

---

# ANALYSE VON BEWEGUNGSFELDERN DREIDIMENSIONALER STARRKÖRPER DURCH EIN CLUSTERING VERFAHREN

---

Internationale Medieninformatik  
an der Hochschule für Technik und Wirtschaft Berlin

## Abschlussarbeit

zur Erlangung des akademischen Grades  
Bachelor of Science

vorgelegt von

**Sophie-Charlotte Bolinski**

geboren am 04.04.1991 in Berlin

26. April 2016 - 16. August 2016

**Erstprüfer:** Herr Prof. Dr. Klaus Jung  
**Zweitprüferin:** Frau Dr. Anna Hilsmann





# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>5</b>
<b>2</b>	<b>Grundlagen</b>	<b>7</b>
2.1	Homogene Koordinaten . . . . .	7
2.2	Transformation . . . . .	9
2.3	Projektion . . . . .	11
2.4	Homographie . . . . .	12
2.5	Clustering . . . . .	15
2.5.1	k-means . . . . .	16
2.5.2	Fuzzy C-means . . . . .	17
2.6	Finden von Ausreißern . . . . .	19
<b>3</b>	<b>Analyse und Konzeption</b>	<b>21</b>
3.1	Vorclustering mit Fuzzy C-means . . . . .	23
3.2	Homographie als Information für das Clustering . . . . .	24
3.3	Konzeption eines Multi-Frame Clusterings . . . . .	25
<b>4</b>	<b>Implementierung und Umsetzung</b>	<b>27</b>
4.1	Framework zur Erzeugung von Testdaten . . . . .	27
4.2	Clustering Methoden und Multi-Frame Clustering . . . . .	28
4.3	Umgang mit Problemen und Testing . . . . .	29
<b>5</b>	<b>Ergebnisse</b>	<b>31</b>
5.1	Vorclustering durch Fuzzy C-means . . . . .	32
5.2	Homographie als Information für das Clustering . . . . .	34
5.2.1	Ausreißer Erkennung . . . . .	34
5.2.2	Modifikation für das Multi-Frame Clustering . . . . .	38
5.3	Ergebnisse des Multi-Frame Clusterings . . . . .	40
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>45</b>
6.1	Zusammenfassung . . . . .	45
6.2	Ausblick . . . . .	45
<b>7</b>	<b>Anhang</b>	<b>47</b>



# 1 Einleitung

Motivation dieser Bachelorarbeit ist es, artikulierte Objekte in Videomaterial zu finden. Artikuliert heißt, dass die Objekte aus sich unterschiedlich bewegenden Komponenten bestehen, die durch Gelenkpunkte verbunden sind. Ähnlich wie beim menschlichen Skelett oder einer Lampe, bei der starre Bewegungskomponenten ebenfalls durch Gelenkpunkte verknüpft sind. Um die Komponenten des Skeletts zu erkennen, wird versucht, das Bewegungsmuster der Komponenten zu untersuchen und sie darüber ausfindig zu machen. Das Erkennen der Bewegungsmuster erfolgt über Datenpunkte auf dem untersuchten Objekt. Diese Punkte können durch Markierungen in Form von Trackingpunkten fest vorgegeben sein oder dynamisch durch einen Bildanalyse Algorithmus gefunden werden. Die dynamisch gefundenen Punkte heißen *interest points* und können beispielsweise durch Peaks im Farbhistogramm ausgemacht werden. Durch die Verfolgung der Änderung der Punkte von einem zum nächsten Frame und die Analyse durch ein mathematisches Modell können die Bewegungsregionen eindeutig ermittelt werden. Die Methode das Bewegungsmodell zu schätzen heißt Homographie. Die Punkte, die einem geschätzten Bewegungsmodell folgen, werden einem Cluster zugeordnet. Eine mögliche Form, wie das Clustering eines artikulierten Objekts aussehen könnte, ist in Abbildung 1.1 zu sehen. Die Bewegungskomponenten und die Gelenkkomponenten sind hier in unterschiedliche Cluster unterteilt worden.

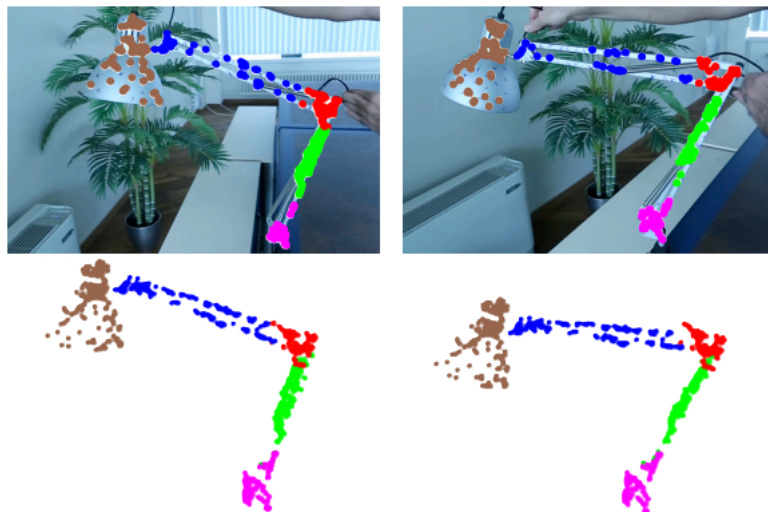


Abbildung 1.1: Ermittlung der Skelettstruktur einer Lampe [1]

Dreidimensionale Objekte aus einer Bildsequenz zu erkennen ist seit mehreren Jahrzehnten ein wichtiges Forschungsgebiet, insbesondere die Struktur der Bewegungen zu Erkennen und zuzuordnen. Diese Technik wird bereits in einem breiten Spektrum von Anwendungen

genutzt. Beispiele sind Positionsermittlung, Roboter Navigation, Erstellung von Modellen dreidimensionaler Körper durch zweidimensionales Bildmaterial, Augmented Reality, Video Stabilisierung oder das Nachbilden von gesamten Städten durch unsortierte Bildsammlungen [1] [2].

In dieser Arbeit soll anstatt aus echtem Bildmaterial aus synthetisch erzeugten Bildfolgen ein eigenes Verfahren zum Erkennen von artikulierten Objekten entwickelt werden, wie artikulierte Objekte erkannt werden können. Die synthetischen Daten bieten den Vorteil, dass genau bekannt ist, welche Daten richtig sind. Dadurch kann bei der Entwicklung des Verfahrens stets überprüft werden, wie sich die Qualität des Clusterings entwickelt. Die Daten der Bildfolgen oder Frames der sich bewegenden Objekte werden durch ein in MATLAB entwickeltes Framework erzeugt und analysiert. Zur Vereinfachung werden in dieser Arbeit Starrkörper in Form von Ebenen betrachtet. Starrkörper sind nicht verformbar, das heißt, dass die Ebenen sich nicht in sich selbst verformen, sondern nur äußeren Transformationen unterliegen. Diese Ebenen sollen die Bewegungskomponenten der simulierten, artikulierten Bewegung darstellen. Durch das Framework können die Ebenen beliebig im dreidimensionalen Raum bewegt werden. Abhängig davon wie viele Frames pro Sekunde gesetzt sind, werden die Daten des synthetischen Videos erzeugt. Diese Daten werden ebenfalls durch das Framework vom dreidimensionalen in den zweidimensionalen Raum projiziert, um Aufnahmebilder einer Kamera zu simulieren. Die so erzeugten Daten bilden den Ausgangspunkt für das Clustering Verfahren, dass in dieser Arbeit entwickelt worden ist. Als interessanter Fall wird das Überschneiden von Ebenen in einem fiktiven Gelenkpunkt untersucht. Für die Entwicklung des Clusterings wurde ein auf Wahrscheinlichkeiten basierender Algorithmus gewählt, der zulässt, dass Punkte der Ebene zu einem bestimmten Grad auch zu einem anderen Cluster gehören. Dadurch können sich überlappende Daten besser erkannt werden, als wenn die Zuordnung eines Punktes zu einem Cluster absolut wäre. Von dieser Basis aus soll das gefundene Clustering mit Hilfe von Homographie verbessert werden. Homographie analysiert die Veränderung des Objekts von einem Frame zum nächsten, aus denen Punkt-korrespondenzen erkannt werden. Verschiedene Ansätze werden diskutiert und abgewogen die Homographie in das Clustering mit einzubeziehen. Aus dem letztendlichen Clustering Verfahren wird ein Multi-Frame Clustering entwickelt, dass iterativ über alle Framepaare des Videos angewandt wird. Als Ergebnis können alle Clusterings betrachtet und die Körper zu einem gewissen Grad an Genauigkeit bestimmt werden. Um so mehr Bilder untersucht werden, desto genauer ist die Abschätzung welche Punkte zu einer Ebene gehören, weil die Fehler aus jedem Framepaar berücksichtigt werden können. Die Bachelorarbeit ist so strukturiert, dass zunächst auf die theoretischen Grundlagen eingegangen wird, um das Fundament für das Verständnis der weiteren Überlegungen zu legen. Grundkenntnisse in linearer Algebra werden im Allgemeinen vorausgesetzt. Es folgt eine Diskussion über verschiedene Ansätze das Problem der artikulierten Objekterkennung anzugehen und welche Clusteringmethoden für diese Aufgabe am besten geeignet sein könnten. Aus diesen Überlegungen wird ein Clustering Verfahren entwickelt und umgesetzt. Zum Abschluss werden die Probleme des entwickelten Verfahrens diskutiert und auf Möglichkeiten der Weiterentwicklung des umgesetzten Verfahrens eingegangen.

## 2 Grundlagen

In diesem Kapitel wird auf wichtige, grundlegende Konzepte zum Verständnis dieser Arbeit eingegangen. Die mathematischen Konzepte der Transformation und Projektion fließen vorwiegend in die Erstellung von synthetischen Daten durch ein eigen entwickeltes Framework ein. Das Verständnis von Homographien und der Clustering Algorithmen ist wiederum für das eigentliche Verfahren zum Analysieren der Framepaare und synthetischen Videodaten von Wichtigkeit.

### 2.1 Homogene Koordinaten

Es ist allgemein bekannt, dass beim Betrachten unserer Umgebung gerade, parallele Linien am Horizont zusammen zu laufen scheinen. Um Abbildungen des dreidimensionalen Raums korrekt im zweidimensionalen darzustellen müssen die Regeln der Perspektive berücksichtigt werden. Um diese Abbildungsregeln korrekt zu beschreiben ist ein mathematisches Modell nötig, dass Punkte im unendlichen oder Fluchtpunkte erlaubt. Jede Linie dieses Systems hat einen Schnittpunkt, auch parallele Linien. Sie schneiden sich am Horizont im Fluchtpunkt.

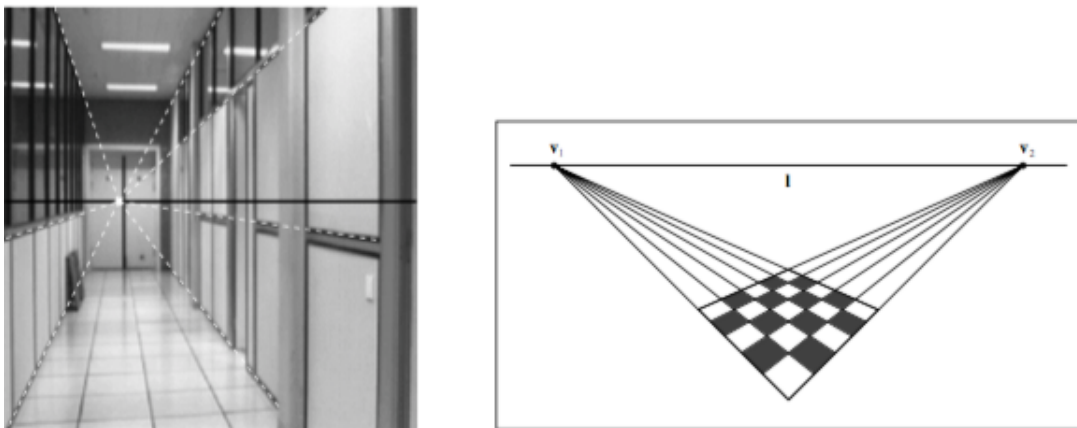


Abbildung 2.1: Darstellung von Perspektive [3]

Dieses Modell nennt sich der projektive Raum und seine Koordinaten homogen. In diesem Raum werden die üblichen Koordinaten des euklidischen Raums um eine weitere Dimension  $w$  ergänzt, um Tiefe darstellbar zu machen. Diese Transformation kann jeder Zeit durch Normalisierung mit der zusätzlichen Dimension rückgängig gemacht werden. Die folgenden Formeln zeigen die Transformation von zweidimensionalen kartesischen Koordinaten in homogene Koordinaten und die Rücktransformation von homogenen in kartesische Koordinaten

durch Normierung mit dem Wert der zusätzlichen Dimension  $w$ .

$$\begin{bmatrix} x \\ y \end{bmatrix} \mapsto \begin{bmatrix} x * w \\ y * w \\ w \end{bmatrix} \text{ bzw. } \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}, w \neq 0 \quad (2.1)$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \text{ bzw. } \begin{bmatrix} x'/w \\ y'/w \\ w/w \end{bmatrix} \mapsto \begin{bmatrix} x \\ y \end{bmatrix}, w \neq 0 \quad (2.2)$$

Homogene Koordinaten haben den Vorteil, dass sie im Gegensatz zu inhomogenen Koordinaten einen Punkt nicht eindeutig bestimmen. Jeder homogene Punkt liegt auf einer zugehörigen Ursprungsgerade im dreidimensionalen Raum, wobei der Punkt nicht der Nullpunkt sein darf. Wenn die zusätzliche Dimension  $w$  von  $-\infty$  bis  $\infty$  durchlaufen wird, werden alle Punkte der zugehörigen Gerade beschrieben. Alle Punkte der zugehörigen Gerade werden auf denselben Punkt der Projektionsebene, die standardmäßig bei  $w = 1$  liegt, abgebildet.

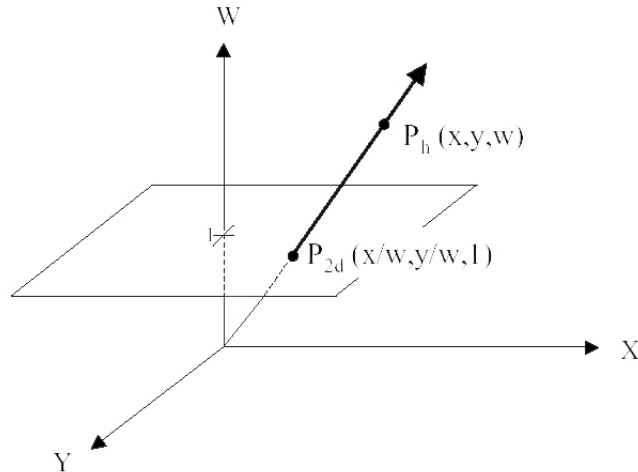


Abbildung 2.2: Veranschaulichung von homogenen Koordinaten [4]

Eine Konsequenz der Nutzung von homogenen Koordinaten ist, dass die Koordinaten gegenüber der Tiefe invariant sind. Die Dualität von Punkt und Gerade ermöglicht es Elemente des projektiven Raums einheitlich darzustellen. Dadurch können alle Projektionen als Matrizen darstellen. Durch Normierung lassen sich die Punkte wieder wie in Gleichung (2.2) in inhomogene Koordinaten zurück rechnen [5] [3].



## 2.2 Transformation

Einfache Transformationen beschreiben Rotation, Scherung, Skalierung und Translation von Körpern. Es gibt verschiedene Arten der Transformation, die nach ihrer Anzahl von verwendeten Freiheitsgrade definiert werden. Eine isometrische Transformation bewahrt Längen und Flächen eines Körpers, es wird um einen Punkt auf dem Körper gedreht und es gibt drei Freiheitsgrade entsprechend dem Rotationswinkel und der Translation in x- und y-Richtung, wie in Abbildung 2.3. Ist die Transformation ähnlich, dann kommt ein Skalierungsfaktor als vierter Freiheitsgrad dazu, zu sehen in Abbildung 2.4. Eine affine Transformation wiederum ermöglicht eine Variation der Winkel, wobei die Parallelität und die Flächenverhältnisse erhalten bleiben. Es ergeben sich sechs Freiheitsgrade, siehe Abbildung 2.5. Die projektive Transformation erweitert durch die Bedingung der Parallelität die Freiheitsgrade um acht, dargestellt in Abbildung 2.6.

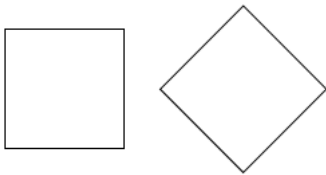


Abbildung 2.3: isometrisch[6]



Abbildung 2.4: ähnlich [6]

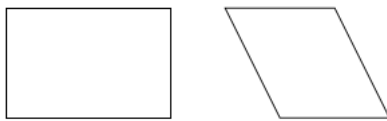


Abbildung 2.5: affin [6]

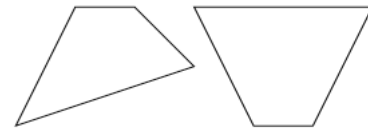


Abbildung 2.6: projektiv [6]

Die Transformationen mit denen sich diese Arbeit in diesem Kontext näher beschäftigen will, sind Rotation und Translation im dreidimensionalen Raum. Transformationen wie Scherung und Skalierung ergeben sich aus Rotation und Translation bei der Abbildung der dreidimensionalen Ebenen in den zweidimensionalen Raum [3]. Beispielsweise wird ein Objekt, das weiter von der Kamera entfernt ist als ein anderes in der Projektion kleiner erscheinen als das andere Objekt, welches näher an der Kamera ist.

