

Liste der noch zu erledigenden Punkte

abstract schreiben	V
noch nicht ganz fertig	2
fazit einleitung	46



ENTZERRUNG VON KEGELOBERFLÄCHEN AUS EINER EINKAMERAANSICHT BASIEREND AUF PROJEKTIVER GEOMETRIE

BACHELORARBEIT
zur Erlangung des akademischen Grades
BACHELOR OF SCIENCE

Westfälische Wilhelms-Universität Münster
Fachbereich Mathematik und Informatik
Institut für Informatik

Betreuung:
Dimitri Berh

Erstgutachten:
Prof. Dr. Xiaoyi Jiang

Zweitgutachten:
Prof. Dr. Klaus Hinrichs

Eingereicht von:
Lars Haalck

Münster, September 2016

abstract schrei-
ben

Zusammenfassung

Zur Bewegungsanalyse von Kriechtieren insbesondere *Drosophila melanogaster*, wurde ein Untersuchungsaufbau konstruiert, so dass diese in einem Kegelstumpf mit einer Weitwinkelkamera untersucht werden können. In der vorliegenden Arbeit stellen wir zwei Verfahren zur Entzerrung von Kegeloberflächen vor bla bla ??

- Problemstellung und Ziel
- zugrunde liegende Theorie
- Ergebnisse

Inhaltsverzeichnis

1. Einleitung	2
2. Theoretische Grundlagen	4
2.1. Kegel	4
2.2. Lineares Ausgleichsproblem	9
2.3. Kamerakalibrierung	10
2.4. Blob-Detektor	12
2.5. Hough-Transformation	12
2.6. RANSAC	13
2.7. Ellipsen	14
2.7.1. Definition	14
2.7.2. Abstand: Punkt zu Ellipse	17
2.8. Analytical Deformable Templates	20
3. Methodik	21
3.1. Kalibrierungsmuster	21
3.1.1. Aufbau des Kalibrierungsmusters	21
3.1.2. Anzahl der Samples	21
3.2. Intrinsische Kamerakalibrierung	23
3.3. Detektion der charakteristischen Punkte	23
3.4. Ellipsen-Detektion	24
3.4.1. RANSAC	24
3.4.2. Analytical Deformable Templates	26
3.5. Zuordnung der Punkte	28
3.6. Weltkoordinaten bestimmen	29
3.7. Entfaltung	30
3.7.1. Vorwärtsentfaltung	30
3.7.2. Rückwärtsentfaltung	31
4. Implementierung	33
5. Analyse	35
5.1. Vergleich Vorwärtsentfaltung und Rückwärtsentfaltung	35
5.2. Einfluss der intrinsischen Kalibrierung	39
5.3. Einfluss der Rotation der Kamera	40
5.4. Laufzeit der Entfaltung	40
5.5. Evaluierung der Ellipsendetektion mittels Deformable Templates	41
5.6. Evaluierung der Ellipsendetektion mittels RANSAC	42
6. Fazit und Ausblick	46
6.1. Parallelisierung	46
6.2. Verbesserung der Linien-Detektion	46
6.3. Verbesserung der Vorwärtsentfaltung	46
6.4. Verbesserung der Rückwärtsentfaltung	47

6.5. Verbesserung der Ellipsendetektion mittels Deformable Templates	48
A. Inhalt der CD	49
Abbildungsverzeichnis	50

1. Einleitung

noch nicht
ganz fertig

Drosophilae melanogaster, im Volksmund auch Fruchtfliege genannt, und ihre Larven sind schon seit Jahrzehnten ein wichtiges Untersuchungsobjekt der Biologie. Sie zeichnen sich aus durch einen schnellen Generationswechsel, eine hohe Nachkommenszahl, eine einfache Chromosomenstruktur und können darüber hinaus leicht gehalten und gezüchtet werden.

In den vergangenen Jahren wurden verschiedene Versuchsaufbauten zur Untersuchung des Bewegungsmusters von Drosophilae entwickelt. Als problematisch erwies sich, dass das Verhalten der Larven beim Herausnehmen aus ihren Aufzuchtröhrchen, bedingt durch den ausgesetzten Stress, negativ beeinflusst wurde. Es wurde daher ein Aufbau entwickelt, in dem die Larven direkt in ihrem Aufzuchtröhrchen beobachtet werden konnten, wie in Abbildung 1.1(a) schematisch dargestellt. Die Larven befinden sich hierbei in einem Zylinder, der von insgesamt fünf Kameras beobachtet wird, die jeweils an einen Raspberry Pi angeschlossen sind. Die Kameras nehmen nun simultan ein Bild auf, wobei hier eine Synchronisation notwendig ist. Auf Grund der Krümmung des Zylinders, müssen diese nun entsprechend entzerrt werden. Anschließend werden die fünf entzerrten Bilder zu einem Gesamtbild zusammengefügt.

Um ein besseren Verhältnis zwischen der Anzahl der Kameras und des Aufzuchtgeheges zu erreichen, wäre eine Reduktion der Kameraanzahl wünschenswert. Statt eines Zylinders benutzen wir in einem neuen Aufbau einen Kegel, wie in Abbildung 1.1(b) zu sehen, und beobachten das Gehege mit einer einzigen Weitwinkelkamera. Eine Synchronisation, sowie ein Zusammenfügen von Einzelbildern, ist hier also nicht mehr notwendig. Wie auch bei dem bisherigen Aufbau muss hier jedoch eine Entzerrung durchgeführt werden. Die Larven, die sich am oberen Rand des Kegels befinden scheinen sonst größer, als jene, die sich im unteren Bereich aufhalten. Ein relativer Vergleich der Larven wäre somit nicht möglich.

In dieser Arbeit werden wir zwei Verfahren zum Entzerrn solcher Kegeloberflächen vorstellen, sodass die relative Vergleichbarkeit der Larven gewährleistet ist. Dazu werden wir die Kegeloberfläche mit Hilfe einer geeigneten Abbildung auf die Mantelfläche abbilden.

In Kapitel 2 dieser Arbeit werden zunächst die notwendigen theoretischen Grundlagen erarbeitet. Anschließend stellen wir in Kapitel 3 beide Verfahren zur Entzerrung von Kegeloberflächen vor und erläutern ihre Funktionsweise, woraufhin wir in Kapitel 4 auf den implementierten Kalibrierungsassistenten eingehen, der diesen Prozess vereinfachen soll. In Kapitel 5 stellen wir beide Verfahren gegenüber und untersuchen Einflussfaktoren auf die Präzision und Qualität der Entzerrung. Im letzten Kapitel dieser Arbeit ziehen wir ein Fazit und geben einen Ausblick auf mögliche Verbesserungsmaßnahmen für die verwendeten Verfahren.

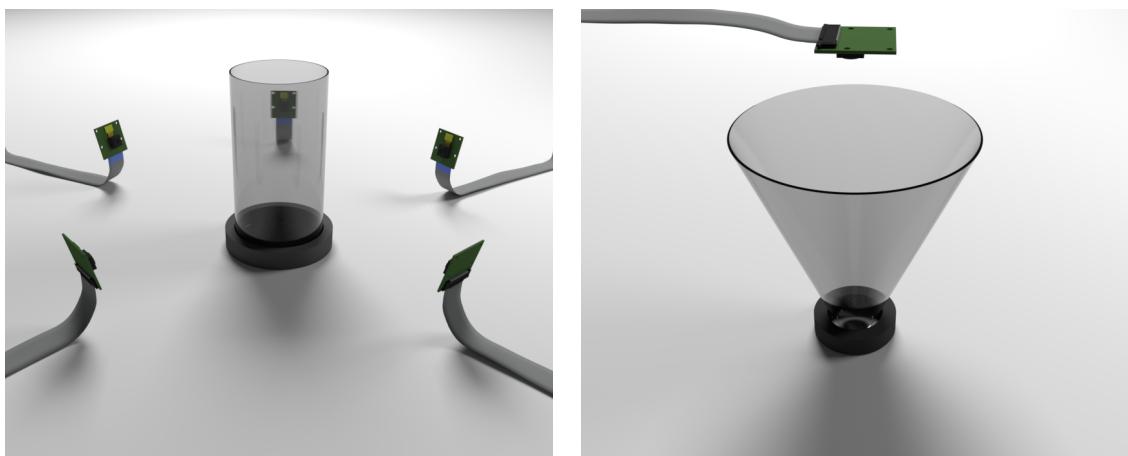


Abbildung 1.1.: beide Ansätze im Vergleich

2. Theoretische Grundlagen

2.1. Kegel

2.1.1 Definition (Kegel)

Ein Kegel ist ein geometrischer Körper, der durch eine beliebige Grundfläche, sowie einen Punkt definiert wird. Ein Kegel mit Kreis (auch Grundkreis genannt) als Grundfläche wird als Kreiskegel bezeichnet. Liegt die Kegelspitze auf einer Geraden durch die Normale der Grundebene, bezeichnen wir den Kegel als gerade.

In der weiteren Arbeit betrachten wir nur gerade Kreiskegel. Ein Kegel mit Spitze $T(0, 0, 0)$, Radius R und Höhe H , dessen Zentrum des Grundkreises in $(0, H, 0)$ liegt, kann parametrisch beschrieben werden als:

$$\begin{aligned}x &= \frac{u}{H} R \cos\theta \\y &= u \\z &= \frac{u}{H} R \sin\theta\end{aligned}\tag{2.1}$$

mit $u \in [0, H]$ und $\theta \in [0, 2\pi]$

S bezeichne hierbei die Seitenhöhe und sei definiert durch das rechtwinklige Dreieck mit den Seitenlängen H , R und S (siehe Abbildung 2.1). Es gilt $S = \sqrt{H^2 + R^2}$.

