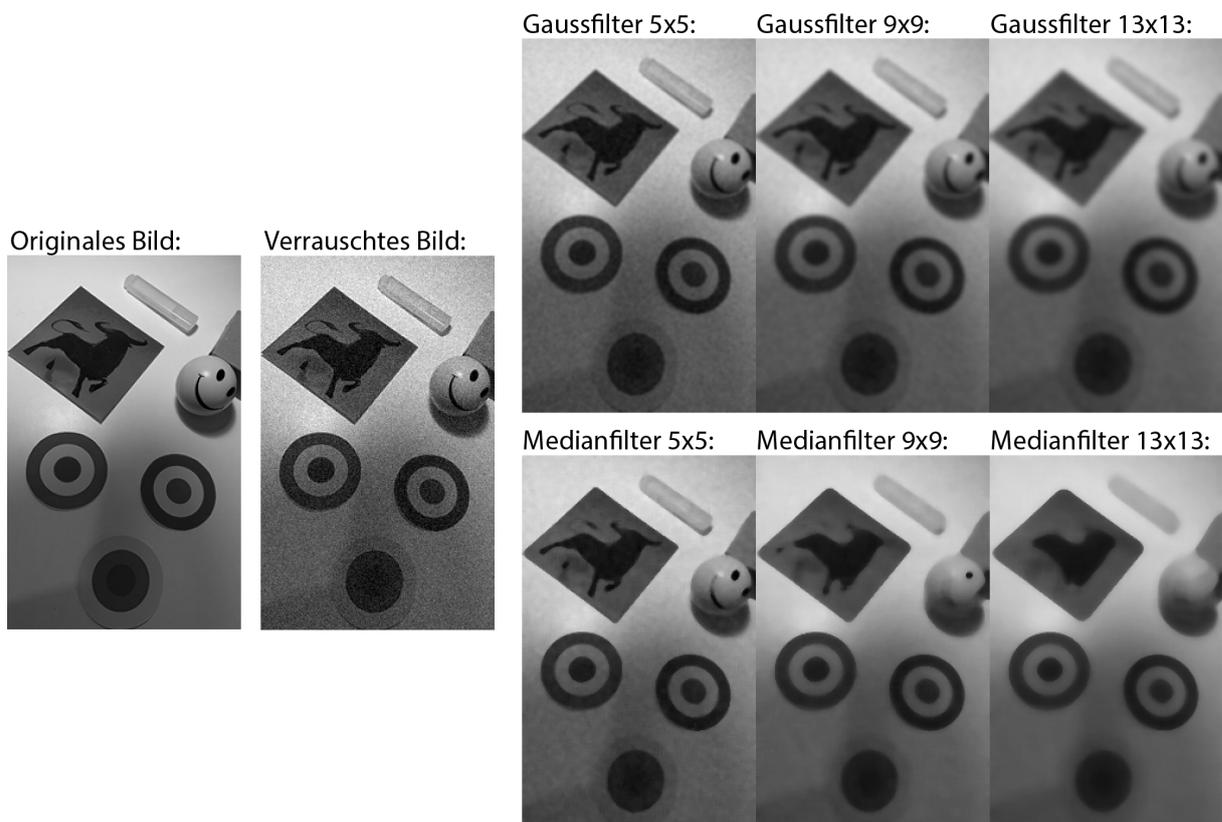


Abbildung 4: Ergebnis der Filterung durch Median- und Gaußfilter versch. Größe

2.1.3 Binäre morphologische Operationen

Binäre morphologische Operationen werden auf binäre Bilder angewendet, um diese für eine bestimmte Anwendung zu optimieren. Hierzu werden Strukturelemente angewendet. Ein Strukturelement besteht aus einer zweidimensionalen, binären Matrix, wobei die Form durch die Einträge mit dem Wert eins definiert wird. Weiterhin ist in jedem Strukturelement der Ankerpunkt definiert. In Abbildung 5 sind beispielhaft drei Strukturelemente dargestellt.

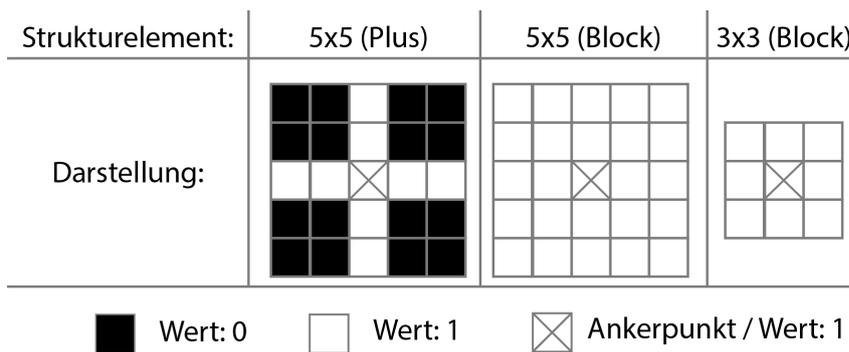


Abbildung 5: Mögliche Strukturelemente für morphologische Operationen

Die für diese Arbeit relevanten Operationen sind **Dilatation**, **Erosion** und **Opening**. Bei der **Dilatation** wird der Ankerpunkt eines Strukturelementes **S** auf jeden Eintrag m_i des binären Bildes **M** gelegt. Hierbei bezeichnet $i = 1, \dots, n$ den Index jeder Position in **M** und $n = h \cdot w$ die Anzahl der Pixel in **M**, wobei h die Anzahl der Zeilen und w die Anzahl der Spalten des Bildes bezeichnen. Befindet sich der Ankerpunkt über dem Pixel m_i , dann seien m_j die Pixel in **M**, die sich mit einem Eintrag des Strukturelementes mit dem Wert eins überlappen. Für m_i gilt dann:

$$m_i = \max(m_j). \tag{4}$$

Folglich wird das Pixel unter dem Ankerpunkt auf eins gesetzt, wenn mindestens ein Eintrag des Strukturelementes mit dem Wert eins auf ein Pixel des Bildes mit dem Wert eins trifft, anderenfalls auf null. Abbildung 6 zeigt die Anwendung der **Dilatation** auf ein binäres Bild mit einem Strukturelement der Größe 3×3 . Weiße Kacheln kennzeichnen den Wert eins, schwarze den Wert null. Grün sind jene Positionen eingefärbt, deren Wert sich von null auf eins ändert.

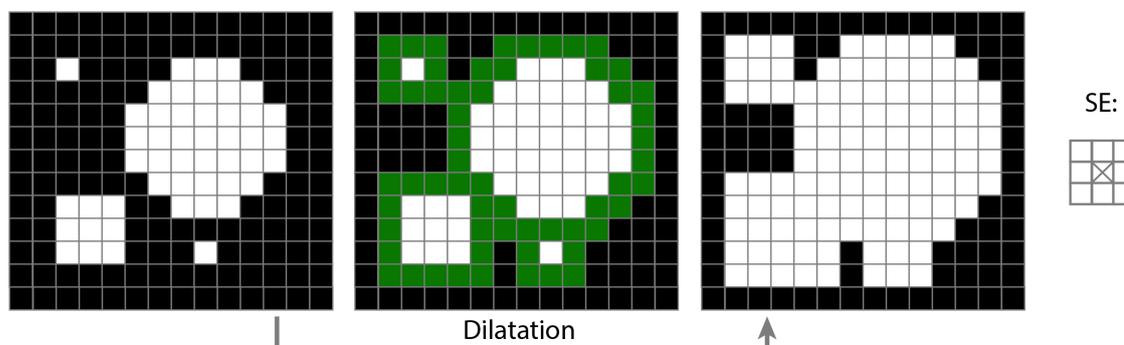


Abbildung 6: Beispiel für Dilatation

Die Operation **Erosion** funktioniert ähnlich wie die **Dilatation**, mit dem Unterschied, dass der neue Wert für m_i dem Minimum der Einträge m_j entspricht:

$$m_i = \min(m_j). \tag{5}$$

Folglich wird das Pixel unter dem Ankerpunkt auf eins gesetzt, wenn alle Einträge des Strukturelementes mit dem Wert eins auf ein Pixel des Bildes mit dem Wert eins treffen, anderenfalls auf null. Abbildung 7 zeigt das Resultat einer **Erosion** mit einem Strukturelementes der Größe 3×3 . Hierbei sind alle Positionen rot eingefärbt, deren Wert sich von eins auf null ändert.

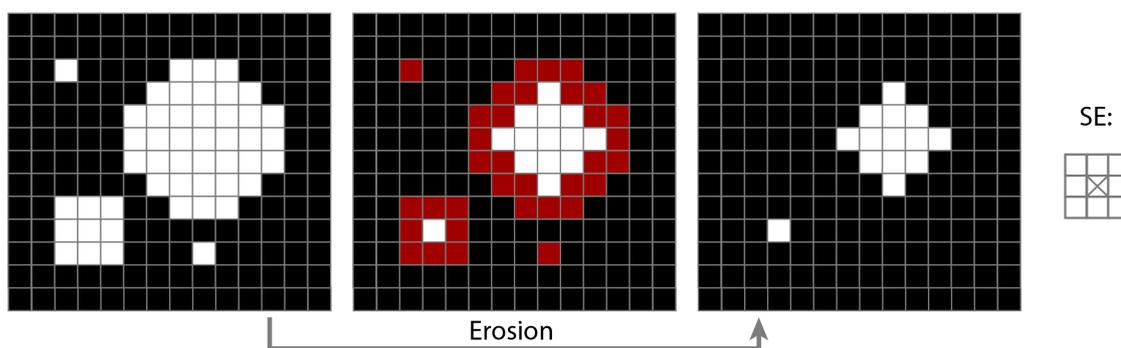


Abbildung 7: Beispiel für Erosion

Das **Opening** bezeichnet die Kombination von **Erosion** und **Dilatation**. Diese Operation entfernt kleine, unzusammenhängende Bestandteile der Maske, wie in Abbildung 8 zu sehen ist. Ebenfalls entfernt werden Stege, was beispielhaft in Abbildung 9 dargestellt ist. Eine Liste der morphologischen Operationen sowie weitere Erläuterungen zu diesen findet man in [Szeliski, 2011].

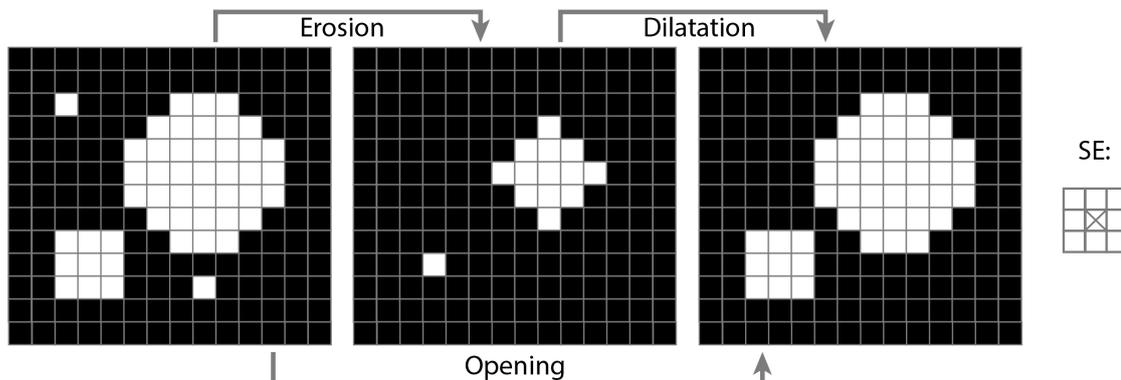


Abbildung 8: Beispiel für Opening (Entfernen kleiner Bereiche)

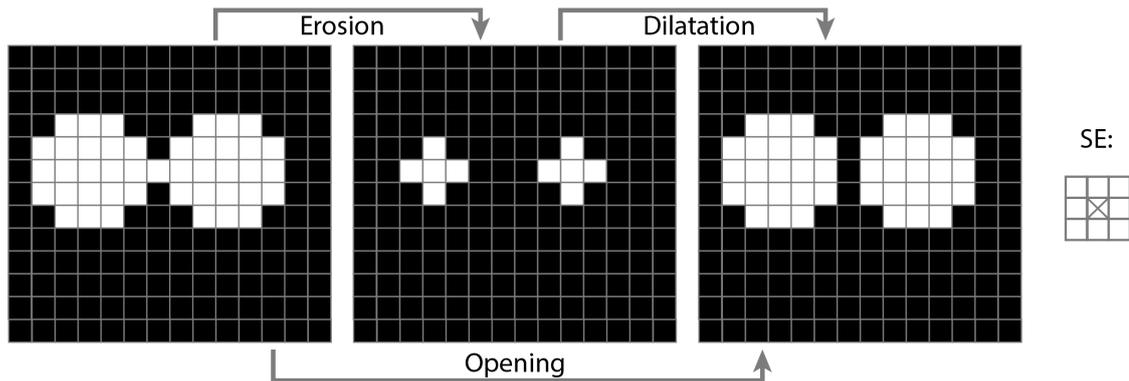


Abbildung 9: Beispiel für Opening (Entfernen eines Stegs)

2.2 Rotationsmatrizen

Die Orientierung eines Koordinatensystemes im Raum (auch als Lage bezeichnet) kann auf mehrere Weisen beschrieben werden. In dieser Arbeit wird sie durch drei Winkel ω, φ, κ ausgedrückt (Euler-Winkel oder auch Lagewinkel). Diese Winkel beschreiben Elementardrehungen um die x-, y- und z-Achse, welche verkettet werden, um die vollständige Rotation des Koordinatensystemes gegenüber einem übergeordneten Koordinatensystem zu definieren. Die Elementardrehung, dargestellt als Matrix, lautet für die Rotation um die x-Achse

$$\mathbf{R}_x(\omega) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \omega & \sin \omega \\ 0 & -\sin \omega & \cos \omega \end{pmatrix}, \quad (6)$$

um die y-Achse

$$\mathbf{R}_y(\varphi) = \begin{pmatrix} \cos \varphi & 0 & -\sin \varphi \\ 0 & 1 & 0 \\ -\sin \varphi & 0 & \cos \varphi \end{pmatrix} \quad (7)$$

und um die z-Achse

$$\mathbf{R}_z(\kappa) = \begin{pmatrix} \cos \kappa & \sin \kappa & 0 \\ -\sin \kappa & \cos \kappa & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (8)$$

Diese Matrizen werden von links nach rechts multipliziert, was bedeutet, dass zunächst um die x-Achse, dann um die mitgedrehte y-Achse und zuletzt um die mitgedrehte z-Achse rotiert wird. Hieraus resultiert die zusammengesetzte Rotationsmatrix

$$\mathbf{R}(\omega, \varphi, \kappa) = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} = \mathbf{R}_x(\omega) \cdot \mathbf{R}_y(\varphi) \cdot \mathbf{R}_z(\kappa)$$

$$= \begin{pmatrix} c_\varphi \cdot c_\kappa & -c_\varphi \cdot s_\kappa & s_\varphi \\ c_\omega \cdot s_\kappa + s_\omega \cdot s_\varphi \cdot c_\kappa & c_\omega \cdot c_\kappa - s_\omega \cdot s_\varphi \cdot s_\kappa & -s_\omega \cdot c_\varphi \\ s_\omega \cdot s_\kappa - c_\omega \cdot s_\varphi \cdot c_\kappa & s_\omega \cdot c_\kappa + c_\omega \cdot s_\varphi \cdot s_\kappa & c_\omega \cdot c_\varphi \end{pmatrix} \quad (9)$$

mit

$$\begin{aligned} c_\omega &:= \cos(\omega), c_\varphi := \cos(\varphi), c_\kappa := \cos(\kappa), \\ s_\omega &:= \sin(\omega), s_\varphi := \sin(\varphi), s_\kappa := \sin(\kappa), \end{aligned}$$

welche die Lage eines Objektes im Raum beschreibt. Rotationsmatrizen sind orthogonal, was bedeutet, dass

$$\mathbf{R}^T \cdot \mathbf{R} = \mathbf{R} \cdot \mathbf{R}^T = \mathbf{I} \quad (10)$$

und

$$\det(\mathbf{R}) = 1 \quad (11)$$

gilt. Aus der Orthogonalität folgt weiterhin, dass

$$\mathbf{R}^T = \mathbf{R}^{-1} \quad (12)$$

gilt. Das Rotieren eines Vektors \mathbf{v} erfolgt durch Multiplikation mit \mathbf{R} . Der resultierende, rotierte Vektor \mathbf{v}' kann durch Multiplikation mit \mathbf{R}^T wieder zurück gedreht werden. Folglich gilt

$$\mathbf{v}' = \mathbf{R} \cdot \mathbf{v} \quad (13)$$

und

$$\mathbf{v} = \mathbf{R}^{-1} \cdot \mathbf{v}' = \mathbf{R}^T \cdot \mathbf{v}'. \quad (14)$$

Die verwendete Konvention für Rotationsmatrizen folgt der in [McGlone et al., 2004] vorgestellten Konvention.

2.3 Parameterschätzung

In folgendem Abschnitt werden Methoden der Parameterschätzung im Sinne der Methode der kleinsten Quadrate vorgestellt, die in dieser Arbeit angewendet werden. Die Parameterschätzung wird zunächst anhand des Gauß-Markov-Modelles erläutert. Die Aufgabe einer Parameterschätzung wird wie folgt definiert. Der Vektor $\mathbf{x} = (x_1, x_2, \dots, x_u)^T$ beinhaltet die u unbekannt GröÙen eines Systems, die es zu ermitteln gilt; er wird daher im Folgenden als Unbekanntenvektor bezeichnet. Darüber hinaus sollen auch die Unsi-

cherheiten sowie die Korrelationen der ermittelten Unbekannten geschätzt werden. Unsicherheiten und Korrelationen werden durch die Varianz-Kovarianz-Matrix (VKM) $\Sigma_{\mathbf{xx}}$ des Unbekanntenvektors dargestellt. $\Sigma_{\mathbf{xx}}$ hat den Aufbau

$$\Sigma_{\mathbf{xx}} = \begin{pmatrix} \sigma_{x_1}^2 & \cdots & \sigma_{x_1 x_u} \\ \vdots & \ddots & \vdots \\ \sigma_{x_u x_1} & \cdots & \sigma_{x_u}^2 \end{pmatrix} \quad (15)$$

Hierbei ist $\sigma_{x_i}^2$ die Varianz der i -ten Unbekannten (Quadrat der Standardabweichung σ_{x_i}). Die Kovarianz $\sigma_{x_i x_j}$ der Unbekannten x_i und x_j hängt mit der Korrelation dieser beiden Unbekannten zusammen, wobei der Korrelationskoeffizient ρ_{x_i, x_j} mit der Gleichung

$$\rho_{x_i, x_j} = \frac{\sigma_{x_i x_j}}{\sigma_{x_i} \cdot \sigma_{x_j}} \quad (16)$$

berechnet wird. Weiterhin sei $\mathbf{l} = (l_1, l_2, \dots, l_n)^T$ der Beobachtungsvektor, welcher die n gemessenen, mit Unsicherheit behafteten Größen beinhaltet, aus welchen die Unbekannten geschätzt werden sollen. Die Varianz-Kovarianz-Matrix des Beobachtungsvektors $\Sigma_{\mathbf{ll}}$ beschreibt die Unsicherheiten und Korrelationen der Beobachtungen, wobei der Aufbau wie analog zu $\Sigma_{\mathbf{xx}}$ ist. Weiterhin seien $\mathbf{f} = (f_1, f_2, \dots, f_n)^T$ Gleichungen, die den expliziten Zusammenhang zwischen den Beobachtungen und dem Unbekanntenvektor beschreiben. Diese Gleichungen der Form

$$l_i = f_i(\mathbf{x}) \quad \text{mit: } i = 1, \dots, n \quad (17)$$

sind im allgemeinen nicht linear und werden funktionales Modell des Systems genannt. Die Differenz aus Anzahl der Beobachtungen und der Unbekannten wird Redundanz genannt. Die Redundanz f berechnet sich folglich zu

$$f = n - u. \quad (18)$$

Gilt $f = 0$, ist das System nicht redundant und es existiert eine eindeutige Lösung. Im Falle $f > 0$ liegen mehr Beobachtungen vor, als notwendig wären um die Unbekannten eindeutig zu bestimmen. In diesem Fall ist das funktionale Modell nur gültig, wenn die sogenannten Residuen $\mathbf{v} = (v_1, v_2, \dots, v_n)^T$ eingeführt werden. Diese bezeichnen die Abweichungen zwischen den eingehenden Beobachtungen und den Werten die sich aus dem funktionalen Zusammenhang und den Unbekannten für die Beobachtungen ergeben. Die

Gleichungen des funktionalen Modelles lauten somit

$$l_i + v_i = f_i(\mathbf{x}) \quad \text{mit: } i = 1, \dots, n. \quad (19)$$

Im Sinne der Methode der kleinsten Quadrate der Residuen gilt es nun, den Zustandsvektor so zu wählen, dass die gewichtete Summe der Quadrate aller Residuen minimal wird. Die Gewichtung erfolgt hierbei über die Gewichtsmatrix $\mathbf{P} = \Sigma_{\mathbf{v}}^{-1}$ welche bewirkt, dass genauere Beobachtungen stärker gewichtet werden. Für den optimalen, ausgeglichenen Unbekanntenvektor $\hat{\mathbf{x}}$ gilt folglich, dass er die Quadratsumme der gewichteten Residuen

$$\Omega = \mathbf{v}^T \cdot \mathbf{P} \cdot \mathbf{v} \quad (20)$$

minimiert, woraus

$$\hat{\mathbf{x}} = \operatorname{argmin}_{\mathbf{x}} \Omega \quad (21)$$

folgt. Falls das funktionale Modell ausschließlich aus linearen Gleichungen besteht, kann der Zusammenhang zwischen Beobachtungen und Unbekannten durch Einführen der Designmatrix \mathbf{A} hergestellt werden. Es gilt der Zusammenhang

$$\mathbf{l} + \mathbf{v} = \mathbf{A} \cdot \mathbf{x} \quad (22)$$

der zu

$$\mathbf{v} = \mathbf{A} \cdot \mathbf{x} - \mathbf{l} \quad (23)$$

umgeformt werden kann. Aus Gleichung 23 und 20 folgt

$$\Omega = (\mathbf{A} \cdot \mathbf{x} - \mathbf{l})^T \cdot \mathbf{P} \cdot (\mathbf{A} \cdot \mathbf{x} - \mathbf{l}) \quad (24)$$

$$= \mathbf{x}^T \cdot \mathbf{A}^T \cdot \mathbf{P} \cdot \mathbf{A} \cdot \mathbf{x} - 2\mathbf{x}^T \cdot \mathbf{A}^T \cdot \mathbf{P} \cdot \mathbf{l} + \mathbf{l}^T \cdot \mathbf{P} \cdot \mathbf{l} \quad (25)$$

Die Minimierung von Ω erfolgt in Form einer Extremwertaufgabe. Der Vektor der ausgeglichenen Unbekannten folgt zu

$$\hat{\mathbf{x}} = (\mathbf{A}^T \cdot \mathbf{P} \cdot \mathbf{A})^{-1} \cdot \mathbf{A}^T \cdot \mathbf{P} \cdot \mathbf{l} \quad (26)$$

und die VKM der ausgeglichenen Unbekannten zu

$$\Sigma_{\hat{\mathbf{x}}\hat{\mathbf{x}}} = (\mathbf{A}^T \cdot \mathbf{P} \cdot \mathbf{A})^{-1} \quad (27)$$

Die ausgeglichenen Beobachtungen $\hat{\mathbf{I}}$ werden definiert zu

$$\hat{\mathbf{I}} = \mathbf{1} + \hat{\mathbf{v}} = \mathbf{f}(\hat{\mathbf{x}}) \quad (28)$$

wobei $\hat{\mathbf{v}}$ die ausgeglichenen Residuen sind, die sich aus den Beobachtungen und dem ausgeglichenen Unbekanntenvektor ergeben.

Falls das funktionale Modell mindestens eine nicht lineare Gleichung beinhaltet, kann der Zusammenhang zwischen Beobachtungen und Unbekannten nicht wie in Gleichung 22 dargestellt werden. Die Berechnung von $\hat{\mathbf{x}}$ und $\hat{\mathbf{I}}$ erfolgt dann durch eine Linearisierung der Gleichungen f_1, \dots, f_n in Form einer Taylor-Entwicklung mit Abbruch nach dem ersten Glied wofür Näherungswerte \mathbf{x}_0 für den ausgeglichenen Vektor der Unbekannten bekannt sein müssen. Der Zusammenhang zwischen Beobachtungen und Unbekannten lautet

$$l_i + v_i = f_i(\mathbf{x}_0) + \frac{\delta f_i(\mathbf{x})}{\delta \mathbf{x}}. \quad (29)$$

Da $\hat{\mathbf{x}}$ und $\Sigma_{\hat{\mathbf{x}}\hat{\mathbf{x}}}$ nicht mehr direkt berechnet werden können erfolgt die Berechnung iterativ. Im Folgenden kennzeichnet $j = 1, \dots, J$ die Nummer der Iteration. Die Designmatrix \mathbf{A}_j ergibt sich im nicht linearen Fall aus den Ableitungen der funktionalen Gleichungen f_1, \dots, f_n nach den Unbekannten an der Stelle \mathbf{x}_{j-1} .

$$\mathbf{A}_j = \left. \frac{\delta f_i}{\delta \mathbf{x}} \right|_{\mathbf{x}_{j-1}} \quad (30)$$

Hieraus folgt, dass der Zusammenhang aus Gleichung 26 im nicht linearen Fall für den gekürzten Beobachtungsvektor

$$\Delta \mathbf{l}_j = \mathbf{1} - f(\mathbf{x}_{j-1}) \quad (31)$$

und den gekürzten Unbekanntenvektor

$$\Delta \mathbf{x}_j = \mathbf{x}_j - \mathbf{x}_{j-1} \quad (32)$$

gilt. Für die Residuen folgt analog zu Gleichung 23

$$\mathbf{v}_j = \mathbf{A}_j \cdot \Delta \mathbf{x}_j - \Delta \mathbf{l}_j \quad (33)$$

Somit kann der gekürzte Beobachtungsvektor zu

$$\Delta \mathbf{x}_j = (\mathbf{A}_j^T \cdot \mathbf{P} \cdot \mathbf{A}_j)^{-1} \cdot \mathbf{A}_j^T \cdot \mathbf{P} \cdot \Delta \mathbf{l}_j \quad (34)$$

und der verbesserte Zustandsvektor zu

$$\mathbf{x}_j = \mathbf{x}_{j-1} + \Delta \mathbf{x}_j \quad (35)$$

berechnet werden. Die Iteration wird abgebrochen, sobald sich der Zustandsvektor nicht mehr maßgeblich ändert, also der Betrag von $\Delta \mathbf{x}_j$ unter einen Grenzwert fällt. Dann gilt

$$\hat{\mathbf{x}} = \mathbf{x}_j \quad (36)$$

und

$$\Sigma_{\hat{\mathbf{x}}\hat{\mathbf{x}}} = (\mathbf{A}_j^T \cdot \mathbf{P} \cdot \mathbf{A}_j)^{-1} \cdot \hat{\sigma}_0^2 \quad (37)$$

mit der empirischen Varianz der Gewichtseinheit

$$\hat{\sigma}_0^2 = \frac{\mathbf{v}^T \cdot \mathbf{P} \cdot \mathbf{v}}{n - u} \quad (38)$$

Die VKM des ausgeglichenen Unbekanntenvektors, die in der Ausgleichung berechnet wird, legt dar, mit welchen Genauigkeiten die Zustände bestimmt wurden und ob Korrelationen der Zustände vorliegen. Dies wird als innere Genauigkeit bezeichnet und ist neben der äußeren Genauigkeit, die durch den Vergleich mit einer Referenz entsteht, ein Maß für die Güte der Ausgleichung bzw. der Beobachtungen. Die Theorie der Methode der kleinsten Quadrate sowie das Gauß-Markov-Modell wird in [McGlone et al., 2004] erläutert.

2.3.1 Gauß-Helmert-Modell

Das Gauß-Helmert-Modell (GHM) beschreibt den allgemeinsten Fall der Ausgleichungsrechnung. Im Gegensatz zu dem in Abschnitt 2.3 vorgestellten Gauß-Markov-Modell basiert das GHM nicht auf einem expliziten, sondern einem impliziten Zusammenhang zwischen den Unbekannten und den Beobachtungen. Das funktionale Modell wird durch p linear unabhängige Bedingungsgleichungen f_1, \dots, f_p der Form

$$0 = f_i(\mathbf{l}, \mathbf{x}) \quad (39)$$

formuliert. Die Anzahl der Freiheitsgrade f berechnet sich als Differenz aus der Anzahl Bedingungsgleichungen p und der Anzahl der Unbekannten u zu

$$f = p - u. \quad (40)$$

Wie im nicht linearen GMM wird $\hat{\mathbf{x}}$ iterativ durch eine Linearisierung der Bedingungsgleichungen bestimmt, weswegen erneut j zur Bezeichnung der Iteration genutzt wird. Neben

den Residuen wird im GHM der Widerspruchsvektor $\mathbf{w}_j = (w_{j,1}, \dots, w_{j,p})^T$ definiert, dessen Einträge sich zu

$$w_{j,i} = f_i(\mathbf{l}, \mathbf{x}_{j-1}) \quad (41)$$

ergeben. Die Linearisierung erfolgt durch Differenzieren der Bedingungsgleichungen nach den Unbekannten (Designmatrix \mathbf{A}) bzw. den Beobachtungen (Bedingungsmatrix \mathbf{B}):

$$\text{Designmatrix : } \mathbf{A}_j = \frac{\delta f_i}{\delta \mathbf{x}} \Big|_{\mathbf{l}_j, \mathbf{x}_{j-1}} \quad (42)$$

$$\text{Bedingungsmatrix : } \mathbf{B}_j = \frac{\delta f_i}{\delta \mathbf{l}} \Big|_{\mathbf{l}_j, \mathbf{x}_{j-1}}. \quad (43)$$

Durch die Linearisierung kann das linearisierte funktionale Modell des GHM aufgestellt werden:

$$0 = \mathbf{w} + \mathbf{B}^T \cdot \mathbf{v} + \mathbf{A} \cdot \Delta \mathbf{x}. \quad (44)$$

Die Schätzung des optimalen Unbekanntenvektors $\hat{\mathbf{x}}$ erfolgt iterativ, wobei für die erste Iteration erneut Näherungswerte für \mathbf{x}_0 benötigt werden. Die Berechnungen jeder Iteration werden im Folgenden aufgeführt. Zunächst muss \mathbf{A}_j und \mathbf{B}_j bestimmt werden. Anschließend wird die Matrix

$$\Sigma_{\mathbf{w}\mathbf{w},j} = \mathbf{B}_j \cdot \Sigma_{\mathbf{l}} \cdot \mathbf{B}_j^T \quad (45)$$

berechnet, mit der die Normalgleichungsmatrix

$$\mathbf{N}_j = (\mathbf{A}_j^T \cdot \Sigma_{\mathbf{w}\mathbf{w},j}^{-1} \cdot \mathbf{A}_j) \quad (46)$$

ermittelt werden kann. Der Vektor der Unbekannten wird anhand der Gleichung

$$\mathbf{x}_j = \mathbf{N}_j^{-1} \cdot \mathbf{A}_j^T \cdot \Sigma_{\mathbf{w}\mathbf{w},j}^{-1} \cdot (-\mathbf{w}_j) \quad (47)$$

verbessert. Zusätzlich werden die Beobachtungen verbessert, wozu zunächst der Korrelationsvektor

$$\mathbf{k}_j = \Sigma_{\mathbf{w}\mathbf{w},j}^{-1} \cdot (\mathbf{I} - \mathbf{A}_j \cdot \mathbf{N}_j^{-1} \cdot \mathbf{A}_j^T \cdot \Sigma_{\mathbf{w}\mathbf{w},j}^{-1}) \cdot (-\mathbf{w}_j) \quad (48)$$

und aus diesem die Residuen

$$\mathbf{v}_j = \Sigma_{\mathbf{l}} \cdot \mathbf{B} \cdot \mathbf{k}_j \quad (49)$$

berechnet werden. Die Verbesserung der Beobachtungen erfolgt, indem die Residuen auf die ursprünglichen Beobachtungen addiert werden:

$$\mathbf{l}_j = \mathbf{l}_{j-1} + \mathbf{v}_j \quad (50)$$

Es wird iteriert, bis das Abbruchkriterium eintritt. Wie beim nicht linearen GMM kann hierzu geprüft werden, ob der Betrag der Verbesserung des Unbekanntenvektors gegenüber dem des vorangegangenen Iterationsschrittes unter einen Grenzwert fällt. Der in der letzten Iteration berechnete Vektor der Unbekannten \mathbf{x}_j ist der gesuchte, ausgeglichene Unbekanntenvektor $\hat{\mathbf{x}}$. Gleichmaßen ergibt sich die VKM des ausgeglichenen Unbekanntenvektors aus der letzten Iteration zu

$$\hat{\Sigma}_{\hat{\mathbf{x}}\hat{\mathbf{x}}} = \hat{\sigma}_0^2 \cdot \mathbf{N}^{-1} \quad (51)$$

mit der empirischen Varianz der Gewichtseinheit $\hat{\sigma}_0^2$ entsprechend Gleichung 38. Der Algorithmus des GHM ist in [McGlone et al., 2004] dokumentiert.