Die Rotation eines Punktes im  $\mathbb{R}^3$  um eine Rotationsachse  $\mathbf{n}$ , mit  $\|\mathbf{n}\| = 1$  mit dem Winkel  $\alpha$  erfolgt über die Multiplikation mit einer Drehmatrix, die von einem festen Koordinatensystem ausgeht. Um eine Rotation um eine beliebige Ursprungsgerade oder Achse durchzuführen, deren Richtung und Orientierung durch einen beliebigen normierten Vektor  $\mathbf{n} = [n_x, n_y, n_z]$ ,  $\|\mathbf{n}\| = 1$  definiert wird existiert die allgemeine Gleichung (2.3) [7]. Die Gleichung (2.3) ist die allgemeine Darstellung der Transformation des Koordinatensystems des Körpers, sodass es in der Form des Weltkoordinatensystems vorliegt. Ein Weltkoordinatensystem wird ebenfalls als kartesisches Koordinatensystem repräsentiert. In diesem Koordinatensystem werden relative Koordinatensysteme, wie die des betrachteten Körpers, referenziert. Es wird um eine bekannte Achse des Weltkoordinatensystems gedreht und an-

schließlich das Koordinatensystem des Körpers wieder zurück transformiert. Damit wurde um eine beliebige Achse durch den Ursprung gedreht.

$$R_n(\theta) = \begin{bmatrix} n_x^2 + (1 - n_x^2)\cos\theta & n_x n_y(1 - \cos\theta) - n_z \sin\theta & n_x n_z(1 - \cos\theta) + n_y \sin\theta \\ n_x n_y(1 - \cos\theta) + n_z \sin\theta & n_y^2 + (1 - n_y^2)\cos\theta & n_y n_z(1 - \cos\theta) + n_x \sin\theta \\ n_x n_z(1 - \cos\theta) + n_y \sin\theta & n_y n_z(1 - \cos\theta) + n_x \sin\theta & n_z^2(1 - \cos\theta) + n_x \sin\theta \end{bmatrix} \quad (2.3)$$

Die Translation eines Punktes durch eine einfache Addition des Verschiebungsvektors  $\mathbf{t} = [t_1, t_2, t_3]$  erreicht werden. Soll ein Körper verschoben werden, muss die Verschiebung auf alle Punkte des Körpers angewandt werden. Wenn homogene Koordinaten verwendet werden, kann die Rotation und Translation in einer einzigen Transformationsmatrix dargestellt werden und mit den Punkten des Körpers multipliziert werden. Gleichung (2.4) veranschaulicht das allgemeine Prinzip für eine Transformationsmatrix mit Rotation.

$$T = [R \ t] = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \quad (2.4)$$

Soll um einen beliebigen Punkt im Raum rotiert werden, müssen zunächst alle Punkte des Körpers so verschoben werden, dass der Rotationsmittelpunkt im Koordinatenursprung liegt. Dieser Vorgang erfolgt durch die Matrix  $T$ . Es wird beliebig um einen Winkel rotiert und der Körper wieder so zurück verschoben, dass sich der Rotationsmittelpunkt wieder an seiner ursprünglichen Position befindet. Es wird also  $T$  rückgängig gemacht, dies kann durch die Invertierung von  $T$  als  $T^{-1}$  beschrieben werden. Die zusammengesetzte Transformationsmatrix  $M$  ergibt sich zu (2.5).  $R$  beschreibt die Rotationsmatrix.

$$x''' = M \cdot x = T^{-1} \cdot R \cdot T \cdot x \quad (2.5)$$

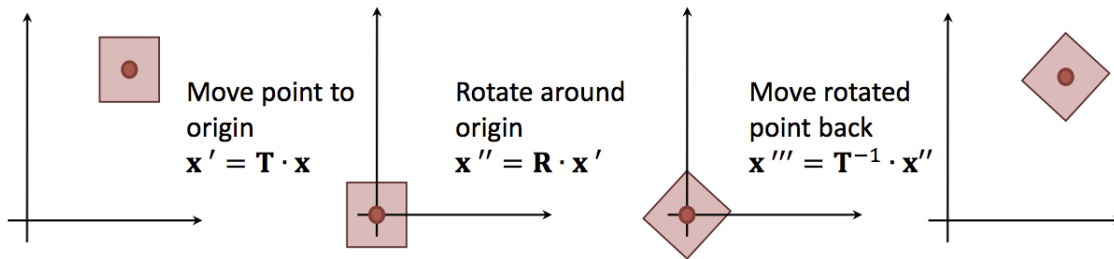


Abbildung 2.7: Vorgehen bei Rotation um einen beliebigen Punkt [7]

## 2.3 Projektion

Bei der Projektion geht es nun darum, unsere im dreidimensionalen Raum transformierten Starrkörper im zweidimensionalen Raum perspektivisch korrekt darzustellen. Die drei Koordinatensysteme für die Weltkoordinaten, Kamera und Bild spielen dabei eine wichtige Rolle. Die Kamera bildet die dreidimensionalen Koordinaten eines Objekts auf den zweidimensionalen Raum ab. Es gibt verschiedene Herangehensweisen um diese planaren Projektionen zu erzeugen. Die wesentlichen sind die perspektivische Projektion (Center of Projection, COP) und die parallele Projektion (Projektionsrichtung, DOP). Der Unterschied zwischen beiden Darstellungen ist das Verhalten von Parallelen. Bei der perspektivischen Darstellung laufen die Projektionsstrahlen auf einen Punkt zu, bei der parallelen Projektion bleiben die Strahlen frei von einem Fluchtpunkt und werden parallel auf der Bildebene abgebildet. Die Problemstellung dieser Arbeit beschäftigt sich vorwiegend mit der perspektivisch korrekten Darstellung des dreidimensionalen Raums.

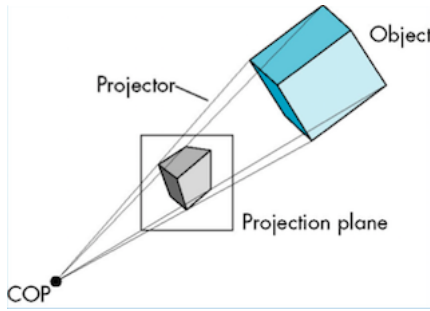


Abbildung 2.8: Perspektivische Projektion [8]

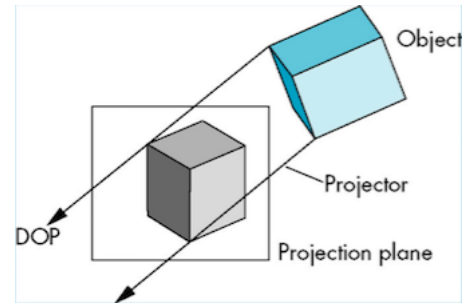


Abbildung 2.9: Parallele Projektion [8]

Perspektivisch korrekte Projektionen werden durch eine Projektionsmatrix  $P$  berechnet. Diese Matrix setzt sich aus der Kamera Kalibrierungsmatrix  $K$  und der Lage der Kamera im Raum, beschrieben durch  $[R \ t]$  zusammen. Gleichung (2.6) zeigt eine Form der Projektion mit homogenen Koordinaten, sie ist allgemein zur Darstellung des Bildes in Pixelkoordinaten gedacht. Die Parameter der Kameramatrix  $f_x$  und  $f_y$  beschreiben die Skalierung des Bildes in x-Richtung und y-Richtung.  $u_0$  und  $v_0$  geben den so genannten *principal point* an, in dem die optische Achse die Bildebene schneidet. Um die Projektionsdaten zu erhalten, wird die Matrix  $P$  auf die homogenen Datenpunkte angewandt.

$$P = K * [R \ t] = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \quad (2.6)$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = P * \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} \quad (2.7)$$

Die folgenden Darstellungen verdeutlichen, was in der obigen Gleichung berechnet wird und was die verwendeten Parameter bedeuten. Die rechte Abbildung in 2.10 zeigt die Z-Achse als optische Achse, die die Bildebene im *principal point* schneidet und beispielhaft, wie eine Gerade mit dem Ortsvektor eines Punktes  $x$  als Richtungsvektor, die Bildebene schneidet und so seine Projektionsposition ermittelt wird. Die rechte Seite in 2.10 zeigt die auf der linken Seite gezeigte Bildebene von vorne. Ergänzend ist das Kamerakoordinatensystem mit den Achsenbezeichnungen  $u$  und  $v$  eingezeichnet. Der *principal point* in der Bildmitte trägt die Bezeichnung  $(u_0, v_0)$ .

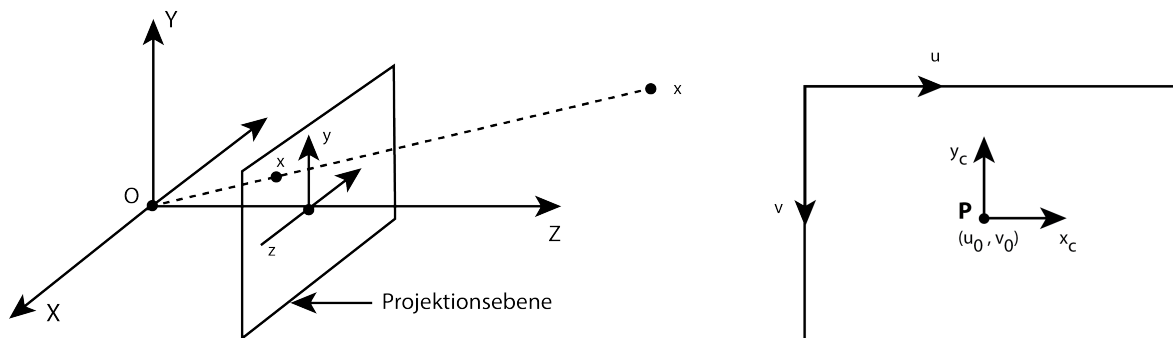


Abbildung 2.10: Projektion von 3D auf 2D (links) Principal Point und Kamerakoordinaten, Bildursprung oft oben links (rechts) [6]

Im Laufe der Arbeit werden die hier beschriebenen Prinzipien wichtig, da dreidimensionale Testdaten durch Projektion einer Kameramatrix in den zweidimensionalen Raum abgebildet werden um Bilddaten einer Kameraaufnahme zu simulieren. Diese Bilddaten liefern den Ausgangspunkt der folgenden Untersuchungen.

## 2.4 Homographie

Wird eine Ebene im Raum starr bewegt und in die Bildebene projiziert, ergibt sich in der Bildebene eine Homographie zwischen den projizierten Ebenen. Eine Homographie kann durch Punktkorrespondenzen von mindestens vier Punkten eine Transformationsmatrix schätzen, die auf den Körper oder in unserem Fall auf eine Ebene angewendet worden ist. Dadurch ist es zum Beispiel möglich, eine Ebene im projektiven Raum auf jede andere projektive Darstellung der Ebene zurückrechnen. Ein Anwendungsfall dieser Technik ist zum Beispiel die Frontalansicht eines in Form einer Ebene ausgewählten Bildbereiches zu erzeugen, um ihn zu Referenzmaterial zu vergleichen. Im Anwendungsfall dieser Arbeit wird sich jedoch auf die Identifizierung von Bewegungen konzentriert, die Punktkorrespondenzen von Frame zu Frame ermittelt und so die Homographie geschätzt. Die Qualität der Schätzung kann durch ihre Abbildungsfehler bestimmt werden. Der Abbildungsfehler gibt wieder, wie gut die Zielposition der Ausgangspunkte getroffen worden ist. Dadurch, dass mit synthetischen Daten gearbeitet wird, sind die Punktkorrespondenzen klar.

Im Folgenden eine schematische Grafik, die das Prinzip der Homographie in unserem Anwendungsfall verdeutlichen soll. Es wird von einer stationären Kamera ausgegangen, die den



Abbildung 2.11: Auswahl des perspektivisch verzerrten Bereichs [3]

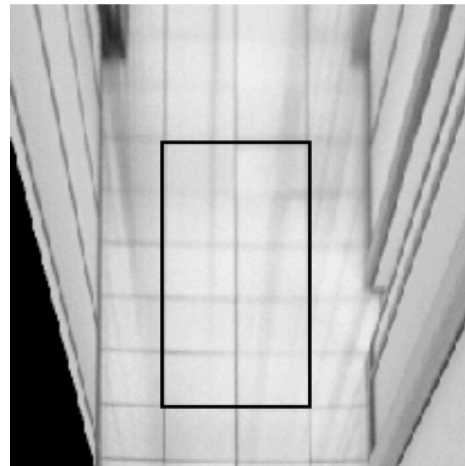


Abbildung 2.12: Vogelperspektive des ausgewählten Bereichs durch Homographie [3]

Ursprung für das Weltkoordinatensystem der Ebenen bildet. Die Ebenen bewegen sich im Bildbereich der Kamera, der Fall, dass sich die Ebenen aus dem Bildbereich bewegen, wird hier vernachlässigt. In der Darstellung 2.13 wird nur eine Punktkorrespondenz von Punkt  $\mathbf{x}$  der Ausgangsebene  $\pi$  zum Punkt  $\mathbf{x}'$  der transformierten Ebene  $\pi'$  gezeigt, die eine der mindestens vier Gleichungen des Gleichungssystems (2.15) zur Schätzung der Transformationsmatrix oder Homographiematrix  $H$  liefert.

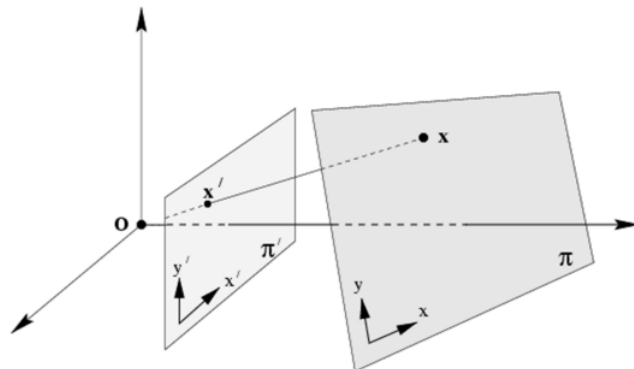


Abbildung 2.13: Bewegung einer Ebene und Ermittlung ihrer Punktkorrespondenzen. [9]

Bei einer planaren Homographie sind acht Freiheitsgrade und damit Unbekannte existent (siehe 2.6). Die neunte Komponente ist die Skalierung in der zusätzlichen Dimension, die durch die homogenen Koordinaten eingeführt wurde und ist daher vernachlässigbar.

Das Ziel ist wie schon erwähnt, aus Punktkorrespondenzen eine Homographiematrix  $H$  wie in Form (2.8) zu schätzen. Wenn die Gleichung (2.8) durch Matrixmultiplikation umgeschrieben wird, ergeben sich die Gleichungen (2.9) und (2.10), wenn die zusätzliche Dimension  $w = 1$  ist. Durch Umformung erhält man die Gleichungen (2.11) und (2.12) und schließlich für jede

Punktkorrespondenz eine Gleichung der Form (2.13) und (2.14), die in einem Gleichungssystem zusammengefasst werden können. Dieses Gleichungssystem ist in (2.15) dargestellt. Es wird von den minimal vier korrespondierenden Punkten  $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$  der Ausgangsebene  $\pi$  zur Zielebene  $\pi'$  der Form  $(x'_1, y'_1), (x'_2, y'_2), (x'_3, y'_3), (x'_4, y'_4)$  ausgegangen. Es können dem System beliebig viele Korrespondenzen übergeben werden. Jede Korrespondenz gibt Aufschluss über die zu bestimmenden Freiheitsgerade. Das lineare Gleichungssystem ist nur eindeutig lösbar, wenn alle Punkte der gleichen Transformation unterliegen. Ansonsten muss die Homographie durch die Least squares Methode geschätzt werden.

$$\begin{bmatrix} x' \\ y' \\ w' \end{bmatrix} \sim \begin{bmatrix} h1 & h2 & h3 \\ h4 & h5 & h6 \\ h7 & h8 & h9 \end{bmatrix} \begin{bmatrix} x \\ y \\ w \end{bmatrix} \quad (2.8)$$

$$x' = \frac{h1x + h2y + h3}{h7x + h8y + 1} \quad (2.9)$$

$$y' = \frac{h4x + h5y + h6}{h7x + h8y + 1} \quad (2.10)$$

$$x'(h7x + h8y + 1) = h1x + h2y + h3 \quad (2.11)$$

$$y'(h7x + h8y + 1) = h4x + h5y + h6 \quad (2.12)$$

$$x' = h1x + h2y + h3 - h7x'x - h8x'y \quad (2.13)$$

$$y' = h4x + h5y + h6 - h7y'x - h8y'y \quad (2.14)$$

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1x_1 & -y'_1y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2x_2 & -x'_2y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -y'_2x_2 & -y'_1y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & 1 & 0 & 0 & 0 & -x'_nx_n & -x'_ny_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -y'_nx_n & -y'_ny_n \end{bmatrix} \begin{bmatrix} h1 \\ h2 \\ h3 \\ h4 \\ h5 \\ h6 \\ h7 \\ h8 \end{bmatrix} = \begin{bmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ \vdots \\ x'_n \\ y'_n \end{bmatrix} \quad (2.15)$$

Es ergeben sich die Elemente der perspektivischen Transformationsmatrix beziehungsweise unserer Homographie, die unsere Abbildung schätzt. Die Homographiematrix wird später beim Clustering zur Bestimmung eines Abbildungsfehlers, der gemäß der Methode in Kapitel 3 unter Abschnitt 3.2 genutzt.