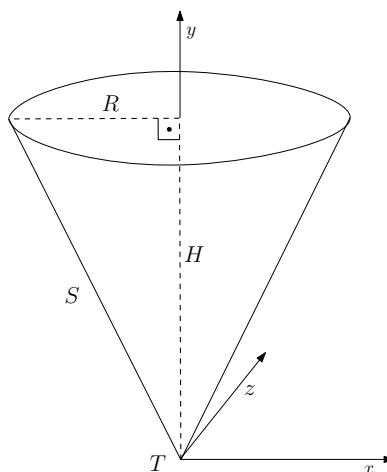


Abbildung 2.1.: Gerader Kreiskegel

2.1.2 Definition (Kegelstumpf und Ergänzungskegel)

Ein Kegelstumpf entsteht als Schnitt eines geraden Kreiskegels mit einer zur Grundfläche parallelen Ebene (siehe Abbildung 2.2). Der Kegel, der definiert ist durch die Schnittfläche und die Spitze des ursprünglichen Kegels bezeichnen wir als Ergänzungskegel. Die Differenz des eigentlichen Kegels und des Ergänzungskegels bezeichnen wir als Kegelstumpf. Die Grundfläche des Kegelstumpfs wird als

Grundkreis, die Schnittfläche als Deckkreis bezeichnet.

H, R, S sind dabei weiterhin die Höhe, Radius und Seitenhöhe des gesamten Kegels. Hinzu kommen h, r, s als Höhe, Radius und Seitenhöhe des Ergänzungskegels. Die Höhe, sowie die Seitenhöhe des Kegelstumpfs werden durch die Differenzen $\Delta S = S - s$, $\Delta H = H - h$ charakterisiert (siehe Abbildung 2.3).

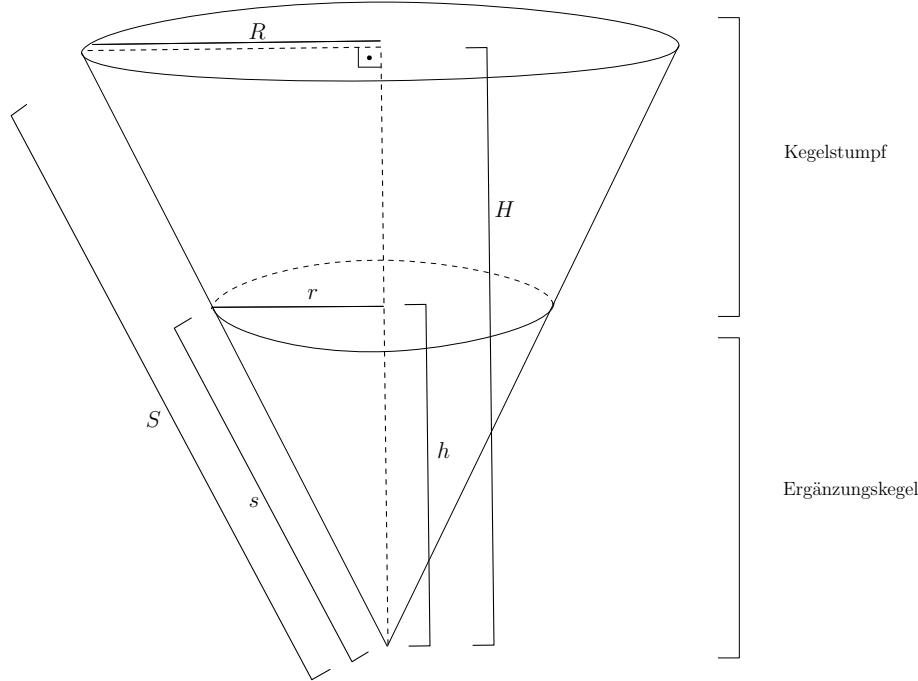


Abbildung 2.2.: Kegelstumpf und Ergänzungskegel

Wir definieren einen Kegelstumpf mit Zentrum $(0, 0, 0)$ des Deckkreises und Zentrum $(0, \Delta H, 0)$ des Grundkreises durch folgende Parametrisierung:

$$\begin{aligned} x &= (r + \frac{u}{\Delta H}(R - r)) \cos\theta \\ y &= u \\ z &= (r + \frac{u}{\Delta H}(R - r)) \sin\theta \end{aligned} \tag{2.2}$$

mit $u \in [0, \Delta H]$ und $\theta \in [0, 2\pi]$. Die parametrische Form eines Kegelstumpfs ist somit ein Verallgemeinerung der Parametrisierung von klassischen Kegeln (siehe Gleichung 2.1), wobei beim klassischen Kegel $r = 0$ gilt. Wie in Abbildung 2.3 zu sehen, wollen wir bei Höhe $u = 0$ einen Radius von r erreichen. Wir führen also eine Skalierung des Intervalls $[0, R]$ auf das Intervall $[r, R]$ durch. Dies erreichen wir durch den Term $(r + \frac{u}{\Delta H}(R - r))$.

Die Mantelfläche des Kegelstumpfs aus Abbildung 2.4 kann dann in Polarkoordinaten parametrisch beschrieben werden als

$$\begin{aligned} x &= -(s + \frac{u}{\Delta H}(S - s)) \sin\phi \\ y &= (s + \frac{u}{\Delta H}(S - s)) \cos\phi \end{aligned} \tag{2.3}$$

mit $u \in [0, \Delta H]$ und $\phi \in [0, \alpha] \subseteq [0, 2\pi]$.

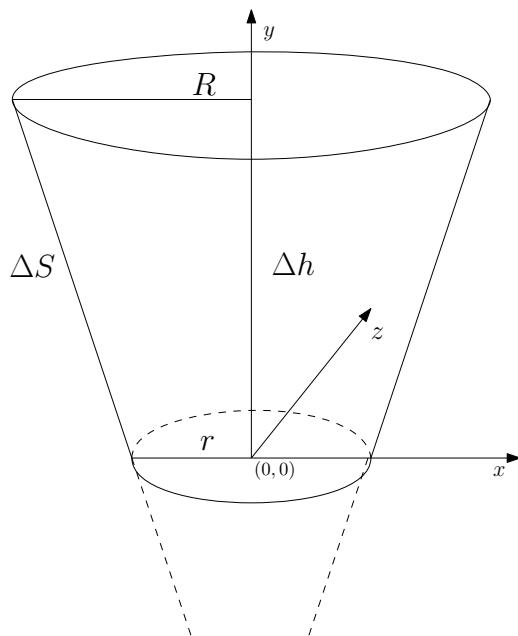


Abbildung 2.3.: Kegelstumpf

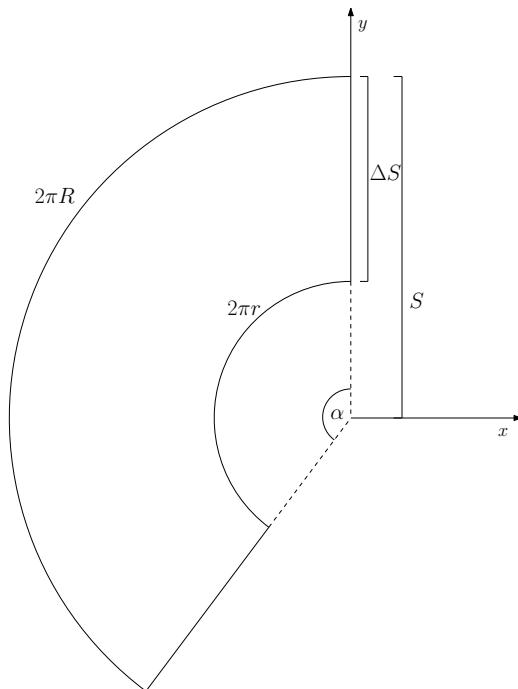


Abbildung 2.4.: Kegelmantelfläche

Die Mantelfläche ergibt sich durch Entrollen des Kegelstumpfs. Da der Umfang des Kreises mit Radius R , $2\pi R$ beträgt, muss, wie in Abbildung 2.4 zu sehen, der äußere Kreisbogen die Bogenlänge $2\pi R$ haben. Analog beträgt die Bogenlänge des inneren Kreisbogens $2\pi r$. Für den Winkel α muss demnach gelten $\alpha S = 2\pi R$, also $\alpha = 2\pi \frac{R}{S}$. Da sich die Seitenhöhe, wie in Abbildung 2.5 zu sehen, linear zur Höhe des Kegelstumpfs

verhält, kann die Seitenhöhe durch die Höhe ausgedrückt werden mit $(s + \frac{u}{\Delta H}(S - s))$. Bei Höhe $u = 0$ ergibt sich somit, wie erwartet die Seitenhöhe s , bei Höhe $u = \Delta H$ die Seitenhöhe S .

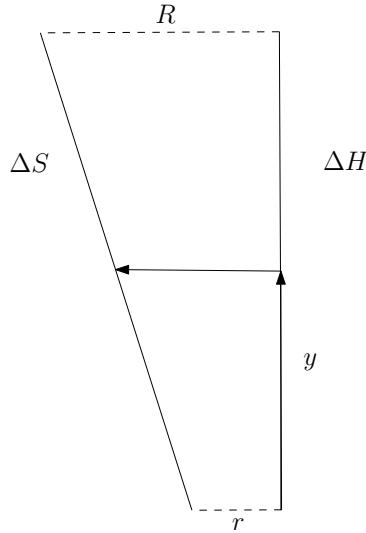


Abbildung 2.5.: Abbildung der Kegelstumpfhöhe auf die Seitenhöhe

Ein Punkt auf der Oberfläche des Kegelstumpfs kann eindeutig einem Punkt auf der Mantelfläche (und umgekehrt) zugeordnet werden. Dazu konstruieren wir folgende Abbildung und ihr Inverses:

Sei dazu ein Punkt $C(x, y, z)$ auf der Oberfläche des Kegelstumpfs gegeben. Gegeben durch die parametrischen Gleichung 2.2, hat C die Form

$$C(x, y, z) = (r + \frac{u}{\Delta H}(R - r)) \cos \theta, u, (r + \frac{u}{\Delta H}(R - r)) \sin \theta$$

für ein $u \in [0, \Delta H]$ und $\theta \in [0, 2\pi]$.