2.3.2 Extended Kalman Filter

Der Algorithmus, der in dieser Arbeit genutzt wird, um den Zustand der Kugel zu verfolgen, ist der Extended Kalman Filter (EKF) Algorithmus. Dieser dient der optimalen Schätzung eines zeitlich veränderlichem Zustandsvektors $\mathbf{x} = (x_1, \dots, x_u)^T$, der den Zustand eines durch u Parameter beschriebenen physikalischen Systems beschreibt. Die Zeit wird hierbei diskret behandelt, wozu feste Zeitpunkte, die sogenannten Epochen eingeführt werden. Das physikalische Modell ermöglicht es, den Zustandsvektor der Epoche k aus dem Zustandsvektor der vorangegangenen Epoche $k - 1$ zu prädictieren. Gleichzeitig werden Beobachtungen hinzugezogen, die mit den Parametern des physikalischen Systems zusammenhängen. Die Beobachtungen stammen aus einem Sensor (z.B. einer Kamera). Der Zusammenhang zwischen den Beobachtungen und dem Zustandsvektor wird durch die Funktionen $\mathbf{h}^{(k)} = (h_1, \dots, h_n)$ beschrieben, wobei n die Anzahl der Beobachtungen in der Epoche k ist. Das EKF berücksichtigt sowohl die Prädiktion als auch die Beobachtungen, wobei die Gewichtung gemäß einem stochastischen Modell erfolgt. Die Prädiktion erfolgt durch das physikalische Bewegungsmodell $\mathbf{f} = (f_1, \dots, f_n)$. Folglich beinhaltet \mathbf{f} die u Gleichungen, mit denen jedes Element des Zustandsvektors prädictiert wird. Hier geht ein Störterm \mathbf{w} mit ein, der das Prozessrauschen und somit die Abweichung des Bewegungsmodells von der Realität beschreibt. Die Prädiktion erfolgt zu

$$\mathbf{x}^{(k)} = \mathbf{f}(\mathbf{x}^{(k-1)}, \mathbf{w}^{(k-1)}). \quad (52)$$

Zwischen den Beobachtungen und dem Zustandsvektor der Epoche k gilt der Zusammenhang

$$\mathbf{l}^{(k)} = \mathbf{h}^{(k)}(\mathbf{x}^{(k)}, \mathbf{v}^{(k)}), \quad (53)$$

wobei erneut die Residuen \mathbf{v} Anwendung finden. Es wird angenommen, dass Störterm und Residuen gemäß

$$\mathbf{w}^{(k)} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\mathbf{w}\mathbf{w}}^{(k)}) \quad (54)$$

und

$$\mathbf{v}^{(k)} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{\mathbf{v}\mathbf{v}}^{(k)}) \quad (55)$$

einer Normalverteilung mit dem Erwartungswert $E = 0$ entstammen. Die fortlaufende Berechnung des Zustandsvektors erfolgt abwechselnd durch den Prädiktions- und Updateschritt. Bei der Prädiktion wird der Zustandsvektor \mathbf{x}_- sowie dessen VKM $\boldsymbol{\Sigma}_{\mathbf{x}\mathbf{x},-}$ anhand des zuletzt ermittelten Zustandes prädiziert. Im Updateschritt werden die in der entsprechenden Epoche vorliegenden Beobachtungen genutzt, um den Zustandsvektor zu verbessern. Nach dem Updateschritt liegt der Zustandsvektor \mathbf{x}_+ sowie dessen VKM $\boldsymbol{\Sigma}_{\mathbf{x}\mathbf{x},+}$ vor. Da beim EKF im Gegensatz zum regulären Kalman Filter nicht vorausgesetzt wird, dass \mathbf{f} und $\mathbf{h}^{(k)}$ ausschließlich lineare Funktionen beinhalten, werden diese Funktionen durch eine Taylor-Entwicklung erster Ordnung approximiert. Die Differenzierung der Gleichungen des Bewegungsmodelles nach den Beobachtungen ergibt die Transitionsmatrix

$$\boldsymbol{\Phi}^{(k-1)} = \left. \frac{\delta \mathbf{f}}{\delta \mathbf{x}} \right|_{\mathbf{x}_+^{(k-1)}}. \quad (56)$$

Die Stellmatrix $\mathbf{G}^{(k-1)}$ wird durch Differenzieren des Bewegungsmodelles nach dem Störterm berechnet:

$$\mathbf{G}^{(k-1)} = \left. \frac{\delta \mathbf{f}}{\delta \mathbf{w}} \right|_{\mathbf{x}_+^{(k-1)}}. \quad (57)$$

Die Stell- und die Transitionsmatrix werden aus den Differentialen an der Stelle $\mathbf{x}_+^{(k-1)}$ berechnet, da die Matrizen zur Prädiktion genutzt werden. Analog zu Gleichung 30 ergibt sich die Designmatrix \mathbf{A} , indem der Zusammenhang zwischen den Beobachtungen und den Zuständen nach dem Zustandsvektor differenziert wird:

$$\mathbf{A}^{(k)} = \left. \frac{\delta \mathbf{h}^{(k)}}{\delta \mathbf{x}} \right|_{\mathbf{x}_-^{(k)}}. \quad (58)$$

Zuletzt ergibt sich die Matrix $\mathbf{M}^{(k)}$, indem \mathbf{h} nach den Residuen differenziert wird:

$$\mathbf{M}^{(k)} = \left. \frac{\delta \mathbf{h}^{(k)}}{\delta \mathbf{v}} \right|_{\mathbf{x}_-^{(k)}}. \quad (59)$$

Die Matrizen \mathbf{A} und \mathbf{M} werden für den Updateschritt verwendet, weswegen die Differentiale an der Stelle $\mathbf{x}_-^{(k)}$ berechnet werden.

Nach der Initialisierung von $\mathbf{x}_+^{(0)}$ und $\Sigma_{\mathbf{xx},+}^{(0)}$, dem angenommenen Zustandsvektor der ersten Epoche sowie dessen VKM, wird in jeder Epoche zunächst der Zustandsvektor gemäß dem Bewegungsmodell prädiziert:

$$\mathbf{x}_-^{(k)} = \mathbf{f}(\mathbf{x}_+^{(k-1)}, \mathbf{w}^{(k-1)}) \quad (60)$$

Die Prädiktion der VKM des Zustandsvektors erfolgt zu

$$\Sigma_{\mathbf{xx},-}^{(k)} = \Phi^{(k-1)} \cdot \Sigma_{\mathbf{xx},+}^{(k-1)} \cdot \Phi^{(k-1)\mathbf{T}} + \mathbf{G}^{(k-1)} \cdot \Sigma_{\mathbf{ww}}^{(k-1)} \cdot \mathbf{G}^{(k-1)\mathbf{T}}. \quad (61)$$

Hierdurch erhöht sich die Unsicherheit der Zustände, da das Prozessrauschen addiert wird. Nach der Prädiktion erfolgt der Updateschritt, in welche die Beobachtungen und deren VKM verwendet werden um den Zustandsvektor und dessen VKM zu verbessern. Hierzu wird zunächst die Kalman-Matrix \mathbf{K} zu

$$\mathbf{K}^{(k)} = \Sigma_-^{(k)} \cdot \mathbf{A}^{(k)\mathbf{T}} \cdot (\mathbf{A}^{(k)} \cdot \Sigma_{\mathbf{xx},-}^{(k)} \cdot \mathbf{A}^{(k)\mathbf{T}} + \mathbf{M}^{(k)} \cdot \Sigma_{\mathbf{ll}}^{(k)} \cdot \mathbf{M}^{(k)\mathbf{T}})^{-1} \quad (62)$$

berechnet. Zuletzt wird \mathbf{K} genutzt, um den gefilterten Zustandsvektor

$$\mathbf{x}_+^{(k)} = \mathbf{x}_-^{(k)} + \mathbf{K}^{(k)} \cdot (\mathbf{l}^{(k)} - h(\hat{\mathbf{x}}_-^{(k)}, \mathbf{0})) \quad (63)$$

und dessen VKM

$$\Sigma_{\mathbf{xx},+}^{(k)} = (\mathbf{I} - \mathbf{K}^{(k)} \cdot \mathbf{A}^{(k)}) \cdot \Sigma_{\mathbf{xx},-}^{(k)} \quad (64)$$

zu berechnen. Für weitere Berechnungen genutzt bzw. ausgewertet werden Zustandsvektor und VKM jeweils nach dem Updateschritt. Der vollständige EKF-Algorithmus kann in [Throne et al., 2006] nachvollzogen werden.

2.4 Zentralprojektion

Um den Zusammenhang zwischen einer Szene und dem von einer Kamera aufgenommenen Abbild der Szene mathematisch zu beschreiben, wird das Modell der Zentralprojektion verwendet, das in diesem Abschnitt erläutert wird. In diesem Modell gelten die Kollinearitätsgleichungen, die den Zusammenhang zwischen einem Objektpunkt im Raum und den Koordinaten des abgebildeten Punktes im Bild einer Kamera beschreiben. Abweichungen zwischen dem idealisierten Modell der Zentralprojektion und der Realität werden durch Modelle für die Kameraverzeichnung berücksichtigt. Die Parameter der Kamera und der Verzeichnung werden in einer Kamerakalibrierung ermittelt. Das Modell der Zentralprojektion wird auch in der Bündelblockausgleichung und dem räumlichen Rückwärtsschnitt genutzt. Diese Verfahren werden ebenfalls in diesem Abschnitt behandelt.

2.4.1 Geometrie und Kollinearitätsgleichungen

Die Zentralprojektion basiert auf der Annahme, dass sich alle Lichtstrahlen, die von einem Objekt reflektiert werden und auf den Sensor der Kamera fallen, in einem gemeinsamen Punkt schneiden, dem Projektionszentrum. In Positivstellung liegt die Bildebene zwischen dem Projektionszentrum und der Szene, man sieht gewissermaßen durch das Bild auf das Objekt. Nun gilt, dass jeder Pixel $\mathbf{p} = (x, y)^T$ in der Bildebene, der korrespondierende Objektpunkt $\mathbf{P} = (X, Y, Z)^T$ sowie das Projektionszentrum auf einer Geraden liegen, welche als Bildstrahl bezeichnet wird. Das kartesische Koordinatensystem der Szene mit den Achsen X, Y und Z wird als Objektkoordinatensystem bezeichnet. Die Achsen x, y und z definieren das Sensorkoordinatensystem mit der Einheit Pixel, wobei Bildpunkte stets auf der Bildebene ($z = 0$) liegen. Im Objektraum ist die Position des Projektionszentrums als $\mathbf{PZ} = (X_0, Y_0, Z_0)^T$, im Sensorkoordinatensystem zu (x_0, y_0, c) definiert. Hierbei ist $\mathbf{H} = (x_0, y_0, 0)^T$ der Bildhauptpunkt und c die Kammerkonstante. Die Parameter x_0, y_0 und c werden als innere Orientierung bezeichnet. Die Lage des Sensorkoordinatensystems im Objektraum wird durch die Rotationsmatrix $\mathbf{R}(\omega_0, \varphi_0, \kappa_0)$ beschrieben. Diese Lagewinkel werden zusammen mit X_0, Y_0 und Z_0 als äußere Orientierung der Kamera definiert. Abbildung 10 stellt die Geometrie der Zentralprojektion dar.

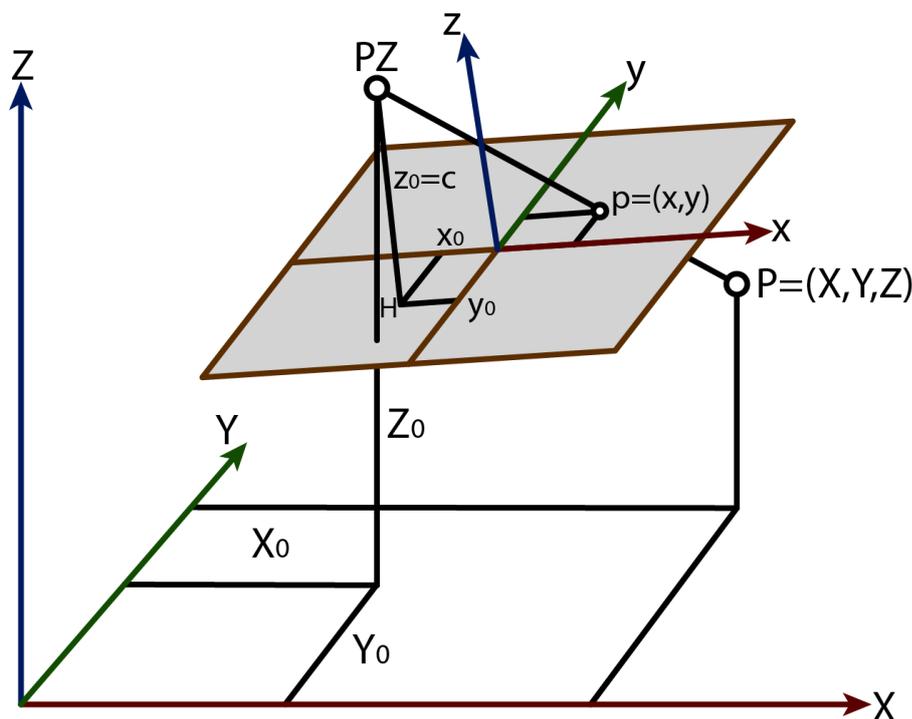


Abbildung 10: Geometrie der Zentralprojektion

Der Zusammenhang zwischen einem Bildpunkt \mathbf{p} und dem korrespondierenden Objektpunkt \mathbf{P} folgt aus dem Modell der Zentralprojektion zu

$$\begin{pmatrix} x - x_0 \\ y - y_0 \\ -c \end{pmatrix} = \frac{1}{m} \cdot \mathbf{R}(\omega_0, \varphi_0, \kappa_0)^T \cdot \begin{pmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{pmatrix} \quad (65)$$

bzw.

$$\begin{pmatrix} x - x_0 \\ y - y_0 \\ -c \end{pmatrix} = \frac{1}{m} \cdot \begin{pmatrix} r_{11} \cdot (X - X_0) + r_{21} \cdot (Y - Y_0) + r_{31} \cdot (Z - Z_0) \\ r_{12} \cdot (X - X_0) + r_{22} \cdot (Y - Y_0) + r_{32} \cdot (Z - Z_0) \\ r_{13} \cdot (X - X_0) + r_{23} \cdot (Y - Y_0) + r_{33} \cdot (Z - Z_0) \end{pmatrix} \quad (66)$$

wobei r_{jk} die Einträge der Rotationsmatrix $\mathbf{R}(\omega_0, \varphi_0, \kappa_0)$ und m der Maßstabsfaktor zwischen den Vektoren

$$\begin{pmatrix} x - x_0 \\ y - y_0 \\ -c \end{pmatrix} \quad \text{und} \quad \begin{pmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{pmatrix}$$

ist, der für jedes zusammengehörige Paar aus Objekt- und Bildpunkten variiert. Um m aus den Gleichungen zu eliminieren, werden die ersten beiden Zeilen jeweils durch die dritte Zeile geteilt. Nach Umformung erhält man die Kollinearitätsgleichungen

$$x = x_0 - c \cdot \frac{r_{11} \cdot (X - X_0) + r_{21} \cdot (Y - Y_0) + r_{31} \cdot (Z - Z_0)}{r_{31} \cdot (X - X_0) + r_{23} \cdot (Y - Y_0) + r_{33} \cdot (Z - Z_0)} \quad (67)$$

$$y = y_0 - c \cdot \frac{r_{12} \cdot (X - X_0) + r_{22} \cdot (Y - Y_0) + r_{32} \cdot (Z - Z_0)}{r_{31} \cdot (X - X_0) + r_{23} \cdot (Y - Y_0) + r_{33} \cdot (Z - Z_0)}, \quad (68)$$

mit denen ein Objektpunkt direkt in die Bildebene projiziert werden kann, sofern die innere und äußere Orientierung der Kamera bekannt sind. Für eine umfassende Erläuterung der Zentralprojektion sowie der Kollinearitätsgleichungen wird auf [Kraus et al., 2004] verwiesen.

2.4.2 Verzeichnung

Herkömmliche Digitalkameras, wie auch die in dieser Arbeit eingesetzte, weichen in der Realität von dem zuvor beschriebenen Modell einer Lochkamera ab. Die auf den Sensor fallenden Lichtstrahlen werden von mehreren Linsen gebrochen. Dies führt zu einer Abweichung zwischen dem Bildpunkt entsprechend der Zentralprojektion und dem wahren Bildpunkt. Diese Abweichung wird Verzeichnung genannt. Um die Verzeichnung zu berechnen und sie somit in den Kollinearitätsgleichungen zu berücksichtigen, existieren verschiedene Modelle. Gemeinsam haben die meisten, dass radiale und tangentielle Verzeichnung unterschieden wird, wobei die radiale Verzeichnung den Bildpunkt in Richtung

des Bildhauptpunktes oder von diesem weg verschiebt. Tangentiale Verzeichnung ist die Komponente senkrecht darauf. Das hier vorgestellte und später genutzte Modell folgt der Konvention, die in `OpenCV` Anwendung findet. Um die Bildkoordinaten eines Objektpunktes $(X, Y, Z)^T$ zu ermitteln, wird zunächst der Vektor, der vom Projektionszentrum auf den Objektpunkt zeigt, in das Koordinatensystem der Kamera rotiert. Es resultiert der Vektor

$$\begin{pmatrix} x^c \\ y^c \\ z^c \end{pmatrix} = \mathbf{R}^T(\omega_0, \varphi_0, \kappa_0) \cdot \begin{pmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{pmatrix}, \quad (69)$$

aus dem die reduzierten Koordinaten

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \frac{x^c}{z^c} \\ \frac{y^c}{z^c} \end{pmatrix} \quad (70)$$

berechnet werden. Aus diesen folgen die verzeichnungsfreien, reduzierten Koordinaten zu

$$x'' = x' \cdot (1 + k_1 \cdot r^2 + k_2 \cdot r^4 + k_3 \cdot r^6) + 2p_1 \cdot x' \cdot y' + p_2 \cdot (r^2 + 2x'^2) \quad (71)$$

$$y'' = y' \cdot (1 + k_1 \cdot r^2 + k_2 \cdot r^4 + k_3 \cdot r^6) + p_1 \cdot (r^2 + 2y'^2) + 2p_2 \cdot x' \cdot y' \quad (72)$$

mit

$$r^2 = x'^2 + y'^2. \quad (73)$$

Die Parameter der Verzeichnung lauten folglich

$$\mathbf{C}_{dist} = (k_1, k_2, k_3, p_1, p_2). \quad (74)$$

Wobei k_1 , k_2 und k_3 der radialen Verzeichnung zugeordnet werden und p_1 sowie p_2 der tangentialen Verzeichnung. Weiterhin muss beachtet werden, dass in der Konvention, die in `OpenCV` eingesetzt wird, die Kammerkonstante entsprechend den Achsen in c_x und c_y unterschieden wird. Die Berechnung der Bildkoordinaten folgt zu

$$x = -c_x \cdot x'' + x_0 \quad (75)$$

$$y = -c_y \cdot y'' + y_0 \quad (76)$$

wohingegen das in dieser Arbeit genutzte Modell der Zentralprojektion die gemeinsame Kammerkonstante c nutzt. Hieraus folgen die Gleichungen:

$$x = -c \cdot x''' + x_0 \quad (77)$$

$$y = -c \cdot y''' + y_0. \quad (78)$$

Nun wird $c = c_x$ definiert, woraus

$$x'' = x''' \quad (79)$$

folgt. Durch Gleichsetzen folgt aus den Gleichungen 76 und 78, dass

$$y''' = y'' \cdot \frac{c_y}{c_x} = y'' \cdot m_y \quad (80)$$

gilt, wobei m_y der Maßstabsfaktor der y-Achse ist.

Mit den Gleichungen 69 - 80 kann ein Objektpunkt unter Berücksichtigung der Verzeichnung in die Bildebene projiziert werden. Bei der Projektion eines beobachteten Bildpunktes in den Objektraum, muss die Verzeichnung ebenfalls berücksichtigt werden. Seien $\mathbf{b}''' = (x''', y''')^T$ die reduzierten, beobachteten Bildkoordinaten, $\mathbf{b}' = (x', y')^T$ die korrigierten Bildkoordinaten und $\Delta\mathbf{b}_{\mathbf{b}'}$ die Korrektur an der Position \mathbf{b}' . Da \mathbf{b}' zunächst unbekannt ist und nicht direkt berechnet werden kann, erfolgt die Berechnung von \mathbf{b}' iterativ. Hierbei wird angenommen, dass die Korrektur an der Position \mathbf{b}' in etwa der Korrektur an der Stelle \mathbf{b}''' entspricht:

$$\Delta\mathbf{b}_{\mathbf{b}'} \approx \Delta\mathbf{b}_{\mathbf{b}'''} \quad (81)$$

$\Delta\mathbf{b}_{\mathbf{b}'''}$ wird mit den zuvor aufgeführten Gleichung berechnet. Mit dieser Korrektur wird

$$\tilde{\mathbf{b}}' = \mathbf{b}''' - \Delta\mathbf{b}_{\mathbf{b}'''} \quad (82)$$

berechnet, die genäherte Position des korrigierten, reduzierten Bildpunktes. Nun wird iterativ die Korrektur $\Delta\mathbf{b}_{\tilde{\mathbf{b}}'}$ an der genäherten Position berechnet und von den beobachteten, reduzierten Koordinaten subtrahiert:

$$\tilde{\tilde{\mathbf{b}}}' = \mathbf{b}_{\mathbf{b}'''} + \Delta\mathbf{b}_{\tilde{\mathbf{b}}'}. \quad (83)$$

In der Regel reichen wenige Iterationen aus, um \mathbf{b}' ausreichend genau anzunähern. Als Abbruchkriterium kann die Norm der Änderung von $\tilde{\mathbf{b}}'$ gegenüber der vorangegangenen Iteration genutzt werden, indem für diese ein Schwellwert festgelegt wird.

Abbildung 11 zeigt die Auswirkung der Verzeichnung. Links ist das aufgenommene Bild dargestellt, rechts das korrigierte Bild. Hierzu wurde für jeden Bildpunkt die korrigierte Position berechnet, und der entsprechende Farbwert gesetzt. Weitere Informationen zur Kameraverzeichnung findet man in [Szeliski, 2011]. Das hier vorgestellte Modell für die radiale und tangential Verzeichnung ist in [ope, 2014] dokumentiert.

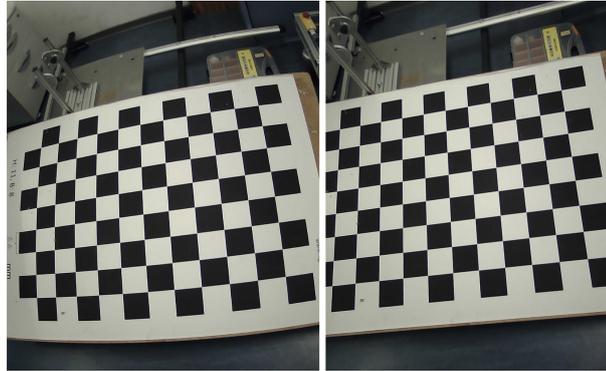


Abbildung 11: Bild vor (links) und nach (rechts) Korrektur der Verzeichnung

2.4.3 Räumlicher Rückwärtsschnitt

Der räumliche Rückwärtsschnitt (RRWS) ist ein Verfahren, mit welchem die äußere Orientierung eines Bildes auf Basis von Passpunkten berechnet werden kann. Für diese Aufgabe existieren verschiedene Algorithmen. Hier vorgestellt und in dieser Arbeit genutzt wird ein iterativer Algorithmus, der die Summe der quadratischen Rückprojektionsfehler minimiert. Dies kann in einem GMM dargestellt werden. Der Beobachtungsvektor beinhaltet die korrigierten Bildkoordinaten $\mathbf{p}_i = (x_i, y_i)$ nach Korrektur der Verzeichnung der bekannten Passpunkte $\mathbf{P}_i = (X_i, Y_i, Z_i)$. Den funktionalen Zusammenhang stellen die Kollinearitätsgleichungen dar. Die Verzeichnung wird durch die Funktionen

$$x = corr_x(x_c) \quad (84)$$

$$y = corr_y(y_c) \quad (85)$$

dargestellt. Für jeden im Bild detektierten Passpunkt folgen die Zusammenhänge

$$x_i + v_{x_i} = corr_x(x_0 - c \cdot \frac{r_{11} \cdot (X_i - X_0) + r_{21} \cdot (Y_i - Y_0) + r_{31} \cdot (Z_i - Z_0)}{r_{31} \cdot (X_i - X_0) + r_{23} \cdot (Y_i - Y_0) + r_{33} \cdot (Z_i - Z_0)}) \quad (86)$$

und

$$y_i + v_{y_i} = corr_y(y_0 - c \cdot \frac{r_{12} \cdot (X_i - X_0) + r_{22} \cdot (Y_i - Y_0) + r_{32} \cdot (Z_i - Z_0)}{r_{31} \cdot (X_i - X_0) + r_{23} \cdot (Y_i - Y_0) + r_{33} \cdot (Z_i - Z_0)}) \quad (87)$$

Hierbei ist r_{jk} der entsprechende Eintrag in der Rotationsmatrix $\mathbf{R}(\omega_0, \varphi_0, \kappa_0)$ welche die Lage der Kamera im Objektraum beschreibt. Die zu ermittelnden Unbekannten sind die Parameter der äußeren Orientierung

$$\mathbf{x} = (X_0, Y_0, Z_0, \omega_0, \varphi_0, \kappa_0)^T. \quad (88)$$

Die aus der Kamerakalibrierung resultierenden Parameter der inneren Orientierung werden als konstant angenommen, wobei diese ebenfalls berechnet werden können. Im Fall, dass nur die sechs Parameter der äußeren Orientierung ermittelt werden, folgt die Redundanz zu

$$f = 2 * n - 6 \quad (89)$$

wobei n die Anzahl der Passpunkte ist. Hieraus wird ersichtlich, dass mindestens drei im Bild detektierte Passpunkte notwendig sind um die äußere Orientierung zu bestimmen. Bei mehr als drei Passpunkten ist das System redundant, womit eine Aussage über die innere Genauigkeit getroffen werden kann. Anzumerken ist, dass das Modell nicht linear ist, wodurch die ungefähre Pose der Kamera bekannt sein muss. Diese kann durch lineare Ansätze wie der in [Gao et al., 2003] vorgestellten Methode ermittelt oder, falls möglich, manuell angegeben werden. Die Berechnung folgt den Gleichungen aus Abschnitt 2.3. Für eine Übersicht der Algorithmen, die zur Bestimmung der Kamerapose genutzt werden können, wird auf [Szeliski, 2011] verwiesen.

2.4.4 Bündelblockausgleichung

Die Bündelblockausgleichung (BBA) ist ein Verfahren zur simultanen Bestimmung der Orientierung aller Bilder eines Blockverbandes und der Objektkoordinaten aller unbekannt Objekte in einem Guss. Das Verfahren beruht auf der Methode der kleinsten Quadrate. Als Beobachtungen gehen in die BBA zunächst die Bildkoordinaten von Pass- und Verknüpfungspunkten ein. Passpunkte sind Objekte, deren Objektkoordinaten durch anderweitige Messungen bestimmt werden. Die Objektkoordinaten der Passpunkte gehen ebenfalls als Beobachtungen ein. Verknüpfungspunkte sind Objekte mit unbekannt Objektkoordinaten. Als zusätzliche Beobachtungen können auch Distanzen zwischen Objektpunkten eingeführt werden. Neben den Beobachtungen werden die Korrespondenzen zwischen gleichen Pass- und Verknüpfungspunkten in den verschiedenen Bildern benötigt. Die mathematische Modellierung erfolgt in einem nicht linearen GMM. Der Beobachtungsvektor beinhaltet die zuvor genannten Beobachtungen. Der Vektor der Unbekannten beinhaltet die äußeren Orientierungen der Bilder und die Objektkoordinaten der Verknüpfungspunkte. Zusätzlich können auch die innere Orientierung der Kamera sowie die Parameter der Verzerrung geschätzt werden (vgl. Ab-

schnitt 2.4.5). Benötigt werden Näherungswerte für die äußere Orientierungen der Bilder, die mit der in [Gao et al., 2003] vorgestellten Methode ermittelt werden können. Näherungswerte für die Verknüpfungspunkte können durch Triangulation ermittelt werden (vgl. [McGlone et al., 2004]).

Der funktionale Zusammenhang zwischen den Objekt- und Bildkoordinaten entspricht den Gleichungen 86 und 87. Für jede beobachtete Distanz d_{ij} zwischen den Objektpunkten $\mathbf{P}_i = (X_i, Y_i, Z_i)^T$ und $\mathbf{P}_j = (X_j, Y_j, Z_j)^T$ lautet der Zusammenhang

$$d_{ij} + v_{d_{ij}} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2 + (Z_i - Z_j)^2}. \quad (90)$$

Bei der Bündelblockausgleichung sind die Koordinaten der Passpunkte sowohl Teil des Unbekanntenvektors als auch des Beobachtungsvektors. Dies ist notwendig, da die Koordinaten hierdurch einerseits verbessert werden können und es anderenfalls nicht möglich wäre einen expliziten Zusammenhang zwischen den Beobachtungen und den Unbekannten herzustellen. Mindestens drei der Objektpunkte müssen gleichzeitig Passpunkte mit bekannten Koordinaten sein, um auch die Skalierung der rekonstruierten Szene zu ermitteln. Eine beispielhafte geometrische Anordnung einer BBA zeigt Abbildung 12.

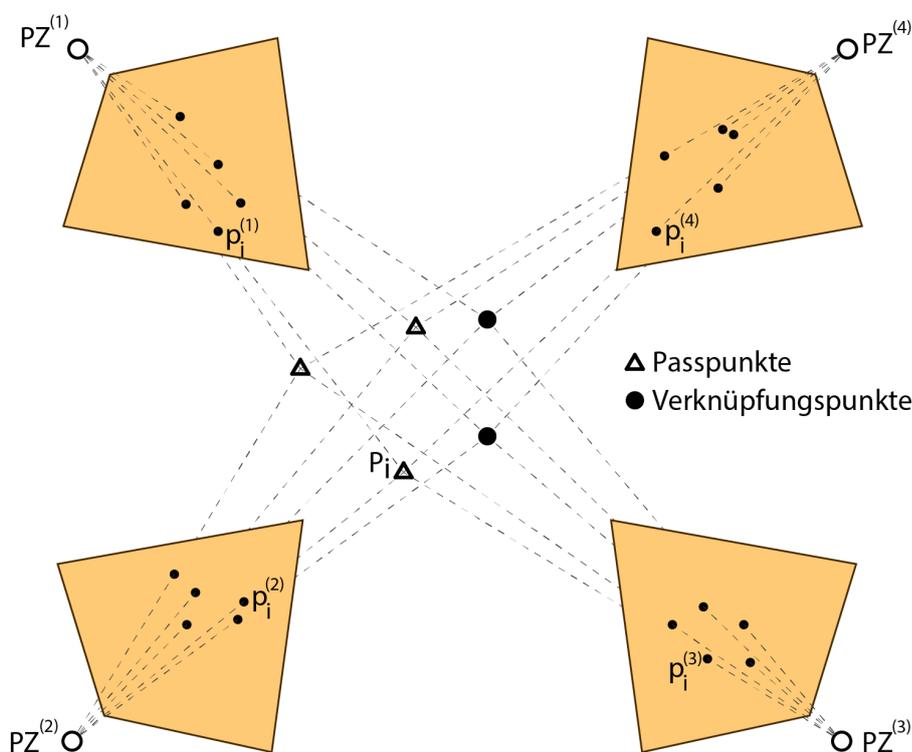


Abbildung 12: Beispielhafte Geometrie einer Bündelblockausgleichung

Dargestellt ist eine Szene mit fünf Objektpunkten (drei Passpunkte und zwei Verknüpfungspunkte), die aus vier Perspektiven fotografiert wird. Eingezeichnet sind Objekt-, und Bildpunkte sowie die Projektionszentren, die Bildstrahlen sowie (in orange) die Bildebenen. Die Bündelblockausgleichung wird ausführlich in [Kraus, 2004] beschrieben.