## 2.5 Clustering

Clustering Verfahren erkennen Strukturen in ungeordneten Daten, um sie so in Gruppen oder Cluster zu unterteilen, sodass die Daten der einen Gruppe untereinander ähnlicher sind als zu den Daten in den anderen Gruppen. In Abbildung 2.14 ist das allgemeine Prinzip von Clustering in seiner einfachsten Form zu sehen. Zum Finden von Clustern gibt es viele unterschiedliche Ideen und Ansätze, wie zum Beispiel die Dichte von Daten oder ihre Wahrscheinlichkeitsverteilung. Clustering ist kein zu automatisierender Prozess, sondern ein Lernprozess, der Erfolg und Misserfolg mit einschließt. Clustering ist insgesamt ein spezifischer Algorithmus sondern eine Zielsetzung, die erfüllt werden soll. Mehrere Algorithmen, die sehr unterschiedlichen Prinzipien folgen, können Teil davon sein das gesetzte Ziel des Clusterings zu erreichen [10].

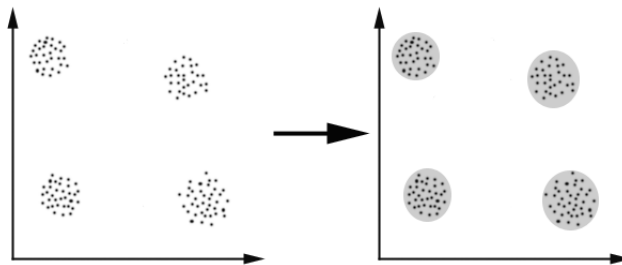


Abbildung 2.14: Allgemeine Verdeutlichung von Clustering [11]

Ein Clustering Verfahren sortiert Daten nach Parametern der Gruppenzugehörigkeit, die gegebenenfalls verschieden gewichtet werden können. Nicht immer ist die Zuordnung eindeutig, wenn zum Beispiel Kriterien auf einen Cluster genauso zutreffen, wie auf einen anderen. Um dennoch eine Zuordnung vorzunehmen, können Prioritäten für die Clusterparameter festgelegt werden. Das richtige Setzen der Parameter für das Clustering hängt jeweils stark von der Problemstellung, dem Datensatz und der Weiterverarbeitung der Daten ab. Die Parameter müssen den Umständen entsprechend so angepasst werden, dass sie der Problemstellung genügen.

Die Sortierung von Daten erfolgt beim Clustering meistens nicht instantan sondern erst nach mehreren Iterationen, bei denen die Bedingungen des Clusterings jedes Mal wiederholt geprüft werden. Wenn der Fehler der Zuordnung einen bestimmten Wert erreicht hat, war das Clustering erfolgreich und es ergibt sich eine eindeutige Zuordnung. Komplizierter wird es, wenn es zur Überschneidung der Daten kommt [11].

Die in den folgenden Abschnitten vorgestellten Verfahren, die die Basis für das hier entwickelte Clustering Verfahren bilden, optimieren nach Abstand [11] [10]. Viele beliebte Clustering Verfahren funktionieren auf diese Art und Weise, wie zum Beispiel in den nächsten Abschnitten vorgestellten k-means und Fuzzy C-means Algorithmen. Es werden hier nur diese zwei Verfahren des Clusterings vorgestellt, da sie für das weitere Verständnis der Arbeit wichtig sind.

### 2.5.1 k-means

Der k-means Algorithmus gehört durch seine Effizienz und Schlichtheit zu einem der beliebtesten Clustering Algorithmen. Die grundlegende Idee ist  $k$  Clustermittelpunkte initial zufällig zu setzen (Abbildung 2.15). Ziel ist die Minimierung des mittleren Abstandes jedes Clusterpunkts  $\mathbf{x}_i$  zu den jeweiligen Clustermittelpunkten  $\boldsymbol{\mu}_j$  (Abbildung 2.16). Die Anzahl der Cluster muss dem k-means Algorithmus mitgegeben werden. Der Algorithmus berechnet nach dieser ersten Initialisierung die neuen Clustermittelpunkte durch den Mittelwert der zugeteilten Clusterpunkte. Anschließend wird eine erneute Berechnung der Abstände zum neuen Clustermittelpunkt vorgenommen und anhand dessen eine neue Einteilung der Datenpunkte zu den Clustern vorgenommen (Abbildung 2.18). Die letzten beiden Schritte des Algorithmus werden so lange wiederholt bis der Algorithmus konvergiert und damit ein endgültiges Clustering vorliegt. Zusammenfassend wird der Abstand zu allen existierenden Clustermittelpunkten nach Gleichung (2.16) minimiert und nach diesem Kriterium einem neuen Cluster zugeordnet. Die Anzahl der Datenpunkte ist  $n$ .

$$J_m = \sum_{j=1}^k \sum_{i=1}^n \| \mathbf{x}_i - \boldsymbol{\mu}_j \|^2 \quad (2.16)$$

Vom k-means Algorithmus gibt es viele verschiedene Variationen um das Clustering zu optimieren. Beliebte Variationen sind zum Beispiel der k-medoid Algorithmus, bei dem jedes neue Clusterzentrum ein Datenwert aus der Datenmenge ist. Der k-means++ startet das Clustering hingegen mit Clusteringzentren, die den gleichen Abstand zueinander haben, um den Erfolg des Clusterings zu garantieren. Bei der Clusterzuteilung bezeichnet man k-means, seine Variationen und andere Algorithmen dieser Art als *hartes Clustering*. Die Daten werden eindeutig einem Cluster zugeordnet, das heißt es gibt pro Iteration über den Datensatz nur eine aktuelle Möglichkeit für jeden Datenwert zu einem Cluster zu gehören, bis der Algorithmus konvergiert. Hier eine Zusammenfassung des k-means Algorithmus:

1. Platziere  $k$  initiale Clustermittelpunkte zufällig im Bereich der zu clusternden Daten
  2. Ordne jedem Datenwert den Cluster zu, dessen Mittelpunkt am nächsten ist
  3. Wenn alle Werte zugeordnet sind, berechne die Position der  $K$  Clustermittelpunkte neu.
  4. Wiederhole Schritt 2 und 3 bis die Clustermittelpunkte sich nicht mehr verändern.
- [11]



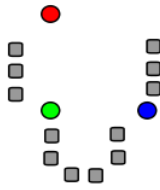
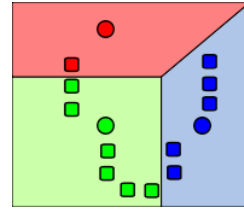

 Abbildung 2.15: 1. Schritt (hier  $k=3$ )[12]


Abbildung 2.16: 2. Schritt [12]

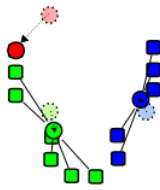


Abbildung 2.17: 3. Schritt [12]

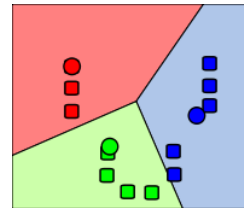


Abbildung 2.18: Ergebnis 4. Schritt [12]

### 2.5.2 Fuzzy C-means

Der Fuzzy C-means Algorithmus nutzt ein *weiches* Clustering und ist vom k-means abgeleitet. In Abbildung 2.19 und 2.20 ist der direkte Unterschied zwischen einem auf Wahrscheinlichkeiten basierendem Algorithmus und einem harten Clustering zu sehen. K-means ordnet Daten direkt und eindeutig einem Cluster zu, Fuzzy C-means ordnet die Daten nur zu einem gewissen Grad einem Cluster zu. Die Wahrscheinlichkeiten der Werte auf der x-Achse sind auf der m-Achse aufgetragen, m steht für *member* oder Mitglied. Das harte Clustering in Abbildung 2.19 ist durch einen abrupten Sprung gekennzeichnet. Alle Werte unterhalb und oberhalb dieser Grenze gehören zu unterschiedlichen Clustern. Beim weichen Clustering zeigt die Zuordnungsfunktion in Abbildung 2.20, dass es für die Daten möglich ist zu mehr als einem Cluster zu gehören. Diese Zuordnung kippt zugunsten des einen oder anderen Clusters, sobald die Wahrscheinlichkeit einen gesetzten Wert annimmt. Die Grenze, ab der die Daten zum Cluster A oder B gehören ist durch eine gestrichelte Linie dargestellt.

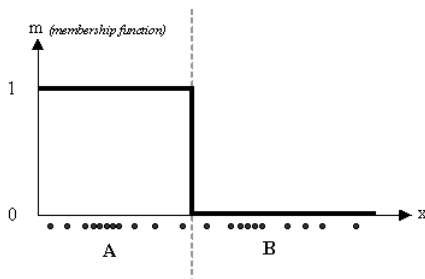


Abbildung 2.19: Die Daten gehören pro Iteration fest zu einem Cluster A oder dem anderen Cluster B [11]

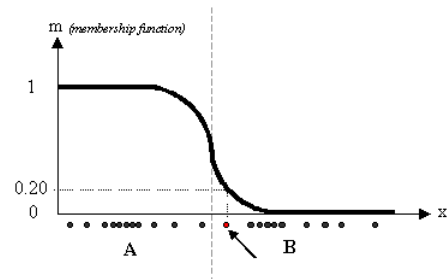


Abbildung 2.20: Weiches Clustering der Daten, Zugehörigkeit zu verschiedenen Clustern A und B ist möglich [11]

Jedem Punkt im Cluster wird ein Satz von Gewichten zugeordnet, die zusätzlich zur Abstandsberechnung herangezogen werden. Die zugeordneten Gewichte pro Punkt werden Fuzzy-Zahlen genannt und entsprechen den Wahrscheinlichkeiten. Wie beim k-means bezeichnet  $\mathbf{x}_i$  den aktuellen Datenwert und  $\boldsymbol{\mu}_j$  das aktuelle Clusterzentrum. Die Zugehörigkeit  $u_{ij}$  und die Clusterschwerpunkte  $c_j$  werden gemäß der Funktion (2.17) optimiert. Der Faktor  $u_{ij}$  gewichtet den Abstand zum Clustermittelpunkt je nachdem, wie wahrscheinlich es ist, dass der Wert zu dem Cluster gehört. Fuzzy C-means wird nun ähnlich wie der k-means so lange iteriert, bis sich die Clusterzentren nicht mehr ändern beziehungsweise keine neuen Datenwerte mehr hinzukommen. Im folgenden Schema ist dargestellt, wie der Algorithmus funktioniert.

$$J_m = \sum_{j=1}^k \sum_{i=1}^n u_{ij}^m \| \mathbf{x}_i - \boldsymbol{\mu}_j \|^2, 1 \leq m \leq \infty \quad (2.17)$$

1. Initialisierung der Wahrscheinlichkeitsmatrix  $U = [u_{ij}], U^{(0)}$
2. Berechnung der Clustermittelpunkte pro Cluster  $k$  ist  $C^{(k)} = [c_j]$  mit der Wahrscheinlichkeitsverteilung  $U^{(k)}$

$$c_j = \frac{\sum_{i=1}^n u_{ij}^m \cdot x_i}{\sum_{i=1}^n u_{ij}^m} \quad (2.18)$$

3. Update der Wahrscheinlichkeitsverteilung aktuelle  $U^{(k)}$  und der folgenden Wahrscheinlichkeitsverteilung  $U^{(k+1)}$

$$c_j = \frac{1}{\sum_{k=1}^C \left( \frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \quad (2.19)$$

4. Wenn die Differenz aus aktueller Wahrscheinlichkeitsverteilung und nächster Wahrscheinlichkeitsverteilung kleiner als das Abbruchkriterium  $\epsilon$  ist, dann stoppe den Algorithmus. Sonst wiederhole Schritt 2 und 3.

$$\| U^{(k+1)} - U^{(k)} \| < \epsilon \quad (2.20)$$

$n$  Anzahl der Datenwerte

$m$  Anzahl der Mitglieder

$C$  Anzahl aller Cluster

$u_{ij}$  Wahrscheinlichkeit für das aktuelle Cluster

$x_i$  Aktueller Datenwert

$c_j$  Aktuelles Clusterzentrum

$c_k$  Aktuelles Clusterzentrum zur Wahrscheinlichkeitsberechnung

[11]

## 2.6 Finden von Ausreißern

Ein Ausreißer ist ein Wert, der eine starke Abweichung von den anderen Werten des Datensatzes aufweist. Das Verfahren, welches hier zum Finden von Ausreißern genutzt werden soll, heißt MAD (*Median absolute deviation*). MAD nutzt Mediane anstatt von Mittelwerten um den Akzeptanzbereich der Daten zu ermitteln, um robuster gegen starke Ausreißer zu sein. Die auf Ausreißer zu untersuchenden Daten seien  $\mathbf{x} = x_1 \cdots x_n$  mit  $n$  Datenpunkten. MAD berechnet zunächst den Median  $\bar{x}$  des Datensatzes. Der Betrag vom Median der Differenzwerte von jedem Datenpunkt und dem Median des Datensatzes ergibt den  $MAD(x)$  Wert des Datensatzes. Schließlich wird der MAD Score  $M_i$  für jeden Datenwert berechnet, der über die Aussortierung des Datenwertes entscheidet. Die Gleichungen (2.21) und (2.22) fassen das Verfahren zusammen.

$$MAD(\mathbf{x}) = median|\mathbf{x} - \bar{x}| \quad (2.21)$$

$$M_{i=1}^n = \left| \frac{x_i - \bar{x}}{MAD(\mathbf{x})} \right| \quad (2.22)$$

Für einen großen Datensatz mit normalverteilten Daten wird bei einem MAD Score von 3.5 zum Aussortieren geraten [13]. Im Anwendungsfall dieser Arbeit ist von nicht normal verteilten Daten auszugehen, daher gilt es den Standardwert zum Aussortieren der Daten anzupassen.



### 3 Analyse und Konzeption

In diesem Kapitel wird die Konzeption eines Clustering Verfahrens zum Erkennen von artikulierten Objekten in einer Bildsequenz beschrieben. In dieser Bachelorarbeit wird von einem vereinfachten Szenario ausgegangen, bei dem die Bewegungskomponenten durch Ebenen simuliert werden. Grundlage zum Erkennen der Objekte wird der in Kapitel 2 Abschnitt 2.5.2 beschriebene Fuzzy C-means sein. Das Ergebnis des Vorclustering wird durch Homographie Information optimiert, sodass alle Objekte in der Bildsequenz zuverlässig gefunden werden. Artikulierte Objekte zeichnen sich dadurch aus, dass sie aus mehreren Komponenten bestehen, die durch Gelenkpunkte miteinander verbunden sind. Die Komponenten des Objekts können sich starr zueinander bewegen. Zum Entwickeln des vorliegenden Clustering Verfahrens wurden einige Parameter vernachlässigt, um einen einfacheren Fall artikulierter Objekte zu erzeugen. Die betrachteten Objekte sind ausschließlich zweidimensionale Ebenen, die sich im dreidimensionalen Raum bewegen und zur Untersuchung in die zweidimensionale Bildebene projiziert werden. Außerdem ist die Bewegung gleichmäßig und es werden synthetisch erzeugte Bilddaten verwendet. Die synthetischen Daten bieten den Vorteil, dass sie das entwickelte Verfahren stets überprüfbar und analysierbar machen. Zwei Formen der artikulierten Bewegung, die durch das selbst entwickelte Framework zur synthetischen Datenerzeugung simuliert wurden, sind in 3.1 dargestellt.

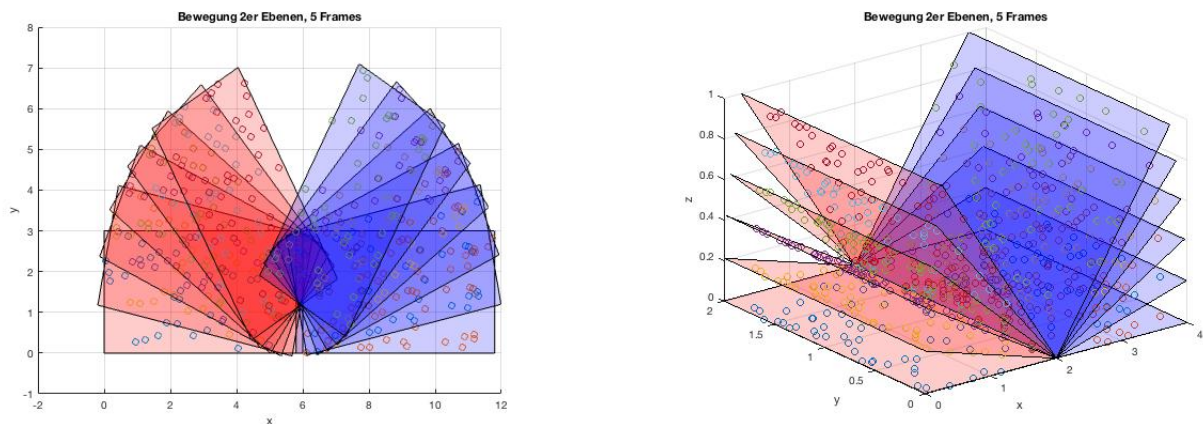


Abbildung 3.1: Beispiele artikulierter Bewegungen, Darstellung aller Framebilder. Die dargestellten Punkte werden pro Ebene und Frame in unterschiedlichen Farben dargestellt.