Aus der y -Koordinate lässt sich direkt die Höhe u ablesen und somit analog zur Gleichung 2.3 die zugehörige Seitenhöhe bestimmen. Diese entspricht dem Radius in der parametrischen Gleichung der Mantelfläche. Wir definieren uns hierfür eine Hilfsfunktion

$$\Sigma(y) := s + \frac{y}{\Delta H}(S - s) \quad (2.4)$$

Da $R, r, \Delta H$ und nun auch die Höhe bekannt sind, können wir den Winkel θ im Kegelstumpf ausrechnen. Anschließend muss dieser noch mit $\frac{R}{S}$ multipliziert werden um ihn auf $[0, \alpha]$ zu skalieren (siehe Gleichung 2.3). Wir wollen in der Gleichung 2.2 nach $\cos \phi$, beziehungsweise nach $\sin \phi$ umstellen und anschließend nach ϕ auflösen. Wir teilen also zunächst beide Gleichungen durch $r + \frac{y}{\Delta H}(R - r)$. Zu beachten ist nun, dass wir nicht die Umkehrfunktionen acos oder asin benutzen können, da \sin und \cos in einer Periode $[0, 2\pi]$ nicht injektiv sind. Die Wertebereiche der Umkehrfunktionen sind also eingeschränkt und wir erhalten somit im Allgemeinen nicht den erwünschten Winkel aus $[0, 2\pi]$. Das gleiche Problem zeigt sich bei $\tan \phi = \frac{\sin \phi}{\cos \phi}$ und der Umkehrfunktion atan . Um dieses Problem zu beheben macht man eine Fallunterscheidung und

definiert wie folgt die sogenannte atan2-Funktion

$$\text{atan2}(y, x) = \begin{cases} \text{atan} \frac{y}{x} & \text{für } x > 0 \\ \text{atan} \frac{y}{x} + \frac{\pi}{2} & \text{für } x < 0, y \geq 0 \\ \text{atan} \frac{y}{x} - \frac{\pi}{2} & \text{für } x < 0, y < 0 \\ \text{atan} + \frac{\pi}{2} & \text{für } x = 0, y > 0 \\ \text{atan} - \frac{\pi}{2} & \text{für } x = 0, y < 0 \\ 0 & \text{für } x = 0, y = 0, \end{cases}$$

die den Winkel ϕ im richtigen Quadranten, also in $[0, 2\pi)$ bestimmt.

Aus diesen Erkenntnissen definieren wieder eine Hilfsfunktion

$$\Phi(x, y, z) := \frac{R}{S} \text{atan2} \left(\frac{z}{r + \frac{y}{\Delta h}(R - r)}, \frac{x}{r + \frac{y}{\Delta H}(R - r)} \right), \quad (2.5)$$

die den korrekten Winkel in Polarkoordinaten liefert und auf das Intervall $[0, \alpha)$ skaliert.

Mit den beiden Hilfsfunktionen 2.4 und 2.5, sowie der Gleichung 2.3 ergibt sich insgesamt

$$\begin{aligned} \Psi: [r, R] \times [0, \Delta H] \times [r, R] &\rightarrow [s, S] \times [s, S] \\ \begin{pmatrix} x \\ y \\ z \end{pmatrix} &\mapsto \begin{pmatrix} -\Sigma(y) \sin \Phi(x, y, z) \\ \Sigma(y) \cos \Phi(x, y, z) \end{pmatrix}. \end{aligned} \quad (2.6)$$

Analog lässt die sich Umkehrabbildung konstruieren:

Sei ein Punkt $L(x, y)$ auf der Mantelfläche des Kegelstumpfs gegeben. Aus der parametrischen Form der Mantelfläche (siehe Gleichung 2.3) ergibt sich

$$L(x, y) = \left(-\left(s + \frac{u}{\Delta H} (S - s) \right) \sin \phi, \left(s + \frac{u}{\Delta H} (S - s) \right) \cos \phi \right)$$

für ein passendes $u \in [0, \Delta H]$ und $\phi \in [0, \alpha) \subseteq [0, 2\pi)$.

Da $L(x, y)$ in Polarkoordinaten gegeben ist, lässt sich der Radius durch $\sqrt{x^2 + y^2}$ bestimmen. Wir können den Winkel ϕ mit inverser Skalierung also analog zu oben durch folgende Hilfsfunktion bestimmen:

$$\Theta(x, y) := \frac{S}{R} \text{atan2} \left(-\frac{x}{\sqrt{x^2 + y^2}}, \frac{z}{\sqrt{x^2 + y^2}} \right) \quad (2.7)$$

Die Höhe im Kegel und somit der Radius lässt sich nun als Umkehrabbildung zu 2.4 bestimmen:

$$H(x, y) := \frac{\left(\sqrt{x^2 + y^2} \right) - s}{S - s} \Delta H \quad (2.8)$$

Insgesamt ergibt sich:

$$\Psi^{-1} : [s, S] \times [s, S] \rightarrow [r, R] \times [0, \Delta H] \times [r, R]$$

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} \left(r + \frac{H(x, y)}{\Delta H}(R - r) \right) \cos(\Theta(x, y)) \\ H(x, y) \\ \left(r + \frac{H(x, y)}{\Delta H}(R - r) \right) \sin(\Theta(x, y)) \end{pmatrix} \quad (2.9)$$

2.2. Lineares Ausgleichsproblem

Gegeben seien m Messdaten $(x_1, y_1), \dots, (x_m, y_m)$, sowie eine lineare Funktion

$$\varphi(x) = \sum_{i=1}^n c_i \varphi_i(x)$$

in x mit Skalaren $c_1, \dots, c_n \in \mathbb{R}$, Funktionen $\varphi_1, \dots, \varphi_n$, $n \leq m$ und $n, m \in \mathbb{N}$.

Gesucht ist eine Lösung welche die mittlere Abweichung

$$\Delta_2 = \min_{c_1, \dots, c_n \in \mathbb{R}} \left(\sum_{j=1}^m (y_j - \varphi(x_j))^2 \right) = \min_{c_1, \dots, c_n \in \mathbb{R}} \left(\sum_{j=1}^m \left(\sum_{i=1}^n y_j - (c_i \varphi_i(x_j)) \right)^2 \right)$$

minimiert [Sto07].

Diese Problem wird als lineares Ausgleichsproblem oder als Methode der kleinsten Quadrate bezeichnet und lässt sich wie folgt formulieren:

$$\min_{c \in \mathbb{R}^n} \|y - Ac\|_2,$$

mit

$$\begin{aligned} c &= (c_1, \dots, c_n)^T \in \mathbb{R}^n \\ x &= (x_1, \dots, x_m)^T \in \mathbb{R}^m \\ y &= (y_1, \dots, y_m)^T \in \mathbb{R}^m \\ A &= (a_{ij}) \in \mathbb{R}^{m \times n} \quad \text{mit} \quad a_{ij} = \varphi_i(x_j) \end{aligned}$$

Das Problem lässt sich auch mit Hilfe der äquivalenten sogenannten Normalengleichung

$$(A^T A)c = A^T y$$

lösen, wobei mindestens eine Lösung existiert und bei mehreren Lösungen, die Lösung mit kleinster 2-Norm ausgewählt wird.

Gilt darüber hinaus $\text{rang}(A) = n$ so ist $A^T A$ invertierbar und es gilt

$$c = \underbrace{(A^T A)^{-1} A^T}_{A^+} y. \quad (2.10)$$

Dabei ist c die eindeutige Lösung mit kleinster 2-Norm. A^+ bezeichnet man auch als Pseudoinverse von A . Im Allgemeinen ist die Matrix $A^T A$ nicht invertierbar. Die Pseudoinverse kann somit nicht einfach mit $(A^T A)^{-1} A^T$ berechnet werden. Stattdessen lässt sie sich durch eine Singulärwertzerlegung (SVD) bestimmen [Sto11]. Die Matrizen $A \in \mathbb{R}^{m \times n}$ und $A^+ \in \mathbb{R}^{n \times m}$ können geschrieben werden als:

$$A = U \Sigma V^T$$

$$A^+ = V \Sigma^+ U^T$$

mit orthogonalen Matrizen $U \in \mathbb{R}^{m \times m}$ und $V \in \mathbb{R}^{n \times n}$ und einer Diagonalmatrix $\Sigma \in \mathbb{R}^{m \times n}$ mit den absteigend sortierten Singulärwerten auf der Diagonale (aufgefüllt mit Nullen), sowie $\Sigma^+ \in \mathbb{R}^{n \times m}$ mit den jeweiligen reziproken Singulärwerten.

Die Lösung des Ausgleichsproblems kann dann mittels SVD angeben werden als:

$$c = V \Sigma^+ U^T y$$

Zu beachten ist, dass dieses Vorgehen auch bei Matrizen A mit $\text{rang}(A) = n$ funktioniert, also wenn $A^T A$ invertierbar ist und darüberhinaus numerisch stabil implementiert werden kann [Sto11].

2.3. Kamerakalibrierung

Um die Parameter einer Kamera korrekt beschreiben zu können, ist ein Kameramodell notwendig. Das wohl bekannteste Kameramodell ist das Lochkamera-Modell. Die Lichtstrahlen der Szene gelangen dabei durch eine kleine Öffnung in die Kamera (siehe dazu Abbildung 2.6).

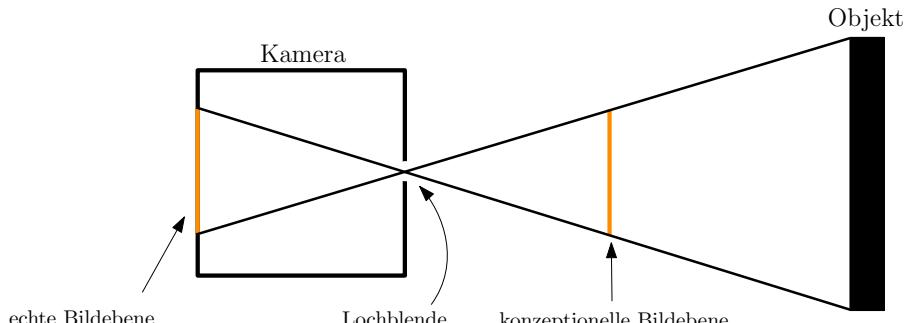


Abbildung 2.6.: Lochkameramodell

Das Bild das an der Rückseite der Kamera entsteht ist dabei um 180° gedreht. Damit man diese Rotation nicht betrachten muss, kann man die Bildebene virtuell vor die Lochblende setzen. Da sich der Abstand zur Blende nicht ändert, ändern sich auch die optischen Eigenschaften des Bildes nicht.