2.4.5 Kamerakalibrierung

Um die Parameter der inneren Orientierung sowie die der Verzeichnung einer Kamera zu bestimmen, wird diese kalibriert. Hierfür existieren mehrere Verfahren. In dieser Arbeit erfolgt das Kalibrieren auf Basis von Bildern eines ebenen Schachbrettmusters mit bekannter Anzahl und Größe der Kacheln. Ein solches Kalibrieremuster zeigt Abbildung 11. Vorteil dieses Musters ist, dass die Ecken der Kanten automatisch, präzise und robust im Bild detektiert werden können. Um die Ecken im Objektraum zu beschreiben, wird das Objektkoordinatensystem in eine Ecke des Musters gelegt, so dass die X- und Y-Achse den Kanten der Kacheln folgen. Die Z-Achse steht senkrecht auf der Ebene, wodurch alle Z-Koordinaten der Eckpunkte zu null werden. Die Punkte, deren Koordinaten im Objektraum bekannt sind und die als fehlerfrei angenommen werden, werden Passpunkte genannt. Um die gesuchten Parameter zu ermitteln wird das Muster aus verschiedenen Perspektiven aufgenommen, wobei mindestens zwei Aufnahmen mit unterschiedlichen Orientierungen notwendig sind. Die Genauigkeit, mit welcher die Parameter der inneren Orientierung und der Verzeichnung ermittelt werden, steigt mit der Anzahl der Aufnahmen. Weiterhin hängt die erreichbare Genauigkeit von der Wahl der Perspektiven ab. Ferner ist darauf zu achten, dass das Muster in jedem Bereich des Bildes mindestens in einer Aufnahme abgebildet wird. Eine Beschreibung günstiger Aufnahmekonfigurationen der Bilder findet man in [Luhmann, 2000]. Im Anhang befindet sich eine Abbildung, die eine beispielhafte Anordnung der Bilder zeigt.

Es folgt die Messung, in der die Eckpunkte der Kacheln automatisch detektiert werden. Die Eckpunkte des Musters ergeben sich aus den Berührungspunkten der detektierten Kacheln. Die so ermittelten Bildkoordinaten dienen als Näherungswerte für eine anschließende Feinmessung. In dieser werden, ausgehend von den genäherten Bildkoordinaten der Eckpunkte, die maximalen Gradienten des entsprechenden Grauwertbildes gesucht. Mit diesem Verfahren können die Bildkoordinaten der Eckpunkte im Subpixel-Bereich bestimmt werden. Aus den Informationen über die Nachbarschaften der Kacheln lassen sich die detektierten Eckpunkte den Objektkoordinaten der Eckpunkte zuordnen. Auf dieser Grundlage werden nun die Parameter der inneren Orientierung sowie die der Verzeichnung, aber auch die äußere Orientierung jedes Bildes so gewählt, dass die Rückprojektionsfehler minimal werden. Der Rückprojektionsfehler bezeichnet die Abweichungen zwischen den detektierten Bildkoordinaten jeder Ecke und den ins Bild zurückprojizierten Koordina-

ten der Eckpunkte. Dieses Optimierungsproblem wird in einer Bündelblockausgleichung gelöst, in welcher die Objekt- und Bildkoordinaten der Eckpunkte als Beobachtungen in Form von Passpunkten eingehen. Der Unbekanntenvektor beinhaltet die Parameter der inneren Orientierung sowie die der Verzeichnung und die äußeren Orientierungen der Bilder. Als Implementierung der Kamerakalibrierung wird `OpenCV` genutzt. Die entsprechenden Module sind in [ope, 2014] dokumentiert.

3 Lösungsansatz

In diesem Kapitel wird der Lösungsansatz vorgestellt. Hierzu wird das Problem zunächst geometrisch betrachtet, um alle für die Zielstellung relevanten Körper sowie deren Pose in einem gemeinsamen Koordinatensystem zu definieren. Im Anschluss wird das Hauptproblem, die Rotation der Kugel zu verfolgen, in Teilprobleme zerlegt und der Ablauf des entwickelten Programmes dargelegt, wobei die komplexeren Teilprobleme genauer in eigenen Kapiteln eingehend erläutert werden.

3.1 Geometrie des Eingabegerätes

Für die geometrische Betrachtung wird das Eingabegerät in die Komponenten Kugel und Rahmen unterteilt. Variablen, die sich auf die Kugel beziehen, werden mit dem Index K versehen. Es wird weiterhin das feste, übergeordnete Koordinatensystem \mathbf{KS}_0 definiert, welches seinen Ursprung im Mittelpunkt der Kugel hat. Die Ausrichtung der Achsen wird durch die Marker auf dem Rahmen definiert. Die Marker bilden die X-Y-Ebene, wobei die Y-Achse ungefähr vertikal ausgerichtet ist. Die Z-Achse steht senkrecht auf der X-Y-Ebene. Weiterhin relevant ist das Sensorkoordinatensystem \mathbf{KS}_S der Kamera, in welchem die Koordinaten von Bildpunkten dargestellt werden. Die x- und y-Achse des \mathbf{KS}_S spannen die Bildebene auf, wobei der Bildpunkt $(0,0)$ in der oberen linken Ecke des Bildes liegt. Die z-Achse des \mathbf{KS}_S zeigt in Blickrichtung der Kamera. Dies führt im Gegensatz zum Lochkameramodell aus Abschnitt 2.4 zu einer negativen Kammerkonstante im \mathbf{KS}_S . Die äußere Orientierung der Kamera wird durch den Vektor $\mathbf{PZ} = (X_0, Y_0, Z_0)^T$ sowie die Lagewinkel $(\omega_0, \varphi_0, \kappa_0)^T$ wiedergegeben. Letztere beschreiben die Verdrehung des \mathbf{KS}_S gegenüber dem \mathbf{KS}_0 . Abschließend wird das an die Kugel gebundene Koordinatensystem \mathbf{KS}_K eingeführt, welches seinen Ursprung ebenfalls im Zentrum der Kugel hat. Die Lagewinkel der Kugel $(\omega_K, \varphi_K, \kappa_K)$ parametrisieren die Rotation des \mathbf{KS}_K gegenüber dem \mathbf{KS}_0 . In Abbildung 13 sind die Koordinatensysteme sowie die Bildebene und das Projektionszentrum der Kamera, sowie die Parameter der inneren Orientierung abgebildet. Zusätzlich sind die Marker auf der Kugel (Blob-Marker) und dem Rahmen (Passpunkte) dargestellt. Zur besseren Unterscheidung werden Koordinaten im \mathbf{KS}_0 mit Großbuchstaben notiert, die Bildkoordinaten im \mathbf{KS}_S mit Kleinbuchstaben.

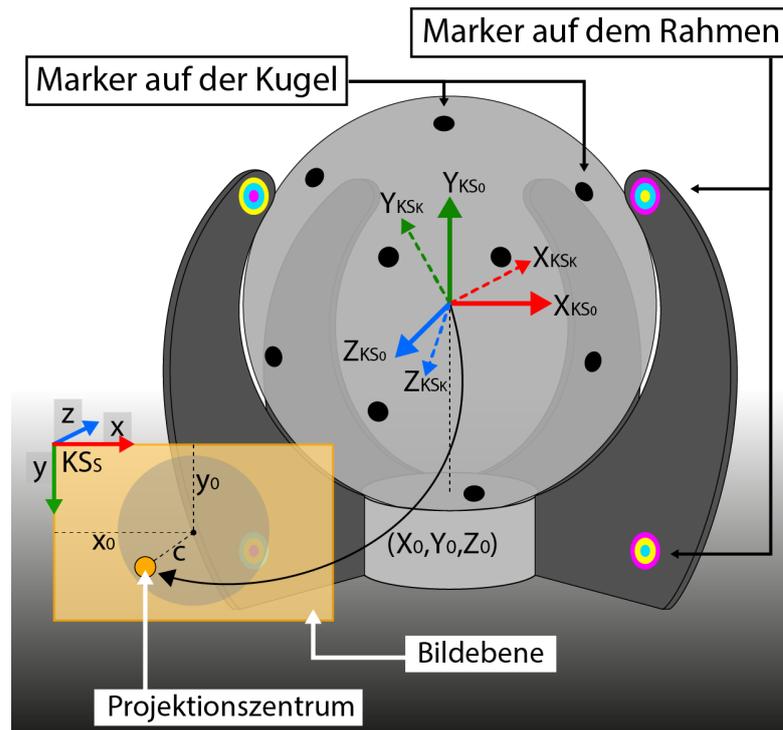


Abbildung 13: Aufbau des Eingabegerätes und Koordinatensysteme

3.2 Ablauf

Der vollständige Ablauf des entwickelten Verfahrens lässt sich in drei Phasen unterteilen: Kalibrierung, Initialisierung und Nutzung. In der Kalibrierungsphase wird die Kamera kalibriert und die Systemkalibrierung durchgeführt. In der Systemkalibrierung wird die Lage des Kugelmittelpunktes bezogen auf den Rahmen sowie der Kugelradius und die Objektkoordinaten der Marker auf dem Rahmen im \mathbf{KS}_0 ermittelt. Die Kalibrierung der Kamera erfolgt mit der in Kapitel 2.4.5 vorgestellten Methode. Das Ermitteln der Kugelparameter erfolgt in zwei Schritten. Im ersten Schritt wird das Eingabegerät von mehreren Standpunkten aus aufgenommen, die Markerzentren der sichtbaren Marker in jedem Bild detektiert, die selben Markerzentren in den verschiedenen Bildern identifiziert und zuletzt alle Unbekannten in einer Bündelblockausgleichung bestimmt. Ermittelt werden hierbei die Objektkoordinaten und VKM der Markerzentren auf dem Rahmen sowie die Objektkoordinaten und VKM der Markerzentren auf der Kugel, die in mindestens zwei Bildern detektiert werden können. Gleichzeitig werden die äußere Orientierung der Bilder sowie die entsprechenden VKM bestimmt. Die farbkodierten Marker auf dem Rahmen fungieren als Passpunkte und definieren das Objektkoordinatensystem. Die innere Orientierung wird in der BBA als konstant angenommen. Die Objektkoordinaten der Markerzentren auf der Kugel werden im zweiten Schritt genutzt um eine ausgleichende Kugel in einem GHM zu schätzen, aus der die Kugelparameter folgen. Zuletzt wird der ermittelte Mittelpunkt der

Kugel genutzt um die reduzierten Objektkoordinaten der Passpunkte zu bestimmen, das heißt deren Koordinaten im \mathbf{KS}_0 . Die Kalibrierung wird einmalig durchgeführt und muss erst wiederholt werden, wenn eine andere Kamera verwendet wird. Die Kalibrierungsphase wird in Abschnitt 3.3 erläutert.

Die zweite Phase ist die Initialisierung, in welcher die Pose der Kamera, das heißt deren äußere Orientierung im Bezug auf das \mathbf{KS}_0 ermittelt wird. Hierzu werden die Zentren der Marker auf dem Rahmen detektiert und der RRWS angewendet. Die Initialisierung ist folglich immer notwendig, wenn sich die Position oder Orientierung der Kamera gegenüber dem Rahmen ändert bzw. wenn dies nicht ausgeschlossen werden kann. Praktisch wird die Initialisierungsphase bei jedem Programmstart durchgeführt und kann auch während der Nutzungsphase wiederholt werden. Die Initialisierungsphase ist ausführlich in Abschnitt 3.4 beschrieben.

Die letzte Phase ist die Nutzungsphase, in welcher der Video-Stream der Kamera genutzt wird, um die Lagewinkel und Drehraten der Kugel zu bestimmen und die Ausgaben für die Zielsoftware zu generieren. Hierzu wird ein EKF eingesetzt, der die Objektkoordinaten der Markerzentren auf der Kugel auswertet, die sich aus dem Schnitt der Bildstrahlen der detektierten Markerzentren und der Kugel ergeben. Die Erläuterung der Nutzungsphase folgt in Abschnitt 3.5.

3.3 Kalibrierung

In der Kalibrierungsphase wird zunächst die Kamera kalibriert. Hierzu wird ein Schachbrettmusters mit bekannten Ausmaßen aus mehreren Perspektiven aufgenommen. Die Bilder werden anschließend mit dem in Abschnitt 2.4.5 vorgestellten Verfahren ausgewertet.¹ Dieses Verfahren liefert die Parameter der inneren und äußeren Orientierung der Kamera, die radialen und tangentialen Verzeichnungparameter sowie die Maßstabskorrektur der y-Achse (siehe Abbildung 14).

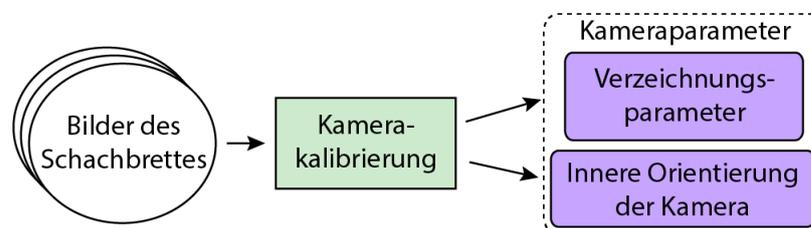


Abbildung 14: Ablauf der Kamerakalibrierung

Die zweite Aufgabe in der Kalibrierungsphase ist die Systemkalibrierung. Diese dient der

¹Als Implementierung wird das OpenCV-Modul `calibrateCamera` verwendet

Die Objektkoordinaten der Blob-Marker und deren VKM werden zuletzt verwendet, um eine ausgleichende Kugel zu schätzen. Diese Schätzung wird in Abschnitt 3.7 erläutert. Es resultieren die Parameter der Kugel $\mathbf{K} = (X_K, Y_K, Z_K, R_K)^T$ (Zentrum und Radius) sowie deren VKM $\Sigma_{\mathbf{K}\mathbf{K}}$. Zuletzt werden die Koordinaten der h Passpunkte $\mathbf{H}_1, \dots, \mathbf{H}_h$, die nach der BBA im $\mathbf{K}\mathbf{S}_B$ vorliegen, in das $\mathbf{K}\mathbf{S}_0$ verschoben. Dies erfolgt durch Subtrahieren der Koordinaten des Kugelzentrums $\mathbf{P}_K = (X_K, Y_K, Z_K)^T$ zu

$$\mathbf{H}_i^{(0)} = \mathbf{H}_i - \mathbf{P}_K. \quad \text{mit: } i = 1, \dots, h \quad (91)$$

Entsprechend wird das Zentrum der Kugel auf

$$\mathbf{P}_K^{(0)} = \mathbf{P}_K - \mathbf{P}_K = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad (92)$$

gesetzt. Den schematischen Ablauf der letzten Teilaufgabe zeigt Abbildung 16.

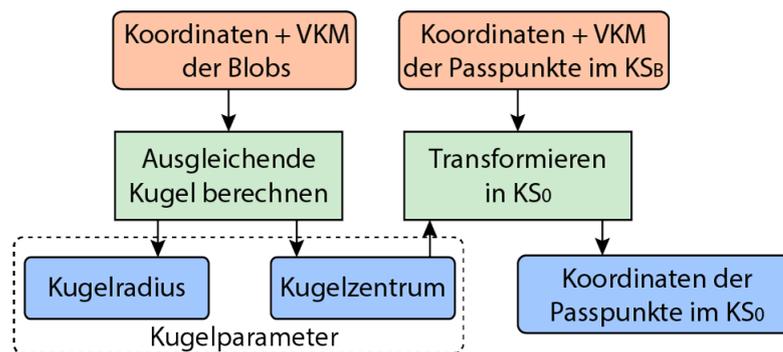


Abbildung 16: Ermitteln der Kugelparameter und Passpunktkoordinaten im $\mathbf{K}\mathbf{S}_0$

3.3.1 Korrespondenzen der detektierten Marker

Für die Durchführung der Bündelblockausgleichung werden die Korrespondenzen zwischen den detektierten Markerzentren in allen Bildern benötigt. Die Punktnummern der auf dem Rahmen angebrachten Marker ergeben sich aus der Farbkodierung (vgl. Abschnitt 3.6.2). Im Folgenden wird erläutert, wie die Nummern der Marker auf der Kugel ermittelt werden. Hierzu wird zunächst der in [Gao et al., 2003] vorgestellte Algorithmus genutzt, um die äußere Orientierung jeder Kameraposition zu ermitteln, wofür die Koordinaten der Marker auf dem Rahmen genutzt werden.³ Für jedes im Bild $b = 1, \dots, n$ detektierte Markerzentrum $m_{i,b}$ mit $i = 1, \dots, M_b$, wobei M_b die Anzahl im Bild b detektierten Markerzentren ist, kann der Bildstrahl $s_{i,b}$ berechnet werden (vgl. Abschnitt 3.8).

³Als Implementierung wird das `OpenCV`-Modul `solvePnP` verwendet

Im Anschluss wird für jedes Paar von Bildstrahlen $s_{i,b}$ und $s_{j,a}$ der unterschiedlichen Bilder b und a der Objektpunkt $P_{i,b,j,a}$ berechnet, der die minimale Distanz $d_{i,b,j,a}$ zu beiden Bildstrahlen aufweist. Für diese Distanz wird eine obere Grenze d_{max} festgelegt. Falls

$$d_{i,b,j,a} < d_{max} \quad (93)$$

erfüllt ist, wird $P_{i,b,j,a}$ der Menge B hinzugefügt, wobei B als leere Menge initialisiert wird. Hierbei wird geprüft, ob es noch keinen weiteren Objektpunkt $P_k \in B$ gibt, für den

$$|P_{i,b,j,a} - P_k|_2 < d_{max} \quad (94)$$

gilt, um die mehrfache Bestimmung möglicher Objektkoordinaten der Markerzentren auszuschließen. Zuletzt werden alle Bildstrahlen identifiziert, deren maximale Distanz zu P_k den Grenzwert d_{max} nicht überschreitet und dem korrespondierenden Markerzentrum im Bild die Nummer k vergeben.

3.4 Initialisierung

Die Initialisierungsphase erfolgt, sobald die Kamera für die Nutzung aufgestellt wurde, so dass sie das Eingabegerät abbilden kann, und das erste Bild der Videosequenz aufgenommen wurde. Die Initialisierung wird bei jedem Start des Programmes wiederholt. Es gilt nun, die äußere Orientierung der Kamera zu ermitteln. Hierzu werden die Zentren der Passpunkte in den ersten n_b Bildern der Videosequenz detektiert. Der RRWS kann schon nach dem ersten Bild durchgeführt werden, jedoch steigt die Anzahl der Beobachtungen bei vier Passpunkten mit einer zunehmenden Anzahl von Bildern bei vier Passpunkten auf $n = 2 \cdot 4 \cdot n_b$ (zwei Bildkoordinaten je Passpunkt und Bild) und somit auch die Redundanz der Ausgleichung. Es ist zu erwarten, dass die erzielte Genauigkeit der äußeren Orientierung durch die steigende Anzahl der Beobachtungen ebenfalls verbessert wird. Die Bildpunkte werden aufgrund der Verzeichnung korrigiert, wie in Abschnitt 2.4.5 beschrieben. Die eindeutige Nummer jedes detektierten Markers wird anhand der Farbkodierung bestimmt. Nun kann der RRWS durchgeführt werden. Aus dem RRWS folgt die äußere Orientierung der Kamera sowie die dazugehörige VKM Σ_{00} . Die ungefähre Pose der Kamera, die in Form von Näherungswerten für den RRWS erforderlich ist, ergibt sich aus dem Aufbau der Konstruktion und der Kamera. Abbildung 17 stellt den Ablauf der Initialisierung dar.

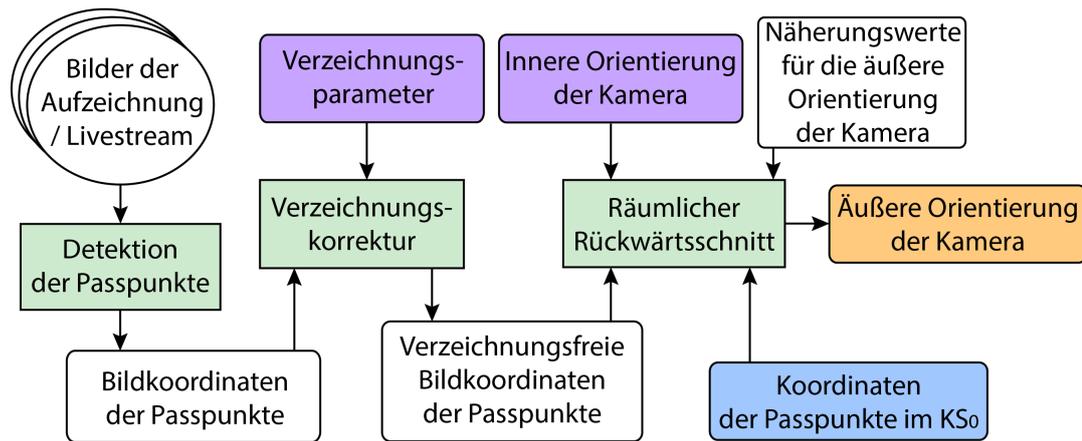


Abbildung 17: Ablauf der Initialisierungsphase

3.5 Nutzung

Sobald die Pose der Kamera durch die Initialisierung ermittelt wurde, kann die Nutzungsphase beginnen. In dieser wird die Videosequenz fortlaufend ausgewertet, um die Lage der Kugel zu verfolgen. Die folgenden Schritte werden für jedes eingehende Bild des Streams ausgeführt. Zunächst werden die auf der Kugel angebrachten Marker im Bild detektiert und um die Verzerrung korrigiert. Mit welcher Genauigkeit die Markerzentren detektiert werden, beschreibt, deren VKM $\Sigma_{\mathbf{bb}}$, wobei diese Matrix mit der Standardabweichung σ_b der Detektion der Markerzentren wie folgt zusammenhängt:

$$\Sigma_{\mathbf{bb}} = \begin{pmatrix} \sigma_b^2 & 0 \\ 0 & \sigma_b^2 \end{pmatrix}. \quad (95)$$

Da zu diesem Zeitpunkt sowohl die Kugelparameter als auch die äußere Orientierung der Kamera inklusive der zugehörigen VKM bekannt sind, können die um die Verzerrung korrigierten Bildkoordinaten der Markerzentern genutzt werden, um die korrespondierenden Objektkoordinaten der Markerzentern zu berechnen. Diese Berechnung besteht im räumlichen Schnitt des Bildstrahles, der sich aus dem Markerzentrum ergibt, und der Kugel. Diese Berechnung wird in Abschnitt 3.8 erläutert. Im Kontext dieser Berechnung wird für jeden Marker die entsprechende VKM der Objektkoordinaten durch Fehlerfortpflanzung ermittelt. Die Objektkoordinaten und VKM der Markerzentern einer Epoche bzw. eines Bildes der Videosequenz werden fortlaufend in einem EKF ausgewertet. Das Filter speichert die Koordinaten der Markerzentern im Koordinatensystem der Kugel. Die Modellierung des EKF sowie dessen Initialisierung und die Datenassoziation wird in Abschnitt 3.9 erläutert. Der EKF liefert die Lagewinkel der Kugel sowie deren zeitliche Änderung (Drehraten). Diese werden zuletzt genutzt um eine Mausbewegung zu berech-

nen, die von der beispielhaft vorgestellten Zielsoftware anstelle der Ausgaben des Trackball genutzt werden kann. die berechnung der Mausbewegung wird ebenfalls in Abschnitt 3.9 behandelt. Der schematische Ablauf der Nutzungsphase ist in Abbildung 18 dargestellt.

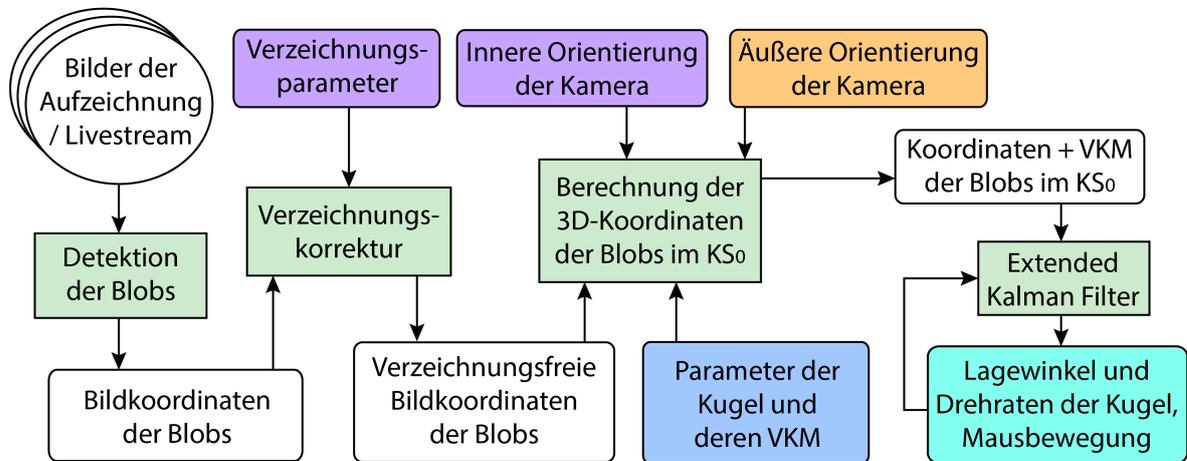


Abbildung 18: Ablauf der Nutzungsphase

3.6 Marker und deren Detektion

Wie aus dem vorangegangenen Abschnitt folgt, werden für die entwickelte Anwendung zwei Typen von Markern verwendet: Zum einen kreisförmige Marker auf der Kugel, zum anderen farbige Ringe auf dem Rahmen. Eine beispielhafte Anordnung der Marker ist in Abbildung 19 dargestellt.

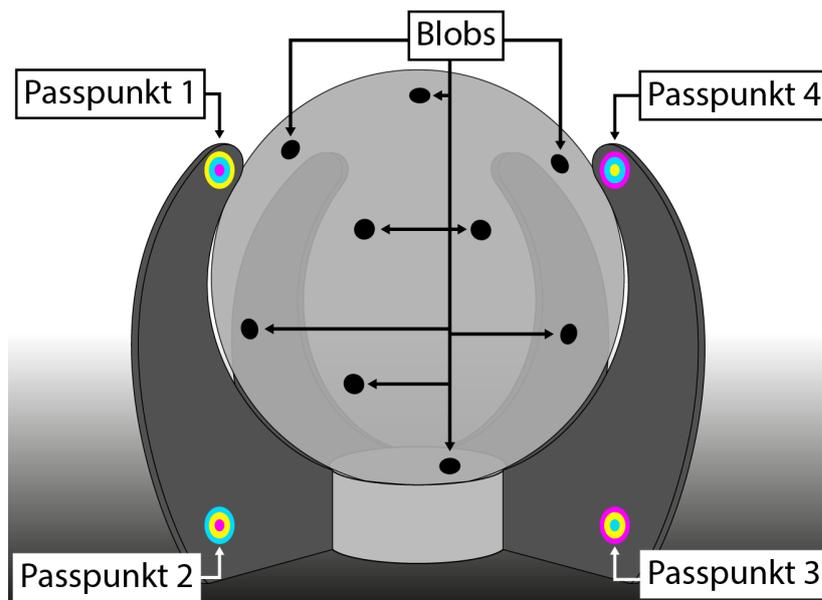


Abbildung 19: Marker auf dem Rahmen und der Vorderseite der Kugel

Die Anforderungen an die Marker und die daraus resultierende Wahl der Markertypen wird in diesem Abschnitt erläutert. Weiterhin wird dargelegt, wie diese Marker im Bild detektiert werden.

3.6.1 Marker auf der Kugel

Da die Kugel über keine nennenswerte Textur verfügt, werden Marker auf ihr platziert. Für diese Marker ergeben sich die folgenden Anforderungen: Der Mittelpunkt jedes Markers muss möglichst genau bestimmbar sein, wobei die Orientierung und Position des Markers einen möglichst geringen Einfluss auf die Genauigkeit haben soll, mit welcher das Markerkzentrum ermittelt wird. Dabei ist die Orientierung der Marker unwesentlich. Weiterhin übernimmt das Filter die Aufgabe, den selben Marker anhand seiner Position in aufeinander folgenden Bildern zu identifizieren, wodurch die Marker zum Zeitpunkt ihrer Detektion nicht unterscheidbar sein müssen. Aufgrund dieser Anforderungen werden schwarze, kreisförmige Marker verwendet (im Folgenden auch als Blob-Marker bezeichnet). Um die Zentren der Blobs im Bild zu detektieren, wird das eingehende Farbbild zunächst in ein Grauwertbild umgewandelt. Das resultierende Bild wird mit einem Medianfilter geglättet, um das Bildrauschen zu reduzieren ohne dabei die Konturen der Marker zu verwischen.⁴ Anhand des geglätteten Grauwertbildes werden nun mehrere binäre Masken erstellt. Diese Masken kennzeichnen alle Pixel, deren Grauwerte innerhalb eines Bereiches liegen. Ziel dieser Maskierung ist es, einen Blob vom Hintergrund zu separieren, um seine Kontur zu detektieren. Hierzu muss entweder der Blob ein Teil der Maske sein, wobei der Bereich um den Blob nicht Teil der Maske sein darf, oder vice versa. Weiterhin muss davon ausgegangen werden, dass, obwohl der Marker selbst einfarbig (schwarz) gedruckt wurde, er aufgrund der Beleuchtung mehrere Grauwerte aufweist. Aus diesem Grund werden m Masken mit jeweils anderen Grenzen für die Grauwerte erstellt.⁵ Ein Pixel ist Teil der Maske M_i , wenn

$$u_M \cdot (i) < p_{\text{grau}} < u_M \cdot i + u_B \quad \text{mit: } i = 0, \dots, m - 1$$

erfüllt ist. Hierbei bezeichnet u_M die Schrittweite und u_B die Breite der Masken. Abbildung 20 zeigt beispielhaft, welche Bereiche die Masken für $u_M = 10, u_B = 30$ und $m = 7$ abdecken.