Es kann angenommen werden, dass Punkte die zu einer Ebene gehören, dicht beieinander liegen. Aus diesem Grund wurde für das Vorclustering ein Fuzzy C-means gewählt, der den

euklidischen Abstand der erzeugten Bildpunktkoordinaten optimiert. Wenn im Folgenden von Fuzzy C-means die Rede ist, wird genau von dieser Modifikation dieses Algorithmus gesprochen. Eine weitere Eigenschaft der Punkte einer Ebene ist, dass sie derselben Projektion unterliegen. Unter der Annahme, dass der Anteil der Überlappung der projizierten Punkte kleiner ist als zu den restlichen Punkten einer Ebene, wurde das Prinzip eines zweistufigen Clusterings entwickelt, welches zunächst auf der Basis der Bildkoordinaten vorclustert und in einem zweiten Schritt untersucht, welcher Homographie die Punkte einer Ebene unterliegen. Um initiale Cluster aus den übergebenen Punkten zu erkennen, ist der Fuzzy C-means daher gut geeignet. Das Verwenden von Wahrscheinlichkeiten macht das Vorclustering robuster im Fall von sich überlappenden Daten, als ein hartes Clustering wie der k-means Algorithmus. Da bei artikulierten Objekten bei Definition Überschneidungen an den Gelenkpunkten auftreten ist der Fuzzy C-means Algorithmus der Wahl für das Vorclustering. Überschneidungen können auch im Laufe der Bewegung auftreten, wenn sich die Bewegungskomponenten perspektivisch überschneiden. Wenn die grobe Aufteilung der Cluster bekannt ist, kommt es mit hoher Wahrscheinlichkeit zu Fehlern in der Cluster Zuordnung im Überschneidungsbereich der Ebenen. Obwohl das Vorclustering in dieser Hinsicht zuverlässiger ist als der k-means, ist seine Zuordnung oft nicht perfekt. Für die Verfeinerung des Clusterings wird nun die Information der Bewegung genutzt. Durch die Erkennung einer Homographie und der Zuordnung aller Punkte zu einer erkannten Homographie sollte das eindeutige Erkennen aller Punkte zu einer Komponente des artikulierten Objekts gewährleisten.

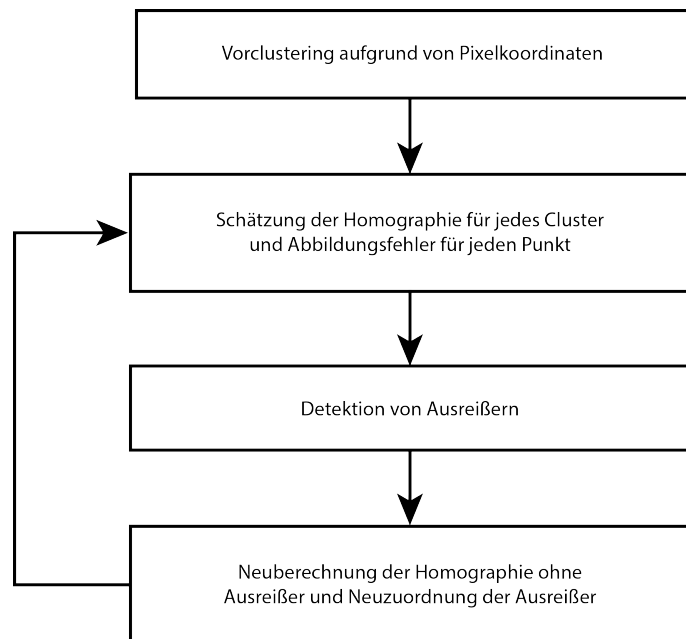


Abbildung 3.2: Prinzip der Verbesserung des Clusterings durch Homographie zusammengefasst.

Um die Homographie schätzen zu können, werden wie in Kapitel 2 Abschnitt 2.4 beschrieben, mindestens vier Punktkorrespondenzen benötigt. Die Korrespondenzen werden aus jeweils

einem Framepaar der Bildsequenz ermittelt. Die Qualität der Bewegungsschätzung ist davon abhängig, ob die richtigen Punktorrespondenzen genutzt worden sind und ob nur Punkte genutzt worden sind, die tatsächlich derselben Bewegung unterliegen. Die Punktkorrespondenzen sind durch die synthetisch erzeugten Daten fehlerfrei. Um die Punkte zu finden, die keiner Bewegung folgen, wird der Abbildungsfehler aller Punkte gemäß Abschnitt 3.2 berechnet. Dabei ist davon auszugehen, dass die meisten Punkte, die an der Schätzung der Homographie beteiligt gewesen sind, einer Bewegung folgen. Sonst ist der Abbildungsfehler ohne Aussage, da dieser Fehler sich auf eine falsche Bewegungsschätzung beziehen würde. Genau aus diesem Grund ist das Vorclustering der Daten sehr wichtig. Abbildung 3.2 zeigt zusammengefasst, wie die Verbesserung des Clusterings gedacht ist. Auf Details dieses Prinzips wird in Abschnitt 3.3 eingegangen.

### 3.1 Vorclustering mit Fuzzy C-means

Der Fuzzy C-means Algorithmus wurde im Vorfeld der Konzeption bereits umgesetzt, um die Qualität des Clusterings bewerten zu können und das Clustering Verfahren besser konzipieren zu können. Wenn die Cluster perfekt gefunden werden, ergeben sich eindeutige Wahrscheinlichkeiten für jeden Punkt. In Abbildung 3.3 ist der optimale Fall der Wahrscheinlichkeitsverteilung für des Vorclustering für zwei Cluster zu sehen. Die Daten wurden nach der Höhe Ihrer Wahrscheinlichkeit sortiert, aufsteigend für das rote Cluster und absteigend für das blaue Cluster. Auf der y-Achse sind die Wahrscheinlichkeiten aufgetragen und auf der x-Achse die nach Wahrscheinlichkeitswert sortierten Datenwerte. Es soll mit den sortierten Datenwerten gezeigt werden, wie allgemein die Wahrscheinlichkeitsverteilung der Werte aussieht, sie trifft keine Aussage über bestimmte Werte. Für jeden Punkt ist eine eindeutige Zuordnung in jedes Cluster zu erkennen, sodass die Zugehörigkeiten ähnlich wie beim k-means in Abbildung 2.19 unter Abschnitt 2.5.2 Kapitel 2 aussehen. Dieser optimale Fall tritt im Allgemeinen nur dann auf, wenn die Körper räumlich voneinander getrennt sind. Andernfalls wäre der Schnittpunkt der Wahrscheinlichkeitsverteilungen verschoben und Werte würden falsch zugeordnet werden. Pro Ebene wurde die gleiche Anzahl von Punkten vergeben, sodass sich der Schnittpunkt stets in der Mitte der Darstellung befinden sollte. Die Ergebnisse des Vorclustering werden in Kapitel 5 Abschnitt 5.1 präsentiert und genauer diskutiert.

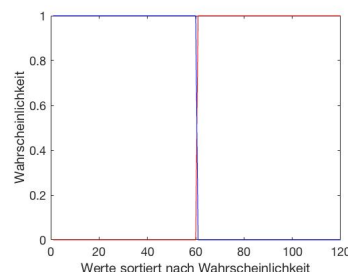


Abbildung 3.3: Optimaler Fall des Vorclustering.

## 3.2 Homographie als Information für das Clustering

Um die Homographie als Parameter des Clusterings zu nutzen, wird zunächst die Homographie aus den Korrespondenzen der Punkte der Cluster aus dem Vorclustering berechnet. Dabei werden die Korrespondenzen zum jeweils nächsten Frame berücksichtigt. Der Fehler der Bewegungsschätzung nennt sich Abbildungsfehler. Dieser Fehlerwert kann darüber berechnet werden, dass die Homographiematrix beziehungsweise geschätzte Transformation  $H$  auf einen Ausgangspunkt  $x$  angewendet wird und die Inverse der Homographiematrix  $H^{-1}$  auf einen Zielpunkt  $x'$  (siehe dazu 2.13). Ergebnis ist jeweils ein geschätzter Zielpunkt  $p'$  und ein geschätzter Startpunkt  $p$ . Die Abweichung des geschätzten Startpunktes und des geschätzten Zielpunktes zu dem eigentlich verwendeten Start- und Zielpunkt ergibt den Abbildungsfehler  $\delta_H$ . Der beschriebene Zusammenhang ist in den Gleichungen (3.1) bis (3.3) dargestellt.

$$p' = H * x \quad (3.1)$$

$$p = H^{-1} * x' \quad (3.2)$$

$$\delta_H = \|x - p\| + \|x' - p'\| \quad (3.3)$$

Unter der Annahme, dass das Vorclustering hinreichend gut ist, wird der Abbildungsfehler der Punkte der Cluster bezüglich ihrer Bewegungsschätzung berechnet. Die Werte mit hohem Fehler werden aus den jeweiligen Clustern aussortiert und gesammelt. Ohne die Werte mit hohem Fehler sollte die Bewegungsschätzung optimiert und der Abbildungsfehler möglichst Null sein. Die optimierte Bewegungsschätzung sollte die Bewegung der Cluster oder Bewegungskomponenten akkurat wiedergeben. Auf der Grundlage der optimierten Bewegungsschätzungen für jedes Cluster kann nun der Abbildungsfehler der aussortierten Werte gegenüber jeder Bewegungsschätzung für jede Komponente berechnet werden. Dem Cluster, dessen Bewegungsschätzung für einen aussortierten Punkt den geringsten Abbildungsfehler erzeugt, wird der aussortierte Punkt zugewiesen. Als Ergebnis sollte eine korrekte Zuweisung aller Punkte und damit eindeutige Cluster vorliegen. Eine optimale Schätzung der Bewegung würde eine Nulllinie für die Darstellung des Abbildungsfehlers erzeugen. Das Aussortieren der Werte kann nach verschiedenen Strategien erfolgen, hier wurde sich für eine Aussortierung der Ausreißer durch das MAD (*Median absolute deviation*) Verfahren entschieden, beschrieben in Kapitel 2 Abschnitt 2.6. Ist die Homographie falsch, ist der erzeugte Abbildungsfehler ohne Aussage. Für die Berechnung der Homographie sind theoretisch nur minimal vier Punkte nötig. Damit sollte ein großzügiges Aussortieren von Ausreißern kein Problem sein. Das Verfahren kann iterativ auf die Daten angewendet werden, um ein optimales Ergebnis für jedes Framepaar zu erhalten. Pro Iteration können weitere auffällige Punkte eliminiert werden, Voraussetzung ist dabei eine richtige Homographie. Das beschriebene Verfahren ist zusammengefasst in Abbildung 3.2 dargestellt.



### 3.3 Konzeption eines Multi-Frame Clusterings

Um ein Multi-Frame Clustering durchzuführen, müssen alle Frames der Bildsequenz untersucht werden, wobei Ziel das eindeutige Erkennen der Ebenen aus der Sequenz ist. Durch das Verwenden von Homographie als Information für das Clustering, werden die Frames immer paarweise betrachtet. Dabei werden die Framepaare überschneidend genutzt. Jeder Frame liefert zunächst die Zielpunkte des vorangegangenen Übergangs und dann die Ausgangspunkte für den nächsten Übergang. Aus den Ausgangs- und Zielpunkten ergeben sich die Punktkorrespondenzen zum Berechnen der Homographie und des Abbildungsfehlers. Das beschriebene Prinzip wird auf alle Frames angewandt, bis auf den ersten und letzten Frame der Bildsequenz. Der Erste Frame hat keinen Vorgänger und liefert daher nur Ausgangspunkte und der Letzte Frame liefert nur Zielpunkte, da er keinen Nachfolger hat. Das beschriebene Prinzip ist in Abbildung 3.4 für die ersten zwei Übergänge angedeutet.

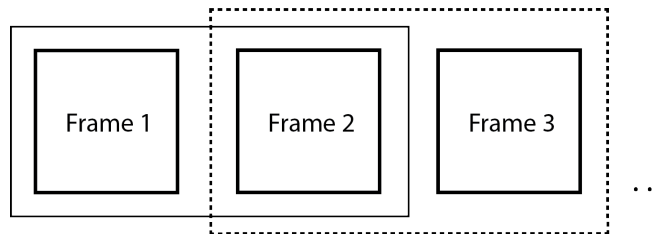


Abbildung 3.4: Prinzip der Homographieschätzung für das Multi-Frame Clustering.

Die Analyse jedes Framepaars soll dem Prinzip aus Abbildung 3.2 folgen. Für jedes Framepaar wird für jeden Punkt und bezüglich jedes Cluster ein Abbildungsfehler berechnet. Das Vorclustering liefert für jedes Framepaar die Ausgangsbasis. Für das Multi-Frame Clustering ist es sinnvoll, das Wissen aus allen Frameübergängen gleichzeitig zu berücksichtigen. Aus dem Wissen aller Frameübergänge kann eine fundierte Aussage über die Zuteilung der Punkte in ein Cluster und damit einem der Körper zugehörig vorgenommen werden. Die Zuordnung der Punkte zu einem Körper oder Cluster bleibt in jedem Frame gleich. Das Wissen aller Frameübergänge kann nun so genutzt werden, dass das Ergebnis aus jedem untersuchten Framepaar in Form des Abbildungsfehlers über die gesamte Bildfrequenz gemittelt wird. Der gemittelte Abbildungsfehler ist das Entscheidungsmaß, aufgrund dem entschieden wird, welchem Cluster die Punkte zugeordnet werden. Das Cluster mit dem geringeren Abbildungsfehler gewinnt. Das Resultat sollte generell relativ robust sein, wenn davon ausgegangen wird, dass das Clustering im Durchschnitt gut funktioniert. Selbst wenn einige Übergänge schlecht geclustert werden, sollte das Endergebnis nicht bedeutend verschlechtert werden.



## 4 Implementierung und Umsetzung

Die Programmierung im Rahmen dieser Arbeit wurde mit MATLAB durchgeführt. MATLAB ist eine kommerzielle Software zur Lösung mathematischer Probleme mit Hilfe von Matrizen und der grafischen Darstellung von Ergebnissen. Es gibt viele Funktionen, die zur Datenverarbeitung genutzt werden können. Dabei werden Daten oft als Einträge von Matrizen verarbeitet. Alle im Rahmen dieser Arbeit entwickelten Funktionen können auch im kostenlosen Octave verwendet werden. Octave ist die Open Source Variante von MATLAB. Alle Funktionen und Skripte unterlagen im Prozess der Bachelorarbeit einer ständigen Entwicklung und Verbesserung.

### 4.1 Framework zur Erzeugung von Testdaten

Zur Erzeugung von Testdaten wurde ein MATLAB Framework implementiert, welches komplett synthetische Daten erzeugt um die untersuchten Clustering Algorithmen zu testen.

Dazu wird in einer Funktion eine Ebene erzeugt, in der sich zufällig erzeugte Datenpunkte befinden, die *interest points* des Objekts simulieren sollen. Um eine zufällige Datenverteilung der Werte auf der Ebenenoberfläche zu erreichen wurde die *random* Funktion von MATLAB genutzt. Dabei wurde durch konditionelle Bedingungen darauf geachtet, dass die Punkte nicht über den Ebenenbereich hinaus streuen. Das Plotten der Ebenen wurde ebenfalls durch eine MATLAB Funktion übernommen. Es wurde sich auf Ebenen als Simulationsobjekt für die Framework Entwicklung konzentriert, da sie in den Vorüberlegungen als Darstellung vereinfachter, artikulierter Objekte gewählt wurden. Der Ebenenbereich, der im Framework farbig dargestellt wurde, dient dabei zur Veranschaulichung des Bereichs, in dem sich die zu untersuchenden Punkte befinden.

Nun sollen sich die Ebenen inklusive der Punkte die sich auf der Ebene befinden im Raum beliebig bewegen können, das heißt in jede Richtung, um jeden Winkel und um jede Achse. Um die Starrkörpertransformation zu implementieren, wurden die Konzepte aus dem Kapitel 2 aus Abschnitt 2.2 in einer Funktion umgesetzt. Der Funktion werden die gewünschten Parameter der Bewegung übergeben und aus ihnen eine Transformationsmatrix berechnet, die auf die Datenwerte angewendet wird.

Alle Punkte werden als homogen betrachtet und die Transformationen in einer Funktion erzeugt und auf die zufällig erzeugten Datenpunkten sowie die Eckpunkte der Ebenen-Plots angewendet. Um sich bewegende *interest points* auf einer Ebene zu simulieren, übernimmt eine weitere Funktion eine Interpolation der Datenwerte von ihrer Ausgangsposition zu einer Zielposition, indem die Anzahl der vorgesehenen Frames pro Sekunde und die Anzahl der Sekunden eingestellt werden kann. Anfangs- und Endwert der Rotation und Translation werden ebenfalls der Funktion übergeben. Aus diesen Angaben wird eine Transformationsmatrix für

jeden Schritt berechnet und durch das Anwenden dieser Matrizen auf die Daten die Werte der synthetischen Bildsequenz erzeugt. Ein Schritt bezieht sich auf einen Frame. In jedem Änderungsschritt werden die Daten gespeichert, sodass sie von der Projektionsfunktion verwendet werden können. Die dreidimensionalen Daten werden durch die Projektionsfunktion auf die zweidimensionale Bildebene der Kamera abbildet. Die Projektionsdaten werden separat gespeichert. Auf die Projektion wird letztendlich ein Clustering Algorithmus angewendet und optional Rauschen hinzugefügt. Der gesamte Ablauf der Funktionsaufrufe ist in einem Skript festgehalten. Für jeden beispielhaften Bewegungsablauf existieren ein eigenes Skript.

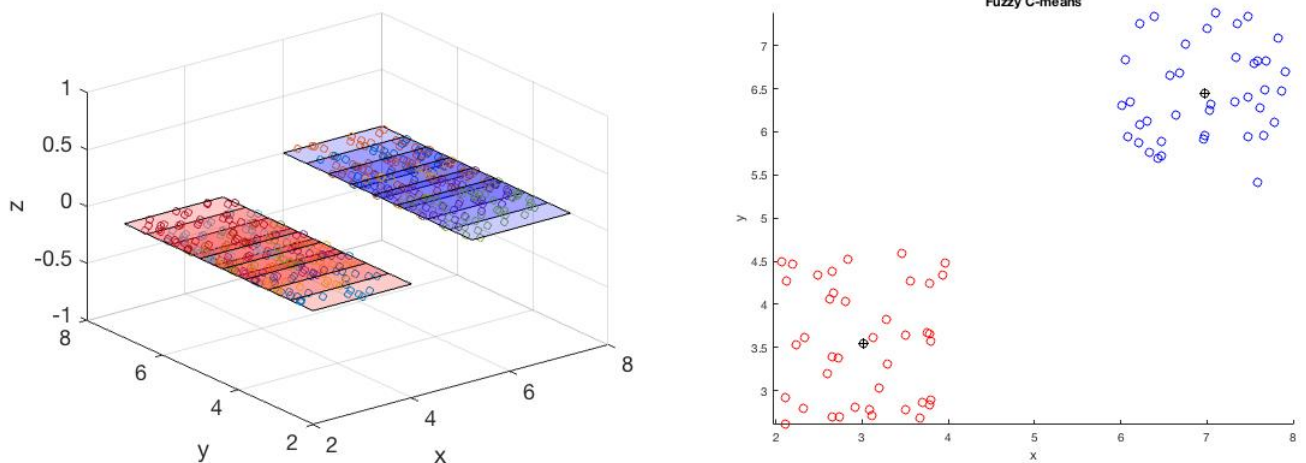


Abbildung 4.1: Visualisierung einer einfachen Translation zweier Ebenen im dreidimensionalen Raum (links) und das Clustering ihrer Positionen aus einem Bildpaar (rechts).