Kamerakalibrierung wird benötigt, um eine Beziehung zwischen Punkten im dreidimensionalen Weltkoordinatensystem und den Punkten auf der zweidimensionalen Bildebene herstellen zu können. Konkret suchen wir eine projektive Abbildung

$$\begin{pmatrix} wu \\ wv \\ w \end{pmatrix} = \underbrace{\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{pmatrix}}_{=:A} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}, \quad (2.11)$$

die einen Punkt $P = (x, y, z)$ in homogenen Koordinaten $\tilde{P} = (x, y, z, 1)$ auf einen Punkt $\tilde{C} = (wu, wv, w)$ beziehungsweise nach der perspektivischen Division $C = (u, v)$ auf die Bildebene abbildet, wobei die a_{ij} von den intrinsischen und extrinsischen Parametern der Kamera abhängen [HS97].

Als intrinsische Parameter einer Kamera, bezeichnen wir die optischen Eigenschaften, sowie die interne Kamerageometrie. Hierzu zählen beispielsweise Linsenverzerrungen, sowie die Brennweite. Extrinsische Parameter beschreiben die dreidimensionale Position und Orientierung der Bildebene relativ zu einem Weltkoordinatensystem [Tsa87].

Man kann die Matrix A mit gegebene Punktkorrespondenzen bestimmen. Wir benötigen dafür Punkte (x_k, y_k, z_k) , $k = 1, \dots, m$ im Weltkoordinatensystem und die korrespondierenden Punkte (u_k, v_k) auf der Bildebene. Wir stellen zunächst das Gleichungssystems 2.11 nach u und v um:

$$u_k = \frac{a_{11}x_k + a_{12}y_k + a_{13}z_k + a_{14}}{a_{31}x_k + a_{32}y_k + a_{33}z_k + a_{34}}$$

$$v_k = \frac{a_{21}x_k + a_{22}y_k + a_{23}z_k + a_{24}}{a_{31}x_k + a_{32}y_k + a_{33}z_k + a_{34}}.$$

Wir wollen nun die a_{ij} als Unbekannte betrachten und formulieren das Gleichungssystem noch einmal um zu:

$$\underbrace{\begin{pmatrix} x_1 & y_1 & z_1 & 1 & 0 & 0 & 0 & 0 & -u_1x_1 & -u_1y_1 & -u_1z_1 \\ 0 & 0 & 0 & 0 & x_1 & y_1 & z_1 & 1 & -v_1x_1 & -v_1y_1 & -v_1z_1 \\ \vdots & \vdots \\ x_m & y_m & z_m & 1 & 0 & 0 & 0 & 0 & -u_mx_m & -u_my_m & -u_mz_m \\ 0 & 0 & 0 & 0 & x_m & y_m & z_m & 1 & -v_mx_m & -v_my_m & -v_mz_m \end{pmatrix}}_{=:M} = \begin{pmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \\ a_{31} \\ a_{32} \\ a_{33} \end{pmatrix} = \begin{pmatrix} u_1 \\ v_1 \\ u_m \\ v_m \end{pmatrix}, \quad (2.12)$$

wobei a_{34} auf eins skaliert werden darf, da wir in homogenen Koordinaten rechnen. Es handelt sich hierbei für $m \geq 6$ um ein überbestimmtes lineares Gleichungssystem, was mittels der Methode der kleinsten Quadrate (siehe Kapitel 2.2) und durch eine Singulärwertzerlegung gelöst werden kann. Diese Verfahren wird auch als Direct Linear Transformation (DLT) bezeichnet.

Das beschriebene Verfahren vernachlässigt dabei Linsenverzerrungen. Solche Verzerrungen entstehen entweder als Produktionsfehler günstiger Linsen, oder werden durch spezielle Linsen, wie bei Weitwinkelkameras, begünstigt. Verzerrungen können in der Regel nicht linear modelliert werden. Der Ansatz über DLT funktioniert hier dementsprechend nicht.

Es wird grundsätzlich zwischen zwei Typen von Verzerrungen unterschieden: radiale Verzerrung und tangentiale Verzerrung. Radiale Verzerrung entsteht durch eine Skalierung des Abstand eines Bildpunktes zum Fokus. Wird der Abstand vergrößert spricht man von einer tonnenförmigen Verzerrung (siehe Abbildung

2.7(b)), wird er verkleinert von einer kissenförmigen Verzerrung (siehe Abbildung 2.7(c)). Radiale Verzerrung kann wie folgt modelliert werden:

$$\begin{aligned}\hat{x}^{(r)} &= x \left[1 + k_1 (x^2 + y^2) + k_2 (x^2 + y^2)^2 + \dots \right] \\ \hat{y}^{(r)} &= y \left[1 + k_1 (x^2 + y^2) + k_2 (x^2 + y^2)^2 + \dots \right],\end{aligned}$$

wobei (x, y) ein verzerrter Bildpunkt, $(\hat{x}^{(r)}, \hat{y}^{(r)})$ der entzerrte Punkt und k_1 und k_2 die Koeffizienten der radialen Verzerrung sind [Zha00].

Tangentiale Verzerrung ist auf eine fehlerhafte Fertigung zurückzuführen und entsteht beispielsweise wenn sich eine Linse nicht parallel zur Bildebene befindet. Sie kann modelliert werden durch:

$$\begin{aligned}\hat{x}^{(t)} &= [2p_1 xy + p_2 ((x^2 + y^2) + 2x^2)] \\ \hat{y}^{(t)} &= [p_1 ((x^2 + y^2) + 2y^2) + 2p_2 xy],\end{aligned}$$

wobei (x, y) wieder ein verzerrter Bildpunkt, $(\hat{x}^{(t)}, \hat{y}^{(t)})$ der entzerrte Punkt und p_1 und p_2 die Koeffizienten der tangentialen Verzerrung sind [HS97].

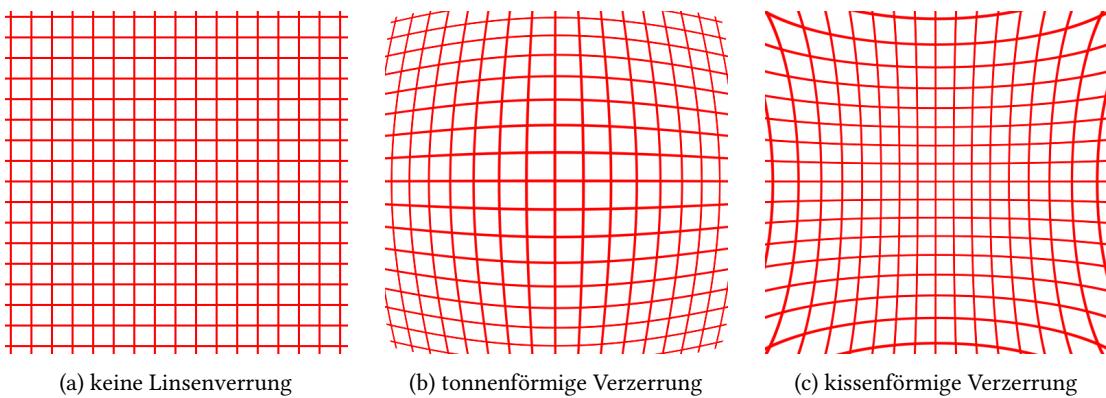


Abbildung 2.7.: Linsenverzerrungen

2.4. Blob-Detektor

2.4.1 Definition (Blob)

Ein Blob ist eine glatte zusammenhängende Region in einem Bild, die sich farblich von ihrer Umgebung abhebt [Lin93].

Ein Blob-Detektor ist also ein Verfahren zum Detektieren solcher Blobs. Die gefundenen Blobs können anschließend nach verschiedenen Kriterien, wie beispielsweise Größe, Farbe, Konvexität oder Rundheit gefiltert werden.

2.5. Hough-Transformation

Hough-Transformation ist ein Verfahren zur Detektion von beliebigen parametrisierbaren Konturen. In dieser Arbeit werden Hough-Transforamtionen benutzt um Geraden zu detektieren.

Eine beliebige zweidimensionale Gerade kann in Polarkoordinaten folgendermaßen implizit ausgedrückt werden:

$$x \cos \phi + y \sin \phi = d, \quad (2.13)$$

wobei $\phi \in [0, 2\pi)$ der Winkel der Geraden mit der x -Achse und $d \geq 0$ der Radius, also der euklidische Abstand der Geraden zum Ursprung des Koordinatensystems ist.

Eine Gerade wird somit als ein Punkt (d, ϕ) in den Parameterraum (auch Hough-Raum) abgebildet.

Um eine Gerade eindeutig zu definieren benötigt man, wie auch bei der klassischen Defintionen $y = mx + b$, zwei Punkte. Beschränkt man sich auf einen Punkt, so lässt sich jedoch die Auswahl von ϕ und d einschränken. Hat man beispielsweise einen Punkt (x_k, y_k) gegeben so lässt sich die Gleichung 2.13 nach d umstellen und man erhält eine sinusförmige Funktion in Abhängigkeit von ϕ .

Um nun beliebige Geraden detektieren zu können, werden ϕ und d zunächst passend diskretisiert:

$$\begin{aligned} \phi_i &= \phi_{min} + \frac{i}{n} \cdot (\phi_{max} - \phi_{min}) \quad \forall i \in [0, n] \\ d_j &= d_{min} + \frac{j}{m} \cdot (d_{max} - d_{min}) \quad \forall j \in [0, m] \end{aligned}$$

Es wird nun ein Akkumulator $\mathcal{H}(\phi, d)$ für alle ϕ_i und d_j auf null gesetzt. Als Nächstes wird ein Kantenbild mittels Canny [Can86] erzeugt. Die Kantenpixel werden mit (x_k, y_k) für $k = 0, \dots, l$ identifiziert. Das eigentliche Verfahren funktioniert nach folgendem Schema:

Algorithmus 1 Hough-Transformation

```

1: for  $k = 0$  to  $l$  do // für alle Kantenpunkte
2:   for  $i = 0$  to  $n$  do // für alle diskreten Winkel  $\phi_i$ 
3:      $d \leftarrow x \cos \phi_i + y \sin \phi_i$ 
4:      $d_i \leftarrow \text{round}(d)$  // runde  $d$  zum nächsten diskreten  $d_i$ 
5:      $\mathcal{H}(\phi_i, d_i) \leftarrow \mathcal{H}(\phi_i, d_i) + 1$ 
6:   end for
7: end for

```

Die Werte im Akkumulator werden oft auch als Votes bezeichnet. Am Ende des Verfahrens sucht man im Akkumulator nach Häufungspunkten. Jeder Häufungspunkt steht dort für einen Geradenkandidaten.