⁴Als Implementierung wird das `OpenCV`-Modul `medianBlur` verwendet

⁵Als Implementierung wird das `OpenCV`-Modul `inRange` verwendet

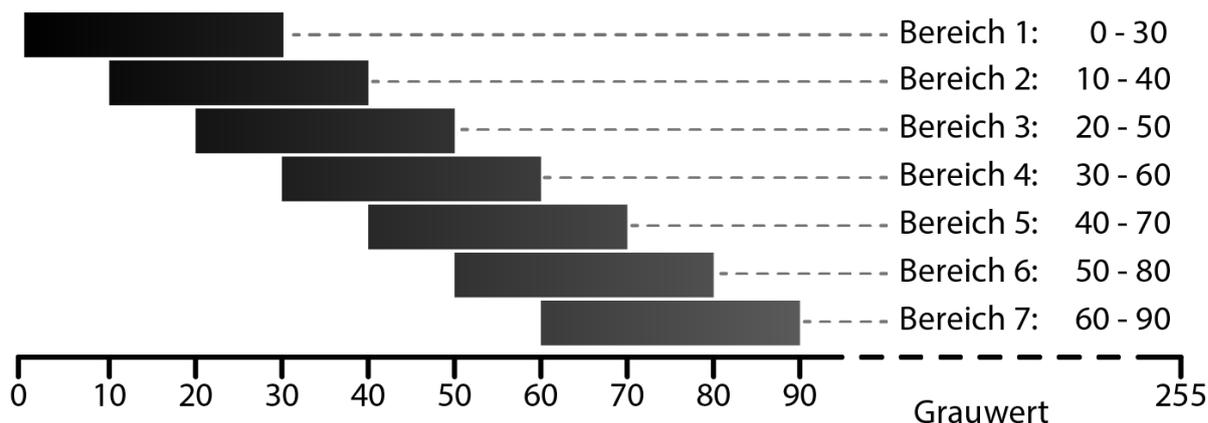


Abbildung 20: Bereiche der Masken für die Detektion von schwarzen Blobs

Auf jede der Masken wird ein **opening** angewendet (**erosion** gefolgt von **dilatation**), um kleine, unzusammenhängende Bereiche zu entfernen.⁶ Auf dem resultierenden binären Bild wird eine Konturensuche durchgeführt, wozu der Algorithmus aus [Suzuki und Abe, 1985] genutzt wird.⁷ Die Momente $m_{a,b}^i$ des Segmentes, das von der Kontur K_k umschlossen wird, berechnen sich zu

$$m_{i,j}^k = \sum_{x,y} (A_k(x,y) \cdot x^i \cdot y^j) \quad \text{wobei gilt:} \quad A_k(x,y) = \begin{cases} 1 & \text{wenn } (x,y) \text{ innerhalb } K_k \\ 0 & \text{wenn } (x,y) \text{ ausserhalb } K_k \end{cases}$$

Das Moment m_{00} entspricht der von K_k umschlossenen Fläche F_k . Weiterhin lässt sich der Schwerpunkt $s_k = (\bar{x}, \bar{y})$ anhand der Momente berechnen:

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \text{und} \quad \bar{y} = \frac{m_{01}}{m_{00}}. \quad (96)$$

Mit diesen Informationen werden einige Bedingungen geprüft, um Konturen zu identifizieren, die einem Blob-Marker zugeordnet werden können. Zunächst wird geprüft, ob F_k größer oder gleich einer unteren Grenze F_{min} für den Flächeninhalt ist. Die erste Bedingung folgt zu:

$$F_k \geq F_{min}. \quad (97)$$

Analog wird geprüft, ob F_k kleiner oder gleich einer oberen Grenze F_{max} für den Flächeninhalt ist. Hieraus folgt die zweite Bedingung zu:

$$F_k \leq F_{max}. \quad (98)$$

⁶Als Implementierung werden die **OpenCV**-Module **erode** und **dilate** verwendet

⁷Als Implementierung wird das **OpenCV**-Modul **findContours** verwendet

Des Weiteren wird die konvexe Hülle der Kontur berechnet und deren Flächeninhalt $F_{k, \text{konvex}}$ ermittelt. Der Quotient

$$q_k = \frac{F_k}{F_{k, \text{konvex}}} \quad \text{mit den Grenzen} \quad 0 < q_k \leq 1$$

gibt Aufschluss darüber, wie sehr der Blob mit dem Index k einer konvexen Form gleicht. Hierbei bedeutet $q_k = 1$, dass die Flächen identisch sind und K_k konvex ist. Hingegen bedeutet $q_k \ll 1$, dass K_k stark von einer konvexen Form abweicht. Für die Kontur eines detektierten Blob-Markers gilt, dass sie im Falle einer fehlerfreien Detektion einer Ellipse entspricht und somit konvex ist. Einflüsse wie Bildrauschen können die Kontur jedoch beeinträchtigen, weswegen ein unterer Grenzwert q_{\min} für q_k festgelegt wird. Daher werden Blobs verworfen, für die gilt:

$$q_k < q_{\min}.$$

Zuletzt müssen mehrfache Detektionen der Konturen verworfen werden, die aus dem selben Blob resultieren. Falls für zwei Konturen K_k und K_l gilt, dass der Schwerpunkt s_k innerhalb K_l liegt und gleichzeitig $A_k \leq A_l$ dann wird A_k aus der Menge der Konturen entfernt. Die Schwerpunkte der verbleibenden Konturen sind die detektierten Markerzentren.

Der Ablauf der Detektion ist in Abbildung 21 anhand eines Beispielbildes dargestellt. Das eingehende Bild zeigt sieben graue Punkte vor einem geringfügig helleren Hintergrund. Sowohl die Punkte als auch der Hintergrund werden von links nach rechts, wie auch von unten nach oben heller, um die Notwendigkeit der mehrfachen Maskierung darzustellen.

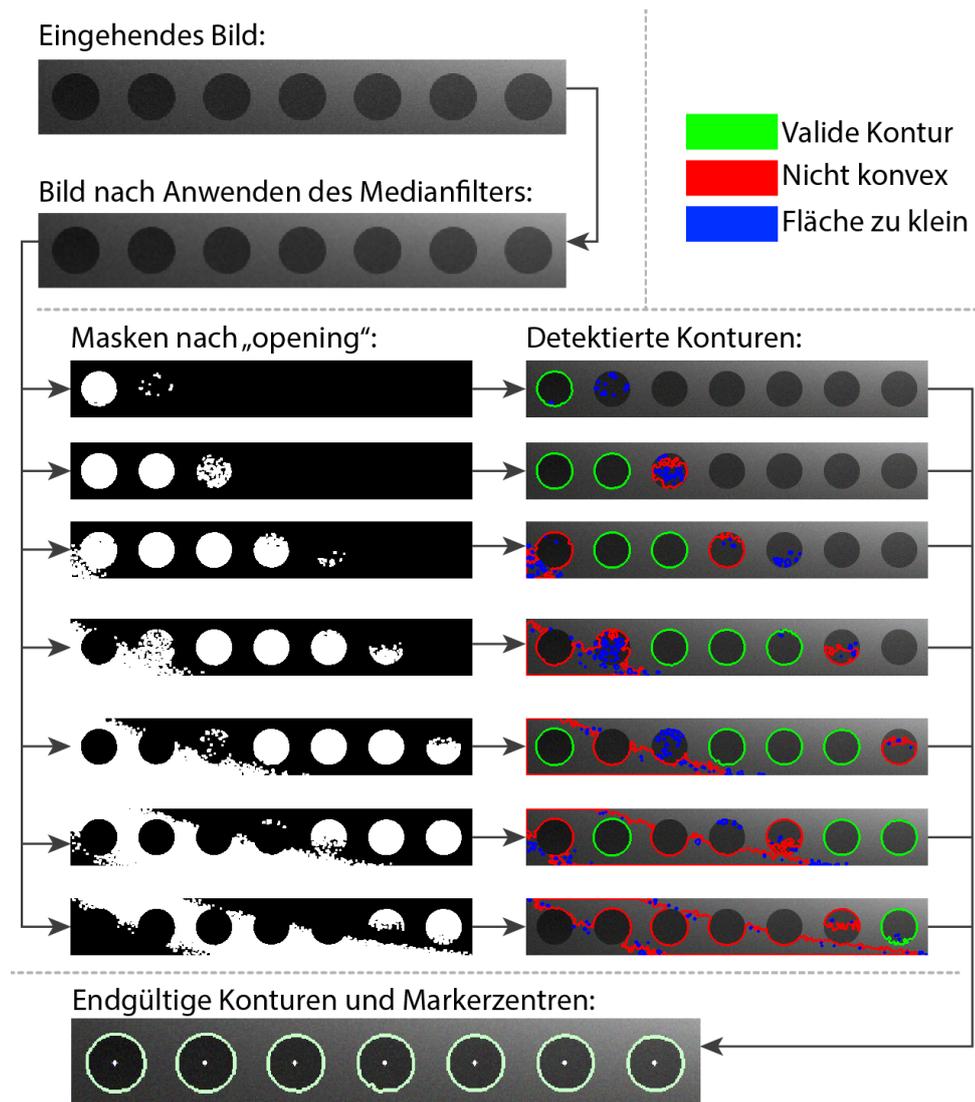


Abbildung 21: Ablauf der Detektion von Blobs und der Bestimmung ihrer Zentren anhand mehrerer Masken und Restriktionen

3.6.2 Marker auf dem Rahmen

Die Marker auf dem Rahmen dienen als Passpunkte für den räumlichen Rückwärtsschnitt und somit der Berechnung der äußeren Orientierung der Kamera. Da hierfür sowohl Objekt- als auch Bildkoordinaten eines Markers benötigt werden, müssen die Marker eindeutig identifizierbar sein. Die Orientierung der Marker zu bestimmen, ist nicht notwendig, da für den RRWS nur die Positionen der Markerzentren relevant sind. Um diese Anforderungen zu erfüllen, werden Marker eingesetzt, die aus einem farbigen Kreis bestehen, der innerhalb zwei andersfarbiger Ringe liegt. Die verwendeten Farben sind Cyan, Magenta und Gelb mit jeweils voller Sättigung und Helligkeit. Diese Farben liegen im Farbraum

möglichst weit auseinander und sind somit optimal unterscheidbar. Wird dem innen liegenden Punkt sowie den umschließenden Kreisen jeweils eine andere Farbe zugeordnet, ergeben sich sechs mögliche Farbkodierungen für die Marker. Die Minimalkonfiguration erfordert drei Marker, es werden jedoch vier Marker eingesetzt, um eine Redundanz bei dem Bestimmen der äußeren Orientierung der Kamera zu erhalten. Die verwendeten Marker sind in Abbildung 22 dargestellt.

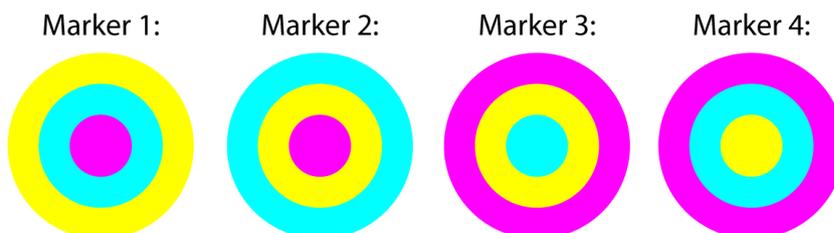


Abbildung 22: Marker auf dem Rahmen für den RRWS

Um die farbkodierten Marker im Bild zu detektieren, wird auf das eingehende Bild zunächst ein Medianfilter angewendet. Im Anschluss werden drei Masken des Bildes erstellt. Diese stellen dar, bei welchen Pixeln sich der **hue**-Wert im Bereich Cyan, Magenta und Gelb befinden. Gleichzeitig werden Grenzwerte für die Sättigung s_{min} und s_{max} sowie für die Helligkeit v_{min} und v_{max} eingeführt. Ein Pixel mit den HSV-Werten (h, s, v) ist Teil der Cyan-Maske, wenn die Bedingungen

$$h_{min,cyan} < h < h_{max,cyan} \quad (99)$$

$$s_{min} < s < s_{max} \quad (100)$$

$$v_{min} < v < v_{max} \quad (101)$$

erfüllt sind. Gleiches gilt für die Masken für den Magenta- und Gelb-Bereich mit den entsprechenden Grenzen für h , s und v . Um die Masken zu verbessern, wird die Methode des **opening** angewandt. Es werden die Konturen jedes zusammenhängenden Bereiches der Maske identifiziert. Diese Konturen beschreiben alle Bereiche, die der jeweiligen Farbe zugeordnet werden können. Unter allen Konturen werden im Anschluss diejenigen ermittelt, welche die Restriktionen der Marker erfüllen. Beispielsweise wird für den Marker 1 die Kontur eines magenta-farbenen Bereiches gesucht, welche innerhalb der Kontur eines cyan-farbenen Bereiches liegt, welche sich wiederum innerhalb der Kontur eines gelben Bereiches befindet. Werden drei Konturen gefunden, welche die Restriktionen eines Markers erfüllen, werden die Momente dieser Konturen berechnet. Aus diesen lassen sich die Schwerpunkte der Konturen berechnen. Es werden jeweils drei Konturen je Marker detektiert, deren Schwerpunkte gemittelt werden, um das Markerzentrum zu bestimmen.

In Abbildung 23 ist der Ablauf der Detektion von den farbkodierten Markern und der Bestimmung der Markerzentren dargestellt. Das Eingangsbild zeigt die vier Marker in einer realen Aufnahme.

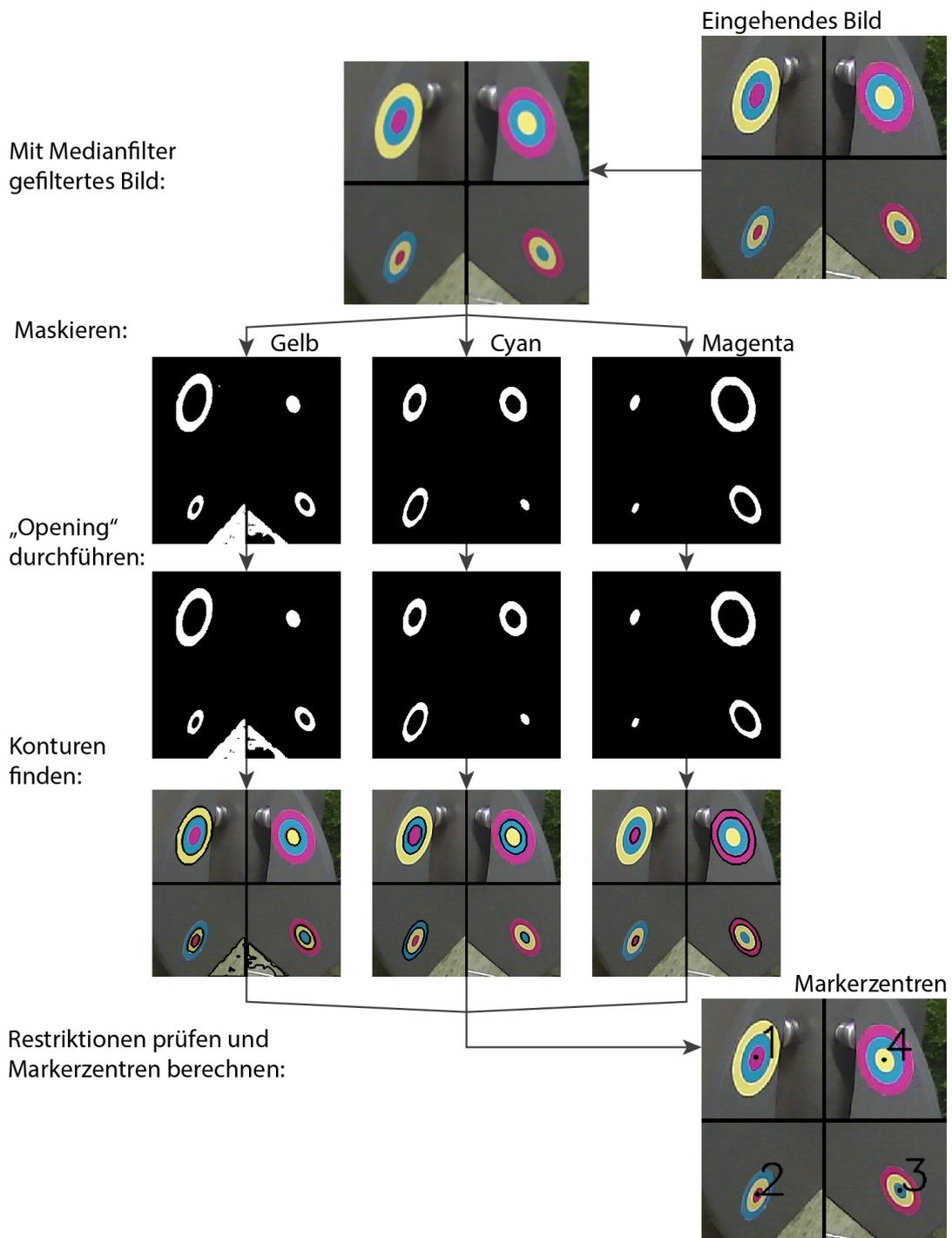


Abbildung 23: Ablauf der Detektion von farbkodierten Markern und der Bestimmung der Markerzentren

3.7 Bestimmen der Kugelparameter

Um den Kugelmittelpunkt und den Radius der Kugel zu ermitteln, wird ein GHM basierend auf den Ergebnissen der Bündelblockausgleichung verwendet. Als Beobachtungen gehen hierbei die Zentren von p Blob-Markern auf der Kugel in Form von deren Objektkoordinaten $(X_{B_i}, Y_{B_i}, Z_{B_i})^T$ ein, mit $i = 1, \dots, p$. Gesucht sind die Position des Kugelmittelpunktes $(X_K, Y_K, Z_K)^T$ sowie der Kugelradius R_K . Beobachtungsvektor \mathbf{l} und Unbekanntenvektor \mathbf{x} lauten somit

$$\mathbf{l} = \begin{pmatrix} (X_{B_1}, Y_{B_1}, Z_{B_1})^T \\ (X_{B_2}, Y_{B_2}, Z_{B_2})^T \\ \vdots \\ (X_{B_p}, Y_{B_p}, Z_{B_p})^T \end{pmatrix} \quad \text{und} \quad \mathbf{x} = \begin{pmatrix} X_K \\ Y_K \\ Z_K \\ R_K \end{pmatrix}. \quad (102)$$

Für jeden Blob-Marker lässt sich eine Bedingungsgleichung aufstellen. Diese fordert, dass die Distanz zwischen dem Markerzentrum und dem Mittelpunkt der Kugel dem Kugelradius entsprechen muss. Die p Bedingungsgleichungen lauten somit:

$$\mathbf{f}_i : \quad 0 = \sqrt{(X_{B_i} - X_K)^2 + (Y_{B_i} - Y_K)^2 + (Z_{B_i} - Z_K)^2} - R_K \quad \text{mit: } i = 1, \dots, p. \quad (103)$$

Für das stochastische Modell wird angenommen, dass keine Korrelation zwischen den einzelnen Blob-Markern existiert. Die VKM der Blob-Marker $\Sigma_{\mathbf{B}_i \mathbf{B}_i}$ entsprechen den Ergebnissen der Bündelblockausgleichung. Die VKM der Beobachtungen $\Sigma_{\mathbf{ll}}$ hat die Struktur

$$\Sigma_{\mathbf{ll}} = \mathbf{Q}_{\mathbf{ll}} = \begin{pmatrix} \Sigma_{\mathbf{B}_1 \mathbf{B}_1} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & \Sigma_{\mathbf{B}_2 \mathbf{B}_2} & 0_{3 \times 3} & \ddots \\ 0_{3 \times 3} & 0_{3 \times 3} & \ddots & 0_{3 \times 3} \\ 0_{3 \times 3} & \ddots & 0_{3 \times 3} & \Sigma_{\mathbf{B}_p \mathbf{B}_p} \end{pmatrix}. \quad (104)$$

Designmatrix \mathbf{A} und Bedingungsmatrix \mathbf{B} der Iteration k ergeben sich zu

$$\mathbf{A}^{(k)} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \Big|_{\mathbf{x}^{(k)}, \mathbf{l}^{(k)}} = \begin{pmatrix} \frac{X_K^{(k)} - X_{B_1}^{(k)}}{d_1^{(k)}} & \frac{Y_K^{(k)} - Y_{B_1}^{(k)}}{d_1^{(k)}} & \frac{Z_K^k - Z_{B_1}^{(k)}}{d_1^{(k)}} & -1 \\ \frac{X_K^{(k)} - X_{B_2}^{(k)}}{d_2^{(k)}} & \frac{Y_K^{(k)} - Y_{B_2}^{(k)}}{d_2^{(k)}} & \frac{Z_K^k - Z_{B_2}^{(k)}}{d_2^{(k)}} & -1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{X_K^{(k)} - X_{B_p}^{(k)}}{d_p^{(k)}} & \frac{Y_K^{(k)} - Y_{B_p}^{(k)}}{d_p^{(k)}} & \frac{Z_K^k - Z_{B_p}^{(k)}}{d_p^{(k)}} & -1 \end{pmatrix} \quad (105)$$

und

$$\mathbf{B}^{(k)} = \frac{\partial \mathbf{f}}{\partial \mathbf{l}} \Big|_{\mathbf{x}^{(k)}, \mathbf{l}^{(k)}} = \quad (106)$$

$$\begin{pmatrix} \frac{X_{B_1}^{(k)} - X_K^{(k)}}{d_1^{(k)}} & \frac{Y_{B_1}^{(k)} - Y_K^{(k)}}{d_1^{(k)}} & \frac{Z_{B_1}^{(k)} - Z_K^{(k)}}{d_1^{(k)}} & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \frac{X_{B_1}^{(k)} - X_K^{(k)}}{d_p^{(k)}} & \frac{Y_{B_1}^{(k)} - Y_K^{(k)}}{d_p^{(k)}} & \frac{Z_{B_1}^{(k)} - Z_K^{(k)}}{d_p^{(k)}} \end{pmatrix} \quad (107)$$

wobei

$$d_i^{(k)} = \sqrt{(X_{B_i}^{(k)} - X_K^{(k)})^2 + (Y_{B_i}^{(k)} - Y_K^{(k)})^2 + (Z_{B_i}^{(k)} - Z_K^{(k)})^2} \quad \text{mit: } i = 1, \dots, p. \quad (108)$$

Der Widerspruchsvektors \mathbf{w} ergibt sich zu

$$w_i^{(k)} = \sqrt{(X_{B_i} - X_K^{(k)})^2 + (Y_{B_i} - Y_K^{(k)})^2 + (Z_{B_i} - Z_K^{(k)})^2} - R_K^{(k)} \quad \text{mit: } i = 1, \dots, p. \quad (109)$$

Die Startwerte für das GHM ergeben sich aus den näherungsweise bekannten Positionen der Marker auf dem Rahmen sowie der Herstellerangabe zum Radius der Kugel. Der iterative Algorithmus des GHM wird wiederholt, bis die größte absolute Änderung des Unbekanntenvektors im Vergleich zur vorherigen Iteration kleiner als ein Mikrometer ist. Sind mindestens fünf Punkte auf der Kugel als Beobachtung vorhanden, gibt es eine Redundanz $f = p - 4 > 0$ und es kann die empirische Varianz der Gewichtseinheit $\hat{\sigma}_0^2$ und mit dieser die a-posteriori VKM der ausgeglichenen Parameter $\hat{\Sigma}_{\mathbf{KK}}$ berechnet werden. Letztere ist zusammen mit den ausgeglichenen Parametern der Kugel $\hat{\mathbf{x}} = \mathbf{K} = (\hat{X}_K, \hat{Y}_K, \hat{Z}_K, \hat{R}_K)$ das Ergebnis der Kugelschätzung.

3.8 Schnitt eines Bildstrahles mit der Kugel

In den vorherigen Abschnitten wurde erläutert, wie die Pose der Kamera sowie der Mittelpunkt und der Radius der Kugel einschließlich deren VKM im \mathbf{KS}_0 ermittelt und die Zentren der Blob-Marker im Bild detektiert und aufgrund der Verzeichnung korrigiert werden. Im folgenden seien $(x_B, y_B)^T$ die Bildkoordinaten des Markerzentrums nach Korrektur aufgrund der Verzeichnung. Im folgenden wird dargelegt, wie die zuvor genannten Informationen zusammengeführt werden, um die 3D-Koordinaten eines detektierten Blob-Markers $\mathbf{B} = (X_B, Y_B, Z_B)$ im \mathbf{KS}_0 zu bestimmen. Im ersten Schritt werden die Objektkoordinaten des Markerzentrums berechnet, im zweiten Schritt die entsprechende VKM. Abbildung 24 zeigt die Geometrie dieser Berechnung.

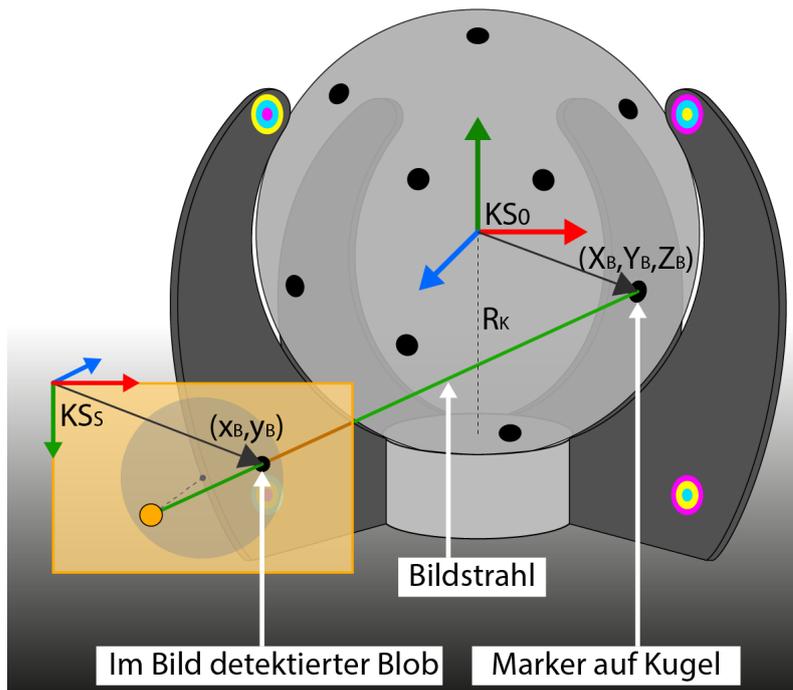


Abbildung 24: Projektion eines detektierten Markers auf die Kugel

3.8.1 Berechnung des Schnittpunktes

Zunächst wird hergeleitet, wie der Bildstrahl eines detektierten Markerzentrums mit der Kugel geschnitten wird, um die Objektkoordinaten des Markerzentrums zu berechnen. Zwischen den Objektkoordinaten und den Korrigierten Bildkoordinaten jedes ermittelten Markerzentrums gilt der Zusammenhang aus Gleichung 65. Formt man diese Gleichung um, resultiert

$$\begin{pmatrix} X_B - X_0 \\ Y_B - Y_0 \\ Z_B - Z_0 \end{pmatrix} = m \cdot \mathbf{R}(\omega_0, \varphi_0, \kappa_0) \cdot \begin{pmatrix} x_B - x_0 \\ y_B - y_0 \\ -c \end{pmatrix}. \quad (110)$$

Hieraus lässt sich der Richtungsvektor

$$\mathbf{r} = \mathbf{R}(\omega_0, \varphi_0, \kappa_0) \cdot \begin{pmatrix} x_B - x_0 \\ y_B - y_0 \\ -c \end{pmatrix} \quad (111)$$

herleiten, der im KS_0 vom Projektionszentrum ausgehend, auf das Markerzentrum zeigt. Der Bildstrahl, auf der sich das gesuchte Markerzentrum befindet, lässt sich nun durch die Geradengleichung

$$\begin{pmatrix} X_B \\ Y_B \\ Z_B \end{pmatrix} = \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix} + \lambda \cdot \mathbf{r} \quad (112)$$

beschreiben. Weiterhin soll das Markerzentrum auf der Kugel liegen, woraus folgt, dass

$$R_K = \sqrt{(X_B - X_K)^2 + (Y_B - Y_K)^2 + (Z_B - Z_K)^2} \quad (113)$$

gilt. Für jeden detektierten Marker liegen vier Gleichungen mit jeweils vier unbekannt-ten Größen X_B, Y_B, Z_B und λ vor. Gibt es keine Lösung der Gleichungen, schneidet der Bildstrahl die Kugel nicht, was als Fehldetektion des Markers gedeutet wird. Tangiert der Bildstrahl die Kugel, folgt genau eine Lösung. Anderenfalls ergeben sich zwei Schnittpunkte S_1 und S_2 mit den Distanzfaktoren λ_1 und λ_2 . Verwendet wird der Schnittpunkt, der näher am Projektionszentrum liegt, also S_1 falls $\lambda_1 < \lambda_2$ und S_2 falls $\lambda_1 > \lambda_2$.

3.8.2 Fehlerfortpflanzung

Die Berechnung der VKM der Objektkoordinaten aller detektierten Markerzentren auf der Kugel ist notwendig, da diese in das stochastische Modell des EKF eingehen. Durch diese zusätzlichen Information wird erwartet, dass die Güte der Filterung steigt und die Ergebnisse eine geringere Abweichung zu den wahren Werten aufweisen. Zusätzlich wird erzielt, dass eine realistischere Aussage über die Genauigkeit der Ergebnisse der Filterung getroffen werden kann.

Erneut wird das GHM Modell genutzt um die VKM $\Sigma_{\mathbf{BB}}$ des Objektpunktes \mathbf{B} zu ermitteln. Als Beobachtungen geht die äußere Orientierung der Kamera in Form von Translation $(X_0, Y_0, Z_0)^T$ und Lagewinkel $(\omega_0, \varphi_0, \kappa_0)^T$ ein. Weitere Beobachtungen sind die Parameter der Kugel $\mathbf{K} = (X_K, Y_K, Z_K, R_K)^T$ und zuletzt die beobachteten, korrigierten Bildkoordinaten des zu projizierenden Blob-Markers (x_B, y_B) . Die Berechnung wird für jeden Marker getrennt durchgeführt. Für die Pose des Projektionszentrums sowie die Kugelparameter liegen jeweils vollständige VKM $\hat{\Sigma}_{00}$ bzw. $\hat{\Sigma}_{\mathbf{KK}}$ vor. Bislang unbekannt ist die Unsicherheit, mit welcher ein Blob-Marker im Bild detektiert wird. Anzumerken ist, dass das Kugelzentrum zwar dem Ursprung des Referenzkoordinatensystems entspricht $X_K = Y_K = Z_K = 0$, dennoch ist es mit Unsicherheit behaftet, die in der Modellierung berücksichtigt werden muss. Der zu schätzende Zustandsvektor beinhaltet die Koordinaten des Markers $(X_B, Y_B, Z_B)^T$ im \mathbf{KS}_0 . Zusammengefasst lauten Beobachtungsvektor und Unbekanntenvektor

$$\mathbf{l} = \begin{pmatrix} (X_0, Y_0, Z_0)^T \\ (\omega_0, \varphi_0, \kappa_0)^T \\ (X_K, Y_K, Z_K, R_K)^T \\ (x_B, y_B)^T \end{pmatrix} \quad \text{und} \quad \mathbf{x} = \begin{pmatrix} X_B \\ Y_B \\ Z_B \end{pmatrix}. \quad (114)$$

Zusätzlich wird die innere Orientierung der Kamera (x_0, y_0, c) benötigt, die aus der Ka-

merakalibrierung hervor geht. Diese Parameter werden hierbei als fehlerfrei angenommen. Die ersten beiden Bedingungsgleichungen des funktionalen Modells ergeben sich durch Umformen der Kollinearitätsgleichungen zu

$$f_1 : \quad 0 = \left(x_0 - c \frac{r_{11}(X_B - X_0) + r_{21}(Y_B - Y_0) + r_{31}(Z_B - Z_0)}{r_{31}(X_B - X_0) + r_{23}(Y_B - Y_0) + r_{33}(Z_B - Z_0)} \right) - x_B \quad (115)$$

und

$$f_2 : \quad 0 = \left(y_0 - c \frac{r_{12}(X_B - X_0) + r_{22}(Y_B - Y_0) + r_{32}(Z_B - Z_0)}{r_{31}(X_B - X_0) + r_{23}(Y_B - Y_0) + r_{33}(Z_B - Z_0)} \right) - y_B. \quad (116)$$

Die dritte Bedingungsgleichung folgt aus der Restriktion, dass sich die Koordinaten des Objektpunktes auf der Kugel befinden müssen, also die Länge des Vektors von Kugelmittelpunkt zum Marker dem Radius der Kugel entsprechen muss, zu

$$f_3 : \quad 0 = R_K - \sqrt{(X_B - X_K)^2 + (Y_B - Y_K)^2 + (Z_B - Z_K)^2}. \quad (117)$$

Für das stochastische Modell wird die VKM $\Sigma_{\mathbf{bb}}$ des detektierten Markerzentrums im Bild eingeführt. Hierbei wird angenommen, dass die Beobachtungen unkorreliert sind und sowohl die x- als auch y-Koordinaten mit einer Standardabweichung σ_b berechnet wurden. Die Wahl von σ_b sowie der Einfluss dieser Wahl auf das Ergebnis der Filterung wird in Abschnitt 4.5 behandelt. Es folgt, dass

$$\Sigma_{\mathbf{bb}} = \begin{pmatrix} \sigma_b^2 & 0 \\ 0 & \sigma_b^2 \end{pmatrix}. \quad (118)$$

Weiterhin wird angenommen, dass Kugelparameter, Kamerapose und Bildpunkt unkorreliert sind. Die vollständige VKM der Beobachtungen ergibt sich zu

$$\Sigma_{\mathbf{ll}} = \begin{pmatrix} \hat{\Sigma}_{00} & 0 & 0 \\ 0 & \hat{\Sigma}_{\mathbf{KK}} & 0 \\ 0 & 0 & \Sigma_{\mathbf{bb}} \end{pmatrix}. \quad (119)$$

Das GHM wird mit der exakten Lösung des Problems initialisiert, wodurch nach einer Iteration die Kofaktormatrix der ausgeglichenen Parameter $\Sigma_{\mathbf{xx}}$ bekannt ist. Mit $\hat{\sigma}_0^2 = 1$ folgt, dass die VKM der Markerkoordinaten $\hat{\Sigma}_{\mathbf{BB}}$ der Kofaktormatrix der ausgeglichenen Parameter entspricht:

$$\hat{\Sigma}_{\mathbf{BB}} = \Sigma_{\mathbf{xx}}. \quad (120)$$

3.9 Extended Kalman Filter

Im Folgenden wird der Filter dargestellt, der fortlaufend aus den Objektkoordinaten der Markerzentren auf der Kugel die Lagewinkel und Drehraten der Kugel bestimmt und aus den Drehraten eine Mausbewegung berechnet, die äquivalent zu den Ausgaben des Trackballs ist. Die grundlegende Idee des Filters ist es, die Bewegung der Blob-Marker im \mathbf{KS}_0 über mehrere Epochen hinweg zu verfolgen. Im Koordinatensystem der Kugel \mathbf{KS}_K sind die wahren Positionen der Marker unveränderlich, wodurch ein Zusammenhang zwischen der Bewegung der Blob-Marker im \mathbf{KS}_0 und der gesuchten Rotation der Kugel gegenüber dem \mathbf{KS}_0 hergestellt werden kann. Es werden zwei Modelle vorgestellt, die sich in der Art unterscheiden, in welcher die Koordinaten der Markerzentren im \mathbf{KS}_K gespeichert werden. Im ersten Modell (EKF_1) werden neu detektierte Marker als Konstanten abgelegt, im zweiten Modell (EKF_2) werden sie in den Zustandsvektor aufgenommen und können sich über die Zeit verändern.