Es existieren zwei Skript Arten im Framework. Einerseits die Skripte, die die Bewegungsdaten erzeugen und andererseits das Multi-Frame Clustering oder Clustering Skript, auf das im nächsten Abschnitt eingegangen wird. Alle Skripte, die die Bewegungsdaten erzeugen folgen dem gleichen Prinzip in der Namensgebung, damit sie alle mit dem Clustering und Multi-Frame Clustering Skript verarbeitet werden können. Die Daten werden bei jedem Aufruf eines Bewegungsskriptes überschrieben. Datensätze, die behalten werden sollen, können in einer .mat Datei gespeichert werden. Es werden alle global definierten Variablen, die sich im Workspace befinden gesichert. Für die Entwicklung des Clustering Verfahrens wurden mehrere Datensätze auf diese Art gespeichert um reproduzierbare Ergebnisse zu erhalten und diese damit vergleichbar zu machen.

## 4.2 Clustering Methoden und Multi-Frame Clustering

Das Clustering erfolgt zwischen allen Framepaaren in überschneidender Form. Es wird also von einem Frame  $n$  jeweils der Übergang des Frames  $n - 1$  zu  $n$  und  $n$  zu  $n + 1$  untersucht, außer für den ersten und letzten Frame der Bildsequenz. Zum Anfang der Entwicklung des

Algorithmus wurde zunächst eine Begrenzung auf einen Frameübergang vorgenommen, da eine nachträgliche Erweiterung nichts an dem Algorithmus ändert, nur an der Art der Daten, die übergeben werden. Das paarweise Clustering aller Frames, wie in Kapitel 3 Abschnitt 3.3 beschrieben, wird im Clustering Skript umgesetzt. Um alle Frames der Bildsequenz zu Clustern, kann das Multi-Frame Clustering Skript ausgeführt werden. Um das Clustering der Videodaten auszuführen, müssen die Daten zunächst über eines der Bewegungsskripte ausgeführt werden oder ein gespeicherter Datensatz verwendet werden. Das paarweise Clustering wird von dem Multi-Frame Clustering Skript so lange durchgeführt, bis über alle Framepaar Übergänge geclustert worden und alle Auswertungsdaten gespeichert sind. Neben den Auswertungsdaten können an verschiedenen Stellen Werte ausgelesen und analysiert werden. In den verschiedenen Skripten werden generell alle verwendeten Funktionen in der richtigen Reihenfolge aufgerufen, die das Clustering betreffen und Variablen so verknüpft, dass die richtigen Werte weitergegeben und verarbeitet werden. Insgesamt befinden sich im Framework mehr Funktionen als sie in den Skripten aufgerufen wurden. Grund dafür ist, dass andere Funktionen, die im Rahmen dieser Arbeit umgesetzt wurden, mit den aktuell verwendeten Funktionen ausgetauscht werden können. Dadurch ist modular untersuchbar, was in verschiedenen Verfahrenskombinationen passiert. Für die Darstellung der Cluster und der Wahrscheinlichkeiten wurden separate Funktionen programmiert um Codedopplungen zu vermeiden. Beim Aufruf des Multi-Frame Clustering Skripts werden automatisch Abbildungen zu den Clustering Ergebnissen der Framepaare ausgegeben. Welche Abbildungen ausgegeben werden sollen ist ebenfalls ein- und ausschaltbar.

### 4.3 Umgang mit Problemen und Testing

Syntaktische Probleme bei der Implementierung waren über den MATLAB Debugger gut zu finden und zu beheben. Logische Probleme sind besonders beim Plot der Daten aufgefallen, oder wenn Daten auf der Konsole ausgegeben worden sind. Das schwierigste beim Entwickeln des vorgestellten Clusterings war die Überlegung, welche Methodiken und Vorgehensweisen am sinnvollsten sind um das gewünschte Ziel zu erreichen. Wenn eine Methodik nicht zum gewünschten Clustering geführt hat, musste neu konzipiert werden. Das Verwerfen und Finden neuer Ansätze waren die größte Herausforderung der Implementierung, neben dem programmieren in MATLAB und das Einfinden in die Denkweise des Programms. Die meisten Verständnisprobleme und Vorgehensweisen konnten in MATLAB Foren und der Dokumentation der Software nachvollzogen werden. Durch den modularen Aufbau des Frameworks können Methoden zur Ausreißerbestimmung oder zur generellen Homographie Verbesserung eingebaut und getestet werden. Diese Eigenschaft war beim gesamten Entwicklungsprozess sehr hilfreich, auch die Möglichkeit beliebige Bewegungen mit verschiedenen Modifikationen für die Testdaten zu generieren. Für jede Idee und Entwicklungsstufe wurde mit verschiedenen Datensätzen getestet und bewertet. Die Vorgehensweise war dabei immer ähnlich der Beschreibung der Ergebnisse in Kapitel 5. Clustering ist ein Lernverfahren, bei dem die Parameter an das gegebene Problem und an die Zielsetzung des Clusterings angepasst werden müssen. Das Ausprobieren der Werte und die Überlegungen, wie diese richtig gesetzt werden könnten waren eins der größten Herausforderungen bei der Entwicklung des Verfahrens.



## 5 Ergebnisse

In diesem Kapitel werden die Ergebnisse des Clusterings von einzelnen Framepaaren und des Multi-Frame Clusterings dargestellt und diskutiert. Es wird gezeigt, wie gut die in Kapitel 3 entwickelte Idee zum Clustern artikulierter Objekte tatsächlich funktioniert hat, Schwächen und Stärken der Methode werden dabei deutlich. Das Ergebnis der einzelnen Stufen des Clustering Verfahrens, bestehend aus Vorclustering und der anschließenden Verbesserung durch Homographie Information wird separat dargestellt. Die separate Darstellung soll die Ergebnisse der einzelnen Clustering Stufen zeigen und analysieren, sodass die Ergebnisse des Multi-Frame Clusterings besser nachvollziehbar werden. Bei der Präsentation der Ergebnisse des Multi-Frame Clusterings werden nur die Ergebnisse nach der zweiten Stufe des Clustering Verfahrens dargestellt, da dies das Ergebnis des Clusterings für jedes Framepaar präsentiert, das zum Auswerten der Bildsequenz genutzt wird.

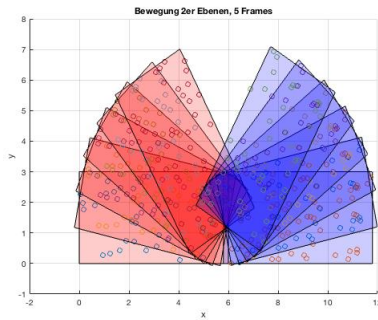


Abbildung 5.1: Bewegung 1

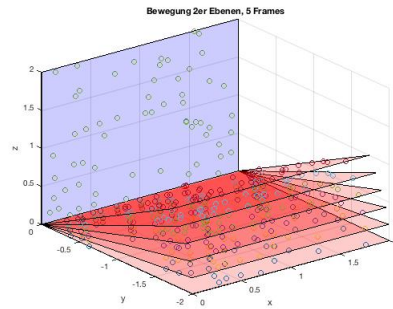


Abbildung 5.2: Bewegung 2

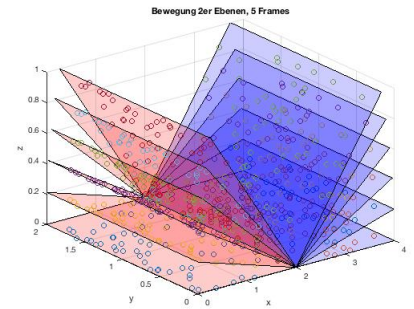


Abbildung 5.3: Bewegung 3

Es wurde im Rahmen dieser Arbeit entschieden, die Bewegungen eins bis drei aus den Abbildungen 5.1 bis 5.3 zu analysieren und diese Ergebnisse zu betrachten. Die Bewegungen sollen eine Steigerung im Schwierigkeitsgrad für das Clustering Verfahren darstellen. Bei Bewegung eins bewegen sich die Ebenen nur in der xy-Ebene gegenläufig, mit enger werdendem eingeschlossenem Winkel. Die xy-Ebene ist die Projektionsebene der Kamera, daher findet für diese Bewegung durch Projektion keine Tiefenverzerrung statt. Tiefenverzerrung findet nur statt, wenn sich die Körper aus der Projektionsebene bewegen, wie bei Bewegung zwei und drei. Bei Bewegung eins aus Abbildung 5.2 bewegt sich die rote Ebene auf die Kamera zu, die andere Ebene bleibt stationär. Projiziert wird wieder die Sicht von oben auf die Bewegung, also aus Sicht der xy-Ebene. Das Gleiche gilt für Bewegung drei aus Abbildung 5.3. Diese Bewegung ist die schwierigste für das Clustering Verfahren, die Ebenen klappen zusammen, so als ob man ein Buch von oben betrachten würde. Beide Buchseiten werden

tiefenverzerrt, daraus ergibt sich eine komplexere Homographie und ein schwierigerer Fall für das Vorclustering.

## 5.1 Vorclustering durch Fuzzy C-means

Wenn das Vorclustering durch den Fuzzy C-means mit Optimierung des euklidischen Abstands der Bildpunktkoordinaten auf überlappende Daten angewandt wird, sind falsche Zuordnungen im Schnittbereich zu beobachten. Dieser Fall ist in Abbildung 5.4 zu sehen. Rechts ist das richtige Clustering, bekannt durch die synthetischen Daten, zu sehen. In der Mitte ist das Ergebnis des Vorclustering dargestellt und rechts das Ergebnis des Vorclustering mit markiertem Unterschied des Ergebnisses zum richtigen Clustering. Der Unterschied ist hier und wird in den folgenden Ergebnissen grün dargestellt.

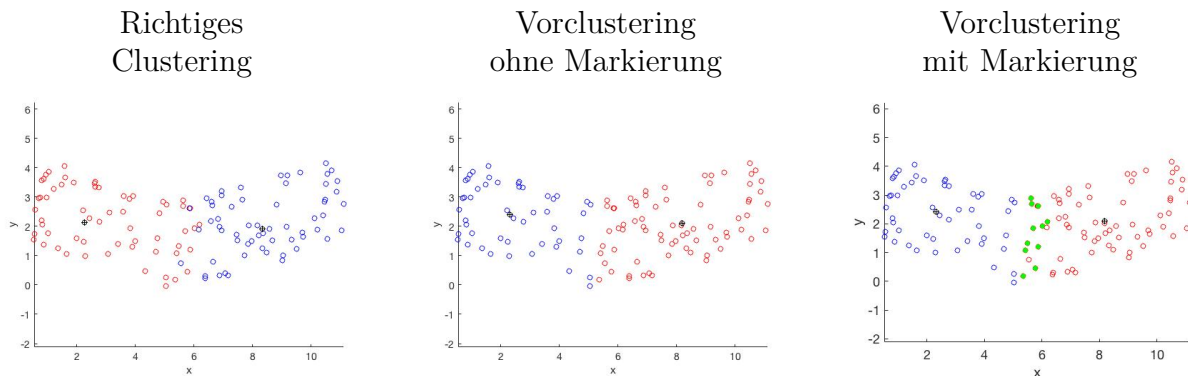


Abbildung 5.4: Verdeutlichung der Zielstellung, hier anhand des Problemfalles von sich überschneidenden Daten. Clustering der Bewegung 1 aus Abbildung 5.1.

Wie in Kapitel 3 Abschnitt 3.1 beschrieben, arbeitet das Vorclustering mit Wahrscheinlichkeiten für jeden Wert. Die folgenden Darstellungen folgen dem gleichen Prinzip, das in dem genannten Abschnitt beschrieben ist. Zusammengefasst wird die Wahrscheinlichkeit aller Punkte dargestellt. Dabei wurden die Wahrscheinlichkeiten jeden Punkt der Cluster der Größe nach sortiert. Für das blaue Cluster absteigend, für das rote Cluster aufsteigend. Für den trivialen Fall von räumlich weit auseinanderliegender Ebenen, die sich parallel aneinander vorbeibewegen, ergibt sich der Optimalfall. Die Abbildung 4.2 zu dieser Bewegung findet sich in Kapitel 4 unter Abschnitt 4.2. Es ist deutlich erkennbar, dass wenige Zwischenwahrscheinlichkeiten vorhanden sind. Umso weniger Zwischenwahrscheinlichkeiten vorhanden sind, desto eindeutiger ist die Einordnung der Werte in die Cluster. Sind die Wahrscheinlichkeiten ungenauer, dann entscheidet der Algorithmus ab dem Schnittpunkt der Wahrscheinlichkeitsverteilungen, ob die Werte zu dem einen oder anderen Cluster gehören. Wie bereits in Kapitel 3 Abschnitt 3.1 vermutet, ordnet das Vorclustering nicht alle Werte korrekt zu. Durch Tiefenverzerrung ändern sich die Abstände unter den Punkten und die Optimierung des euklidischen Abstands der Bildpunktkoordinaten durch den Fuzzy C-means findet nicht alle Zugehörigkeiten. Der euklidische Abstand unter den Punkten ist



nach der Verzerrung kein zuverlässiges Kriterium mehr, Punkte können nach der Verzerrung weiter auseinander liegen aber dennoch zu einem Cluster gehören. Dieses Problem ist in Abbildung 5.6 zu Bewegung zwei zu sehen. Bewegung eins aus Abbildung 5.1 befindet sich in der Projektionsebene. Die Komponenten unterliegen einer Überschneidung. Es finden durch das Vorclustering falsche Zuordnungen statt, dennoch ist das Ergebnis besser als das aus dem Clustering in 5.6. Das Ergebnis des Clusterings ist zum Vergleich in Abbildung 5.4 zu sehen.

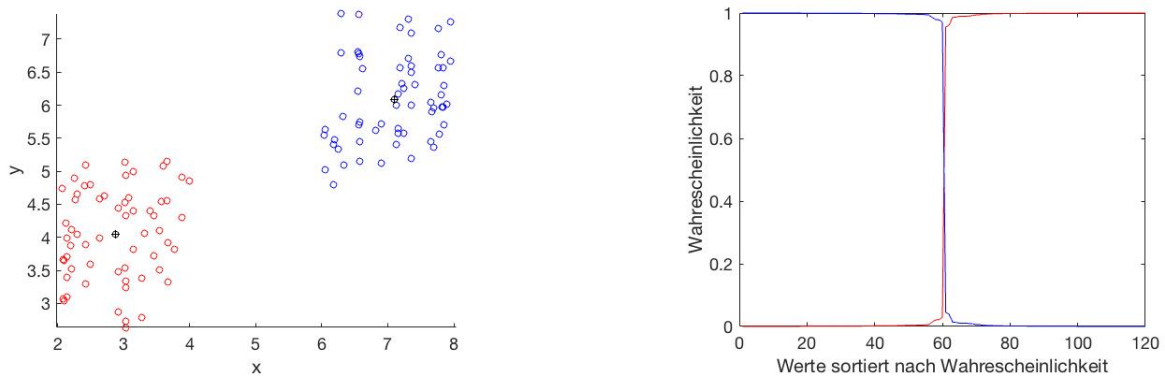


Abbildung 5.5: Fuzzy C-means Algorithmus angewandt auf einen Datensatz von sich parallel aneinander vorbei bewegendem Körper, erzeugt durch das selbst entwickelte MATLAB-Framework.

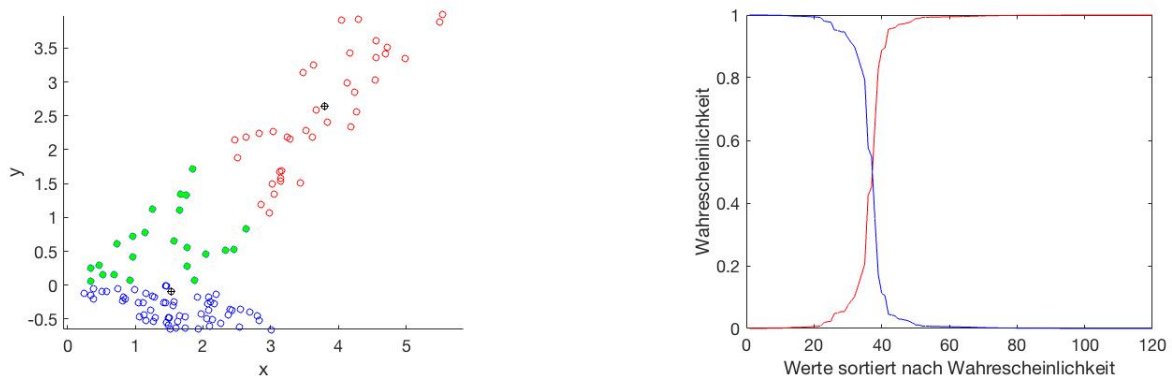


Abbildung 5.6: Fuzzy C-means Algorithmus angewandt auf den Datensatz aus Abbildung 5.2, erzeugt durch das selbst entwickelte MATLAB-Framework.