2.6. RANSAC

Random Sample Consensus (RANSAC) [FB81] ist ein nicht-deterministisches robustes Verfahren zur Parameterschätzung eines Modells bei einer, möglicherweise durch starke Ausreißer, gestörten Messreihe. Im Gegensatz zu Verfahren, wie der Methode der kleinsten Quadrate, die versuchen eine optimale Lösung für alle Messdaten zu bestimmen, werden bei RANSAC nur eine Teilmenge der Messreihe genutzt.

Wir wählen aus der Menge der Messdaten wiederholt zufällig die minimale Anzahl Messdaten aus, die nötig sind um das Modell eindeutig zu beschreiben und prüfen dann, wie gut das geschätzte Modell die restlichen Messdaten beschreibt. Die Güte des Modells wird im Allgemeinen durch ein Distanzmaß, wie zum Beispiel der euklidische Abstand, berechnet. Hat ein Messdatum eine vorher definierte Maximaldistanz zum geschätzten Modell nicht überschritten, wird es ins sogenannte Consensus Set des Modells aufgenommen. Das Modell mit dem größten Consensus Set wird schließlich ausgewählt.

Die Anzahl der Iterationen, die mindestens notwendig sind, um mit einer Wahrscheinlichkeit von $p \in [0, 1]$, bei einem relativen Ausreißeranteil von $\epsilon \in [0, 1)$ und einer Anzahl von k Daten, um das Modell eindeutig zu beschreiben, mindestens einmal eine ausreißerfreie Teilmenge der Messreihe zu erhalten, lässt sich berechnen mit [FB81]

$$n_{min} = \frac{\log(1-p)}{\log\left(1-(1-\epsilon)^k\right)}. \quad (2.14)$$

2.7. Ellipsen

2.7.1. Definition

2.7.1 Definition (Ellipse)

Eine Ellipse wird beschrieben durch diejenigen Punkte, dessen Summe der Abstände zu zwei gegebenen Punkten f_1 und f_2 (Brennpunkte) konstant sind. Der Mittelpunkt der Verbindungslinie der beiden Brennpunkte wird als Zentrum der Ellipse bezeichnet (siehe Abbildung 2.8). Als Hauptachse a bezeichnen wir die Verbindungslinie von Mittelpunkt durch einen der Brennpunkte bis zum Scheitelpunkt (V_1 beziehungsweise V_3) der Ellipse. Die zu ihr rechtwinklige Verbindungslinie durch den Mittelpunkt bis zu einem der anderen Scheitelpunkte (V_2 beziehungsweise V_4) nennen wir Nebenachse b .

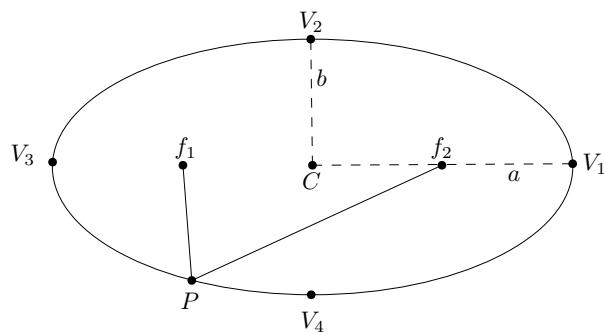


Abbildung 2.8.: Ellipse mit Brennpunkten f_1, f_2 , Zentrum C , Hauptachse a , Nebenachse b und Scheitelpunkten V_1/V_3 und V_2/V_4

In ihrer einfachsten Form liegt die Ellipse im Zentrum des Koordinatensystems und ihre Haupt- und Nebenachse a und b sind achsenausgerichtet. Das heißt ihre Hauptachse liegt auf der x -Achse und ihre Nebenachse auf der y -Achse. Sie kann dann in der impliziten Form

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (2.15)$$

beschrieben werden.

Befindet sich die Ellipse nicht im Ursprung so muss eine Verschiebung beziehungsweise bei einer Rotation ein Drehwinkel (siehe Abbildung 2.9) ergänzt werden. Es gilt

$$\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} = 1 \quad (2.16)$$

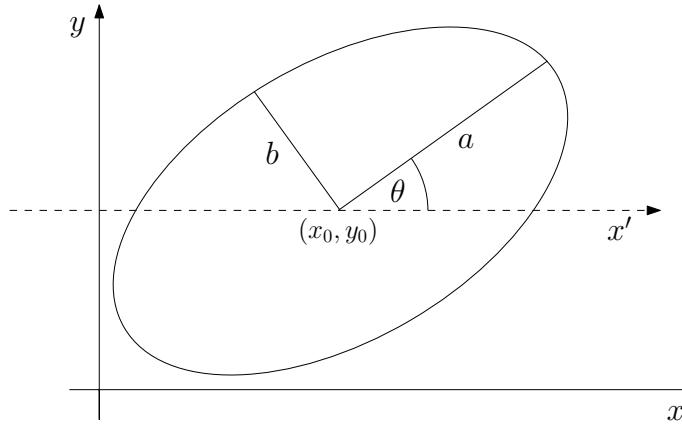


Abbildung 2.9.: Ellipse mit Zentrum (x_0, y_0) , Hauptachse a , Nebenachse b , sowie Drehwinkel θ

beziehungsweise

$$\frac{((x - x_0) \cos \theta + (y - y_0) \sin \theta)^2}{a^2} + \frac{((x - x_0) \sin \theta - (y - y_0) \cos \theta)^2}{b^2} = 1 \quad (2.17)$$

mit Ellipsenzentrum $(x_0, y_0) \in \mathbb{R}^2$, Hauptachsen $a, b \in \mathbb{R}^+$, sowie Drehwinkel $\theta \in [0, 2\pi)$ oder parametrisiert

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_0 + a \cos \phi \cos \theta - b \sin \phi \sin \theta \\ y_0 + a \cos \phi \sin \theta + b \sin \phi \cos \theta \end{pmatrix} \quad (2.18)$$

mit $\phi \in [0, 2\pi)$ und x_0, y_0, a, b, θ wie oben.

In ihrer allgemeinsten Form lässt sich eine Ellipse durch ein implizites Polynom zweiten Grades charakterisieren

$$ax^2 + by^2 + cxy + dx + ey + f = 0 \quad \text{mit} \quad c^2 - 4ab < 0 \quad (2.19)$$

mit $a, b, c, d, e, f \in \mathbb{R}$. Eine Ellipse lässt sich also durch sechs Punkte eindeutig beschreiben (fünf, wenn man f auf eins skaliert).

Die beiden Formen 2.17 und 2.19 sind äquivalent, falls die Ellipse nicht degeneriert ist (ohne Beweis) [Law72]. Die Umformung von 2.17 nach 2.19 ist mit Hilfe einer Hauptachsentransformation möglich. Da wir diese später brauchen, wird sie hier exemplarisch vorgeführt.

Zunächst einmal fällt auf, dass der gemischte Term cxy genau dann null ist, wenn die Ellipse nicht rotiert wurde. Im ersten Schritt versuchen wir also die Rotation der Ellipse rückgängig zu machen, um den Rotationswinkel bestimmen zu können.

Die Gleichung 2.19 kann umgeformt werden zu:

$$\begin{aligned} ax^2 + \frac{c}{2}xy + \frac{c}{2}yx + by^2 + dx + ey + f &= 0 \\ \Leftrightarrow (ax + \frac{c}{2}y) \begin{pmatrix} x \\ y \end{pmatrix} + dy + ey + f &= 0 \\ \Leftrightarrow \underbrace{\begin{pmatrix} x & y \end{pmatrix}}_{=:u^T} \underbrace{\begin{pmatrix} a & \frac{c}{2} \\ \frac{c}{2} & b \end{pmatrix}}_{=:M} \underbrace{\begin{pmatrix} x \\ y \end{pmatrix}}_{=:u} + (d \quad e) \underbrace{\begin{pmatrix} x \\ y \end{pmatrix}}_{=:u} + f &= 0 \\ \Leftrightarrow u^T Mu + (d \quad e) u + f &= 0 \end{aligned}$$

Der gemischte Term wird alleine durch $M = \begin{pmatrix} a & \frac{c}{2} \\ \frac{c}{2} & b \end{pmatrix}$ bestimmt. Die Matrix M ist invertierbar, denn

$$\det M = ab - \frac{c^2}{4}$$

ist nur dann gleich null, wenn $c^2 - 4ab = 0$, was ein Widerspruch zur Annahme in Gleichung 2.19 ist. M hat somit eine von null verschiedene Determinante und somit vollen Rang, hat also zwei von null verschiedene Eigenwerte [Bos06, S. 199]. Insbesondere gibt es also zwei Eigenvektoren von M , die zueinander orthogonal sind, da Eigenvektoren zu verschiedenen Eigenwerten orthogonal zueinander sind.[Bos06, S. 198].

Da die Matrix M außerdem symmetrisch ist, ist sie orthogonal diagonalisierbar [Bos06, S. 278].

Es gilt $M = S^T DS$, wobei $S \in \mathbb{R}^{2 \times 2}$ eine orthogonale Matrix mit den normierten Eigenvektoren als Zeilen und $D = \text{diag}(\lambda_1, \lambda_2) \in \mathbb{R}^{2 \times 2}$ eine Diagonalmatrix mit den beiden Eigenwerten von M auf der Diagonalen ist. Ohne Beschränkung der Allgemeinheit gelte $\lambda_1 \leq \lambda_2$, andernfalls vertausche die Eigenvektoren in S .

Sei nun $v := Su$. So gilt:

$$\begin{aligned} u^T (S^T DS) u + (d - e) \underbrace{(S^T S)}_{=1} u + f &= 0 \\ \Leftrightarrow (Su)^T D(Su) + (d - e) S^T (Su) + f &= 0 \\ \Leftrightarrow v^T Dv + (d - e) S^T v + f &= 0 \\ \Leftrightarrow \lambda_1 v_1^2 + \lambda_2 v_2^2 + (d - e) S^T v + f &= 0 \end{aligned} \tag{2.20}$$

mit $v = (v_1, v_2)^T$. Man sieht, dass der gemischte Teil somit eliminiert wurde. Durch Anwenden der Transformation S wurde u also in das Koordinatensystem, in dem die Ellipse achsenausgerichtet ist, transformiert.