3.9.1 Modell 1: Punkte als Konstante gespeichert

Für die Modellierung des EKF werden zunächst Beobachtungsvektor sowie Zustandsvektor der Epoche k definiert. Beobachtet werden in jeder Epoche die Objektkoordinaten $\mathbf{B}_i^{(k)} = (B_{i,x}^{(k)}, B_{i,y}^{(k)}, B_{i,z}^{(k)})^T$ der Markerzentren auf der Kugeloberfläche im \mathbf{KS}_0 . Die VKM $\Sigma_{\mathbf{B}_i, \mathbf{B}_i}$ der Koordinaten jedes Zentrums sind ebenfalls bekannt (vgl. Kapitel 3.8). Der Zustandsvektor \mathbf{x} soll die Orientierung der Kugel sowie deren Änderung beschreiben. Diese werden über die Lagewinkel und die Drehraten parametrisiert. Der Zustandsvektor der Epoche k lautet daher

$$\mathbf{x}^{(k)} = (\omega_K^{(k)}, \varphi_K^{(k)}, \kappa_K^{(k)}, \dot{\omega}_K^{(k)}, \dot{\varphi}_K^{(k)}, \dot{\kappa}_K^{(k)})^T, \quad (121)$$

wobei der Zustandsvektor mit

$$\mathbf{x}_+^{(0)} = (0, 0, 0, 0, 0, 0)^T \quad (122)$$

und dessen VKM mit

$$\Sigma_{\mathbf{xx}_+}^{(0)} = \begin{pmatrix} 0.0001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.0001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.0001 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.0001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0001 \end{pmatrix} \quad (123)$$

initialisiert wird. Der Störterm entspricht den Winkelbeschleunigungen, da das dynamische System auf der Annahme basiert, dass die Drehraten konstant sind:

$$\mathbf{w}^{(k)} = (\ddot{\omega}_K^{(k)}, \ddot{\varphi}_K^{(k)}, \ddot{\kappa}_K^{(k)})^T. \quad (124)$$

Der Erwartungswert der Störterme ist definitionsgemäß null. Die Normalverteilungen, denen die Störterme folgen, werden durch die gemeinsame Standardabweichung σ_w parametrisiert. Unter der Annahme, dass $\Sigma_{\mathbf{ww}}$ über alle Epochen konstant ist folgt

$$\Sigma_{\mathbf{ww}}^{(k)} = \Sigma_{\mathbf{ww}} = \begin{pmatrix} \sigma_w^2 & 0 & 0 \\ 0 & \sigma_w^2 & 0 \\ 0 & 0 & \sigma_w^2 \end{pmatrix}. \quad (125)$$

Das dynamische System beschreibt, wie der Zustandsvektor aus der vorherigen Epoche prädiziert werden kann. Die entsprechenden Gleichungen lauten

$$f_\omega : \quad \omega_{K,-}^{(k)} = \omega_{K,+}^{(k-1)} + \delta t \cdot \dot{\omega}_{K,+}^{(k-1)} + \frac{\delta t^2}{2} \cdot \ddot{\omega}_K^{(k-1)} \quad (126)$$

$$f_\varphi : \quad \varphi_{K,-}^{(k)} = \varphi_{K,+}^{(k-1)} + \delta t \cdot \dot{\varphi}_{K,+}^{(k-1)} + \frac{\delta t^2}{2} \cdot \ddot{\varphi}_K^{(k-1)} \quad (127)$$

$$f_\kappa : \quad \kappa_{K,-}^{(k)} = \kappa_{K,+}^{(k-1)} + \delta t \cdot \dot{\kappa}_{K,+}^{(k-1)} + \frac{\delta t^2}{2} \cdot \ddot{\kappa}_K^{(k-1)} \quad (128)$$

$$f_{\dot{\omega}} : \quad \dot{\omega}_{K,-}^{(k)} = \dot{\omega}_{K,+}^{(k-1)} + \delta t \cdot \ddot{\omega}_K^{(k-1)} \quad (129)$$

$$f_{\dot{\varphi}} : \quad \dot{\varphi}_{K,-}^{(k)} = \dot{\varphi}_{K,+}^{(k-1)} + \delta t \cdot \ddot{\varphi}_K^{(k-1)} \quad (130)$$

$$f_{\dot{\kappa}} : \quad \dot{\kappa}_{K,-}^{(k)} = \dot{\kappa}_{K,+}^{(k-1)} + \delta t \cdot \ddot{\kappa}_K^{(k-1)}. \quad (131)$$

Wobei δt der Zeitspanne zwischen den Epochen k und $k - 1$ entspricht. Da mit dem EKF ein Videostream ausgewertet wird, entspricht δt der inversen Bildrate der Kamera und kann als konstant angesehen werden. Differenziert man diese Gleichungen nach dem Zustandsvektor, ergibt sich die Transitionsmatrix $\Phi^{(k)}$ zu

$$\Phi^{(k)} = \begin{pmatrix} 1 & 0 & 0 & \delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (132)$$

Werden die Gleichungen des dynamischen Modells nach dem Störvektor differenziert,

resultiert die Stellmatrix $\mathbf{G}^{(k)}$ zu

$$\mathbf{G}^{(k)} = \begin{pmatrix} \frac{\delta t^2}{2} & 0 & 0 \\ 0 & \frac{\delta t^2}{2} & 0 \\ 0 & 0 & \frac{\delta t^2}{2} \\ \delta t & 0 & 0 \\ 0 & \delta t & 0 \\ 0 & 0 & \delta t \end{pmatrix}. \quad (133)$$

Als nächstes wird der Zusammenhang zwischen den Beobachtungen und dem Zustandsvektor hergestellt. Sei $\mathbf{P}_j = (P_{j,x}, P_{j,y}, P_{j,z})^T$ ein Punkt der die Koordinaten eines gespeicherten Markerzentrums im Koordinatensystem der Kugel \mathbf{KS}_K beschreibt, so gilt für den beobachteten Punkt $\mathbf{B}_i^{(k)}$ der dem gespeicherten Punkt \mathbf{P}_j zugeordnet wird und dessen Koordinaten im \mathbf{KS}_0 vorliegen der Zusammenhang

$$\mathbf{B}_i^{(k)} + v_i = R_{K,-}^{(k)} \cdot \mathbf{P}_j = \mathbf{R}(\omega_{K,-}^{(k)}, \varphi_{K,-}^{(k)}, \kappa_{K,-}^{(k)}) \cdot P_j \quad (134)$$

woraus sich die Beobachtungsgleichungen \mathbf{h} ergeben:

$$h_{B_{i,x}} : \quad B_{i,x}^{(k)} + v_{i,x} = r_{11} \cdot P_{j,x} + r_{12} \cdot P_{j,y} + r_{13} \cdot P_{j,z} + v_{i,x} \quad (135)$$

$$h_{B_{i,y}} : \quad B_{i,y}^{(k)} + v_{i,y} = r_{21} \cdot P_{j,x} + r_{22} \cdot P_{j,y} + r_{23} \cdot P_{j,z} + v_{i,y} \quad (136)$$

$$h_{B_{i,z}} : \quad B_{i,z}^{(k)} + v_{i,z} = r_{31} \cdot P_{j,x} + r_{32} \cdot P_{j,y} + r_{33} \cdot P_{j,z} + v_{i,z}. \quad (137)$$

Hierbei sind v_i die Residuen, welche die Abweichung der Beobachtung zum erwarteten Wert ausdrücken.

Der Beobachtungsvektor jeder Epoche setzt sich folglich aus den Koordinaten der $m^{(k)}$ assoziierten Punkte zusammen und hat die Länge $n^{(k)} = 3 \cdot m^{(k)}$. Die Datenassoziation wird in Abschnitt 3.9.2 erläutert. Entsprechend kann auch die VKM des Beobachtungsvektors aus den VKM der Beobachteten Punkte $\mathbf{B}_1, \dots, \mathbf{B}_i, \dots, \mathbf{B}_m$ aufgebaut werden zu

mit

$$\begin{aligned} c_\omega &:= \cos(\omega), c_\varphi := \cos(\varphi), c_\kappa := \cos(\kappa), \\ s_\omega &:= \sin(\omega), s_\varphi := \sin(\varphi), s_\kappa := \sin(\kappa), \end{aligned}$$

wobei \mathbf{P}_j der gespeicherte Punkt ist, der mit dem Punkt \mathbf{B}_i assoziiert wurde. Die Jacobi-Matrix der Beobachtungsgleichungen $\mathbf{M}^{(k)}$ bezüglich der Residuen ist eine Identitätsmatrix der Dimension $n \times n$, wobei n die Anzahl der assoziierten Beobachtungen ist. Mit den dargelegten Gleichungen und Matrizen kann der EKF-Algorithmus wie in Abschnitt 2.3.2 beschrieben durchgeführt werden.

3.9.2 Assoziation der Beobachtungen

Für den Ablauf des Filters ist es notwendig, dass die im \mathbf{KS}_0 beobachteten Koordinaten der Marker mit den richtigen im \mathbf{KS}_K gespeicherten Punkten assoziiert werden. Hierzu werden die Lagewinkel der Kugel jeweils nach der Prädiktion genutzt, um die Koordinaten der gespeicherten Punkte vom \mathbf{KS}_K in das \mathbf{KS}_0 zu rotieren. Ein im \mathbf{KS}_0 beobachteter Objektpunkt $\mathbf{B}_i^{(k)}$ der Epoche k wird genau dann mit dem im \mathbf{KS}_K gespeicherten Punkt \mathbf{P}_j assoziiert, wenn die Distanz zwischen dem rotierten, gespeicherten Punkt und dem beobachteten Markerzentrum unter dem Grenzwert g liegt, folglich wenn

$$|\mathbf{R}(\omega_{K,-}^{(k)}, \varphi_{K,-}^{(k)}, \kappa_{K,-}^{(k)}) \cdot \mathbf{P}_j - \mathbf{B}_i^{(k)}|_2 < g \quad (149)$$

erfüllt ist. Hierbei bezeichnet j die Indizes der gespeicherten Punkte und i die Indizes der detektierten Markerzentren. Mit anderen Worten werden die im \mathbf{KS}_K gespeicherten Markerzentren in das \mathbf{KS}_0 rotiert und es wird überprüft, ob sich ein beobachteter Punkt in der umliegenden Kugel mit dem Radius g befindet

Denkbar sind nun drei Fälle: Der erste, eben beschriebene Fall tritt ein, wenn $\mathbf{B}_i^{(k)}$ mit dem Punkt \mathbf{P}_j assoziiert wird. Im zweiten Fall wird ein Punkt beobachtet, der keinem gespeicherten Punkt zugeordnet werden kann. Dann wird der beobachtete Objektpunkt $\mathbf{B}_i^{(k)}$ in das \mathbf{KS}_K rotiert und die Koordinaten von \mathbf{P}_i gespeichert. Die Transformation erfolgt hierbei durch Multiplikation der transponierten Rotationsmatrix zu

$$\mathbf{P}_i = \mathbf{R}_{K,+}^{(k)T} \cdot \mathbf{B}_i^{(k)}. \quad (150)$$

Anzumerken ist, dass die Rotationsmatrix aus den Lagewinkeln im Zustandsvektor nach dem Updateschritt gebildet wird, weswegen sie mit einem Plus gekennzeichnet ist. Tritt der letzte Fall ein, dass nämlich kein beobachteter Punkt mit dem gespeicherten Punkt \mathbf{P}_j assoziiert werden kann, so wird \mathbf{P}_j aus der Menge der gespeicherten Punkte entfernt. Die

genannten Fälle sind in Abbildung 25 am Beispiel einer sich drehenden Scheibe dargestellt.

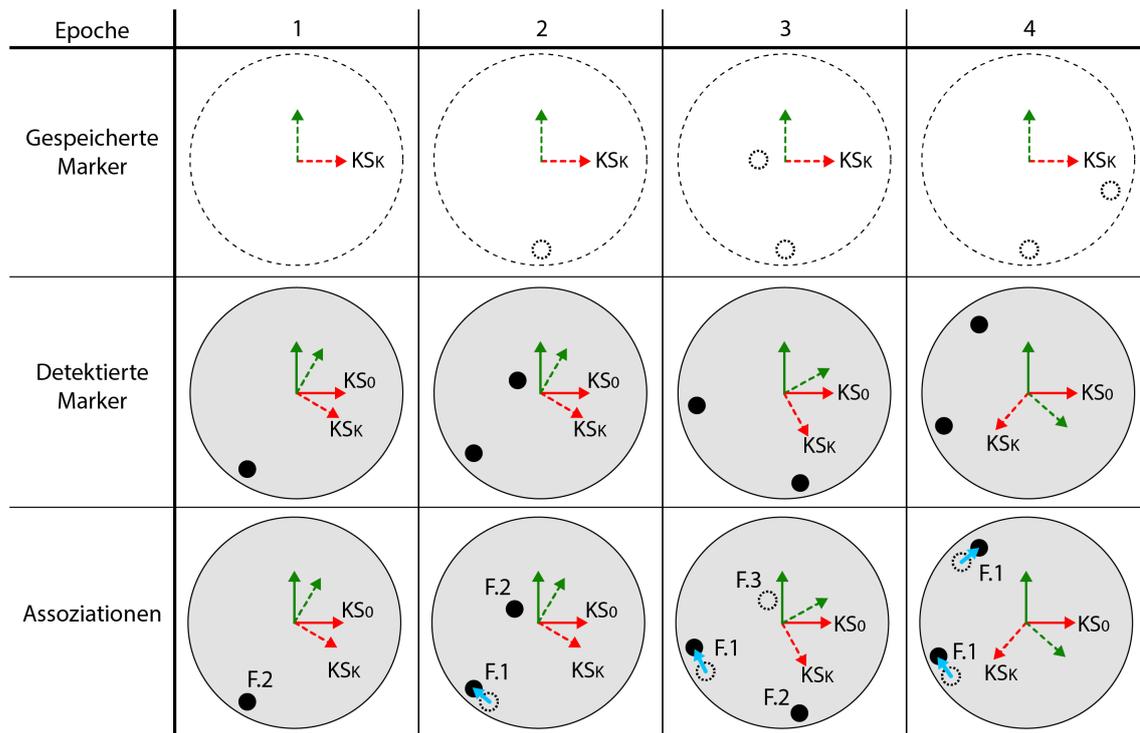


Abbildung 25: Assoziationen zwischen Beobachtungen und gespeicherten Punkten über mehrere Epochen und Fälle der Datenassoziation

Es sind die gespeicherten Marker, die detektierten Marker sowie die resultierenden Assoziationen in vier aufeinanderfolgenden Epochen dargestellt. In der ersten Epoche sind noch keine Punkte gespeichert daher kommen auch keine Assoziation zustande. Der beobachtete Punkt wird in das KS_K rotiert und gespeichert (Fall 2). In der darauf folgenden Epoche werden zwei Punkte detektiert, wobei der untere mit dem gespeicherten Punkt assoziiert wird (Fall 1). Der zweite Punkt wird wiederum zur Menge der gespeicherten Punkte hinzugefügt. Dieser Punkt wird in der dritten Epoche keiner Beobachtung zugeordnet, weswegen er nach der dritten Epoche gelöscht wird (Fall 3). In der vierten Epoche werden beide gespeicherten Punkte mit den beobachteten Punkten assoziiert.

3.9.3 Modell 2: Punkte im Zustandsvektors gespeichert

Das im vorangegangenen Abschnitt präsentierte Filtermodell wird in diesem Modell wie folgt erweitert. Im KS_K gespeicherte Punkte werden nicht mehr als Konstante abgelegt, sondern sind Bestandteil des Zustandsvektors. Dies hat zur Folge, dass einerseits die Unsicherheiten und Korrelationen der Koordinaten dieser Punkte modelliert werden und andererseits die gespeicherten Punkte nicht länger aus den Beobachtungen einer einzelnen Epoche resultieren, sondern fortlaufend durch assoziierte Beobachtungen verbessert wer-

den können. Der erweiterte Zustandsvektor der Epoche k , in dem $s^{(k)}$ Punkte gespeichert wurden, lautet

$$\mathbf{x}^{(k)} = \begin{pmatrix} (\omega_K^{(k)}, \varphi_K^{(k)}, \kappa_K^{(k)}, \dot{\omega}_K^{(k)}, \dot{\varphi}_K^{(k)}, \dot{\kappa}_K^{(k)})^T \\ (P_{1,x}^{(k)}, P_{1,y}^{(k)}, P_{1,z}^{(k)}, \dots, P_{s,x}^{(k)}, P_{s,y}^{(k)}, P_{s,z}^{(k)})^T \end{pmatrix}. \quad (151)$$

Die Anzahl u der Parameter des Zustandsvektors bei m gespeicherten Punkten berechnet sich zu $u = 6 + 3 \cdot m$. Das dynamische System muss um drei Funktionen für jeden gespeicherten Punkt erweitert werden, die deren Prädiktion zu ermöglichen. Für einen gespeicherten Punkt \mathbf{P}_i gelten hierbei die Zusammenhänge

$$f_{P_{i,x}} : P_{i,x,-}^{(k)} = P_{i,x,+}^{(k-1)} \quad (152)$$

$$f_{P_{i,y}} : P_{i,y,-}^{(k)} = P_{i,y,+}^{(k-1)} \quad (153)$$

$$f_{P_{i,z}} : P_{i,z,-}^{(k)} = P_{i,z,+}^{(k-1)}. \quad (154)$$

Hierdurch wird modelliert, dass sich die wahre Position von \mathbf{P}_i im Koordinatensystem der Kugel nicht verändert. Ferner hat kein Störterm Einfluss auf die gespeicherten Koordinaten, was praktisch bedeutet, dass die modellierte Genauigkeit der gespeicherten Punkte durch die Prädiktion unverändert bleibt. Die Struktur von Transitionsmatrix $\Phi^{(k)}$ und Stellmatrix $\mathbf{G}^{(k)}$ erweitern sich zu

$$\Phi_{u \times u}^{(k)} = \begin{pmatrix} 1 & 0 & 0 & \delta t & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \delta t & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & \delta t & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \ddots & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (155)$$

Anderenfalls lauten die Einträge von $\tilde{\mathbf{A}}_{ji}^{(k)}$

$$\tilde{a}_{11,ij} = c_\varphi \cdot c_\kappa \quad (160)$$

$$\tilde{a}_{12,ij} = -c_\varphi \cdot s_\kappa \quad (161)$$

$$\tilde{a}_{13,ij} = s_\varphi \quad (162)$$

$$\tilde{a}_{21,ij} = c_\omega \cdot s_\kappa + s_\omega \cdot s_\varphi \cdot c_\kappa \quad (163)$$

$$\tilde{a}_{22,ij} = c_\omega \cdot c_\kappa - s_\omega \cdot s_\varphi \cdot s_\kappa \quad (164)$$

$$(165)$$

$$\tilde{a}_{23,ij} = -s_\omega \cdot c_\varphi \quad (166)$$

$$\tilde{a}_{31,ij} = s_\omega \cdot s_\kappa - c_\omega \cdot s_\varphi \cdot c_\kappa \quad (167)$$

$$\tilde{a}_{32,ij} = s_\omega \cdot c_\kappa + c_\omega \cdot s_\varphi \cdot s_\kappa \quad (168)$$

$$\tilde{a}_{33,ij} = c_\omega \cdot c_\varphi. \quad (169)$$

mit

$$c_\omega := \cos(\omega), c_\varphi := \cos(\varphi), c_\kappa := \cos(\kappa),$$

$$s_\omega := \sin(\omega), s_\varphi := \sin(\varphi), s_\kappa := \sin(\kappa).$$

Zuletzt wird betrachtet, wie sich der Zustandsvektor und seine VKM verändert, wenn ein neuer Punkt gespeichert oder ein gespeicherter Punkt gelöscht wird. Diese Änderungen erfolgen nach dem Updateschritt, folglich an $\mathbf{x}_+^{(k)}$ bzw. $\Sigma_{\mathbf{xx},+}^{(k)}$. Im Fall eines gespeicherten Punktes, der mit keinem beobachteten Punkt assoziiert wurde, werden die entsprechenden drei Spalten und Zeilen, die den Koordinaten des gespeicherten Punktes zugeordnet werden, aus $\Sigma_{\mathbf{xx}}$ sowie die korrespondierenden Einträge des Zustandsvektors gelöscht. Im Fall eines beobachteten Punktes \mathbf{B}_i , dem kein gespeicherter Punkt zugeordnet wird, werden seine Koordinaten in das \mathbf{KS}_K transformiert und an den Zustandsvektor angehängt. Analog dazu wird auch $\Sigma_{\mathbf{xx},+}^{(k)}$ um drei Zeilen und Spalten erweitert. Sei $\tilde{\mathbf{x}}_+^{(k)}$ der Zustandsvektor und $\tilde{\Sigma}_{\mathbf{xx},+}^{(k)}$ die VKM vor besagter Erweiterung und $\mathbf{R}_{K,+}^{(k)}$ die Rotationsmatrix gebildet aus den Lagewinkeln nach dem Updateschritt, so erfolgt die Erweiterung durch den nicht assoziierten Punkt $\mathbf{B}_i = (B_{i,x}, B_{i,y}, B_{i,z})^T$ mit der VKM $\Sigma_{\mathbf{B}_i\mathbf{B}_i}$ des Zustandsvektors zu

$$\mathbf{x}_+^{(k)} = \begin{pmatrix} \tilde{\mathbf{x}}_+^{(k)} \\ \mathbf{R}_{K,+}^{(k)T} \cdot \mathbf{B}_i \end{pmatrix} \quad (170)$$

und der VKM zu

$$\Sigma_{\mathbf{xx},+}^{(k)} = \begin{pmatrix} & & & \vdots & & \vdots & & \vdots \\ & \tilde{\Sigma}_{\mathbf{xx},+}^{(k)} & & 0 & & 0 & & 0 \\ & & & 0 & & 0 & & 0 \\ \dots & 0 & 0 & & & & & \\ \dots & 0 & 0 & & \mathbf{R}_{K,+}^{(k)T} \cdot \Sigma_{\mathbf{B}_i\mathbf{B}_i} \cdot \mathbf{R}_{K,+}^{(k)} & & & \\ \dots & 0 & 0 & & & & & \end{pmatrix}. \quad (171)$$

Hierdurch werden zwei Annahmen ausgedrückt: einerseits, dass die VKM des neu gespeicherten Punktes der VKM des beobachteten Punktes entspricht (transformiert von \mathbf{KS}_0 nach \mathbf{KS}_K), andererseits, dass zum Zeitpunkt der Speicherung keine Korrelationen zwischen den neuen und alten Zuständen existiert.

3.9.4 Umrechnung in Mausbewegung

Um eine Ausgabe analog zum Trackball zu erzeugen, werden die Rotationsmatrizen der Epochen k und $k - 1$ jeweils nach dem Updateschritt genutzt. Gesucht ist die Änderung der Position eines Punktes C , der sich zum Zeitpunkt $k - 1$ unter dem Mittelpunkt der Kugel befindet. Die Definition von \mathbf{C} folgt aus der Definition des \mathbf{KS}_0 , dessen Y-Achse ungefähr vertikal ist. Die Distanz zwischen \mathbf{C} und dem Kugelzentrum sei c , woraus sich die Koordinaten des Punktes \mathbf{C} im \mathbf{KS}_0 in der Epoche $k - 1$ zu

$$\mathbf{C}^{(k-1)} = (0, -c, 0)^T \quad (172)$$

ergeben. Nach der inkrementellen Rotation der Kugel lauten sie

$$\mathbf{C}^{(k)} = \mathbf{R}_{K,+}^{(k)} \cdot \mathbf{R}_{K,+}^{(k-1)T} \cdot \mathbf{C}^{(k-1)}. \quad (173)$$

Der Maus-Output $\Delta \mathbf{m}^{(k)} = (\Delta m_x^{(k)}, \Delta m_y^{(k)})^T$ der Epoche k ergibt sich zu

$$\begin{pmatrix} \Delta m_x^{(k)} \\ \Delta m_y^{(k)} \end{pmatrix} = \begin{pmatrix} C_x^{(k)} \\ -C_z^{(k)} \end{pmatrix}. \quad (174)$$

Diese Vorgangsweise beruht auf der Annahme, dass die Änderung der Y-Koordinate von \mathbf{C} aufgrund der inkrementellen Rotation vernachlässigt werden kann. Der Fehler dieser Vereinfachung steigt mit der Größe der inkrementellen Rotation (größere Drehraten oder längere Zeitspanne zwischen den Epochen), sowie mit einem kleiner werdenden Kugelradius. Der Parameter c beeinflusst die Skalierung des Ausgabevektors $\Delta \mathbf{m}^{(k)}$, jedoch nicht dessen Richtung. Das Problem der Skalierung besteht auch bei dem beispielhaften Eingabegerät, da der Ausgabevektor des Trackballs vom Betriebssystem skaliert wird. In der Beispielapplikation kann die eingehende Mausbewegung erneut skaliert werden, um dies abzufangen.

4 Experimente

Um die Funktionalität des entwickelten Programmes zu validieren, sowie seine Parameter und weitere Einflussfaktoren (Anzahl der Marker auf der Kugel, Bildrauschen und Belichtung, Kamerakonfiguration und Filtermodell) zu untersuchen, werden mehrere Datensätze ausgewertet. In diesem Kapitel werden zunächst die untersuchten Datensätze vorgestellt und die Kalibrierung des Systems dargelegt. In Abschnitt 4.4 werden die berechneten, inneren Genauigkeiten der Objektkoordinaten der Markerzentren betrachtet, die aus der stochastischen Herangehensweise resultieren. Anschließend wird die Durchführung der Experimente beschrieben und die Ergebnisse werden präsentiert und analysiert. Abschließend werden die Ergebnisse der Experimente zusammengefasst.

4.1 Datensätze

Die in den Experimenten genutzten Datensätze lassen sich in zwei Gruppen unterteilen. Der erste Datensatz entstammt einem realen Versuchsaufbau, die weiteren Datensätze wurden mit der Rendering-Software **Blender** [Blender Foundation, 1995] erzeugt und sind somit synthetischen Ursprungs. Der reale Datensatz umfasst die Bilder des Schachbrettmusters mit denen die Kamera kalibriert wird, die Aufnahmen des Eingabegerätes und die gemessenen Distanzen zwischen den Passpunkten welche für die 3D-Rekonstruktion verwendet werden sowie ein Video eines Spieldurchlaufes. Jeder synthetischer Datensatz besteht aus dem gerenderten Video sowie den Parametern der inneren und äußeren Orientierung der Kamera, den Koordinaten der Marker auf dem Rahmen sowie den Kugelparametern jeweils im \mathbf{KS}_0 .

4.1.1 Reale Aufnahme

Der erste Datensatz wurde unter realen Bedingungen erzeugt. Hierzu wurde das in Kapitel 1 vorgestellte Eingabegerät verwendet. Auf die Kugel wurden 32 schwarze Punkte geklebt, wobei jeder Punkt einen Durchmesser von 4 cm bemisst und die Entfernung zwischen benachbarten Punkten etwa 40 cm beträgt. Bei der Platzierung wurde grob darauf geachtet die Punkte möglichst gleichmäßig auf der Kugel zu verteilen. Zusätzlich wurden vier farbkodierte Marker auf die Innenseiten der vorderen beiden Stabilisatoren des Rahmens geklebt. Der Versuchsaufbau ist in Abbildung 26 dargestellt.

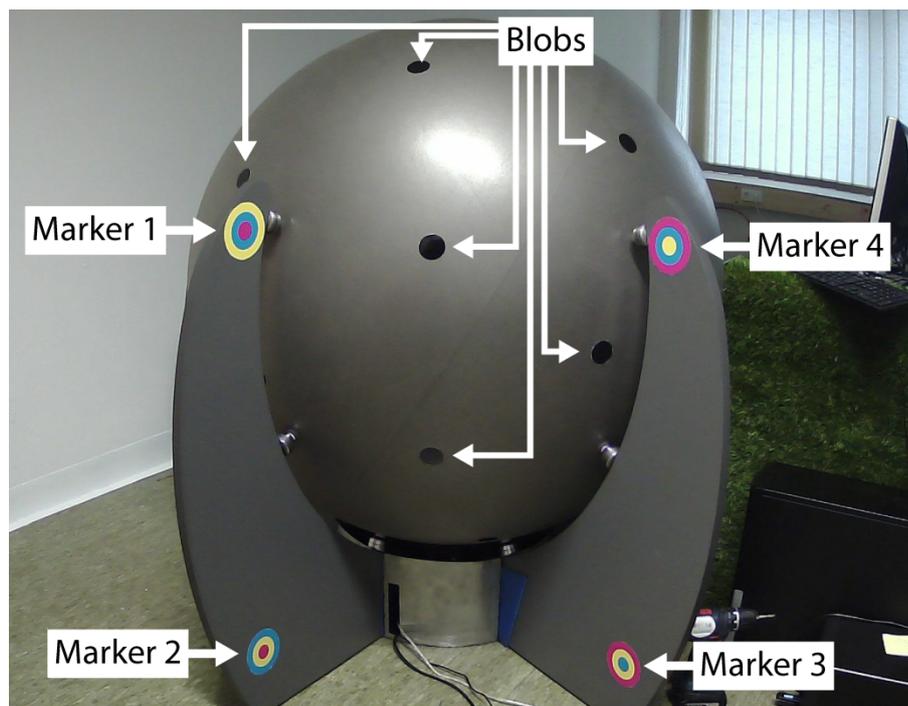


Abbildung 26: Eingabegerät mit Markern

Die eingesetzte Kamera der Marke 'ELP' (ELP-USBFHD01M-MF40) liefert maximal 60 Bilder pro Sekunde bei einer Auflösung von 1920×1080 Pixeln, wobei die Bildwiederholrate und die Auflösung konfigurierbar sind. Mit dieser Kamera wurden für die Kamera- kalibrierung Aufnahmen eines Schachbrettmusters angefertigt und das Eingabegerät aus mehreren Standpunkten aus aufgenommen. Der Datensatz umfasst weiterhin die Entfernungen zwischen den Markern auf dem Rahmen. Diese wurden mit einem Messband gemessen und sind in Tabelle 1 aufgelistet. Bei der Messung wurde eine Messungenauigkeit (Standardabweichung) von einem Millimeter angenommen. Die Nummerierung bezieht sich auf Abbildung 26. Die Durchführung der Kamera- und Systemkalibrierung wird in Abschnitt 4.3 erläutert.