Mit dieser Menge falsch zugeordneter Punkte für einige Fälle könnte es für die Homographie Stufe schwierig werden, alle Punkte richtig zuzuordnen. Durch die synthetischen Daten ist bekannt, dass sechzig Punkte pro Cluster gefunden werden sollten. Der Schnittpunkt der Wahrscheinlichkeitsverteilung in Abbildung 5.6 sollte genau bei sechzig liegen. Der Schnittpunkt ist bei der betrachteten Verteilung stark verschoben, umso größer die Verschiebung, desto mehr falsche Werte wurden zugeordnet. An den Ergebnissen ist zu sehen, dass das

Vorclustering die grundlegende Clusterstruktur findet, aber es in bestimmten Fällen zu starken Abweichungen von den zu erkennenden Daten gibt. Wie ausreichend die Resultate des Vorclusterings sind, wird im nächsten Abschnitt untersucht.

## 5.2 Homographie als Information für das Clustering

Wird die Homographie Optimierung wie in Kapitel 3 Abschnitt 3.2 auf das Vorclustering in Abschnitt 5.1 angewendet, ergibt sich für den trivialen Fall auseinanderliegender Objekte, wie in Abbildung 5.5, immer ein optimales Ergebnis, da das Vorclustering schon optimal gewesen ist. Interessant ist wieder der Problemfall von sich überschneidenden Daten. Das Ergebnis der Verbesserung des Clusterings durch die Information der Homographie hängt stark von der Qualität des Vorclusterings und Homographie Berechnung ab. Umso mehr Punkte im Vorclustering falsch erkannt worden sind, desto schlechter war das Ergebnis der Verbesserung, da die Homographie sensibel gegenüber falsch zugeordneten Werten ist. Diese Schlussfolgerung ergab sich durch Vergleich der Ergebnisse des entwickelten Clustering Verfahrens zu dem durch die synthetischen Daten bekannten Clustering. Auf dieser Basis konnten die Falsch zugeordneten Werte unter verschiedenen Aspekten analysiert werden. In den folgenden Abschnitten erfolgt eine Analyse und Diskussion der Cluster Verbesserung durch Homographie Information. Dabei wird untersucht, wie sich der Abbildungsfehler aller Werte der jeweiligen Cluster vor und nach der Verbesserung verhalten und wie der Abbildungsfehler der falsch erkannten Werte aussieht. Aus diesen Beobachtungen kann beurteilt werden, wie gut das Verfahren die zu erkennenden Werte selektiert und richtig zuordnet.

### 5.2.1 Ausreißer Erkennung

In diesem Abschnitt wird untersucht, wie sich die Auswahl der Ausreißer in den Abbildungsfehlern auf das Clustering auswirkt. Es wird deutlich, wie sensibel die Homographie Schätzung gegenüber falschen Werten ist, die für die Berechnung herangezogen wurden. Die folgenden Abbildungen konzentrieren sich auf die Bewegung zwei aus Abbildung 5.2. An diesem Beispiel werden die Auswirkungen eines zu hoch oder zu niedrig gewählten MAD Score untersucht. Außerdem welche Konsequenzen sich ergeben, wenn mehrere Iterationen bei einer falschen Wahl des MAD Scores oder schwachen Homographien durchgeführt werden. Die Ergebnisse dieses Abschnitts dienen als Fundament für die Wahl des MAD Scores für das Multi-Frame Clustering. Alle falsch zugeordneten Werte durch die Verbesserung mit Homographie Information werden in magenta dargestellt, die durch das Vorclustering falsch zugeordneten Werte in grün.

Abbildung 5.7 zeigt die Grenzwerte der Abbildungsfehler der beiden Cluster, die sich aus den gewählten MAD Scores ergeben. Die Grenzwerte sind durch senkrechte Balken gekennzeichnet, dabei bezieht sich der rote Balken auf den Grenzwert des roten Clusters und der blaue Balken auf den Grenzwert des blauen Clusters. Alle Werte unter dem eingezeichneten Grenzwert werden für eine Homographie Berechnung herangezogen, auf deren Basis ein Abbildungsfehler für die aussortierten Daten berechnet wird. Alle Werte, die über dem Grenzwert liegen, werden aussortiert. Die Werte unter dem Grenzwert werden behalten und aus

ihnen eine Homographie für jeden Punkt bezüglich jeden Clusters berechnet. Diese Homographie der optimierten Daten ist die Ausgangsbasis für den Abbildungsfehler der aussortierten Daten. Aufgrund der sich so ergebenden Fehlerwerte findet eine Neuordnung der aussortierten Werte in das Cluster statt, welches den geringsten Abbildungsfehler erzeugt. Die in Abbildung 5.7 dargestellten Werte und Grenzwerte sind die Basis für die folgenden Untersuchungen der Kombinationen aus Iteration und MAD Score und deren Auswirkung auf das Clustering.

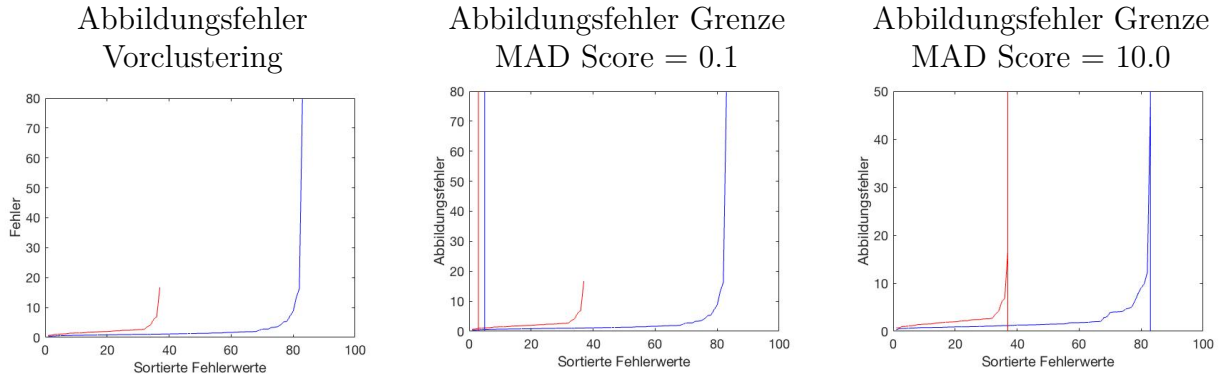


Abbildung 5.7: Grenzwerte des Vorclustering und der MAD Scores 0,1 und 10,0.

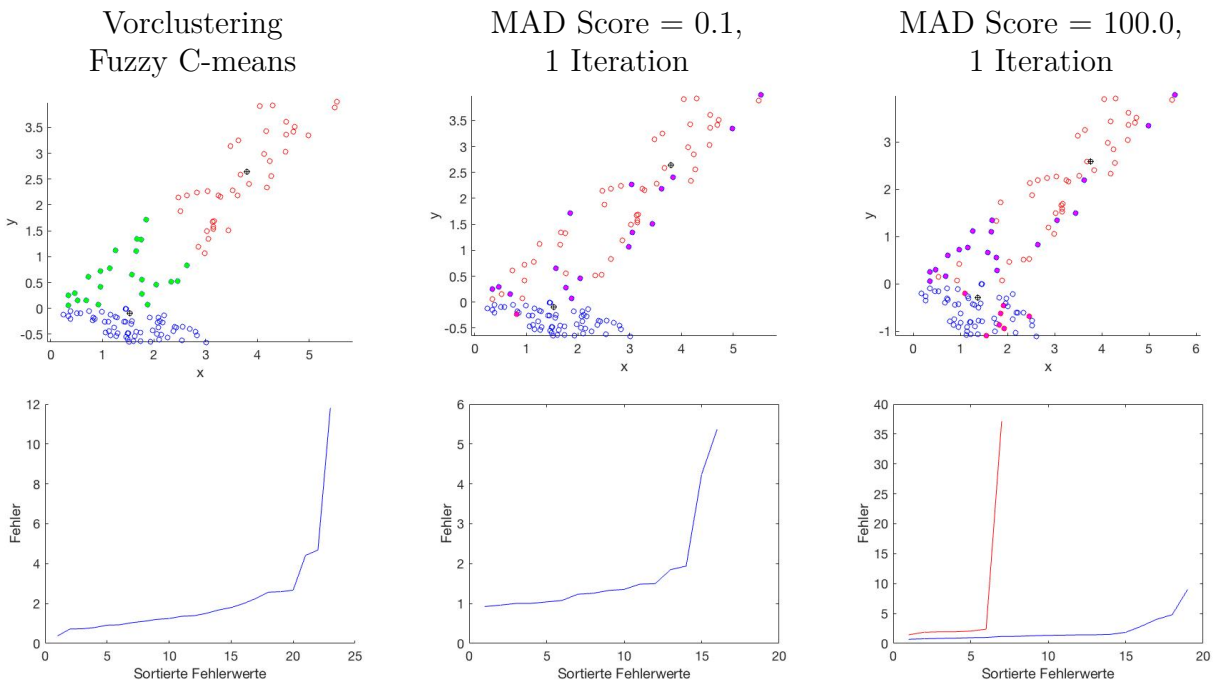


Abbildung 5.8: Verhalten bei zu niedrigem MAD Score, hier 0,1. Plot der jeweils falsch erkannten Werte, bekannt durch die synthetischen Daten. Clustering Ergebnisse und Sortierte Abbildungsfehler. Ein Framepaar.

Abbildung 5.8 zeigt, was bei einem sehr niedrigen und sehr hohen MAD Score, beziehungsweise Grenzwert für den Abbildungsfehler passiert. Dabei wurde in dieser Abbildung nur einmal iteriert. Ein sehr niedriger MAD Score von 0,1 zeigt hier ein relativ gutes Ergebnis, da die Abbildungsfehler unter dem Grenzwert relativ gering sind. Das heißt, sie folgen der Homographie, berechnet aus den Vorclustern. Dennoch scheinen die ausgewählten Werte mit niedrigem Abbildungsfehler dennoch keine solide Basis für die Schätzung der Homographie zu sein, da immer noch einige Werte falsch erkannt wurden. Jedoch ist die Homographie Schätzung wegen der falschen Zuordnungen aus dem Vorclustering fehlerhaft und der Abbildungsfehler nur bezüglich dieser falschen Homographie niedrig. Ist die Homographie falsch, dann wird durch die enge Auswahl nur Punkte gewählt, die der falschen Homographie entsprechen. Richtige Punkte werden eventuell falsch zugeordnet und falsch zugeordnete Werte nicht aussortiert. Die Frage ist, in welchem Maß die Homographie falsch ist und wie eng der Grenzwert gesetzt werden muss, um falsch zugeordnete Werte zu finden. Wenn davon ausgegangen wird, dass die Homographie richtig ist, können die Werte sehr eng ausgewählt werden, da die Fehlerwerte gegen Null die richtigen Werte repräsentiert. Ist die Homographie falsch, ist der Fall nicht so einfach, der Abbildungsfehler ist in diesem Fall nicht so aussagekräftig und kann keine konkrete Aussage über die Clusterzugehörigkeit treffen.

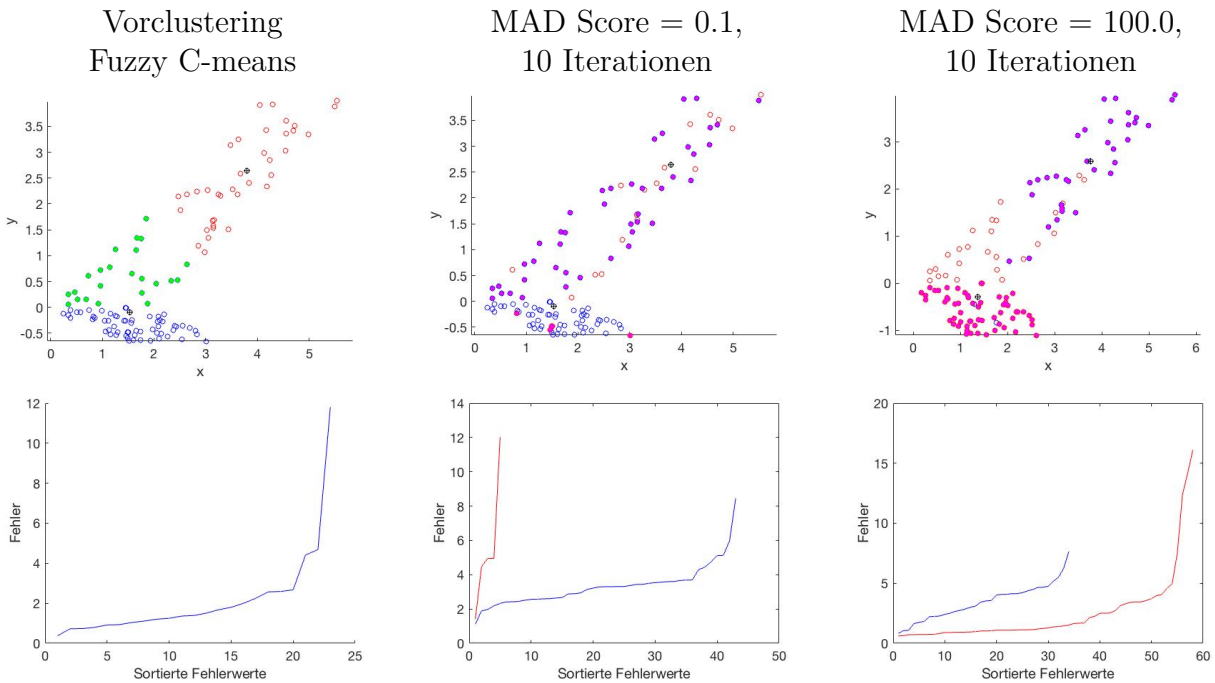


Abbildung 5.9: Verhalten bei zu niedrigem MAD Score, hier 0,1. Plot der jeweils falsch erkannten Werte, bekannt durch die synthetischen Daten. Clustering Ergebnisse und Sortierte Abbildungsfehler der falsch zugeordneten. Ein Framepaar.

Werden die Werte großzügig ausgewählt, können auch jetzt richtige Werte aber auch falsche Werte an die neue Basis für die Homographie Schätzung bilden. In Abbildung 5.8 weisen die Clusterings mit hohem und niedrigem MAD Score ein ähnliches Ergebnis auf, es werden

ähnlich viele falsche Werte durch beide Varianten erkannt, nur anders verteilt. Diese Tatsache weist darauf hin, dass das Vorclustering zu fehlerhaft für die Homographie Schätzung ist und der Abbildungsfehler nicht als Entscheidungskriterium für die Einteilung in die Cluster ausreicht. Die Homographie Berechnung scheint anfällig für falsche Werte im Vorclustering zu sein.

In Abbildung 5.9 zeigt die Ergebnisse der Verbesserung mit Homographie Information in der gleichen Modifikation wie in Abbildung 5.8, nur dass hier insgesamt zehn Iterationen für die Ausreißer Detektion durchgeführt wurden. Wenn der Abbildungsfehler aus oben genannten Gründen keine Aussage über die richtige Zuordnung der Punkte in ein Cluster trifft, wird das Ergebnis des Clusterings mit mehreren Iterationen eher schlechter als besser. Insgesamt liefern mehrere Iterationen nur eine Verbesserung, wenn durch vorherige Iterationen tatsächlich eine Verbesserung in der Clustereinteilung stattgefunden hat, also mit robuster Homographie. Sonst wird der Fehler der Einteilung immer größer. Nur unter glücklichem Zufall kann eine Verbesserung stattfinden. Da reproduzierbare Ergebnisse wünschenswert sind, stellt sich die Frage, wie der MAD Score mit der aktuellen Methode am sinnvollsten ist.

Falsch erkannte Werte, MAD Score = 0.1											
	V	i = 1	i = 2	i = 3	i = 4	i = 5	i = 6	i = 7	i = 8	i = 9	i = 10
B1	11	0	0	0	0	0	0	0	0	0	0
B2	23	17	41	61	51	31	43	28	25	28	48
B3	18	21	53	58	78	75	78	91	75	74	73

Tabelle 5.1: i steht für Iterationen der Cluster Verbesserung durch Homographie Information. B1-B3 steht für die Bewegungen aus den Abbildungen 5.1 bis 5.3. V steht für das Vorclustering.

Falsch erkannte Werte, MAD Score = 100.0											
	V	i = 1	i = 2	i = 3	i = 4	i = 5	i = 6	i = 7	i = 8	i = 9	i = 10
B1	11	1	0	0	0	0	0	0	0	0	0
B2	23	26	36	70	72	48	63	58	21	25	32
B3	18	22	14	11	13	16	17	22	24	21	18

Tabelle 5.2: i steht für Iterationen der Cluster Verbesserung durch Homographie Information. B1-B3 steht für die Bewegungen aus den Abbildungen 5.1 bis 5.3. V steht für das Vorclustering.

Die Darstellung der Ergebnisse der verschiedenen Bewegungen in den Tabellen 5.1 und 5.2 verdeutlicht das Schwanken der falsch zugeordneten Werte und was bei einer robusten Homographie Schätzung passiert.

### 5.2.2 Modifikation für das Multi-Frame Clustering

Für das Multi-Frame Clustering wurde sich auf Grund der Ergebnisse aus 5.2 dafür entschieden, dass nur eine Iteration für die Homographie Verbesserung durchgeführt wird und als Grenzwert der Median der MAD Scores aller Punkte verwendet wird. Die Wahl dieses MAD Score verhindert, dass zu wenige oder zu viele falsche Punkte in die neue Schätzung der Homographie und deshalb in die Entscheidung für eine neue Cluster Einteilung einbezogen werden. Wenn die Homographie Schätzung robuster wäre, wäre ein iteratives Verfahren, wie es in der Konzeption angedacht war, zu empfehlen. Besonders bei den Bewegungen, die in der Projektion zusätzlich einer Tiefenverzerrung unterlagen, lieferte das Vorclustering schlechtere Ergebnisse. Bewegung 5.2 und 5.3 sind die untersuchten Bewegungen, die Tiefenverzerrung unterliegen, da sie sich aus der xy-Projektionsebene auf die Kamera zu bewegen. Die Bewegung aus Abbildung 5.1 findet nur in der Projektionsebene statt. In diesem Fall ist die Homographie auch einfacher bestimmbar als bei den anderen Fällen und damit weniger Fehleranfällig. Alle falsch zugeordneten Werte durch die Verbesserung mit Homographie Information werden hier wieder Magenta dargestellt.