Eine Rotationsmatrix mit Rotationswinkel θ ist definiert durch:

$$R = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \tag{2.21}$$

Es gilt offenbar $S = R$ für ein geeignetes θ , da die Eigenvektoren normiert und orthogonal zueinander sind θ kann also einfach ausgerechnet werden, denn es gilt:

$$\theta = \text{atan2}(\sin \theta, \cos \theta) = \text{atan2}(S_{2,1}, S_{1,1})$$

Formt man nun Gleichung 2.20 weiter um, ergibt sich:

$$\begin{aligned}
& \lambda_1 v_1^2 + \lambda_2 v_2^2 + \underbrace{(d \quad e) S^T v + f}_{=: (d', e')} = 0 \\
\Leftrightarrow & \lambda_1 v_1^2 + \lambda_2 v_2^2 + d' v_1 + e' v_2 + f = 0 \\
\Leftrightarrow & (\lambda_1 v_1^2 + d' v_1) + (\lambda_2 v_2^2 + e' v_2) + f = 0 \\
\Leftrightarrow & (\lambda_1 v_1^2 + d' v_1) + \left(\frac{d'^2}{4\lambda_1} - \frac{d'^2}{4\lambda_1} \right) + (\lambda_2 v_2^2 + e' v_2) + \left(\frac{e'^2}{4\lambda_2} - \frac{e'^2}{4\lambda_2} \right) + f = 0 \tag{2.22} \\
\Leftrightarrow & \left[\lambda_1 \left(v_1^2 + \frac{2d'}{2\lambda_1} v_1 + \frac{d'^2}{4\lambda_1^2} \right) - \frac{d'^2}{4\lambda_1} \right] + \left[\lambda_2 \left(v_2^2 + \frac{2e'}{2\lambda_2} v_2 + \frac{e'^2}{4\lambda_2^2} \right) - \frac{e'^2}{4\lambda_2} \right] + f = 0 \\
\Leftrightarrow & \underbrace{\lambda_1 (v_1 + \frac{d'}{2\lambda_1} v_1)^2}_{=-x'_0} + \underbrace{\lambda_2 (v_2 + \frac{e'}{2\lambda_2} v_2)^2}_{=-y'_0} - \underbrace{\left(\frac{d'^2}{4\lambda_1} + \frac{e'^2}{4\lambda_2} - f \right)}_{=:\sigma} = 0,
\end{aligned}$$

da $\lambda_1, \lambda_2 \neq 0$.

Vergleichen wir diese Gleichung mit Gleichung 2.16, so können wir das Zentrum der transformierten Ellipse (x'_0, y'_0) aus Gleichung 2.22 einfach ablesen. Um das Zentrum der eigentlichen Ellipse zu bestimmen, müssen wir mit der inversen Rotation S^T multiplizieren:

$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = S^T \begin{pmatrix} x'_0 \\ y'_0 \end{pmatrix}$$

Die Gleichung 2.22 lässt sich anschließend weiter vereinfachen:

$$\begin{aligned}
& \lambda_1 (v_1 - x'_0)^2 + \lambda_2 (v_2 - y'_0)^2 = \sigma \\
\Leftrightarrow & \frac{\lambda_1}{\sigma} (v_1 - x'_0)^2 + \frac{\lambda_2}{\sigma} (v_2 - y'_0)^2 = 1 \tag{2.23}
\end{aligned}$$

wobei $\sigma \neq 0$, wenn die Ellipse nicht zum Punkt entartet ist (folgt aus dem Determinantenkriterium [Law72]). Vergleichen wir nun die beiden Gleichungen 2.23, sowie 2.16, so können wir folgende Beziehung herleiten:

$$\begin{aligned}
& \frac{\lambda_1}{\sigma} = \frac{1}{a^2} \quad \text{und} \quad \frac{\lambda_2}{\sigma} = \frac{1}{b^2} \\
\Leftrightarrow & \sqrt{\frac{\sigma}{\lambda_1}} = a \quad \text{und} \quad \sqrt{\frac{\sigma}{\lambda_2}} = b \tag{2.24}
\end{aligned}$$

Wir dürfen hierbei die Wurzel ziehen, da $\frac{\sigma}{\lambda_i} > 0$ für nicht entartete reelle¹ Ellipsen (folgt aus dem Determinantenkriterium [Law72]) gilt. Da die Transformation in das Koordinatensystem der Ellipse nur Rotationen enthält, bleiben die Längen der Hauptachsen erhalten. Es gilt wie erwartet $a \geq b$, da $\lambda_1 \leq \lambda_2$.

2.7.2. Abstand: Punkt zu Ellipse

Das hier beschriebene Verfahren zur Bestimmung der kürzesten euklidischen Distanz eines Punktes zu einer Ellipse stammt aus der Arbeit von David Eberly [Ebe13]. Wir betrachten nur Ellipsen im Ursprung, die achsenausgerichtet sind und darüber hinaus nur Punkte im ersten Quadranten. Ansonsten wird die Ellipse in den Ursprung verschoben und um ihren entgegengesetzten Drehwinkel rotiert. Der untersuchte Punkt

¹Man kann Ellipsen auch mit imaginären Achsen definieren [Law72].

wird anschließende analog zur Ellipse transformiert. Da die Ellipse dann bezüglich der x - und y -Achse symmetrisch ist, kann der Punkt einfach durch Spiegelung in den richtigen Quadranten verschoben werden. Der Abstand zur Ellipse ändert sich dadurch nicht.

Wir bezeichnen von nun an $Q = (y_0, y_1)$ als einen Punkt, dessen Distanz zur Ellipse wir berechnen wollen und $E = (x_0, x_1)$ als denjenigen eindeutigen Punkt, welcher auf der Ellipse liegt und die kürzeste euklidische Distanz zum Punkt Q hat.

Aufgrund dieser Forderungen können wir ohne Beschränkung der Allgemeinheit folgende Aussagen treffen:

- Die Ellipse kann stets durch die implizite Gleichung

$$\frac{x_0^2}{a^2} + \frac{x_1^2}{b^2} = 1 \quad (2.25)$$

mit $a \geq b \geq 0$ beziehungsweise in der parametrischen Form

$$\mathcal{E}(\theta) = (a \cos \phi, b \sin \phi) \quad \phi \in [0, 2\pi)$$

beschrieben werden.

- Es gilt $y_0, y_1, x_0, x_1 \geq 0$.

Für die quadrierte Distanz von einem beliebigen Punkt Q zu einem Punkt $\mathcal{E}(\theta)$ auf der Ellipse gilt dann

$$F(\theta) = |\mathcal{E}(\theta) - Q|^2. \quad (2.26)$$

Wir wollen F minimieren und betrachten daher die Ableitung

$$F'(\theta) = 2(\mathcal{E}(\theta) - Q) \cdot \mathcal{E}'(\theta). \quad (2.27)$$

F' wird null, wenn $(\mathcal{E}(\theta) - Q)$ und $\mathcal{E}'(\theta)$ zu einander orthogonal sind. Daraus folgt, dass der Vektor von Q zu E senkrecht zur Ellipse stehen muss (siehe Abbildung 2.10).

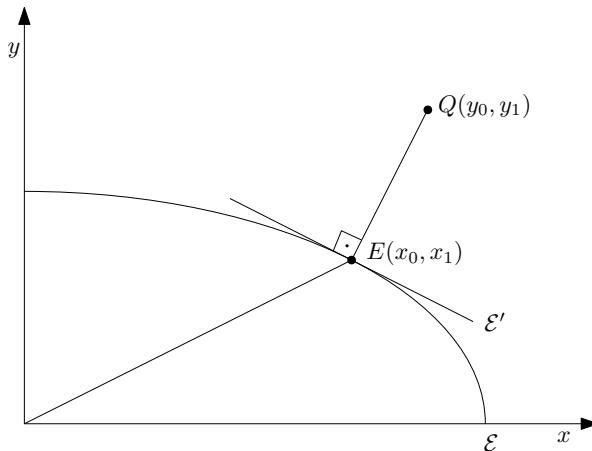


Abbildung 2.10.: Ellipsenausschnitt im ersten Quadranten mit Abfragepunkt Q und eingezeichneter kürzester Distanz zum Ellipsenpunkt E . Die Ellipse wird beschrieben durch \mathcal{E}

Betrachten wir die Funktion $G(z_0, z_1)$ basierend auf der impliziten Darstellung aus Gleichung 2.25

$$G(z_0, z_1) = \frac{z_0^2}{a^2} + \frac{z_1^2}{b^2} - 1. \quad (2.28)$$

Die Nullstellen von G beschreiben somit die Ellipse. Sei nun solch eine Nullstelle gegeben als (x_0, y_0) . Der Gradient ∇G ist dann in (x_0, x_1) ein Normalenvektor der Ellipse. Somit ist auch der halbe Gradient $\nabla G(x_0, x_1)/2$ ein Normalenvektor. Da aus Gleichung 2.27, die Orthogonalität von \overrightarrow{QE} zur Ellipse folgt, muss der Vektor \overrightarrow{QE} ein skalares Vielfaches des halben Gradienten sein. Es gilt somit:

$$(y_0, y_1) - (x_0, x_1) = t \frac{\nabla G(x_0, x_1)}{2} = t \left(\frac{x_0}{a^2}, \frac{x_1}{b^2} \right) \quad (2.29)$$

für ein $t \in \mathbb{R}$.