Marker ID (von)	Marker ID (nach)	Distanz [mm]
1	2	750 ± 1
1	4	650 ± 1
2	3	650 ± 1
3	4	750 ± 1

Tabelle 1: Distanzen zwischen Markern

Mit der Kamera wurde weiterhin ein Durchlauf des Spiels in Form einer 30 Sekunden langen Videodatei aufgezeichnet. Um die Verarbeitungszeit zu verkürzen, wurde die Auflösung der Kamera hierbei auf 640×480 Pixel und die Bildwiederholrate auf 30 Bil-

der pro Sekunde reduziert. Die Videosequenz umfasst folglich ca. 900 Bilder. Abbildung 27 zeigt ein Bild der laufenden Auswertung des realen Datensatzes (im Folgenden als Datensatz R_1 abgekürzt). Eingezeichnet sind die Konturen der detektierten Blob-Marker in weiß, die Zentren der farbkodierten Marker in grün, sowie die in das Bild projizierten Koordinaten der gespeicherten Blob-Marker der letzten Epochen in rot, wobei kleinere Punkte weiter zurückliegende Epochen darstellen.

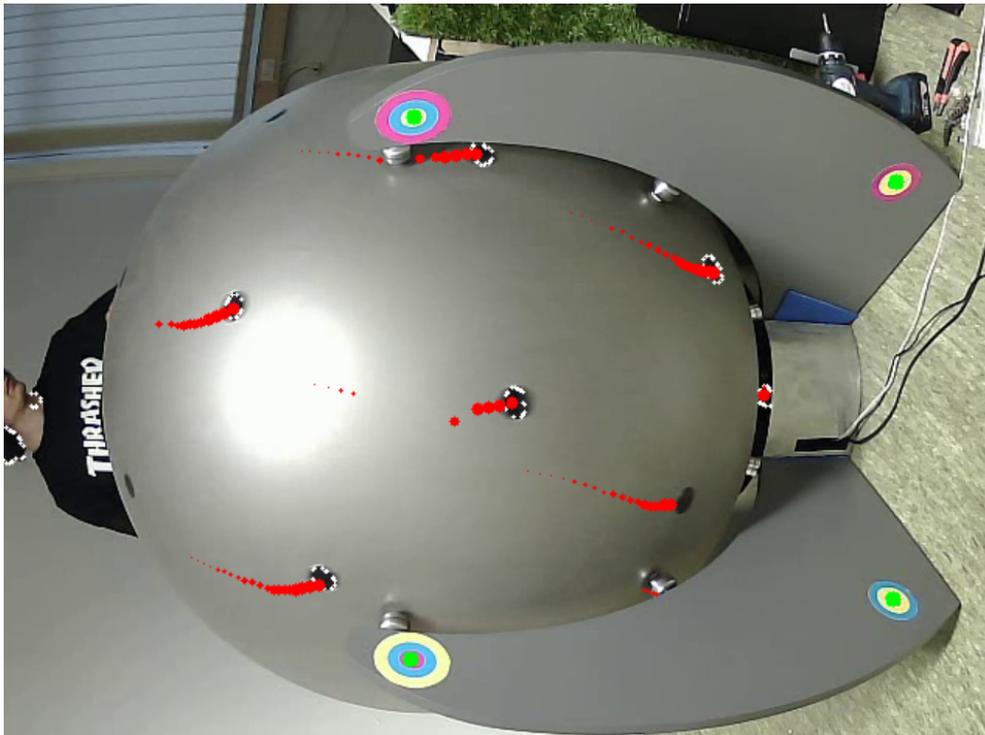


Abbildung 27: Bild der laufenden Auswertung des Datensatzes R_1

Als Referenz für die Ergebnisse der Auswertung des Datensatzes R_1 dient die vom Trackball aufgezeichnete Mausebewegung. Die Mausebewegung in Pixeln ist jedoch abhängig von der Datenrate, mit der die Mausebewegungen eingehen. Da diese Datenrate bei den Referenzbewegungen nicht konstant ist, werden die Referenzbewegungen sowie die berechneten Mausebewegungen durch die jeweiligen Zeitspannen dividiert, die zwischen den aufgezeichneten bzw. berechneten Daten liegen. Die hieraus resultierenden Mausgeschwindigkeiten in Pixeln pro Sekunde sind unabhängig von der Aufnahmezeit, was einen aussagekräftigen Vergleich erlaubt. Zusätzlich sind die gefilterten Mausgeschwindigkeiten gegenüber der Referenz zeitlich um einen Wert Δt versetzt und es liegt die unbekannte Skalierung s zwischen den berechneten Werten und der Referenz vor. Dies ist schematisch in Abbildung 28 abgebildet.

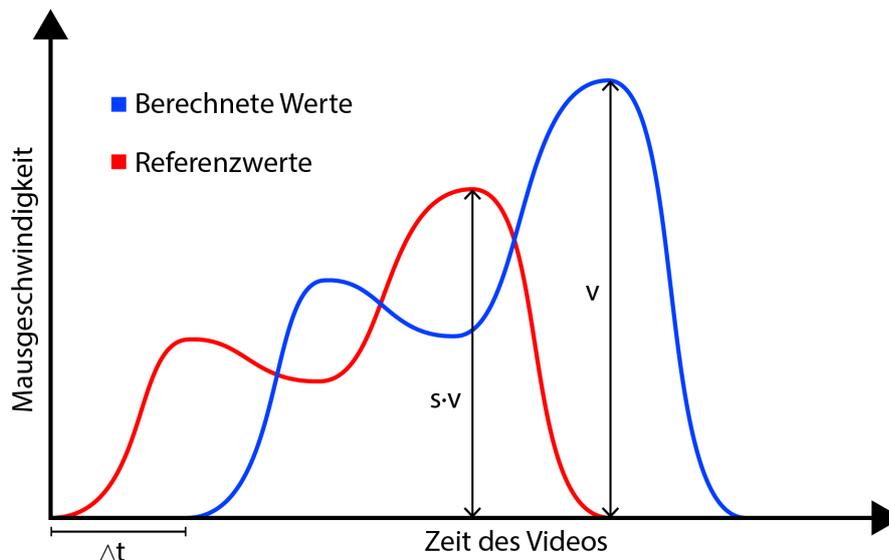


Abbildung 28: Darstellung der zeitlichen Verschiebung und der Skalierung zwischen Referenz und berechneten Ergebnissen

Aus diesem Grund werden die Ergebnisse zunächst mit verschiedenen Werten für Δt auf der Zeitachse verschoben und jeweils eine lineare Regression durchgeführt. Hierbei wird angenommen dass die n gemessenen Daten m_i und die Referenzwerte r_i mit $i = 1, \dots, n$ den Zusammenhang

$$r_i = a + s \cdot m_i \quad (175)$$

aufweisen. Die lineare Regression liefert die optimalen Werte für a und s sowie den Korrelationskoeffizient nach Pearson ρ zwischen m_i und r_i ([Cramer und Howitt, 2004]). Der Korrelationskoeffizient nach Pearson erlaubt eine Aussage darüber, wie gut die Datensätze für eine zeitliche Verschiebung von Δt übereinstimmen. Es wird angenommen, dass der wahre zeitliche Offset zwischen den berechneten Werten und der Referenz den maximalen positiven Korrelationskoeffizient ρ_{max} erzielt. Dieser wird in den Experimenten als Qualitätsmaß des entsprechenden Durchlaufes genutzt. Für den grafischen Vergleich werden die berechneten Werte für die Skalierung s und das zeitliche Offset Δt angebracht, um die Verläufe aneinander anzupassen.

4.1.2 Synthetische Datensätze

Da für den ersten Datensatz keine direkten Referenzwerte für die Lagewinkel und Drehraten existieren, wurden synthetische Datensätze erzeugt. Mit der Rendering-Engine **Blender** wurde der reale Versuchsaufbau nachgestellt. In der modellierten Szene werden die innere und äußere Orientierung der Kamera und die Bewegung der Kugel vorgegeben. Die Auflösung der modellierten Kamera beträgt entsprechend dem realen Datensatz 640×480 Pixel bei einer Bildrate von 30 Bildern pro Sekunde. Eine Kamerakalibrierung ist nicht notwendig, da die innere Orientierung vorgegeben wird und keine Verzeichnung auftritt. Die vorgegebenen Parameter sind in Tabelle 2 aufgeführt.

Konf.	x_0 [px]	y_0 [px]	c [px]	X_0 [m]	Y_0 [m]	Z_0 [m]	ω_0 [°]	φ_0 [°]	κ_0 [°]
nah	320	240	320	0	-0.45	1.27	-170	0	0
fern	320	240	640	0	0	2.35	180	0	90

Tabelle 2: Vorgegebene innere und äußere Orientierung der Kamera im Modell

Modelliert werden die Konfigurationen **nah** und **fern**, wobei die Konfiguration **fern** näherungsweise der Kamerakonfiguration des realen Datensatzes R_1 entspricht (vgl. Abbildung 27 und Abbildung 29). Abbildung 29 zeigt beispielhaft ein Bild des synthetischen Datensatzes S_2 , wobei die zur Laufzeit eingezeichneten Elemente denen in Abbildung 27 entsprechen. Die Belichtung der Szene ist dem realen Datensatz nachempfunden. Bei der Konfiguration **nah** befindet sich die Kamera näher an der Kugel. Gleichzeitig wurde der Öffnungswinkel der Kamera erhöht, damit alle Marker auf dem Rahmen im Bild sichtbar sind. Die Koordinaten der Marker auf dem Rahmen im \mathbf{KS}_0 sowie die Kugelparameter werden durch den geometrischen Aufbau der Szene festgelegt, wodurch die Kalibrierungsphase für die synthetischen Datensätze entfällt. Auch die äußere Orientierung der Kamera ist bekannt und wird als Referenz für die Ergebnisse der Initialisierungsphase genutzt. Die vorgegebene Bewegung der Kugel, parametrisiert durch deren Lagewinkel, ist in Abbildung 30 dargestellt, wobei die vorgegebenen Lagewinkel aus quadratischen Funktionen zusammengesetzt sind, um die Bewegung möglichst realistisch zu gestalten, da somit keine Sprünge in den Drehraten oder den Lagewinkeln auftreten. Die Dauer der vorgegebenen Bewegung und somit auch die Länge der Videos beträgt 10 Sekunden.

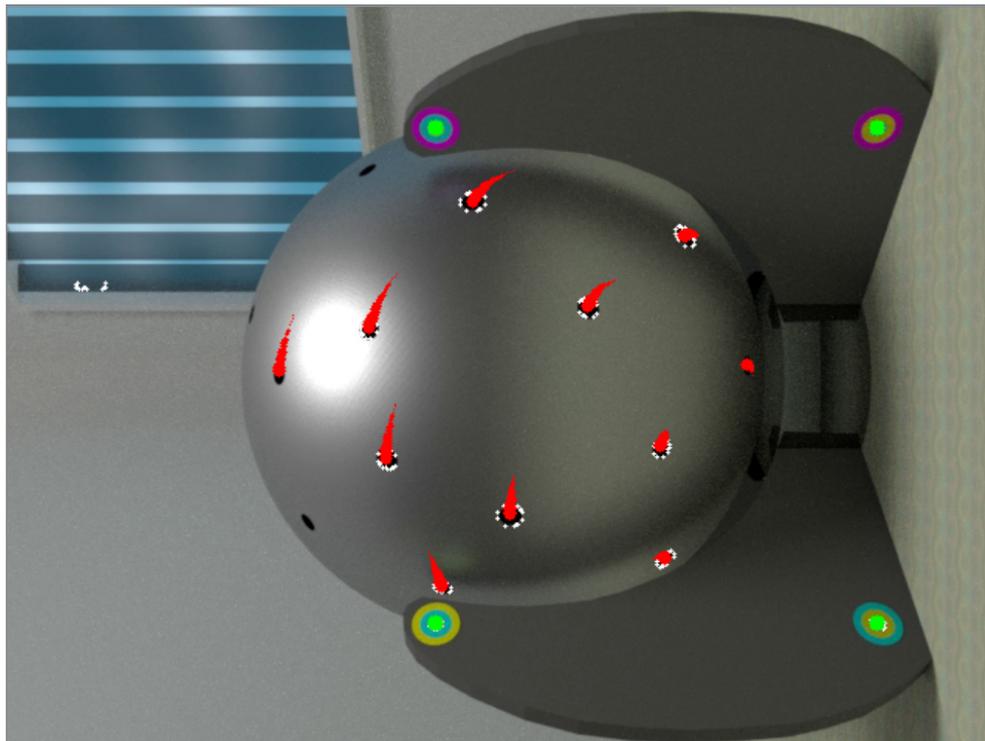


Abbildung 29: Bild der laufenden Auswertung des synthetischen Datensatzes S_2

Aus diesen Kurven lassen sich Referenzwerte für die Lagewinkel, die Drehraten sowie für die Mausgeschwindigkeiten berechnen. Die Bewegung der Kugel ist in allen synthetischen Datensätzen identisch. Die Datensätze unterscheiden sich in der Anzahl der auf der Kugel angebrachten Marker bzw. in der Konfiguration der Kamera. Eine Übersicht über die synthetischen Datensätze liefert Tabelle 3.

Datensatz	Anzahl der Marker	Konfiguration
S_1	26	fern
S_2	50	fern
S_3	74	fern
S_4	50	nah

Tabelle 3: Eigenschaften der synthetischen Datensätze

Als Qualitätsmaß für die Auswertung der synthetischen Datensätze wird analog zur ersten Gruppe der maximale Korrelationskoeffizient ρ_{max} zwischen den berechneten Mausgeschwindigkeiten und den Referenzgeschwindigkeiten genutzt. Im Gegensatz zum ersten Datensatz gibt es jedoch keine zeitliche Verschiebung der berechneten Werte zur Referenz und der Skalierungsfaktor entfällt, weswegen $\Delta t = 0$ und $a = 0$ sowie $s = 1$ gelten. Weiterhin wird die Wurzel des mittleren quadratischen Fehlers (RMSE) als Qualitätsmaß genutzt (vgl. [Cramer und Howitt, 2004]) wobei diese für den Fehler der Lagewinkel sowie

den Fehler der Drehraten berechnet werden. Der RMSE wird für einen Datensatz mit n berechneten Werten und n korrespondierenden Referenzwerten allgemein mit der Gleichung

$$\text{RMSE} = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^n (r_i - m_i)^2} \quad \text{mit:} \quad i = 1, \dots, n \quad (176)$$

berechnet. Hierbei bezeichnet m_i den i -ten, berechneten Wert und r_i den zugehörigen Referenzwert. Weiterhin wird für die Lagewinkel, die Drehraten und die Mausgeschwindigkeiten der kombinierte RMSE als Qualitätsmaß genutzt. Der kombinierte RMSE der Lagewinkel aus den berechneten Lagewinkeln ω_i, φ_i und κ_i und den korrespondierenden Referenzwerten $\omega_{i,r}, \varphi_{i,r}$ und $\kappa_{i,r}$ zu

$$\text{RMSE} = \sqrt{\frac{1}{n} \cdot \sum_{i=1}^n (\omega_i - \omega_{i,r})^2 + (\varphi_i - \varphi_{i,r})^2 + (\kappa_i - \kappa_{i,r})^2} \quad \text{mit:} \quad i = 1, \dots, n. \quad (177)$$

Die Berechnung für den kombinierten RMSE der Drehraten und Mausgeschwindigkeiten erfolgt analog.

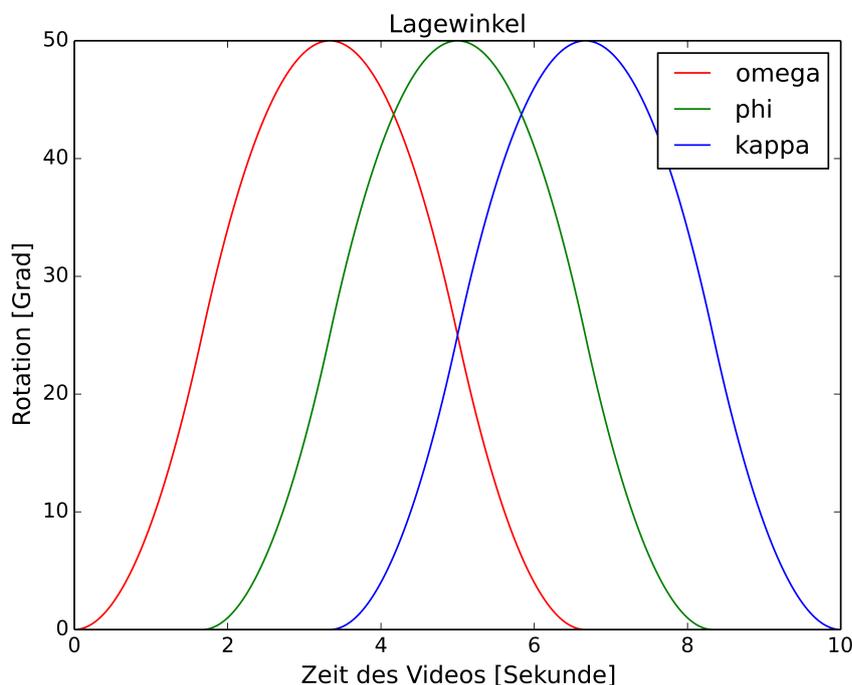


Abbildung 30: Vorgegebene Lagewinkel der synthetischen Datensätze

4.2 Wahl der Programmparameter

In Tabelle 4 werden die Parameter aufgeführt, die bei der Auswertung der Experimente angesetzt werden.

Symbol	Wert	Beschreibung
3D-Rekonstruktion		
d_{max}	10 [cm]	Maximale Distanz für Schnitt der Bildstrahlen
Markerdetektion (Kugel)		
u_M	10 [–]	Schrittweite der Masken
u_B	30 [–]	Breite der Masken
m	10 [–]	Anzahl der Masken
F_{min}	40 [px^2]	Minimaler Flächeninhalt der Blobs
F_{max}	300 [px^2]	Maximaler Flächeninhalt der Blobs
q_{min}	0,95 [–]	Bedingung für die Form
Markerdetektion (Rahmen)		
s_{min}	60 [–]	Minimale Sättigung
s_{max}	255 [–]	Maximaler Sättigung
s_{min}	110 [–]	Minimaler value-Wert
s_{max}	255 [–]	Maximaler value-Wert
Filterung		
g	30 [mm]	Maximale Distanz für Datenassoziation

Tabelle 4: Wahl der Programmparameter

Die hier aufgeführten Parameter wurden empirisch unter Berücksichtigung der Geometrie bestimmt. Der Parameter d_{max} ist so gewählt, dass alle Korrespondenzen der Blob-Marker gefunden werden und gleichzeitig keine ungültigen Schnittpunkte resultieren. Für eine optimale Wahl der Parameter der Markerdetektion wurde die Markerdetektion auf mehrere Bilder der Videosequenz angewendet und die Parameter so gewählt, dass möglichst alle Marker erkannt werden, es jedoch zu möglichst wenig falschen Detektionen kommt. Zuletzt ist g so gewählt, dass die Datenassoziation der Filterung möglichst nur korrekte Assoziationen erzielt. Die Wahl der Parameter ist auf die untersuchten Datensätze angepasst und von der Konfiguration der eingesetzten Kamera sowie Geometrie des Eingabegerätes abhängig. Zuletzt spielt die maximale Drehrate sowie die Bildrate der Kamera eine Rolle bei der Wahl von g . Eine höhere Drehrate hat, wie auch eine geringere Bildrate, zur Folge, dass sich ein Blob-Marker zwischen zwei Epochen weiter bewegt. Hierdurch muss auch g erhöht werden.

4.3 Kalibrierung des untersuchten Eingabegerätes

Die Kalibrierung des untersuchten Eingabegerätes erfolgt wie in Abschnitt 3.3 beschrieben. Im ersten Schritt wird die eingesetzte Kamera (vgl. Abschnitt 4.1.1), wie in Abschnitt 2.4.5 beschrieben, mit mehreren Bildern eines Schachbrettmusters zweifach kalibriert und zwar mit 19 Bildern der Auflösung 1920x1080 Pixel (für die Systemkalibrierung) und ein zweites Mal mit 19 Bildern der Auflösung 640x480 Pixel (für die Initialisierungs- und Nutzungsphase). Das genutzte Schachbrett weist 11×8 Eckpunkte auf, die Kantenlänge einer Kachel beträgt 86 mm. Abbildung 31 zeigt die Positionen der Kamera, aus denen die Bilder mit der Auflösung 640x480 Pixel aufgenommen wurden, sowie die Passpunkte, die den Eckpunkten des Schachbrettmusters gleichen. Die Anordnung der Bilder mit der Auflösung 1920x1080 Pixel folgt der gleichen Struktur.

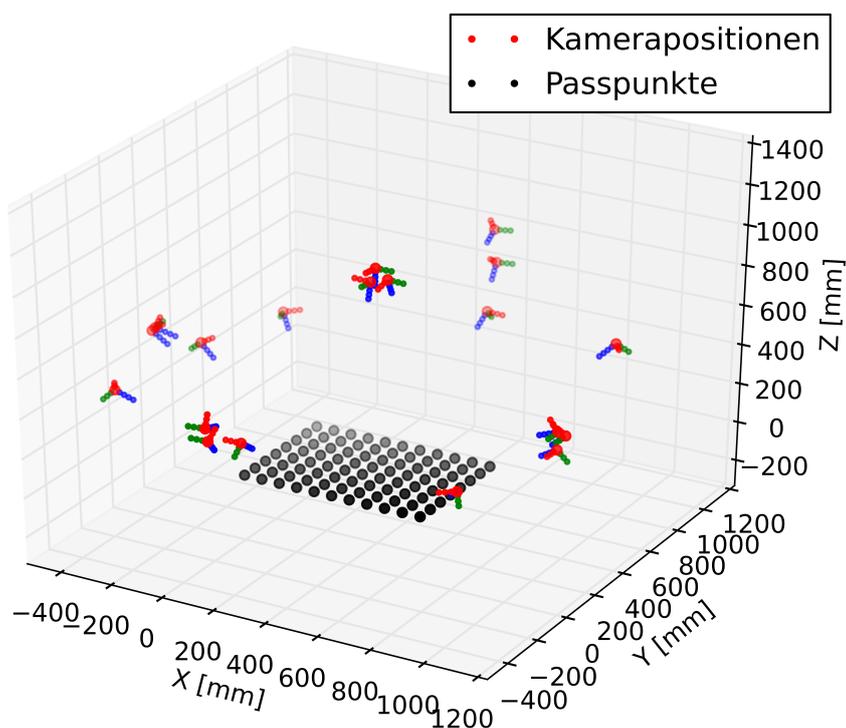


Abbildung 31: Kamerapositionen und Passpunkte (Kamerakalibrierung)

Im Anhang befindet sich eine Darstellung, welche neben den Passpunkten und Kamerapositionen auch die aufgenommenen Bilder umfasst. Die aus der Kalibrierung resultierenden Parameter der inneren Orientierung sowie die der Verzeichnung sind in Tabelle 5 aufgeführt.

Auflösung [px]	x_0 [px]	y_0 [px]	c [px]	m_y [-]	
1920x1080	967,4	665,9	1404,3	0,99816	
640x480	321,1	300,3	624,7	1,00085	
Auflösung [px]	k_1 [-]	k_2 [-]	k_3 [-]	p_1 [-]	p_2 [-]
1920x1080	-0,49121	0,30472	-0,11262	-0,00014	-0,00015
640x480	-0,49141	0,29957	-0,11383	-0,00369	-0,00077

Tabelle 5: Parameter der inneren Orientierung und der Verzeichnung

Um die Qualität der Kamerakalibrierung beurteilen zu können, werden die mittleren Rückprojektionsfehler (MRPF) berechnet. Der MRPF entspricht dem mittleren euklidischen Abstand zwischen den detektierten Bildkoordinaten und den in die Bildebenen projizierten und korrigierten Passpunkten. Die MRPF sind in Tabelle 6 aufgeführt. Für beide Auflösungen ist der MRPF geringer als ein Pixel, was auf eine ausreichend gute Konfiguration der Aufnahmen schließen lässt. Die VKM der berechneten äußeren Orientierungen der aufgenommenen Bilder sowie die VKM der ausgeglichenen Kameraparameter und der Verzeichnung können aufgrund der genutzten Implementierung nicht angegeben werden.

Auflösung [px]	MRPF [px]
1920x1080	0,080
640x480	0,147

Tabelle 6: Mittlere Rückprojektionsfehler der Passpunkte nach der Kamerakalibrierung

Im zweiten Schritt der Kalibrierungsphase wird die Systemkalibrierung durchgeführt. Hierzu wird das Eingabegerät aus sechs Perspektiven mit einer Auflösung von 1920×1080 Pixeln fotografiert. Es wird eine höhere Auflösung als in der Initialisierungs- und Nutzungsphase verwendet, da mit dieser die Markerzentren mit einer höheren Genauigkeit bestimmt werden können und die hiermit verbundene längere Berechnungszeit vertretbar ist, da die Auswertung einmalig durchgeführt wird. Aus den gemessenen Distanzen (vgl. Tabelle 1) ergeben sich die Koordinaten der Passpunkte in dem gemeinsamen Koordinatensystem (vgl. Abschnitt 3.3), wobei die resultierenden, in Tabelle 7 aufgeführten Koordinaten zunächst als fehlerfrei angenommen werden. Es folgt das in Abschnitt 3.3.1

Passpunktnummer	X [mm]	Y [mm]	Z [mm]
1	0	750	0
2	0	0	0
3	650	0	0
4	650	750	0

Tabelle 7: Objektkoordinaten der Passpunkte für die Bestimmung der Punktnummern

vorgestellte Verfahren mit dem die Punktnummern aller detektierter Marker bestimmt werden.

In Abbildung 32 sind die hieraus resultierenden Zwischenergebnisse für alle aufgenommenen Bilder dargestellt.

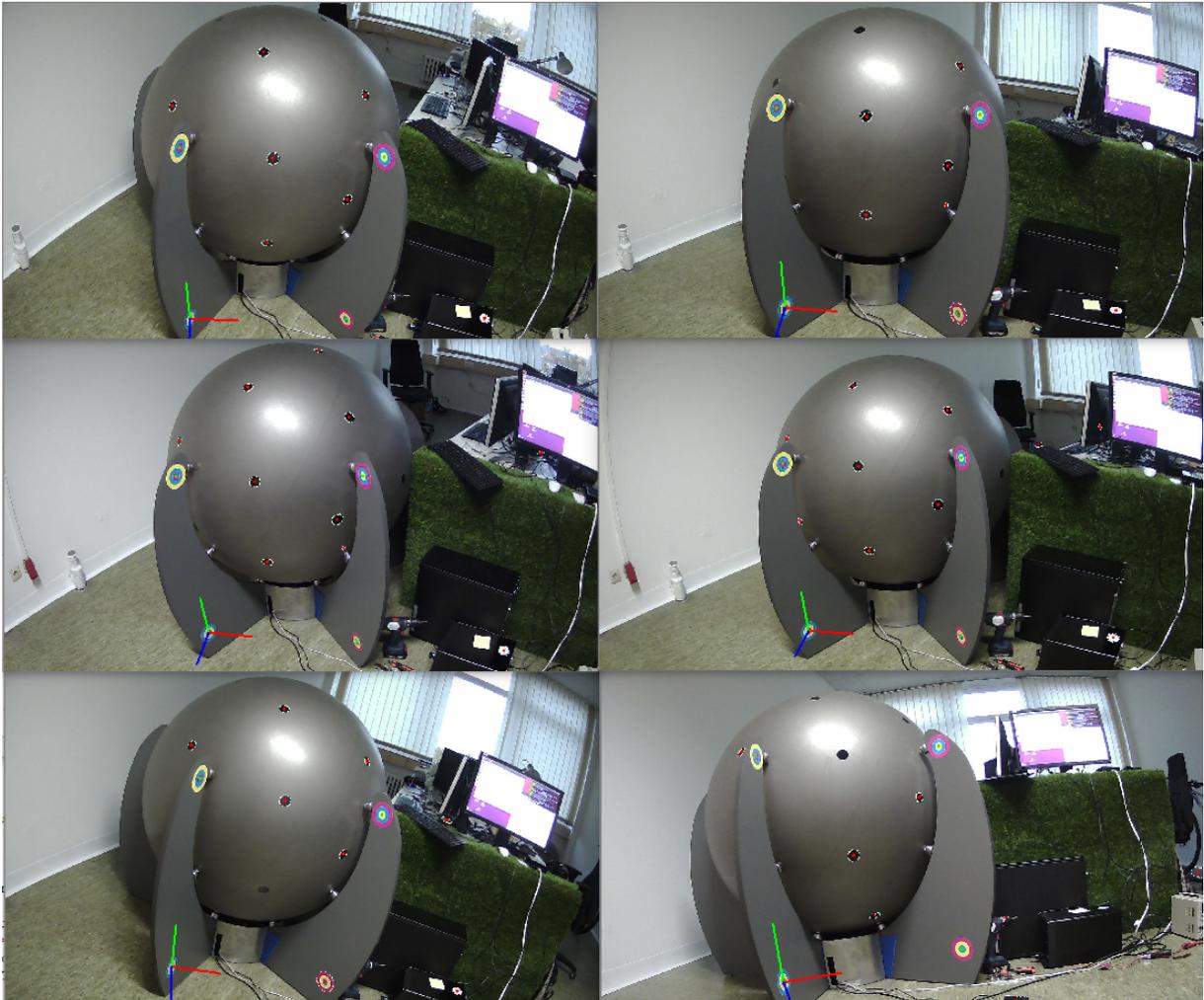


Abbildung 32: Aufnahmen des Eingabegerätes für die 3D-Rekonstruktion

Die Abbildung zeigt die detektierten Markerzentren sowie das gemeinsame Koordinatensystem, das seinen Ursprung aufgrund der Wahl der Objektkoordinaten der Passpunkte zunächst im unteren linken Marker auf dem Rahmen hat. In Abbildung 33 sind die entsprechenden Positionen der Kamera sowie die ermittelten Koordinaten der Markerzentren dargestellt, die im Zuge des Verfahrens zur Findung der Korrespondenzen berechnet werden. Zusätzlich ist die Kugel dargestellt die durch Ausgleichung aus den Näherungswerten der Markerzentren auf der Kugel geschätzt wird.