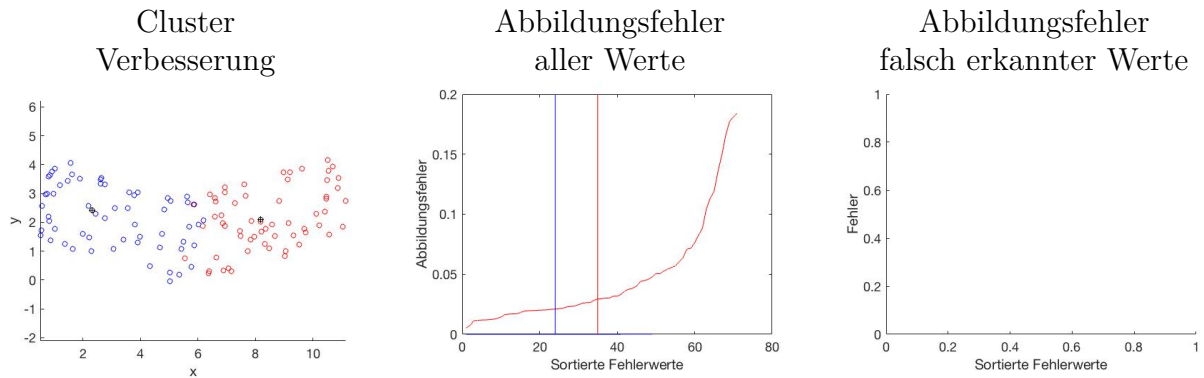


Abbildung 5.10: Verbesserung durch Homographie Information angewandt auf Bewegung 1 aus Abbildung 5.1.

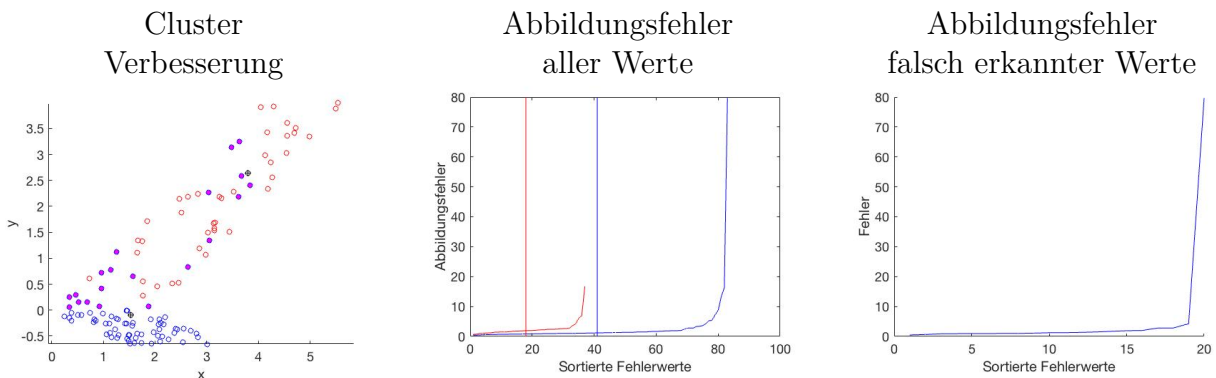


Abbildung 5.11: Verbesserung durch Homographie Information angewandt auf Bewegung 2 aus Abbildung 5.2.

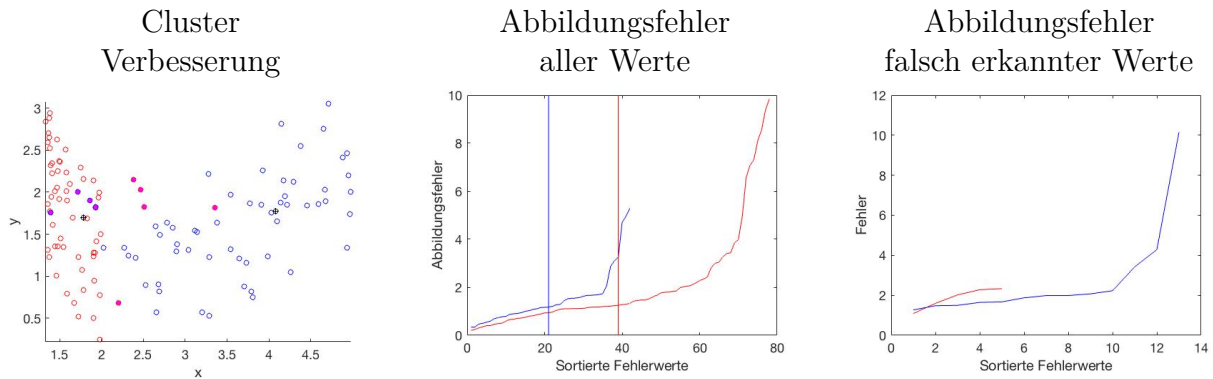


Abbildung 5.12: Verbesserung durch Homographie Information angewandt auf Bewegung 3 aus Abbildung 5.3.

Die Ergebnisse für die Cluster Verbesserung durch Homographie Information wurde in Abbildung 5.10 bis 5.12 exemplarisch für alle Bewegungen für ein Framepaar mit der gewählten Modifikation für das Multi-Frame Clustering durchgeführt. Es ist zu sehen, dass die Abbildungsfehler der falsch zugeteilten Werte etwas geringer sind als in den absichtlich falschen Modifikationen in Abschnitt 5.2.1 und gleichmäßiger verteilt. Dennoch gibt es einige Werte, die durch die gewählte Vorgehensweise nicht gefunden werden, was wieder auf die fehlende Robustheit der Homographie zurückzuführen ist. Durch das Aussortieren mit dem Mittlersten MAD Score ist das Ergebnis unter dem Aspekt, dass die gewählte Homographie Schätzung nicht sehr robust ist, dennoch akzeptabel. Das Ergebnis des Multi-Frame Clustering unter dieser Modifikation wird im folgenden Abschnitt präsentiert. Im Kapitel 6 wird darauf eingegangen, wie die Homographie Schätzung robuster gemacht werden kann und sich so das Clustering der Framepaare verbessert. Das Clustering Ergebnis wird damit ebenfalls optimiert, da der aus dem Clustering der Framepaare resultierende Abbildungsfehler die Grundlage für das Multi-Frame Clustering ist.

## 5.3 Ergebnisse des Multi-Frame Clusterings

Das Multi-Frame Clustering wurde wie in Kapitel 3 Abschnitt 3.3 umgesetzt. Der einzige Unterschied ist, dass aufgrund der Ergebnisse aus Abschnitt 5.2 nur eine Iteration durchgeführt wird. Die Kamera guckt wieder in jedem Fall ist die  $xy$ -Ebene die Projektionsebene. Die Bewegungen 5.2 und 5.3 werden also von oben betrachtet. Die Ergebnisse des Multi-Frame Clusterings der verschiedenen Bewegungen sind in den Abbildungen 5.13 bis 5.15 zu sehen, die jeweils in mehrere Abbildungen unterteilt sind. Es wurden exemplarisch jeweils die Ergebnisse der Analyse von drei Framepaaren der Bewegungen eins bis drei durchgeführt. Die analysierten Framepaare befinden sich über der horizontalen Linie. Es wird von rechts nach links jeweils das Cluster Ergebnis, die initialen Abbildungsfehler aller Werte der jeweiligen Cluster und anschließend die Fehler aller Werte der jeweiligen Cluster nach der Verbesserung dargestellt. Die senkrechten Balken markieren in den initialen Abbildungsfehlern wieder die Grenze, ab der Werte aussortiert worden sind und an der Verbesserung des Clusterings teilgenommen haben. Wenn die Fehlerlinien der beiden Cluster kaum noch sichtbar sind, dann liegen sie übereinander und sind nahezu Null. Der Abbildungsfehler aller Werte für die jeweiligen Cluster sind in diesem Fall gegenüber der aus den Ausgangsdaten berechneten Homographie optimal. Unter der Linie ist das Ergebnis der Analyse des Multi-Frame Clusterings zu sehen. In der Ergebnisdarstellung ist von links nach rechts das Ergebnis Clustering, die Mitglieder pro Iteration und schließlich die falsch erkannten Werte pro Framepaar zu sehen. Die grüne Linie bezieht sich auf die falsch erkannten Werte durch das Vorclustering und die Magenta Linie auf die falsch erkannten Werte nach der Verbesserung durch das Nutzen der Homographie Information. Diesem Prinzip folgen alle Abbildungen, die das Ergebnis des Multi-Frame Clusterings präsentieren.

Die in den vorangegangenen Abschnitten dieses Kapitels beobachteten und diskutierten Probleme gelten hier auch. Es wurde für die Bewegung 5.1 ein perfektes Ergebnis erzielt, so wie das Verfahren gedacht gewesen ist. Die anderen Bewegungen folgen einer komplexeren Homographie und sind aufgrund der fehlerbehafteten Vorcluster schwerer zu erkennen gewesen. Gerade, weil die Homographie Schätzung sehr anfällig für falsche Werte in der Schätzung ist. Positiv ist dennoch, dass die Verbesserung durch Homographie Information insgesamt weniger falsch zugeordnete Punkte erzeugt als das Vorclustering. Das ist an der magenta Linie unter den Ergebnissen im Vergleich zur grünen Linie für das Vorclustering zu erkennen. Die falsch erkannten Werte sind mehr gestreut, als die falsch erkannten Werte durch das Vorclustering, das mit Bildkoordinaten Abständen arbeitet. Das liegt daran, dass die Homographie nicht korrekt und das Aussortieren der Daten nicht repräsentativ und damit beliebig ist. Insgesamt ist aber das Potenzial dieser Vorgehensweise zu erkennen, wenn die Homographie Schätzung robuster gegenüber falsch zugeordneten Werten ist. Das Vorclustering ordnet in komplexeren Fällen zwar falsche Werte zu, findet aber die richtige Tendenz für das Clustering. Der Ansatz alle Informationen aus allen Framepaaren zu nutzen scheint ein mit das beste Ergebnis für die Clusterzuordnung zu liefern.



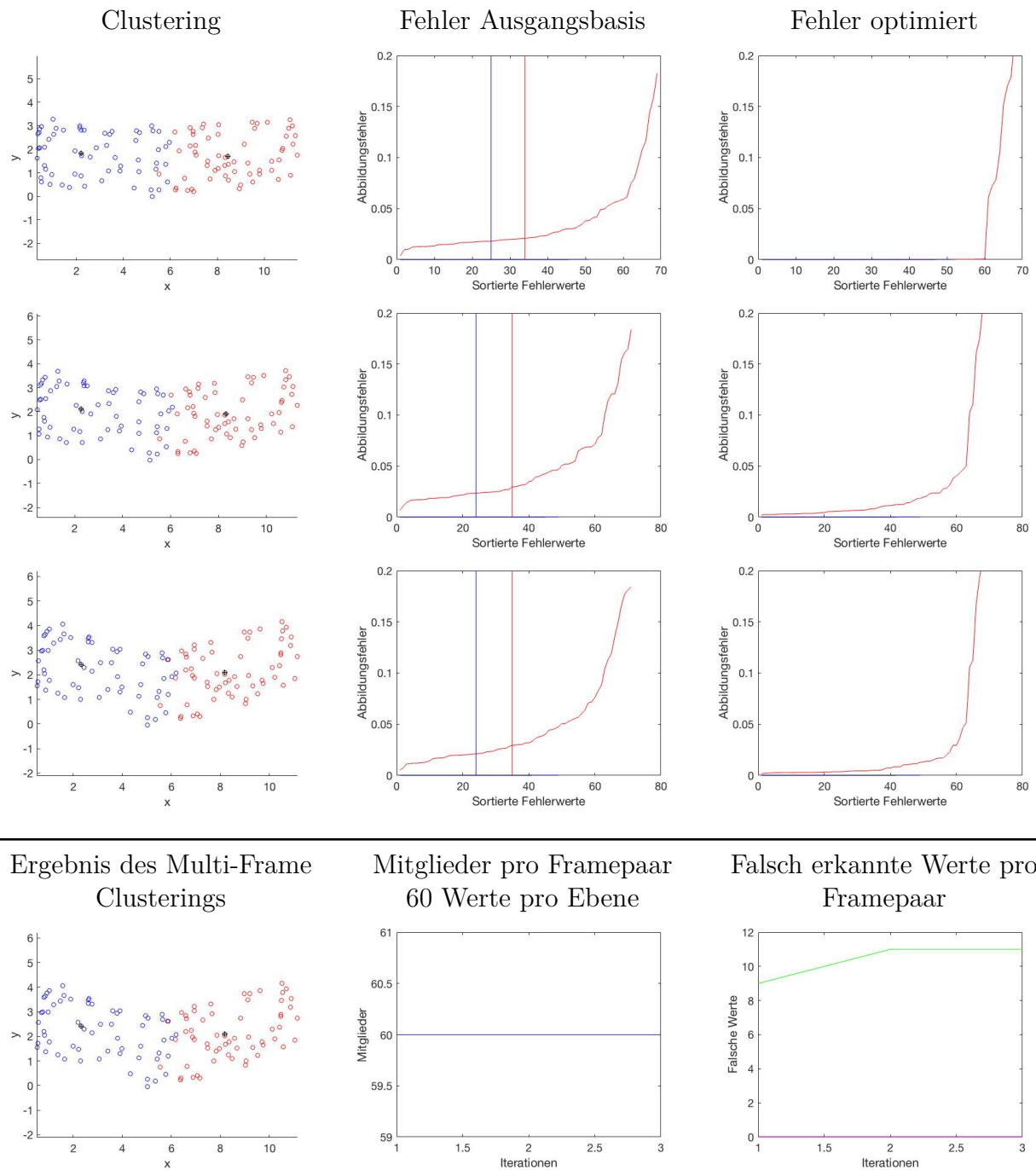


Abbildung 5.13: Bewegung 1: Multi-Frame Clustering Ergebnisse der einzelnen Framepaare und letztendliches Ergebnis nach der Auswertung unter der Linie. 60 Punkte pro Ebene, MAD Schwellwert 1.5.

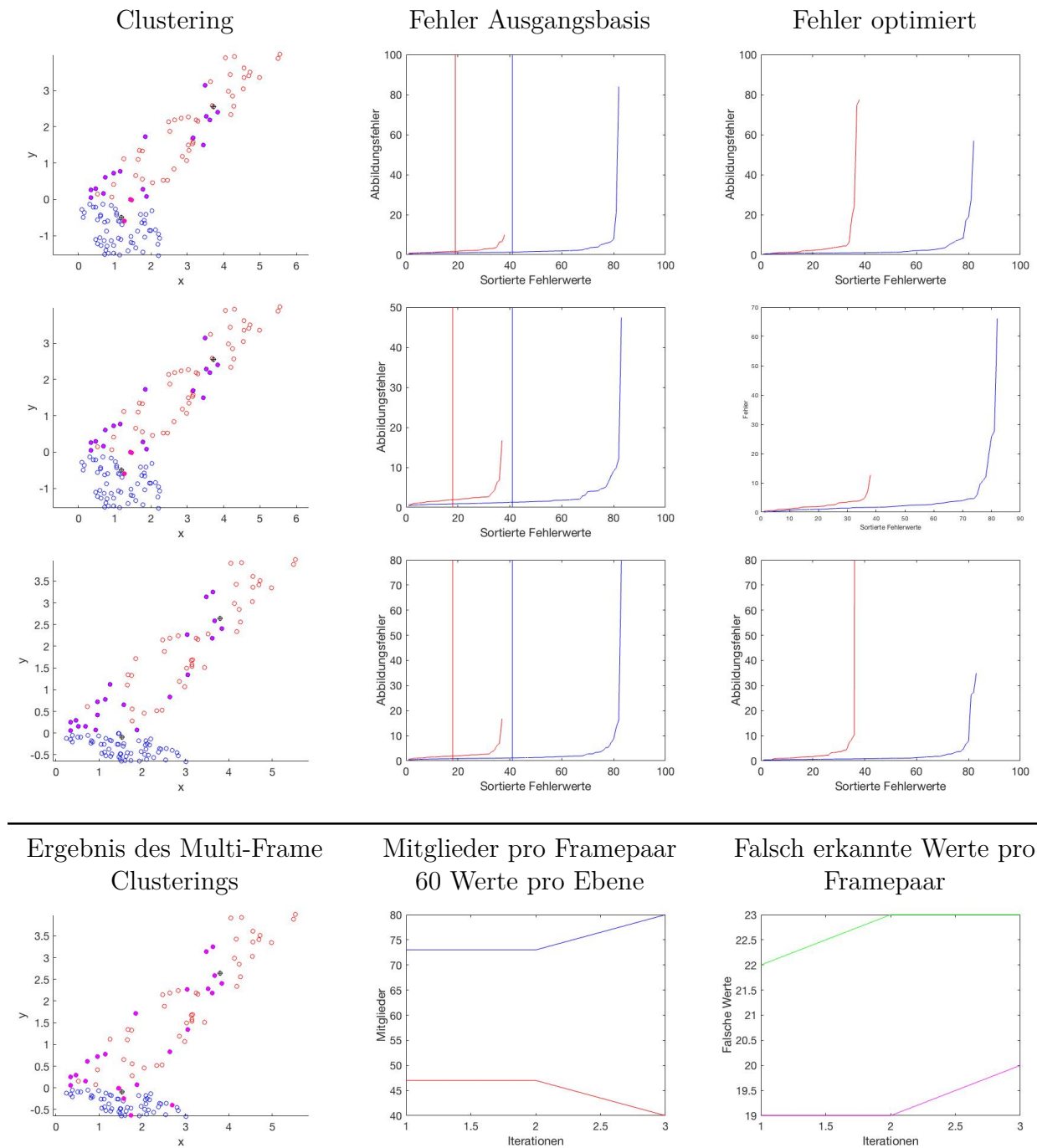


Abbildung 5.14: Bewegung 2: Multi-Frame Clustering Ergebnisse der einzelnen Framepaare und letztendliches Ergebnis nach der Auswertung unter der Linie. 60 Punkte pro Ebene, MAD Schwellwert 1.0.