Umgestellt nach y_0 und y_1 , beziehungsweise nach x_0 und x_1 ergibt sich:

$$y_0 = x_0 \left(1 + \frac{t}{a^2} \right), \quad y_1 = x_1 \left(1 + \frac{t}{b^2} \right) \quad (2.30)$$

$$x_0 = \frac{a^2 y_0}{t + a^2}, \quad x_1 = \frac{b^2 y_1}{t + b^2} \quad (2.31)$$

Man macht nun eine Fallunterscheidung:

1. Der einfachste Fall ist, wenn sich der Punkt Q auf der y -Achse (außer $(0, 0)$) befindet, wenn also gilt $y_0 = 0, y_1 > 0$. Da die Hauptachse nach der y -Achse ausgerichtet ist und $a \geq b$ gilt, ist der Punkt auf der Ellipse mit der kürzesten Distanz zu Q offenbar $E = (0, b)$ und für die Distanz gilt $d = |y_1 - b|$.
2. Als nächstes betrachten wir ein Q auf der x -Achse (einschließlich $(0, 0)$), wenn also gilt $y_0 \geq 0, y_1 = 0$. Es gilt also mit 2.30

$$y_0 = x_0 \left(1 + \frac{t}{a^2} \right), \quad 0 = x_1 \left(1 + \frac{t}{b^2} \right)$$

Wenn $x_1 = 0$ gilt, muss $x_0 = a$ gelten, damit $E(x_0, x_1)$ auf der Ellipse ist. Es gilt analog zum ersten Fall $E = (a, 0)$ mit $d = |y_0 - a|$

Gilt $x_1 \neq 0$, so können wir in der zweiten Gleichung durch x_1 teilen und es folgt $t = -b^2$ und somit $y_0 = x_0 \left(1 - \frac{b^2}{a^2} \right)$. Der Punkt $E(x_0, x_1)$ liegt außerdem oberhalb von $(a, 0)$ und es gilt deshalb, auf Grund der Krümmung der Ellipse, $x_0 < a$.

und somit ergibt sich zusammen mit 2.31 die Ungleichung:

$$\begin{aligned} x_0 &= \frac{a^2 y_0}{a^2 - b^2} < a \\ \Leftrightarrow y_0 &< \frac{a^2 - b^2}{a} < a \end{aligned}$$

Das bedeutet, dass sich der Punkt $Q(y_0, y_1)$ in der Ellipse befindet.

Für Punkte $Q(y_0, 0)$ mit $y_0 \geq \frac{a^2 - b^2}{a}$ muss dementsprechend $x_1 = 0$ gelten und der kürzeste Punkt ist wieder $E(a, 0)$.

Für Punkte $Q(y_0, 0)$ mit $y_0 < \frac{a^2 - b^2}{a}$, muss für $E(x_0, x_1)$ nach Umstellen der Ellipsengleichung 2.25 $x_1 = b \cdot \sqrt{1 - (\frac{x_0}{a})}$ gelten. Die Distanz beträgt dann:

$$d^2 = (x_0 - y_0)^2 + x_1^2 = b^2 \left(1 - \frac{y_0^2}{a^2 - b^2}\right)$$

3. Der letzte Fall, den wir betrachten müssen, ist der allgemeinste Fall. Es gilt $y_0 > 0$, sowie $y_1 > 0$. Da wir uns nur im ersten Quadranten bewegen, gilt darüber hinaus $x_0, x_1 \geq 0$. Mit diesen Eigenschaften und 2.30 lässt sich folgende Einschränkung für t herleiten:

$$\begin{aligned} 0 < y_0 &= x_0 \left(1 + \frac{t}{a^2}\right) \\ \Leftrightarrow -1 \cdot a^2 &< t. \end{aligned}$$

Analog ergibt sich mit y_1 : $-b^2 < t$. Da $a \geq b$ gilt, reicht es, sich nur die zweite Ungleichung anzuschauen, da sie die erste impliziert. Setzt man nun Gleichung 2.31 in Gleichung 2.28 ein, erhält man:

$$\begin{aligned} F(t) &= \left(\frac{a^2 y_0}{a(t+a^2)}\right)^2 + \left(\frac{b^2 y_1}{b(t+b^2)}\right)^2 - 1 \\ &= \left(\frac{ay_0}{t+a^2}\right)^2 + \left(\frac{by_1}{t+b^2}\right)^2 - 1 \end{aligned}$$

Durch Untersuchung der Ableitungen kann nun zeigen, dass diese Funktion auf dem gesamten Intervall $[-b^2, \infty)$ monoton fällt und links gekrümmmt ist [Ebe13]. Darüber hinaus gilt:

$$\lim_{t \searrow -b^2} F(t) = +\infty, \quad \lim_{t \rightarrow \infty} F(t) = -1.$$

Da F stetig ist, muss es also eine Nullstelle geben, die aufgrund des Monotonie- und Krümmungsverhaltens sogar eindeutig ist. Die Nullstelle lässt sich beispielsweise durch Intervallschachtelung oder Newton-Verfahren bestimmen.

2.8. Analytical Deformable Templates

Analytical Deformable Templates ist ein Verfahren zur Detektion von analytischen Kurven. Ein Template ist dabei definiert durch eine Menge von Parametern, die a priori Wissen über die erwartete Form ermöglichen. Als Beispiel sei eine Ellipse genannt, die durch ihr Zentrum, ihren beiden Achsen und den Drehwinkel (x_0, y_0, a, b, ϕ) definiert ist. Es wird dann eine Energiefunktion konstruiert, dessen Terme eine Anziehung des Templates an Hauptmerkmale (wie Kantenstärke, Kantenorientierung, Farbintensität, etc.) des Bildes hervorrufen. Die Energiefunktion wird dabei durch numerische Verfahren, wie dem Gradientenverfahren, minimiert [YHC92].

3. Methodik

In diesem Kapitel stellen wir zwei Vorgehensweisen zur Entzerrung von Kegeloberflächen vor. Zunächst gehen wir auf das verwendete Kalibrierungsmuster ein, worauf hin die einzelnen Schritte der Entzerrung erläutert werden.

Die geometrischen Eigenschaften des Kegelstumpfs ($r, R, \Delta H$) können gemessen und somit als bekannt angenommen werden. Darüber hinaus nehmen wir an, dass sich das Zentrum des kleineren Kreises im Weltkoordinatensystem an der Position $(0, 0, 0)$ befindet (siehe Abbildung 2.3 in Kapitel 2.1). Durch diese Einschränkung gehen jegliche absolute Größenverhältnisse verloren. Die Larven können jedoch weiterhin relativ zu einander verglichen werden.

3.1. Kalibrierungsmuster

3.1.1. Aufbau des Kalibrierungsmusters

Um eine Beziehung zwischen Bildpunkten und Kegelpunkten herstellen zu können, ist ein Kalibrierungsmuster notwendig.

Die Wahl des Kalibrierungsmusters spielt dabei eine entscheidende Rolle bei der Robustheit und Präzision der Entfaltung. Es muss gewährleistet sein, dass die charakteristischen Merkmale des Musters auch bei leichten Abweichungen der Kamera vom Lot und schlechteren Beleuchtungssituationen zuverlässig erkannt werden. Das Muster muss darüber hinaus so entworfen sein, dass beim Zusammenlegen im Kegel, dessen geometrische Eigenschaften nicht verfälscht, sondern realitätsgerecht wiedergegeben werden.

Wir haben uns für ein Muster entschieden, dass in äquidistanten Abständen ΔR , beginnend mit dem kleinen Radius r des Kegelstumpfs (siehe Abbildung 2.3) Kreislinien und in gleichen Winkelabständen $\Delta\alpha$ auf der Seitenhöhe Liniensegmente besitzt. Das zusammengelegte Muster ist in Abbildung 3.1 skizziert, beziehungsweise das entfaltete in 3.2. Die Anzahl der Kreislinien wird mit n gekennzeichnet, die Anzahl sichtbarer Liniensegmente im Kegel mit m . Zu beachten ist, dass bedingt durch das Entfalten, in Abbildung 3.2 ein Liniensegment doppelt zu sehen ist. Die schwarzen Kreise bezeichnen wir als Samples.

Dadurch dass die Geometrie des Kegels bekannt ist, kann jedem Sample ein Punkt auf dem Kegel im Weltkoordinatensystem zugeordnet werden. Da ein Kegel beliebig um die y -Achse rotiert werden kann, ist diese Zuordnung zunächst nicht eindeutig. Dazu nehmen wir an, dass das Liniensegment mit dem kleinsten Winkel zur x -Achse mit dem Kegelwinkel $\theta = 0$ korrespondiert (siehe Gleichung 2.2 in Kapitel 2.1).

Zur Konstruktion des Musters benötigt man ΔS und s , die man aus der Geometrie des Kegels errechnen kann und außerdem den Öffnungswinkel, der gegeben ist als $\alpha = 2\pi \frac{R}{S}$ (siehe Gleichung 2.3 in Kapitel 2.1).

3.1.2. Anzahl der Samples

Die Anzahl der Samples sollte groß genug sein, um möglichst viele geometrische Informationen des Kegels zu erhalten, aber klein genug, dass eine Detektion der Samples problemlos möglich ist. Insbesondere auf dem innersten Kreis, macht sich eine zu hohe Sampleanzahl negativ bemerkbar, da der Abstand zueinander sehr