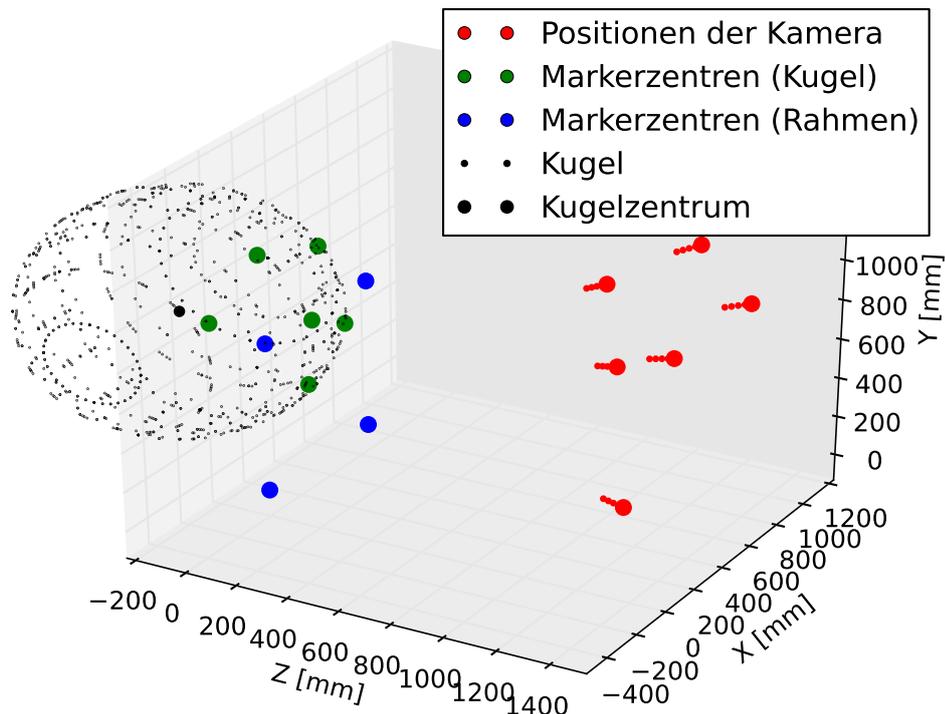


Abbildung 33: Positionen der Kamera und der Markerzentren

Nachdem die Punktnummern bestimmt wurden, wird die BBA durchgeführt. Dabei werden die Marker als Passpunkte eingeführt um den Block zu lagern. Die hierzu verwendeten Koordinaten der Markerzentren auf dem Rahmen sowie deren Standardabweichungen sind in Tabelle 8 dargestellt. Die Unsicherheiten wurden hierbei so gewählt, dass sich eine zwangsfreie Lagerung des Blockes ergibt. Es werden sechs Koordinaten mit einer hohen

Passpunktnummer	X [mm]	Y [mm]	Z [mm]	σ_X [mm]	σ_Y [mm]	σ_Z [mm]
1	0	750	0	50	50	0,1
2	0	0	0	0,1	0,1	0,1
3	650	0	0	50	0,1	0,1
4	650	750	0	50	50	50

Tabelle 8: Objektkoordinaten der Passpunkte für die BBA

Genauigkeit angesetzt, genau so viele, wie notwendig sind, um die Rotation und Translation des gemeinsamen Koordinatensystemes zu definieren. Zusätzlich gehen die gemessenen Distanzen zwischen den Passpunkten als Beobachtungen ein. Die Beobachtungen entsprechen den Distanzen aus Tabelle 1.

Neben den äußeren Orientierungen der Kamera sowie den ausgeglichenen Koordinaten der Passpunkte resultieren die Koordinaten der Markerzentren auf der Kugel, sowie die zugehörigen VKM. In Tabelle 9 ist jeweils die Wurzel der mittleren quadratischen Standardabweichung RMSS der ermittelten Objektkoordinaten für die X-, Y-, und Z-Koordinaten angegeben

RMSS _X [mm]	RMSS _Y [mm]	RMSS _Z [mm]
0,8714	0,6905	1,3093

Tabelle 9: RMSS der Objektkoordinaten der Markerzentren (BBA)

Aus diesen Ergebnissen folgt, dass die Z-Koordinaten der Markerzentren bei der Konstellation der aufgenommenen Bilder am ungenauesten bestimmt werden können. Dies ist plausibel, da alle Aufnahmen von einer Seite erfolgen, damit die Marker auf dem Rahmen in den Bildern sichtbar sind. In Tabelle 10 ist jeweils der RMSS der ermittelten äußeren Orientierung aller Bilder angegeben

RMSS _X [mm]	RMSS _Y [mm]	RMSS _Z [mm]	RMSS _ω [°]	RMSS _φ [°]	RMSS _κ [°]
4,3	3,7	2,6	0,15	0,15	0,05

Tabelle 10: RMSS der Objektkoordinaten der Markerzentren (BBA)

Aus diesen Ergebnissen folgt, dass die Z-Koordinaten der Projektionszentren am genauesten bestimmt werden können, was damit zu begründen ist, dass Änderungen der Z-Koordinaten die größte Verschiebung der Bildkoordinaten der Passpunkte verursachen. Gleiches gilt für die Verdrehung um die Z-Achse, weswegen auch diese im Mittel am genauesten bestimmt werden kann.

Die Koordinaten und VKM der Markerzentren auf der Kugel werden im Anschluss genutzt um eine ausgleichende Kugel zu schätzen. Die inneren Genauigkeiten, die aus dieser Ausgleichung folgen sind in Tabelle 11 aufgeführt.

Korrelation [-]	X_K	Y_K	Z_K	R_K
Y_K	-0,333			
Z_K	-0,460	0,858		
R_K	0,457	-0,858	-0,998	
Standardabweichung [mm]	0,7	0,9	3,3	2,7
Wert [mm]	324,4	595,4	-562,9	584,1

Tabelle 11: Korrelationen, Standardabweichungen und Zahlenwerte der ermittelten Kugelparameter

Die Ergebnisse zeigen, dass die Z-Koordinate des Kugelmittelpunktes und der Radius mit der größten Unsicherheit bestimmt wurden und diese Parameter gleichzeitig stark korreliert sind. Dies ist mit der Verteilung der Blob-Marker zu begründen, die für diese Berechnung verwendet wurden. Die in der BBA verwendeten Bilder (vgl. Abbildung 32) wurden alle von der Vorderseite des Eingabegerätes aufgenommen, weswegen auch die resultierenden Markerzentren auf einer Seite der Kugel liegen. Dies führt zu einer ungenauen Bestimmung des Kugelradius und der Entfernung der Kugel (Z-Komponente) sowie einer starken negativen Korrelation. Eine Untersuchung der äußeren Genauigkeit ist nicht möglich, da für die Kugelparameter keine Referenzwerte vorliegen. Die erzielte Genauigkeit der Kugelparameter ist ausreichend genau, kann jedoch deutlich gesteigert werden, wenn das Eingabegerät für die Systemkalibrierung von allen Seiten aufgenommen wird.

4.4 Innere Genauigkeit der Blob-Marker

Im ersten Experiment wird der Datensatz R_1 ausgewertet, wobei der Fokus zunächst auf die Unsicherheiten und Korrelationen der Zwischenergebnisse gelegt wird. Zunächst werden die Ergebnisse des RRWS betrachtet, mit dem in der Initialisierungsphase die äussere Orientierung der Kamera bestimmt wird. Die Korrelationen und Standardabweichungen der Position und Lage der Kamera sind in Tabelle 12 aufgeführt.

Kor. [-]	X_0	Y_0	Z_0	ω_0	φ_0	κ_0
Y_0	0,060					
Z_0	-0,093	-0,806				
ω_0	-0,083	-0,972	0,899			
φ_0	-0,994	-0,060	0,090	0,083		
κ_0	-0,300	-0,002	0,012	0,032	0,339	
Std.-Abw.	1,9 [mm]	1,3 [mm]	1,1 [mm]	0,07 [°]	0,09 [°]	0,02 [°]

Tabelle 12: Korrelationen und Standardabweichungen der äußeren Orientierung

Die hohen Korrelationen zwischen X_0 und φ_0 sowie zwischen Y_0 und ω_0 sind plausibel, da Änderungen dieser Parameter jeweils ähnliche Auswirkungen auf die Bildkoordinaten der Marker auf dem Rahmen haben. Ebenfalls ist erklärbar, dass die Verschiebung des Projektionszentrums entlang der z-Achse Z_0 bzw. die Rotation um die z-Achse κ_0 mit der höchsten Genauigkeit bestimmt werden, da sie die größten Auswirkungen auf die Bildkoordinaten der Marker haben.

Zuletzt werden die Genauigkeiten und Korrelationen der berechneten Koordinaten der Blob-Marker im \mathbf{KS}_0 betrachtet. Hierzu wird beispielhaft das in Abbildung 34 dargestellte Bild analysiert, in welchem zur Laufzeit fünf Marker auf der Kugel detektiert werden.

Die Objektkoordinaten der Markerzentren sowie die Korrelationskoeffizienten und Standardabweichungen sind in Tabelle 13 aufgeführt.

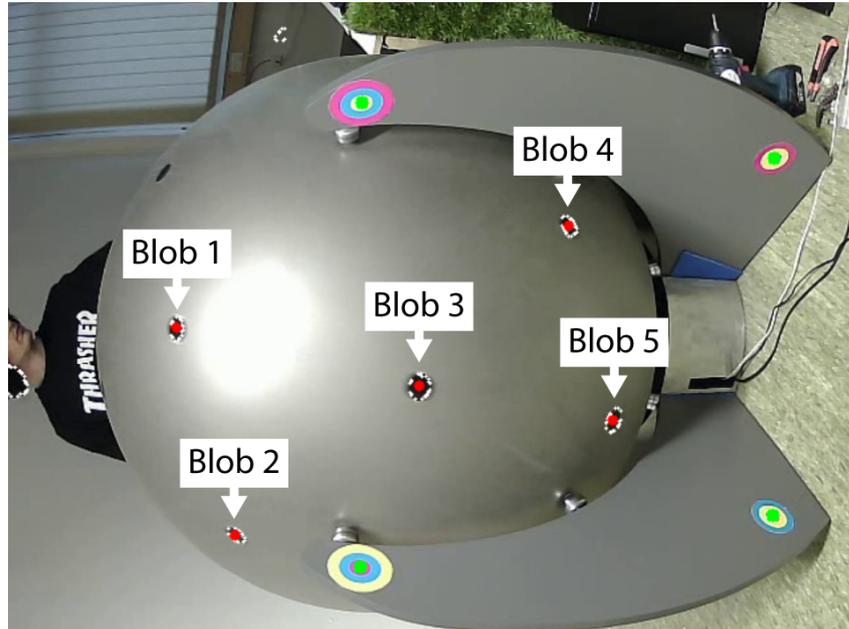


Abbildung 34: Detektierte Marker auf der Kugel

Nr.	X	Y	Z	$\rho_{X,Y}$	$\rho_{X,Z}$	$\rho_{Y,Z}$	σ_X	σ_Y	σ_Z
		[mm]			[–]		[mm]		
1	9,4	401,3	424,4	-0,174	0,136	-0,959	1,2	1,5	1,7
2	-322,1	316,5	370,6	-0,501	0,930	-0,714	2,5	1,2	3,0
3	-73,0	70,5	575,3	0,276	0,652	0,155	1,2	0,9	0,7
4	165,1	-177,3	531,5	-0,297	-0,691	0,756	0,9	1,5	0,8
5	-141,0	-279,7	493,1	0,675	0,708	0,917	1,3	2,1	1,5

Tabelle 13: Korrelationen und Standardabweichungen der Objektkoordinaten der Markerzentren

Es ist ersichtlich, dass die Korrelationskoeffizienten wie auch die Standardabweichungen stark von der Position des Markers abhängen. Je schiefender der Schnitt zwischen Bildstrahl und Kugel desto ungenauer wird die Position bestimmt und desto größer die Korrelation mit der Z-Achse, welche in etwa der Blickrichtung der Kamera entspricht.

4.5 Parameteranalyse

Im folgenden Experiment wird die Auswirkung der Parameter für das Systemrauschen des EKF σ_w sowie die Auswirkung der angenommenen Standardabweichung σ_b , mit welcher die Zentren der Blob-Marker im Bild detektiert werden, auf die Ergebnisse untersucht. Hierzu wird der reale Datensatz R_1 sowie der erste synthetische Datensatz S_1 mit verschiedenen Werten für σ_b und σ_w ausgewertet. Zur Auswertung wird der in Abschnitt 3.9.3 entwickelte EKF genutzt. Als Qualitätsmaß dient zunächst ρ_{max} als Qualitätsmaß der Übereinstimmung zwischen den Ergebnissen und der Referenz, da dieser Wert für beide Datensätze berechnet werden kann. Die Tabellen 14 und 15 zeigt die Korrelationskoeffizienten für verschiedene Kombinationen von σ_w und σ_b , die sich aus der Auswertung der Datensätze R_1 bzw. S_1 ergeben.

ρ_{max}		$\sigma_b [px]$		
		0,1	1	10
σ_w [rad/sek ²]	0,001	0,645	0,594	0,678
	0,01	0,902	0,794	0,527
	0,1	0,962	0,948	0,778
	1	0,947	0,956	0,954

Tabelle 14: Korrelationskoef. für verschiedene Parameterkombinationen (Datensatz R_1)

ρ_{max}		$\sigma_b [px]$		
		0,1	1	10
σ_w [rad/sek ²]	0,001	0,979	0,973	0,965
	0,01	0,995	0,991	0,990
	0,1	0,999	0,998	0,998
	1	0,998	0,996	0,995

Tabelle 15: Korrelationskoef. für verschiedene Parameterkombinationen (Datensatz S_1)

Weiterhin werden für den Datensatz S_1 der kombinierten RMSE für die Lagewinkel, die Drehraten und die berechneten Mausgeschwindigkeiten aller Epochen berechnet. Die Ergebnisse sind in den Tabellen 16, 17 und 18 dargestellt. In den Tabellen ist jeweils das beste Ergebnis fett gedruckt.

Es wird ersichtlich, dass die besten Ergebnisse für die Drehraten und Mausgeschwindigkeiten für die Parameter $\sigma_w = 0,1 \left[\frac{rad}{sek^2}\right]$ und $\sigma_b = 0,1[px]$ erreicht werden. Der Fehler der Lagewinkel wird für die untersuchten Kombinationen der Parameter minimal wenn das Systemrauschen auf $\sigma_w = 1 \left[\frac{rad}{sek^2}\right]$ erhöht wird. Dies ist damit zu begründen, dass mit einem größeren Wert für σ_w das physikalische Modell im EKF weniger gewichtet wird, was in größeren Sprüngen der Lagewinkel resultiert. Dies verringert deren Fehler, verursacht

Komb. RMSE der Lagewinkel [°]		σ_b [px]		
		0,1	1	10
σ_w [rad/sek ²]	0,001	2,68	2,96	3,14
	0,01	0,72	0,89	0,93
	0,1	0,18	0,27	0,27
	1	0.11	0.13	0.15

Tabelle 16: Kombiniertes RMSE der Lagewinkel (Datensatz S_1)

Komb. RMSE der Drehraten [°/sek]		σ_b [px]		
		0,1	1	10
σ_w [rad/sek ²]	0,001	5,42	6,04	6,04
	0,01	2,23	2,98	3,10
	0,1	1,09	1,60	1,61
	1	1,52	1,90	2,12

Tabelle 17: Kombiniertes RMSE der Drehraten (Datensatz S_1)

Komb. RMSE der Mausgeschwindigkeiten [px/sek]		σ_b [px]		
		0.1	1	10
σ_w [rad/sek ²]	0,001	0,49	0,56	0,64
	0,01	0,23	0,30	0,34
	0,1	0,11	0,15	0,15
	1	0,15	0,20	0,21

Tabelle 18: Kombiniertes RMSE der Mausgeschwindigkeiten (Datensatz S_1)

jedoch einen größeren Fehler in den Drehraten. Da für die beispielhafte Applikation die Änderungsraten bzw. Mausgeschwindigkeiten ausschlaggebend sind, wird für die nachfolgenden Experimente $\sigma_w = 0,1$ [$\frac{rad}{sek^2}$] und $\sigma_b = 0,1$ [px] angesetzt. Anzumerken ist, dass die Detektion der Markerzentren mit einer Standardabweichung von 0,1 Pixeln bei einer durchschnittlichen Entfernung der Marker von 1,5 m und einer Kammerkonstante von ca. 630 Pixeln einer Standardabweichung von ungefähr 0,24 mm im Objektraum entspricht, was sehr optimistisch ist. Die geringeren Fehler für die hohe Genauigkeit der Detektion ist damit zu begründen, dass für das Filter nicht die exakten Mittelpunkte der Marker relevant sind, sondern vielmehr dass in aufeinander folgenden Bildern möglichst die gleichen Positionen auf der Kugel detektiert werden. Der Parameter des Systemrauschens, der das Maß der Winkelbeschleunigung modelliert ist mit $\sigma_w = 0,1$ [$\frac{rad}{sek^2}$] $\approx 5,73$ [$\frac{°}{sek^2}$] in einem realistischen Bereich (Die durchschnittliche Winkelbeschleunigung beträgt für den Datensatz R_1 $2,9$ [$\frac{°}{sek^2}$], die maximale Winkelbeschleunigung beträgt $16,5$ [$\frac{°}{sek^2}$]).

4.6 Vergleich der Filtermodelle

In diesem Experiment soll geprüft werden, ob die Erweiterung des Zustandsvektors im EKF_2 gegenüber dem EKF_1 , in dem die Markerzentren als Parameter gespeichert werden (vgl. Abschnitt 3.9), eine Verbesserung der Ergebnisse bewirkt. Hierzu werden die Datensätze S_1 und R_1 mit beiden Filtern ausgewertet. In Tabelle 19 sind die maximalen Korrelationskoeffizienten ρ_{max} zwischen den berechneten Mausgeschwindigkeiten und den Referenzgeschwindigkeiten sowie der kombinierte RMSE der Drehraten und der Mausgeschwindigkeiten für den Datensatz S_1 aufgeführt.

Datensatz	Filter	ρ_{max} [-]	RMSE der Drehraten [$^\circ/sek$]	RMSE der Mausgeschwindigkeiten [px/sek]
R_1	EKF_1	0,962	-	-
	EKF_2	0,963	-	-
S_1	EKF_1	0,998	1,59	0,15
	EKF_2	0,999	1,09	0,11

Tabelle 19: Vergleich der Filtermodelle

Ersichtlich ist, dass das EKF_2 , in dem die Koordinaten der Markerzentren im Zustandsvektor gespeichert werden, marginal bessere Ergebnisse für den Korrelationskoeffizienten erzielt. Die Verbesserung zeigt sich jedoch deutlich im kombinierten RMSE der Änderungsraten für den Datensatz S_1 . Eine Verbesserung ist auch in den resultierenden Mausgeschwindigkeiten zu erkennen. Abbildung 35 zeigt die berechneten sowie die Referenzmausgeschwindigkeiten über die ersten 10 Sekunden des Datensatzes R_1 für beide Filtermodelle. Auffällig ist, dass das EKF_2 weniger Ausreißer aufweist. Beispielfhaft sind in Abbildung 35 zwei Ausreißer in den linken Abbildungen markiert, die beim zweiten Modell nicht auftreten. Die Abbildung zeigt weiterhin, dass die Referenzwerte für die y-Achse stark streuen, sobald hohe Mausgeschwindigkeiten auftreten.

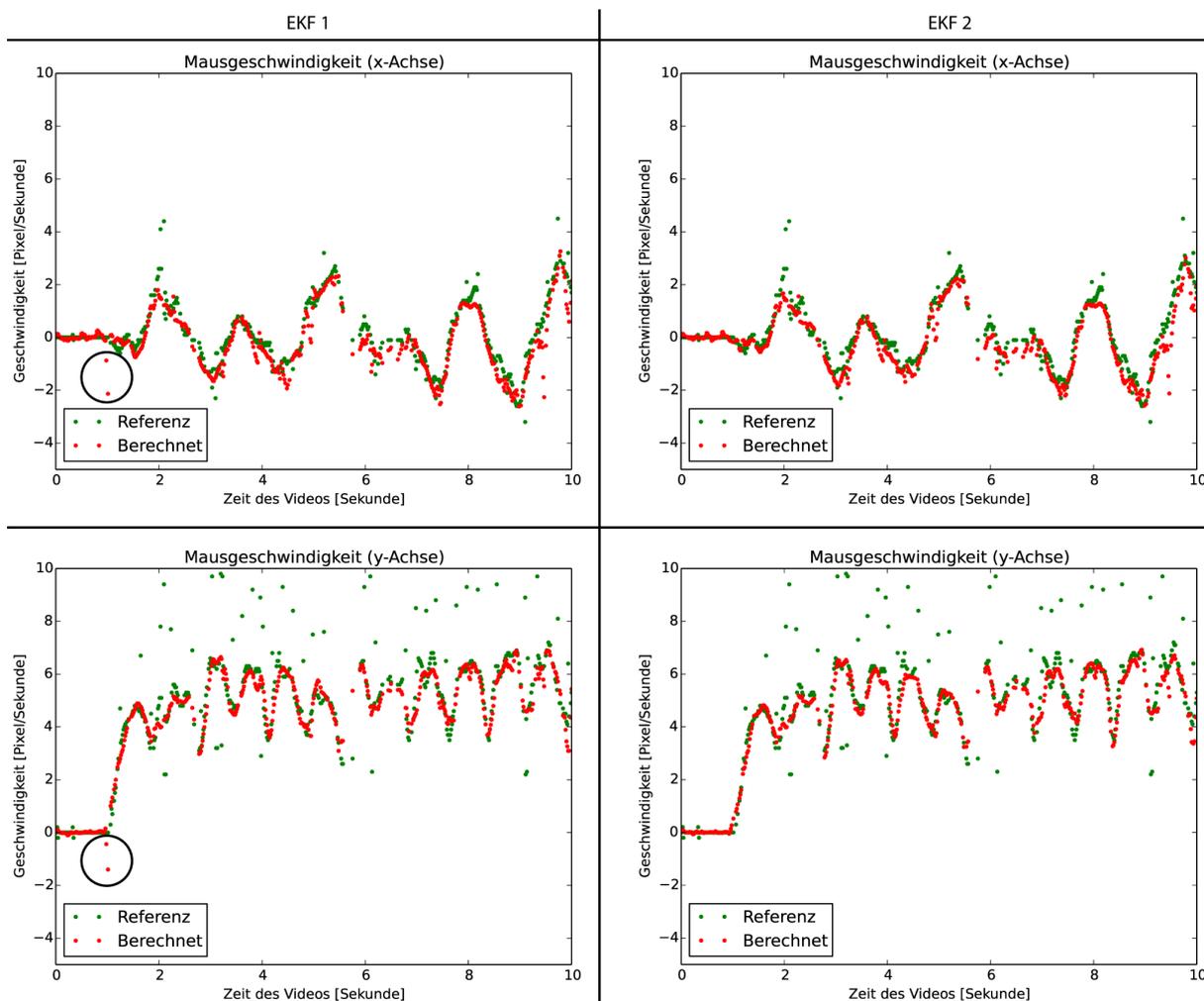


Abbildung 35: Vergleich der Mausgeschwindigkeiten beider Filtermodelle für den Datensatz R_1

Aus diesem Grund werden zusätzlich der Mausgeschwindigkeiten für den synthetischen Datensatz betrachtet, für welche exakte Referenzwerte vorliegen. In den Abbildungen 36 und 37 sind die berechneten Mausgeschwindigkeiten für den synthetischen Datensatz S_1 , die Referenzwerte, sowie die Differenzen jeweils für die x- und y-Achse dargestellt. Es wird ersichtlich, dass die Ergebnisse, die aus dem zweiten entwickelten EKF stammen, ein geringeres Rauschen aufweisen. Aufgrund der besseren Ergebnisse für das zweite EKF wird dieses für die Durchführung der weiteren Experimente genutzt.

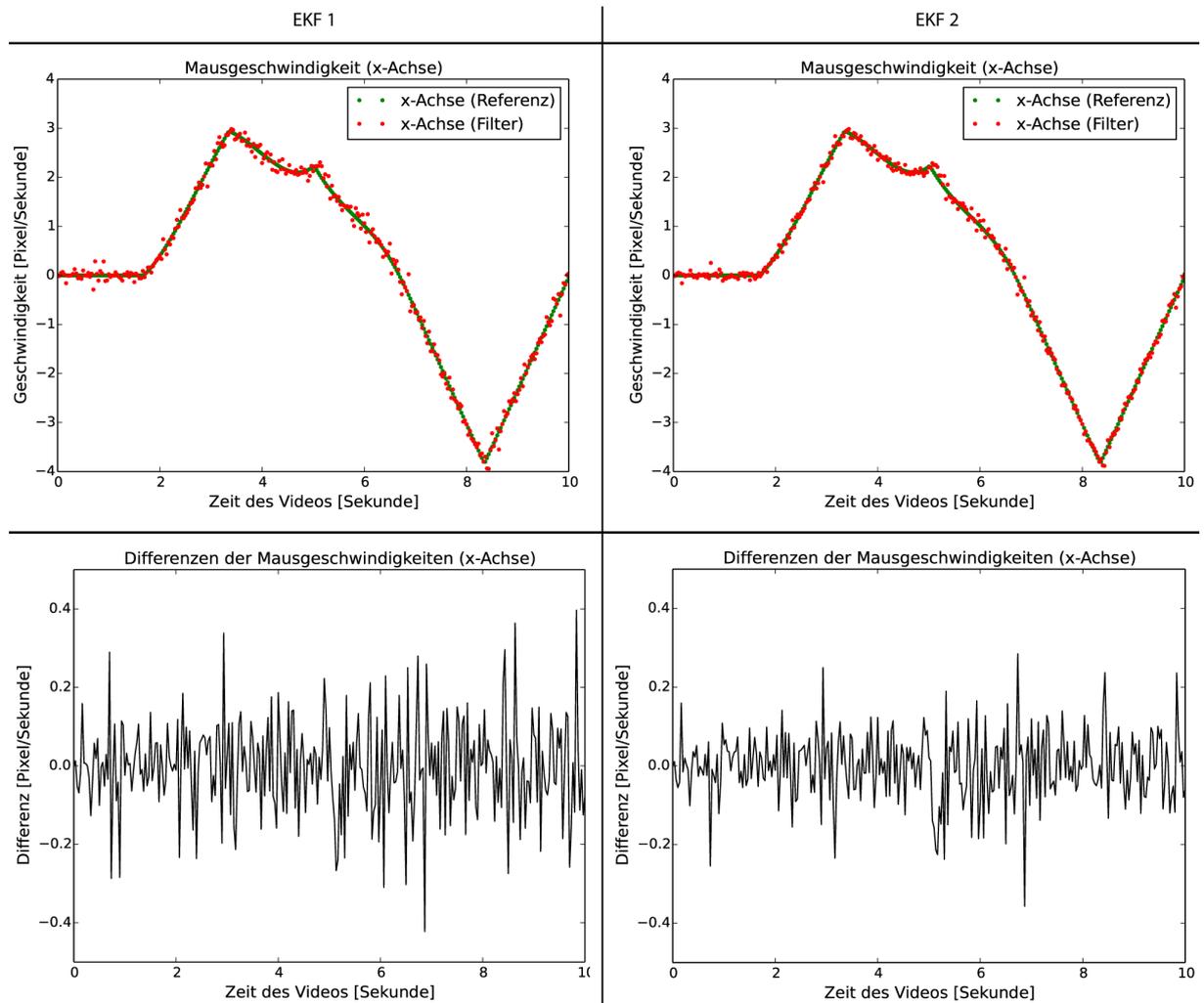


Abbildung 36: Vergleich der Mausgeschwindigkeiten und Geschwindigkeitsdifferenzen bei-
der Filtermodelle für den Datensatz S_1 (X-Achse)

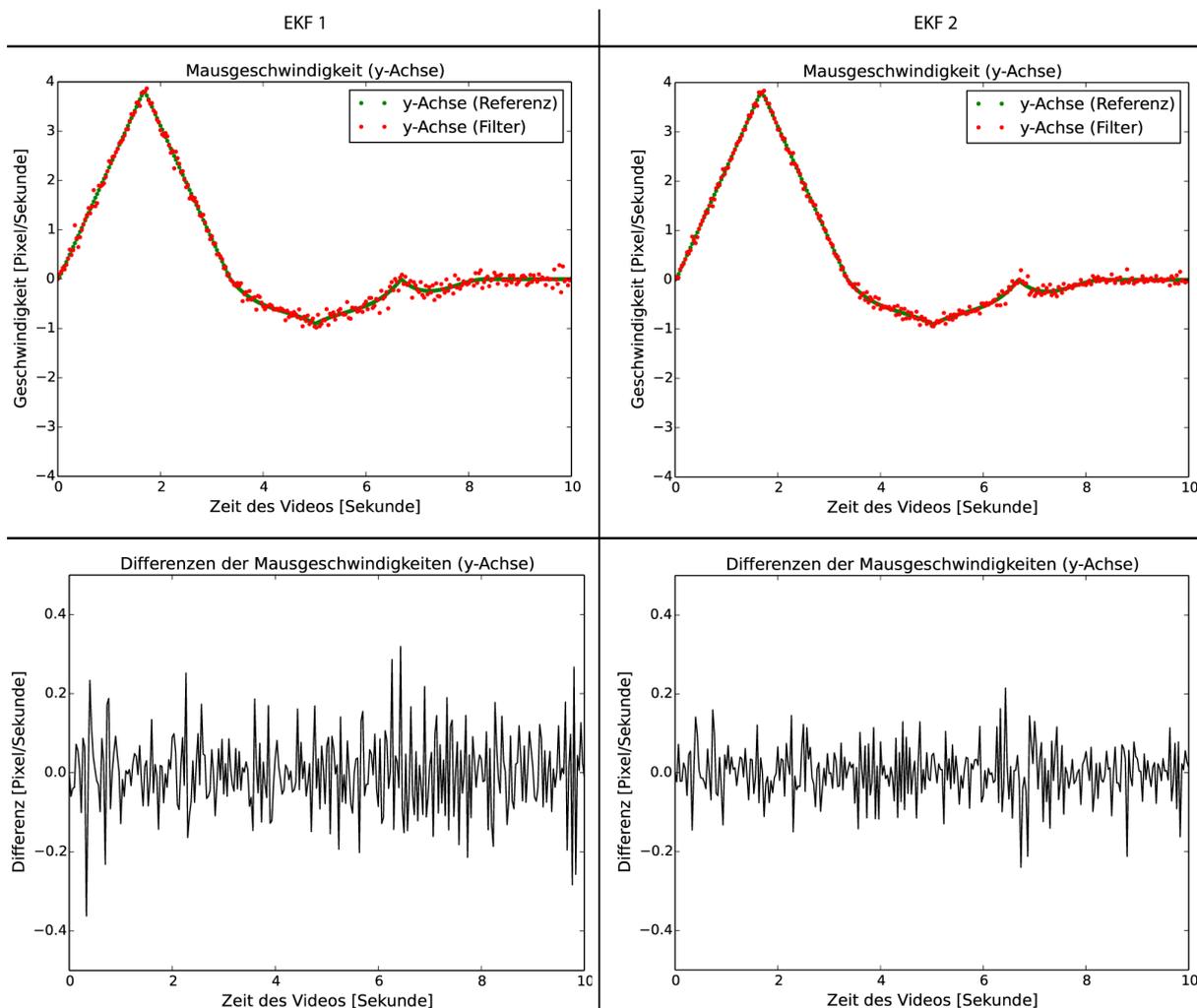


Abbildung 37: Vergleich der Mausgeschwindigkeiten und Geschwindigkeitsdifferenzen bei-
 der Filtermodelle für den Datensatz S_1 (Y-Achse)

4.7 Einfluss der Markeranzahl

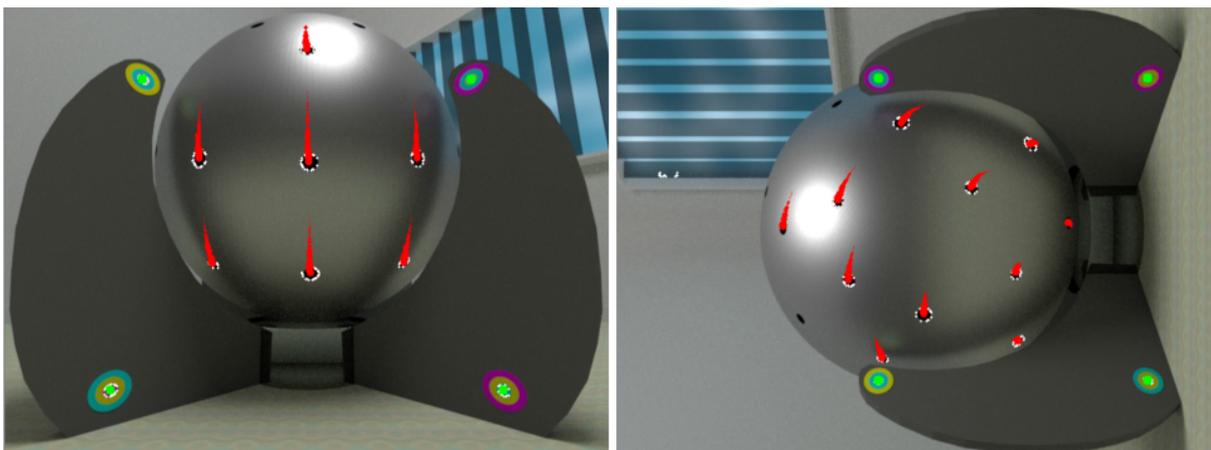
Um die Auswirkung der Anzahl von Markern auf der Kugel auf die Ergebnisse der Filte-
 rung zu untersuchen, werden die Datensätze S_1 , S_2 und S_3 ausgewertet, in denen sich 26,
 50 bzw. 74 Marker auf der Kugel befinden. Die Ergebnisse sind in Tabelle 20 zusamme-
 gefasst. Wie zu erwarten, verringern sich die Fehler mit zunehmender Anzahl von Markern.
 Gleichzeitig können die absoluten Lagewinkel bereits mit 26 Markern über die Laufzeit
 des Videos mit einem RMSE von ungefähr 0.1 Grad bestimmt werden. Die Drehraten sind
 mit einem RMSE von weniger als einem Grad pro Sekunde mit 26 Markern ausreichend
 genau für die derzeitige Verwendung des Eingabegerätes. Weiterhin kann aus der durch-
 schnittlichen Anzahl der gespeicherten Marker abgeleitet werden, dass ungefähr 15 - 20
 % der Marker in jedem Bild detektiert werden.