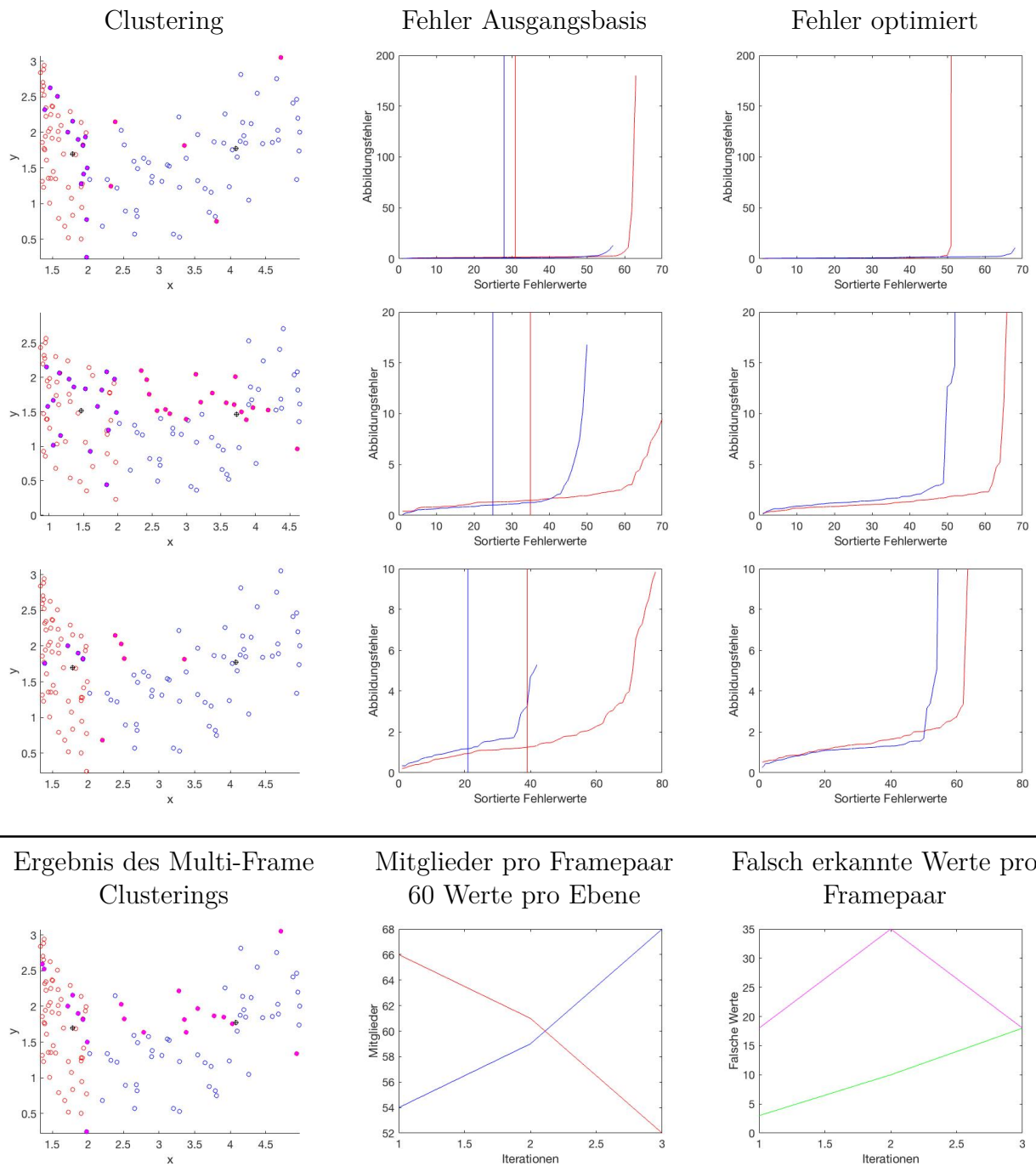


Abbildung 5.15: Bewegung 3: Multi-Frame Clustering Ergebnisse der einzelnen Framepaare und letztendliches Ergebnis nach der Auswertung unter der Linie. 60 Punkte pro Ebene, MAD Schwellwert 2.0.



# 6 Zusammenfassung und Ausblick

## 6.1 Zusammenfassung

Ziel dieser Arbeit war es, ein Clustering Verfahren zu entwickeln, dass artikulierte Objekte in einer Bildsequenz erkennt. Dazu wurde ein Framework entwickelt, dass Testdaten generiert, um die Ergebnisse analysierbar zu machen. Es wurde eine Recherche zu verschiedenen Verfahren und Modellen vorgenommen und ein zweistufiges Clustering Verfahren ausgewählt. Ein auf Wahrscheinlichkeiten basierendes Vorclustering, dass nach dem euklidischen Abstand der Bildkoordinaten optimiert, bildet die Basis für die zweite Stufe der des Verfahrens. Die zweite Stufe nutzt die Information der Homographie, die aus Punktkorrespondenzen eines Bildpaars ein Bewegungsmodell erstellt. Die Abbildungsfehler aller Punkte wurden untersucht und über Ausreißer Detektion wurde das Bewegungsmodell optimiert und auf dieser Grundlage hat sich das Clustering verbessert.

Es hat sich ergeben, dass das entworfene Verfahren anfällig gegenüber zu einem Cluster falsch zugeordneter Punkte ist und daher eine mehrfache Iteration zur Ausreißer Erkennung nur Sinn , wenn eine robustere Homographie Schätzung vorliegt. Für Objekte in der Projektionsebene der Kamera wurden die Ebenen korrekt zugeordnet. Für Objekte, die einer zusätzlichen Tiefenverzerrung unterlagen, ergaben sich Schwierigkeiten beim entwickelten Clustering Verfahren. Wird die Homographie robuster, hat das Verfahren Potenzial artikulierte Objekte in einer Bildsequenz korrekt zu erkennen.

## 6.2 Ausblick

Um das Problem der Homographie Robustheit zu lösen, könnte in Zukunft eine anderer Vorgehensweise gewählt werden. Zum Beispiel der RANSAC (*Random sample consensus*) ist ein bekannter Algorithmus, der das Problem der Homographie Robustheit angeht. Es werden zufällig Daten aus dem übergebenen Datensatz ausgewählt und mit ihnen ein Bewegungsmodell geschätzt. Anschließend werden alle Punkte gesammelt, die diesem Modell entsprechen. Das zufällig geschätzte Bewegungsmodell mit den meisten Punkten gewinnt. Durch dieses Vorgehen sollte die Homographie Schätzung wesentlich besser werden, da der RANSAC Algorithmus robuster gegenüber Ausreißern ist und die Homographie immer wieder überprüft. Derzeit werden alle Punkte aus dem Vorclustering direkt zur Homographie Berechnung genutzt [14].

Wenn das Verfahren mit den synthetischen Daten funktioniert, könnte Rauschen auf die Daten angewandt werden und die Robustheit überprüft werden. Wenn das Verfahren dahingehend funktioniert, könnte die Untersuchung auf andere Starrkörper und dreidimensionale

Objekte ausgeweitet werden. Wenn das Verfahren in allen synthetisch erzeugten Fällen funktioniert, kann das Verfahren mit echten Bilddaten und schließlich Videodaten getestet und hingehend der auftauchenden Probleme verbessert werden. Das Ergebnis ist dann eine zuverlässige Erkennung von artikulierten Objekten in einer Videosequenz, die in verschiedenen Bereichen der Computer Vision Anwendung finden könnte, wie zum Beispiel in der Roboter Navigation oder Augmented Reality [1].

## **7 Anhang**

## Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Abschlussarbeit selbstständig und nur unter Verwendung der von mir angegebenen Quellen und Hilfsmittel verfasst zu haben. Sowohl inhaltlich als auch wörtlich entnommene Inhalte wurden als solche kenntlich gemacht. Die Arbeit hat in dieser oder vergleichbarer Form noch keinem anderem Prüfungsgremium vorgelegen.

Datum: \_\_\_\_\_ Unterschrift: \_\_\_\_\_





## Danksagung

An dieser Stelle möchte insbesondere Frau Dr. Anna Hilsmann für das Thema *Analyse von Bewegungsfeldern dreidimensionaler Starrkörper durch ein Clustering Verfahren* danken, für die Hilfe bei allen Fragestellungen und der wertvollen Kritik beim Schreiben der Bachelorarbeit. Ich habe sehr viel durch das Schreiben dieser Arbeit gelernt und bin sehr dankbar für die tatkräftige Unterstützung und gute Kommunikation. Weiterhin möchte ich Herrn Prof. Dr. Klaus Jung danken, der immer ein offenes Ohr für mich hatte und mir beigestanden hat.

Zu besonderem Dank bin ich auch den wundervollen Menschen verpflichtet, die mir in jeder Phase dieser Arbeit beigestanden und mich unterstützt haben, soweit sie konnten.



# Abbildungsverzeichnis

1.1	Ermittlung der Skelettstruktur einer Lampe [1]	5
2.1	Darstellung von Perspektive [3]	7
2.2	Veranschaulichung von homogenen Koordinaten [4]	8
2.3	isometrisch [6]	9
2.4	ähnlich [6]	9
2.5	affin [6]	9
2.6	projektiv [6]	9
2.7	Vorgehen bei Rotation um einen beliebigen Punkt [7]	10
2.8	Perspektivische Projektion [8]	11
2.9	Parallele Projektion [8]	11
2.10	Projektion von 3D auf 2D (links) Principal Point und Kamerakoordinaten, Bildursprung oft oben links (rechts) [6]	12
2.11	Auswahl des perspektivisch verzerrten Bereichs [3]	13
2.12	Vogelperspektive des ausgewählten Bereichs durch Homographie [3]	13
2.13	Bewegung einer Ebene und Ermittlung ihrer Punktkorrespondenzen. [9]	13
2.14	Allgemeine Verdeutlichung von Clustering [11]	15
2.15	1. Schritt (hier k=3) [12]	17
2.16	2. Schritt [12]	17
2.17	3. Schritt [12]	17
2.18	Ergebnis 4. Schritt [12]	17
2.19	Die Daten gehören pro Iteration fest zu einem Cluster A oder dem anderen Cluster B [11]	17
2.20	Weiches Clustering der Daten, Zugehörigkeit zu verschiedenen Clustern A und B ist möglich [11]	17
3.1	Beispiele artikulierter Bewegungen, Darstellung aller Framebilder. Die dargestellten Punkte werden pro Ebene und Frame in unterschiedlichen Farben dargestellt.	21
3.2	Prinzip der Verbesserung des Clusterings durch Homographie zusammengefasst.	22
3.3	Optimaler Fall des Vorclustering.	23
3.4	Prinzip der Homographieschätzung für das Multi-Frame Clustering.	25
4.1	Visualisierung einer einfachen Translation zweier Ebenen im dreidimensionalen Raum (links) und das Clustering ihrer Positionen aus einem Bildpaar (rechts).	28

5.1	Bewegung 1 . . . . .	31
5.2	Bewegung 2 . . . . .	31
5.3	Bewegung 3 . . . . .	31
5.4	Verdeutlichung der Zielstellung, hier anhand des Problemfalles von sich überschneidenden Daten. Clustering der Bewegung 1 aus Abbildung 5.1. . . . .	32
5.5	Fuzzy C-means Algorithmus angewandt auf einen Datensatz von sich parallel aneinander vorbei bewegender Körper, erzeugt durch das selbst entwickelte MATLAB-Framework. . . . .	33
5.6	Fuzzy C-means Algorithmus angewandt auf den Datensatz aus Abbildung 5.2, erzeugt durch das selbst entwickelte MATLAB-Framework. . . . .	33
5.7	Grenzwerte des Vorclustering und der MAD Scores 0,1 und 10,0. . . . .	35
5.8	Verhalten bei zu niedrigem MAD Score, hier 0,1. Plot der jeweils falsch erkannten Werte, bekannt durch die synthetischen Daten. Clustering Ergebnisse und Sortierte Abbildungsfehler. Ein Framepaar. . . . .	35
5.9	Verhalten bei zu niedrigem MAD Score, hier 0,1. Plot der jeweils falsch erkannten Werte, bekannt durch die synthetischen Daten. Clustering Ergebnisse und Sortierte Abbildungsfehler der falsch zugeordneten. Ein Framepaar. . . .	36
5.10	Verbesserung durch Homographie Information angewandt auf Bewegung 1 aus Abbildung 5.1. . . . .	38
5.11	Verbesserung durch Homographie Information angewandt auf Bewegung 2 aus Abbildung 5.2. . . . .	38
5.12	Verbesserung durch Homographie Information angewandt auf Bewegung 3 aus Abbildung 5.3. . . . .	39
5.13	Bewegung 1: Multi-Frame Clustering Ergebnisse der einzelnen Framepaare und letztendliches Ergebnis nach der Auswertung unter der Linie. 60 Punkte pro Ebene, MAD Schwellwert 1.5. . . . .	41
5.14	Bewegung 2: Multi-Frame Clustering Ergebnisse der einzelnen Framepaare und letztendliches Ergebnis nach der Auswertung unter der Linie. 60 Punkte pro Ebene, MAD Schwellwert 1.0. . . . .	42
5.15	Bewegung 3: Multi-Frame Clustering Ergebnisse der einzelnen Framepaare und letztendliches Ergebnis nach der Auswertung unter der Linie. 60 Punkte pro Ebene, MAD Schwellwert 2.0. . . . .	43

# Literaturverzeichnis

- [1] KAAAN YÜCER, Alexander Sorkine-Hornung Olga Sorkine-Hornung Oliver W. Oliver Wang ; ETH ZURICH, Disney R. (Hrsg.): *Reconstruction of Articulated Objects from a Moving Camera*. 2015
- [2] PRATEEK SINGHAL, N Dinesh Reddy K Madhava K. Aditya Deshpande D. Aditya Deshpande ; INFORMATION TECHNOLOGY, International I. (Hrsg.): *Top Down Approach to Detect Multiple Planes in Images*. 2013
- [3] RICHARD HARTLEY, Andrew Z. ; PRESS, Cambridge U. (Hrsg.): *Multiple View Geometry in Computer Vision*. 2004
- [4] KURTH, Winfried ; GÖTTINGEN, University of (Hrsg.): *Homogene Koordinaten*. [http://www.uni-forst.gwdg.de/~wkurth/cb/html/cg\\_v06a.htm](http://www.uni-forst.gwdg.de/~wkurth/cb/html/cg_v06a.htm). Version: 2016
- [5] PAREIGIS, Bodo: *Analytische und Projektive Geometrie für die Computer-Graphik*. 1990
- [6] FISHER, Bob ; EDINBURGH, The U. (Hrsg.): *3x4 Projection Matrix*. [http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/EPSRC\\_SSAZ/node3.html](http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/EPSRC_SSAZ/node3.html). Version: 1997
- [7] EISERT, Dr. P. ; BERLIN, Humboldt U. (Hrsg.): *Object Transformations*. 2015/2016
- [8] FOLEY, Feiner H. Van Dam D. Van Dam: *Computer Graphics: Principles and Practice*. 1990
- [9] KAISERSLAUTERN, Technische U. ; GABRIELE BLESER, Christiano G. (Hrsg.): *2D projective transformations*. [http://ags.cs.uni-kl.de/fileadmin/inf\\_ags/3dcv-ws11-12/3DCV\\_WS11-12\\_lec04.pdf](http://ags.cs.uni-kl.de/fileadmin/inf_ags/3dcv-ws11-12/3DCV_WS11-12_lec04.pdf). Version: 2016
- [10] PANG-NING TAN, Vipin K. Michael Steinbach S. Michael Steinbach ; MINNESOTA, University of (Hrsg.): *Introduction to Data Mining*. <https://www-users.cs.umn.edu/~kumar/dmbook/ch8.pdf>. Version: 2006
- [11] MATTEUCCI, Matteo: *A Tutorial on Clustering Algorithms*. [http://home.deib.polimi.it/matteucc/Clustering/tutorial\\_html/cmeans.html](http://home.deib.polimi.it/matteucc/Clustering/tutorial_html/cmeans.html). Version: 2016
- [12] WESTON.PACE: *This image is part of an example of the K-means algorithm. This is the first step, where the points and centroids are randomly placed*. [https://en.wikipedia.org/wiki/K-means\\_clustering](https://en.wikipedia.org/wiki/K-means_clustering). Version: 2007

- [13] HILSMANN, Dr. A. ; BERLIN, Humboldt-Universität zu (Hrsg.): *Image-Based Approaches for Photo-Realistic Rendering of Complex Objects*. <http://edoc.hu-berlin.de/dissertationen/hilsmann-anna-2014-03-28/PDF/hilsmann.pdf>. Version: 2013
- [14] DERPANIS, Konstantinos G. ; EECS (Hrsg.): *Overview of the RANSAC Algorithm*. [http://www.cse.yorku.ca/~kosta/CompVis\\_Notes/ransac.pdf](http://www.cse.yorku.ca/~kosta/CompVis_Notes/ransac.pdf). Version: 2010

## **Notizen**