	26 Marker	50 Marker	74 Marker
Durchschnittliche Anzahl gespeicherter Marker [-]	4,9	8,9	14,5
RMSE der Lagewinkel [°]			
x-Achse	0,100	0,070	0,040
y-Achse	0,086	0,067	0,056
z-Achse	0,122	0,083	0,080
RMSE der Drehraten [°/sek]			
x-Achse	0,595	0,551	0,456
y-Achse	0,481	0,407	0,406
z-Achse	0,773	0,624	0,561

Tabelle 20: Ergebnisse für unterschiedliche Anzahl von Markern

4.8 Einfluss der Kamerakonfiguration

In diesem Experiment werden die Auswertungen der Datensätze S_2 und S_4 gegenübergestellt um die Auswirkung der Kamerakonfiguration auf die Qualität der Ergebnisse zu untersuchen. Hierzu wird zunächst die Genauigkeiten und die Fehler betrachtet, mit denen die äußere Orientierung der Kamera bei beiden Konfigurationen bestimmt wird. Im Anschluss werden die Ergebnisse der Filterung für beide Konfigurationen dargelegt. In beiden Datensätzen befinden sich 50 Marker auf der Kugel. Die Kamera ist im Datensatz S_2 näher an der Kugel positioniert. Damit auch die Marker auf dem Rahmen im Bild der Kamera sichtbar sind, ist der Öffnungswinkel größer als bei der weiter entfernten Kamera. Abbildung 38 zeigt jeweils ein Bild beider gerenderten Sequenzen mit Zwischenergebnissen. Die eingezeichneten Elemente entsprechen denen in Abbildung 27.

Abbildung 38: Bilder der laufenden Auswertung mit unterschiedlichen Kamerakonfigurationen. Konfiguration *nah*(links) und *fern*(rechts)

In Tabelle 21 sind die berechneten Korrelationen, die Standardabweichungen sowie die

absoluten Fehler der Parameter der äußeren Orientierung für den Datensatz S_2 aufgeführt. Tabelle 22 stellt diese Ergebnisse für den Datensatz S_4 dar. Die absoluten Fehler ergeben sich aus der Differenz der berechneten äußeren Orientierung und der Kamerapose, die bei der Modellierung vorgegeben wird.

Kor. [-]	X_0	Y_0	Z_0	ω_0	φ_0	κ_0
Y_0	0,000					
Z_0	0,002	-0,751				
ω_0	0,000	-0,998	0,770			
φ_0	-0,998	0,000	-0,001	0,000		
κ_0	0,451	0,001	0,000	0,000	-0,428	
Std.-Abw.	2,81 [mm]	2,72 [mm]	0,75 [mm]	0,081 [°]	0,081 [°]	0,016 [°]
Fehler	0,61 [mm]	2,93 [mm]	0,03 [mm]	0,049 [°]	0,018 [°]	0,021 [°]

Tabelle 21: Kor. und Std.-Abw. der äußeren Orientierung für Datensatz S_2 (**fern**)

Kor. [-]	X_0	Y_0	Z_0	ω_0	φ_0	κ_0
Y_0	0,000					
Z_0	-0,005	0,453				
ω_0	0,001	-0,968	-0,457			
φ_0	-0,976	0,001	0,006	-0,001		
κ_0	0,281	-0,002	-0,003	0,002	-0,272	
Std.-Abw.	0,35 [mm]	0,30 [mm]	0,11 [mm]	0,017 [°]	0,019 [°]	0,007 [°]
Fehler	1,11 [mm]	0,72 [mm]	0,76 [mm]	0,019 [°]	0,008 [°]	0,001 [°]

Tabelle 22: Kor. und Std.-Abw. der äußeren Orientierung für Datensatz S_4 (**nah**)

Die Lagewinkel können bei der Konfiguration **nah** durchweg mit einem kleineren Fehler bestimmt werden. Während die Translation entlang der x-Achse und z-Achse bei der weiter entfernten Kamera einen kleineren absoluten Fehler gegenüber der nahen Konfiguration aufweisen, existiert die größte Abweichung bei der Translation entlang der y-Achse bei der Konfiguration **fern**. In Tabelle 23 sind die Auswirkungen der untersuchten Kamerakonfigurationen auf die Ergebnisse der Filterung aufgeführt. Hier wird ersichtlich, dass die resultierenden Lagewinkel der Kugel bei der Konfiguration **fern** geringere Fehler aufweisen, wohingegen die Drehraten mit einem größeren Fehler bestimmt werden. Neben den RMSE ist in Tabelle 23 die durchschnittliche Anzahl von Markern angegeben die im Filter gespeichert werden. Die weiter entfernte Kamera kann einen größeren Teil der Kugeloberfläche abbilden, wodurch durchschnittlich 1,5 Marker mehr (entspricht ca. 20 %) im Vergleich zur Konfiguration **nah** detektiert werden, was sich positiv auf die Ergebnisse der Filterung auswirkt. Ein Zuwachs von 20% detektierten Markern für die Konfiguration **fern** gegenüber der Konfiguration **nah** ist plausibel, da der rechnerische Zuwachs der

	Konfiguration nah	Konfiguration fern
Durchschnittliche Anzahl gespeicherter Marker [-]	7,4	8,9
RMSE der Lagewinkel [°]		
x-Achse	0,23	0,07
y-Achse	0,13	0,07
z-Achse	0,21	0,08
RMSE der Drehraten [°/sek]		
x-Achse	0,46	0,55
y-Achse	0,35	0,41
z-Achse	0,58	0,62

Tabelle 23: Ergebnisse für unterschiedliche Kamerakonfigurationen

sichtbaren Oberfläche ca 27 % beträgt. Abbildung 39 stellt die rechnerisch sichtbare Kugeloberfläche in Abhängigkeit der Entfernung zwischen dem Projektionszentrum und dem Kugelmittelpunkt dar. Für die Konfiguration **fern** beträgt der Abstand $d = 2,35 [m]$, wodurch eine Fläche von $2,74 [m^2]$ sichtbar ist. Bei der Konfiguration **nah** beträgt der Abstand $d = 1,35 [m]$, wodurch $2,16 [m^2]$ der Oberfläche sichtbar ist. Gleichzeitig wird bei der Konfiguration **fern** ein größerer Teil der Kugel durch den Rahmen verdeckt was die Diskrepanz zwischen dem rechnerischen und dem empirischen Zuwachs der Fläche bzw. der Markeranzahl erklärt.

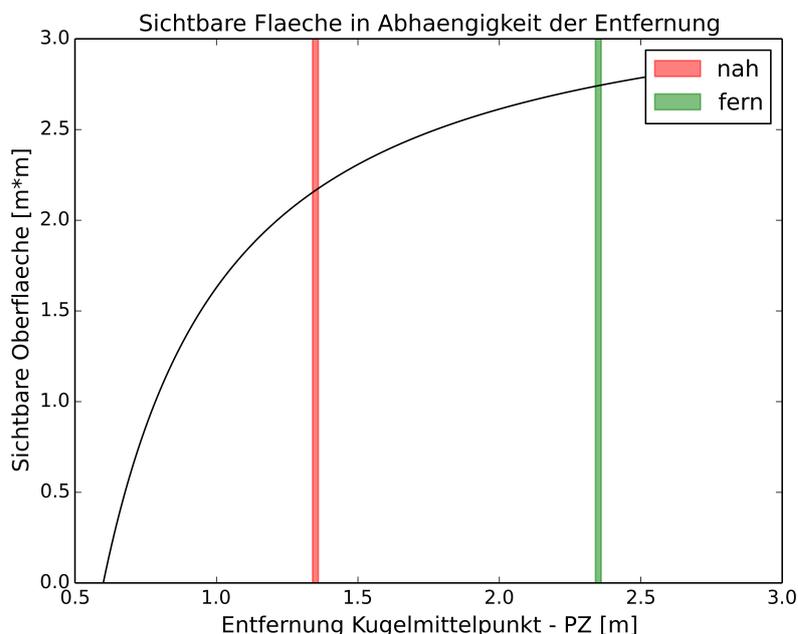


Abbildung 39: Sichtbare Fläche der Kugel in Abh. der Entfernung des Projektionszentrums zum Kugelmittelpunkt

Anzumerken ist, dass für beide Konfigurationen das Verhältnis zwischen dem Abstand der Kugel und dem Projektionszentrum d und der Kammerkonstante c in etwa gleich ist:

$$\frac{d}{c} \approx \text{const.} \quad (178)$$

Dies führt dazu, dass auch die Marker in etwa gleich groß im Bild abgebildet werden. Hieraus kann gefolgert werden, dass die Detektionsgenauigkeit der Markerzentren ebenfalls annähernd gleichbleibend ist. Die unterschiedlichen Genauigkeiten mit denen die äußeren Orientierungen der Kamera bestimmt werden ist folglich mit der Verteilung der Markerzentren zu begründen. Bei der Konfiguration `fern` ist insbesondere die Y-Koordinate und die Lagewinkel des Projektionszentrums mit größeren Fehlern behaftet, weswegen die Drehraten schlechter bestimmt werden. Bei der fortlaufenden Bestimmung der Lagewinkel überwiegt offensichtlich der positive Einfluss der größeren Anzahl der sichtbarer Marker auf der Kugel, weswegen die Lagewinkel mit der Konfiguration `fern` mit einem geringeren Fehler ermittelt werden.

4.9 Einfluss des Bildrauschens und der Belichtung

In diesem Experiment wird untersucht, wie sensitiv das entwickelte Programm auf verstärktes Bildrauschen bzw. eine Änderung der Belichtung ist. Hierzu wird bei der Auswertung des Datensatzes S_1 auf die Kanäle jedes Bildpunktes aller Bilder ein zufälliger, normalverteilter Wert $r \sim \mathcal{N}(0, \sigma_r)$ addiert, um ein Bildrauschen zu simulieren. Hierbei ist σ_r die Standardabweichung des Rauschens. Gleichzeitig wird zur Simulation einer Über- bzw. Unterbelichtung ein konstanter Wert l auf die Farbkanäle aller Bilder addiert. Abbildung 40 zeigt die laufende Auswertung des Datensatzes S_2 mit $l = 50$ und $\sigma_r = 15$. Die Farbinformationen sind mit 8 Bit gespeichert, so dass 0 eine minimale und 255 eine maximale Intensität bedeutet. Als Qualitätsmaß wird der kombinierte RMSE der Drehraten verwendet.

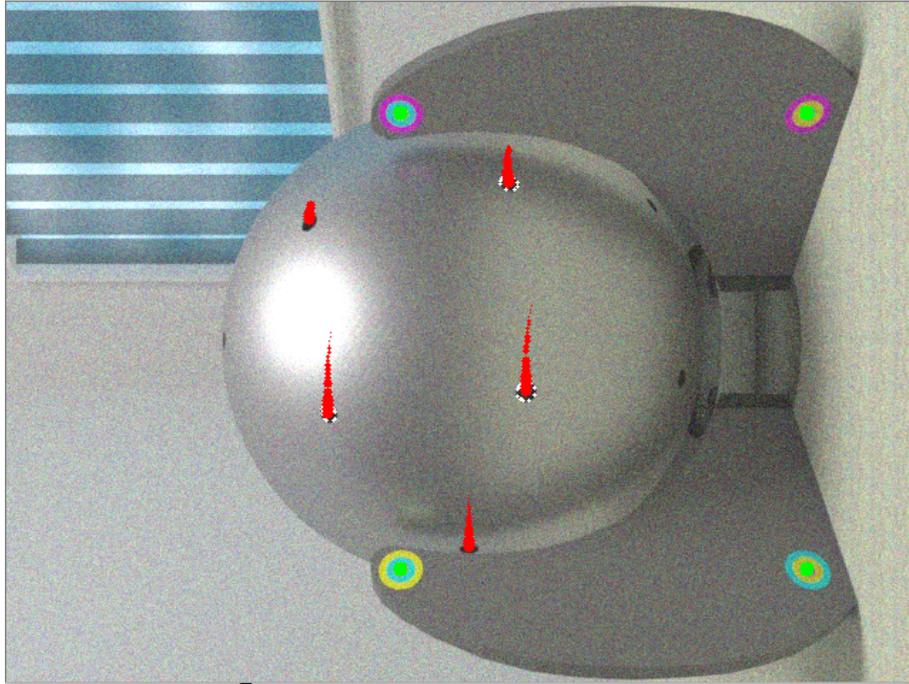


Abbildung 40: Bild der Auswertung des verrauschten und überbelichteten Datensatzes S_1

In Tabelle 24 sind die Korrelationskoeffizienten für verschiedene Standardabweichungen des Bildrauschens und Helligkeitsänderungen aufgeführt.

komb. RMSE der Drehraten [$^{\circ}/sek$]		σ_r [-]			
		1	5	10	15
l [-]	-50	1,20	1,29	1,62	-
	-25	0,98	1,04	1,18	1,47
	25	1,06	1,20	1,25	1,59
	50	1,06	1,04	1,09	1,42

Tabelle 24: Ergebnisse für verschieden starkes Bildrauschen und Helligkeiten

Für $l = -50$ können in Kombination mit $\sigma_r = 15$ nur zwei der Marker auf dem Rahmen detektiert werden, weswegen keine Auswertung möglich ist. Die übrigen Kombinationen liefern kaum schlechtere Ergebnisse als die Auswertung des unveränderten Videos, was zeigt, dass das Programm nicht sensitiv auf die untersuchten Einflüsse ist, solange alle notwendigen Marker detektiert werden können. Exemplarisch sind in Abbildung 41 die Lagewinkel sowie die Differenzen der Lagewinkel zu den Referenzwerten für die Konstellation $l = 50$ und $\sigma_r = 15$ abgebildet.

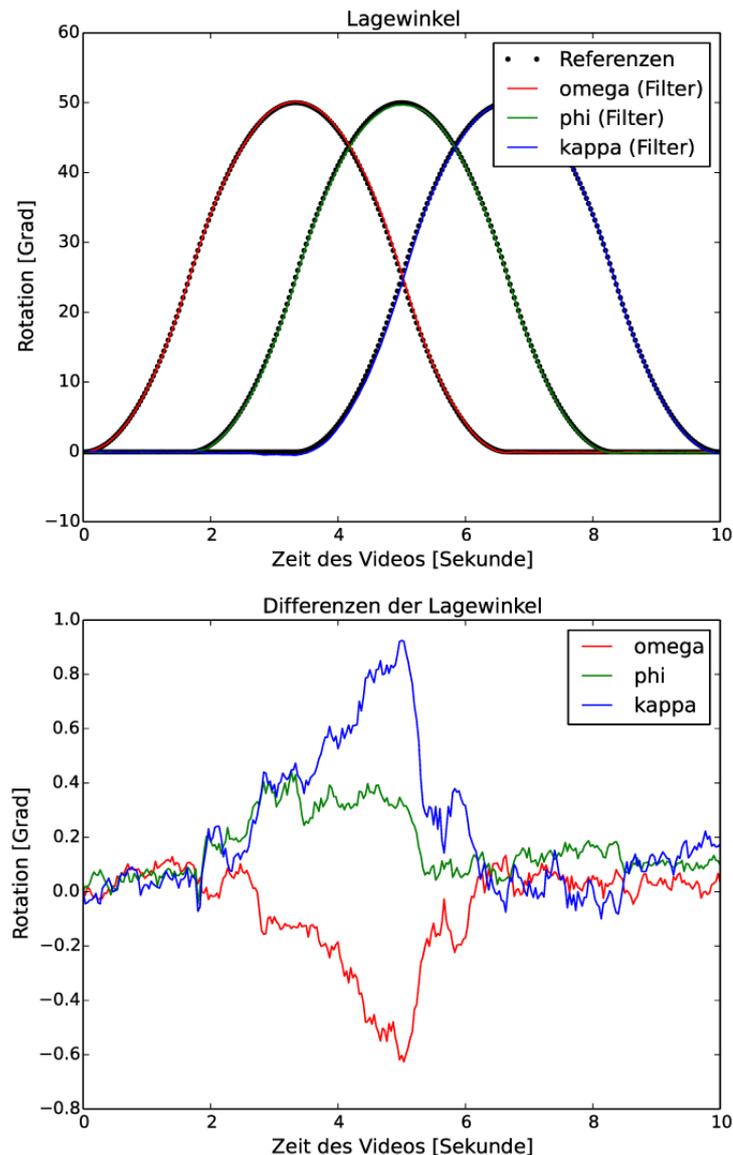


Abbildung 41: Lagewinkel und Differenzen zu Referenzwerten für verrauschten und überbelichteten Datensatz

Über die Laufzeit des Videos sind die Differenzen der Lagewinkel geringer als ein Grad. Anzumerken ist, dass die Lagewinkel nicht direkt beobachtet werden, was normalerweise einen stetig zuwachsenden Fehler verursacht. In Abbildung 41 ist zu sehen, dass die Differenzen aus den berechneten Lagewinkeln und den Referenzwerten zunächst zunimmt, dann jedoch ab der Mitte des Videos wieder abnehmen. Dies ist damit zu begründen, dass sich die Kugel wieder in den Ausgangszustand zurückdreht und sich die Fehler hierdurch ausgleichen.

4.10 Diskussion

Die in Abschnitt 4.5 durchgeführte Analyse der Parameter liefert Parameter für das Prozessrauschen und die angenommenen Genauigkeiten der Markerdetektion, mit denen ein kombinierter RMSE für die Lagewinkel der Kugel über 10 Sekunden von $0,1^\circ$ und für die Drehraten von $1,1^\circ/\text{sek}$ erreicht wird. Der Vergleich mit den anderen, untersuchten Parameterkombinationen zeigt, dass das Programm nicht sensitiv auf diese Parameter ist. Der Vergleich der Filtermodelle in Abschnitt 4.6 zeigt, dass die Erweiterung des Zustandsvektors die Ergebnisse der Filterung verbessert. Für den Datensatz S_1 wird der kombinierte RMSE der Drehraten um ca. 30 % reduziert. Der grafische Vergleich der Ergebnisse beider Filtermodelle ergibt, dass das Filtermodell, in dem die Punkte im Zustandsvektor gespeichert werden einen glatteren Verlauf der berechneten Mausgeschwindigkeiten liefert, was für die beispielhafte Zielapplikation von Vorteil ist. Der in Abschnitt 4.7 untersuchte Einfluss der Markeranzahl zeigt, dass die Ergebnisse mit zunehmender Anzahl der Marker auf der Kugel besser werden, jedoch ist die Verbesserung gering. Der Einfluss des Bildrauschens und der Belichtung, der in Abschnitt 4.9 untersucht wurde, ist ebenfalls gering, lediglich für ein starkes Bildrauschen in Kombination mit einer starken Unterbelichtung, können nicht alle Marker auf dem Rahmen detektiert werden. Dies kann durch eine ausreichende Beleuchtung der Szene vermieden werden. Die beiden untersuchten Kamerakonfigurationen liefern ähnliche Ergebnisse für die RMSE der Drehraten, jedoch liefert die Kamerakonfiguration **fern** bessere Ergebnisse für die Lagewinkel.

Es wird angemerkt, dass die Messung der Distanzen zwischen den Passpunkten, aus denen die Skalierung der rekonstruierten Szene folgt, nicht relevant für den Betrieb des Eingabegerätes ist, sondern lediglich der metrischen Interpretation der Ergebnisse dient. Dies ist damit zu begründen, dass die Lagewinkel und Drehraten der Kugel unabhängig von der Skalierung der Szene sind und die Skalierung der berechneten Mausbewegung ohnehin parametrisiert wird. Abschließend resultiert aus den Ergebnissen der Experimente, dass das entwickelte Verfahren eingesetzt werden kann, um die Pose der Kugel zu bestimmen. Das Verfahren zeigt sich robust im Hinblick auf die untersuchten Einflüsse. Aufgrund der hohen Korrelationen der Ergebnisse mit den Referenzwerten des realen Datensatzes kann gefolgert werden, dass die erzielten Genauigkeiten ausreichen, um die Kugel als Eingabegerät zu nutzen, wobei vorausgesetzt wird, dass die in Abschnitt 4.2 zusammengefassten Parameter dem Aufbau und der eingesetzten Kamera entsprechend angepasst werden.

5 Zusammenfassung und Ausblick

Durch den Einsatz einer Kamera in Kombination mit den verwendeten Markern liefert das entwickelte Verfahren eine Möglichkeit, die Lagewinkel und Drehraten der Kugel zu bestimmen. Die erzeugten Ausgaben können zur Steuerung von Software genutzt werden. Für die beispielhaft untersuchte Zielapplikation können die Drehraten der Kugel in Mausbewegungen umgerechnet werden, wodurch der bislang eingesetzte Trackball ersetzt werden kann, ohne die Zielapplikation anpassen zu müssen.

Die durchgeführten Experimente zeigen, dass die Qualität der erzeugten Ergebnisse zwar von der Wahl der Parameter, der Konfiguration der Kamera sowie der Bildqualität und von der Anzahl von Markern auf der Kugel abhängt, jedoch sind auch die vergleichsweise schlechteren Ergebnisse weitgehend ausreichend für die beispielhafte Applikation. Mit der Aufnahme der gespeicherten Koordinaten der Markerzentren in den Zustandsvektor kann die Güte der berechneten Ausgaben geringfügig gesteigert werden.

Durch die Wahl der Methoden ist das entwickelte Programm echtzeitfähig. Die Auswertungen wurden auf einem Rechner mit einem AMD Prozessor mit vier Kernen und einer Taktrate von 3.2 GHz in Echtzeit durchgeführt. Die eingesetzten Detektionsverfahren für Marker erweisen sich als robust. Der bautechnische Aufwand sowie die notwendigen Vorbereitungen sind gering, da keine aufwändige Vermessung des Eingabegerätes notwendig ist. Die Marker per Augenmaß zu platzieren und die Distanzen mit einem Metermaß zu messen, erweist sich als ausreichend genau. Mit wenigen Bildern des Eingabegerätes und einer kalibrierten Kamera kann die Kalibrierungsphase durchgeführt werden, welche eine manuelle Vermessung der Kugelparameter und Markerkoordinaten in einem gemeinsamen Koordinatensystem ersetzt. Sofern keine Änderungen an der Konstruktion vorgenommen werden, ist das entwickelte System dauerhaft betriebsfähig, sofern die Randbedingungen gegeben sind, dass das Eingabegerät im Bild sichtbar ist, was eine ausreichende Beleuchtung einschließt.

Um das System für die Nutzung zu optimieren, sollten möglichst viele Konfigurationen der Kamera getestet und eine feinere Justierung der Parameter durchgeführt werden. Mit einer höheren Anzahl von Markern auf der Kugel kann die Genauigkeit der Ergebnisse gesteigert werden. Der Einfluss der Größe dieser Marker, sowie die Anzahl und Größe der Marker auf dem Rahmen wurde nicht untersucht. In weiteren Experimenten könnten diese Einflüsse, wie auch der Einsatz anderer Markertypen, untersucht werden. Darüber hinaus könnten Methoden eingesetzt werden, mit denen der Mittelpunkt einer Kontur durch Ausgleichung ermittelt wird, wodurch Informationen über die Detektionsgenauigkeit vorliegen. Beispielsweise können Verfahren genutzt werden, um eine Ellipse einschließlich

deren Zentrum mit der Methode der kleinsten Quadrate zu schätzen. Aus einem solchen Ansatz resultieren Informationen über die Genauigkeit der Messung, wodurch der ange-setzte Parameter der Detektionsgenauigkeit nicht mehr benötigt wird. Weiterhin wurde nicht untersucht, welchen Einfluss die Bildwiederholrate und Auflösung der Kamera auf die Qualität der Ergebnisse hat, wobei anzunehmen ist, dass mit einer höheren Auflösung bzw. Bildwiederholrate zwar die Qualität der Ergebnisse steigt, gleichzeitig sich jedoch die Berechnungszeit der Algorithmen erhöht.

6 Quellenverzeichnis

- [ope, 2014] (2014). The OpenCV Reference Manual. docs.opencv.org/3.0-beta/opencv2refman.pdf Besucht: 17.09.2017.
- [Azad et al., 2011] Azad, P., Münch, D., Asfour, T., und Dillmann, R. (2011). 6-DoF Model-based Tracking of Arbitrarily Shaped 3D Objects. In *IEEE International Conference on Robotics and Automation*, Seiten 5204–5209.
- [Blender Foundation, 1995] Blender Foundation (1995). Blender. blender.org Besucht: 17.09.2017.
- [Chang et al., 2004] Chang, F., Chen, C.-J., und Lu, C.-J. (2004). A linear-time component-labeling algorithm using contour tracing technique. *Computer Vision and Image Understanding*, 93:206–220.
- [Cramer und Howitt, 2004] Cramer, D. und Howitt, D. L. (2004). *The Sage dictionary of statistics: a practical resource for students in the social sciences*. Sage.
- [Fiala, 2005] Fiala, M. (2005). ARTag, a fiducial marker system using digital techniques. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Band 2, Seiten 590–596.
- [Forster et al., 2014] Forster, C., Pizzoli, M., und Scaramuzza, D. (2014). SVO: Fast semi-direct monocular visual odometry. In *IEEE International Conference on Robotics and Automation*, Seiten 15–22.
- [Gander et al., 1994] Gander, W., Golub, G. H., und Strebel, R. (1994). Least-squares fitting of circles and ellipses. *BIT Numerical Mathematics*, 34:558–578.
- [Gao et al., 2003] Gao, X.-S., Hou, X.-R., Tang, J., und Cheng, H.-F. (2003). Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:930–943.
- [Garcia et al., 2011] Garcia, E., Sültrop, C., und Hausotte, T. (2011). Verbesserung der Detektion sphärischer Marker für die optische Navigationschirurgie. *CURAC Jahrestagung*, 10.
- [Hajimolahoseini et al., 2014] Hajimolahoseini, H., Amirfattahi, R., und Khorshidi, S. (2014). Real-Time Pose Estimation and Tracking of Rigid Objects in 3D Space Using Extended Kalman Filter. In *Iranian Conference on Electrical Engineering*, Band 22, Seiten 1545–1549.
- [Halir und Flusser, 1998] Halir, R. und Flusser, J. (1998). Numerically Stable Direct

- Least Squares Fitting of Ellipses. In *International Conference in Central Europe on Computer Graphics and Visualization*, Band 6, Seiten 125–132.
- [Kraus, 2004] Kraus, K. (2004). *Photogrammetry*, Band 1. Walter de Gruyter. Berlin, New York, 7. Auflage.
- [Kraus et al., 2004] Kraus, K., Jansa, J., und Kager, H. (2004). *Photogrammetry*, Band 2. Ferd. Dümmlers Verlag, 3. Auflage.
- [Luhmann, 2000] Luhmann, T. (2000). *Nahbereichsphotogrammetrie*. Herbert Wichmann Verlag, Heidelberg.
- [McGlone et al., 2004] McGlone, J. C., Mikhail, E. M., und Bethel, J. (2004). *Manual of Photogrammetry*. asprs, 5. Auflage.
- [Patidar et al., 2010] Patidar, P., Gupta, M., Srivastava, S., und Nagawat, A. K. (2010). Image De-noising by Various Filters for Different Noise. *International Journal of Computer Applications*, 9(4).
- [Suzuki und Abe, 1985] Suzuki, S. und Abe, K. (1985). Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30:32–46.
- [Szeliski, 2011] Szeliski, R. (2011). *Computer Vision - Algorithms and Applications*. Springer.
- [Thrun et al., 2006] Thrun, S., Burgard, W., und Fox, D. (2006). *Probabilistic Robotics*. The MIT Press. Cambridge, London.

Anhang

Kamerakalibrierung

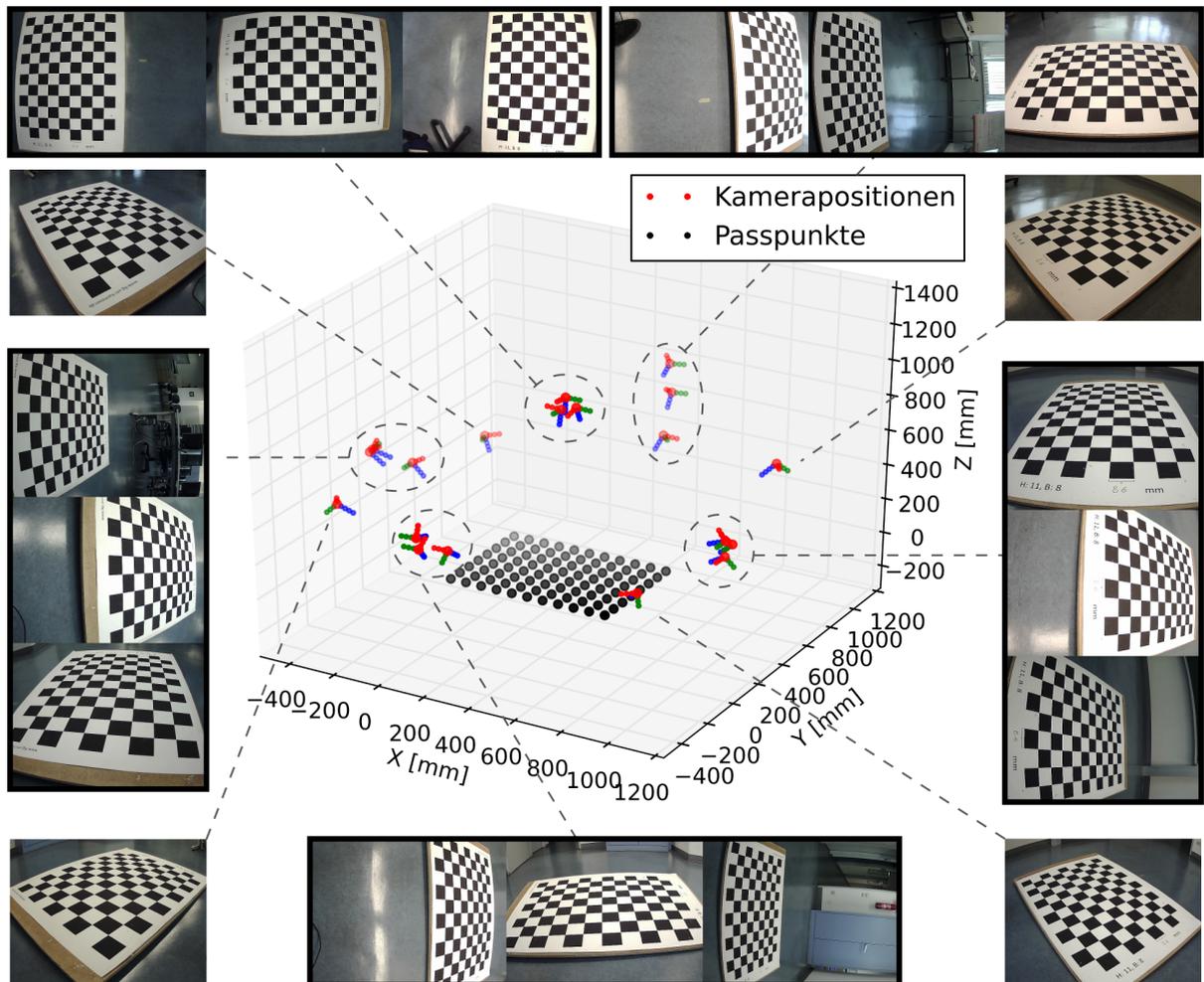


Abbildung 42: Kamerapositionen und Passpunkte bei der Kamerakalibrierung und Darstellung der Aufnahmen (Auflösung der Bilder: 640×480 Pixel)