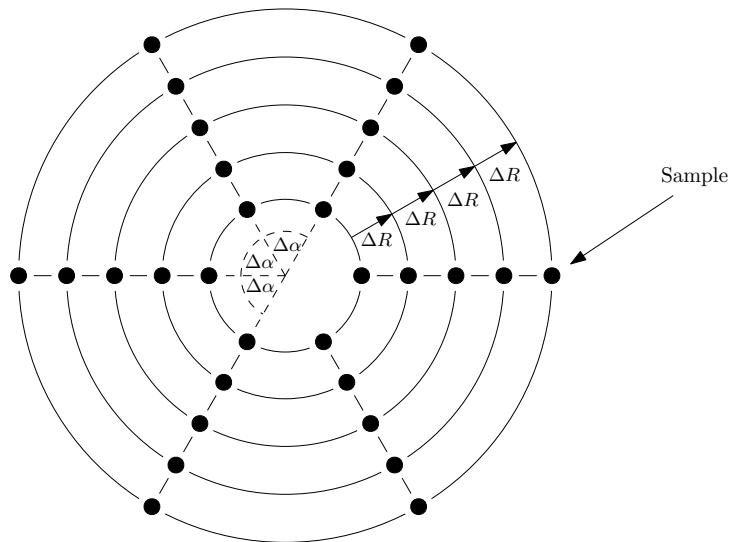


Abbildung 3.1.: Kalibrierungsmuster von oben mit $n = 5, m = 6$

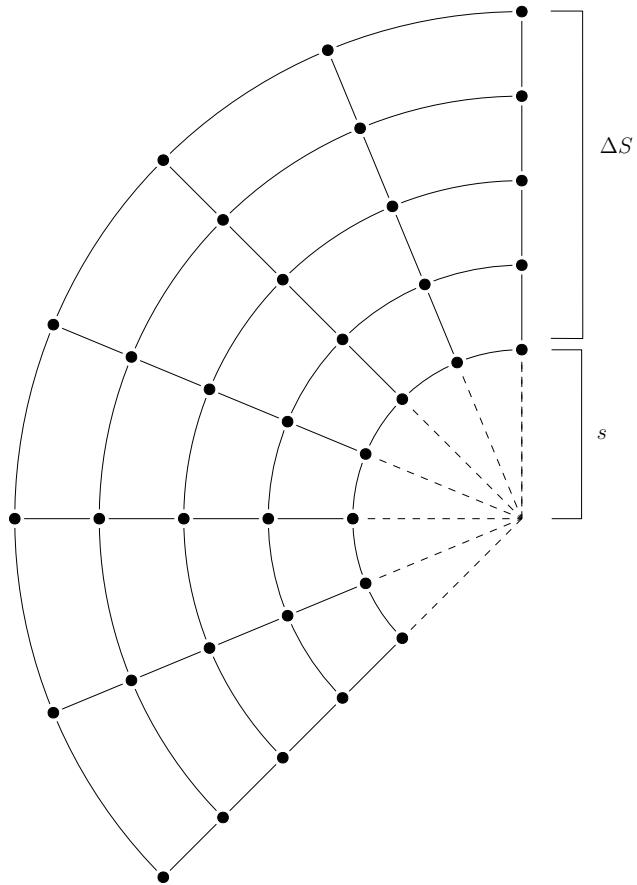


Abbildung 3.2.: Kalibrierungsmuster entfaltet mit $n = 5, m = 6$

klein wird, was eine Detektion erschwert. Des Weiteren sollte noch ein möglichst großer Teil der Kreislinien zu sehen bleiben, da diese für die spätere Ellipsendetektion benötigt werden.

3.2. Intrinsische Kamerakalibrierung

Bedingt durch die Wahl einer Weitwinkelkamera, sind die Bilder von einer starke tonnenförmige (nach außen gewölbte) Verzerrung geprägt.(siehe Abbildung 3.3). Diese muss herausgerechnet werden, da sonst Abstände im Bild nicht mehr der Realität entsprechen und dadurch die Präzision der Entfaltung stark abnimmt (siehe Kapitel 5). Es wird also im ersten Schritt eine Kamerakalibrierung durchgeführt.

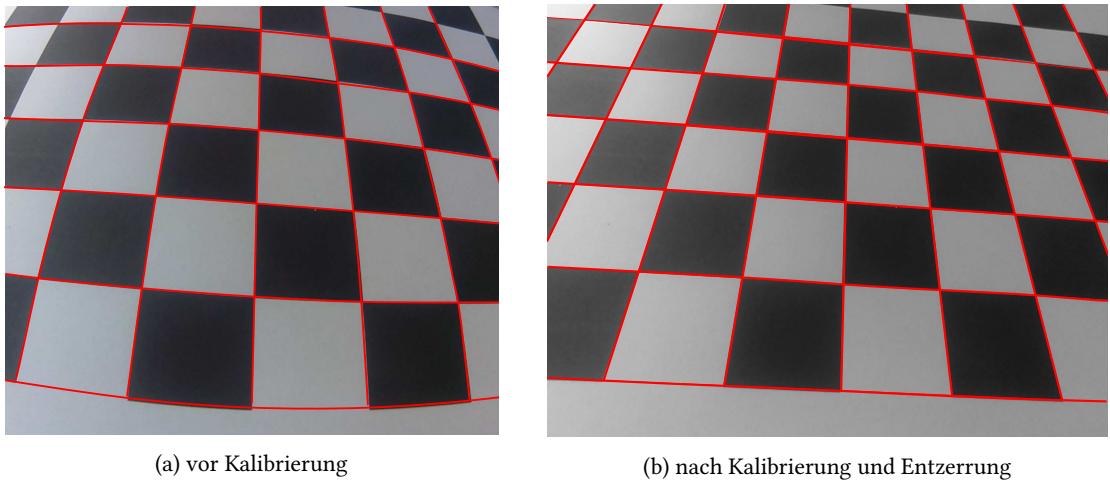


Abbildung 3.3.: Kamerakalibrierung

3.3. Detektion der charakteristischen Punkte

Nach der Kamerakalibrierung und entsprechender Entzerrung werden die Bildkoordinaten der Samples bestimmt. Dazu wird ein Blob-Detektor (siehe Kapitel 2.4) benutzt. Um ein Sample korrekt detektieren zu können, muss sich der Punkt farblich stark von seiner Umgebung abheben (siehe Definition eines Blobs 2.4.1). Das bedeutet, dass ein Sample freigestellt (helle homogene Umgebung) sein muss. Insbesondere dürfen die Kreislinien und Liniensegmente des Kalibrierungsmusters also nicht durchgezogen sein.

Nach der Detektion werden die Blobs nach folgenden Kriterien gefiltert:

- **Fläche:** zu kleine Blobs werden verworfen
- **Rundheit:** zu unrude Blobs werden verworfen. Rundheit ist hier definiert als $circ = \frac{4\pi \cdot \text{Fläche}}{(\text{Umfang})^2} \in [0, 1]$, wobei ein Kreis mit $circ = 1$ maximal rund ist.
- **Konvexität:** zu unkonvexe Blobs werden verworfen. Konvexität ist hier definiert als $conv = \frac{\text{Fläche Blob}}{\text{Fläche konvexe Hülle}}$

In Abbildung 3.4 ist beispielhaft links ein Grauwertbild und rechts die detektierten Blobs (rot) auf dem gleichen Bild nach der Kameraentzerrung zu sehen.

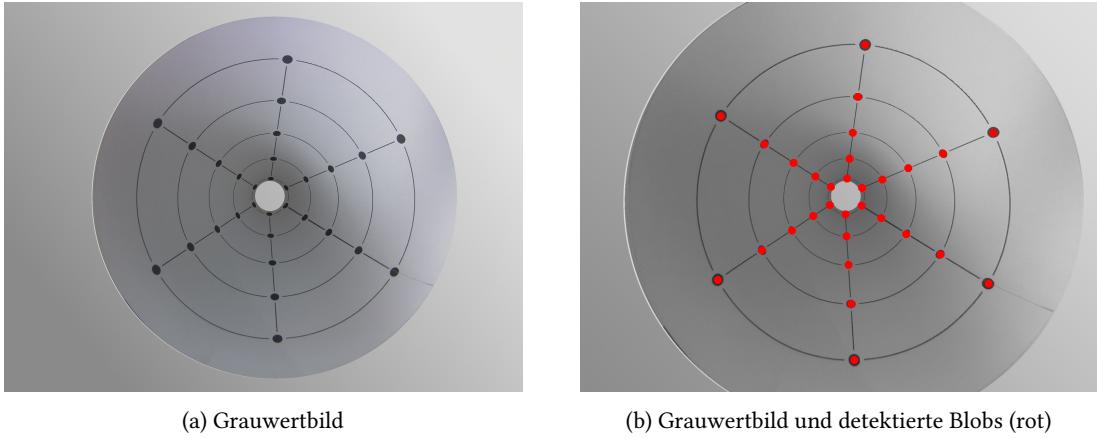


Abbildung 3.4.: Detektion der Samples

3.4. Ellipsen-Detektion

3.4.1. RANSAC

Nachdem die Sample-Positionen bestimmt wurden, muss für jeden Sample entschieden werden, auf welcher der Kreislinien er liegt. Da die Kreise, bedingt durch perspektivische Verzerrung, zu Ellipsen werden, wird eine Verfahren benötigt, dass Ellipsen erkennt.

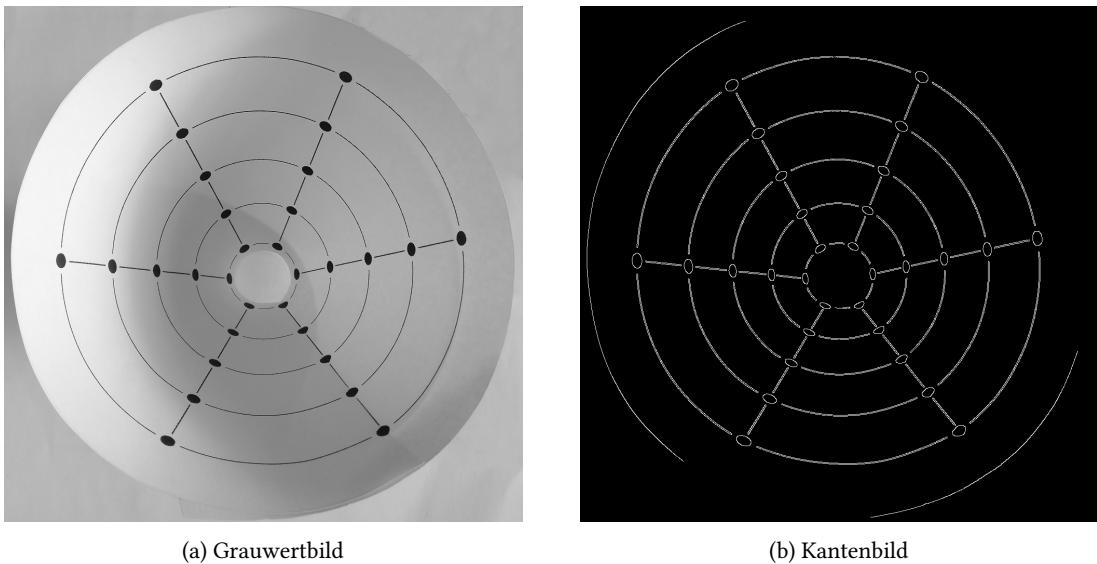


Abbildung 3.5.: Canny-Kantendetektion

Zunächst werden die Kanten mit Hilfe von Canny [Can86] detektiert (siehe Abbildung 3.5). Anschließend versuchen wir möglichst genau das Zentrum der innersten Ellipsen zu schätzen. Wir benutzen dafür Hough-Transformationen (siehe Kapitel 2.5), um Linien im Kantenbild zu detektieren. Es werden anschließend die Schnittpunkte aller Liniensegmente bestimmt. Auf Grund der perspektivischen Verzerrung, schneiden sich die Liniensegmente nicht in einem Punkt. Darüber hinaus werden, auf Grund der Liniendicke auf dem

Kalibrierungsmuster, durch Canny viele Linien als doppelte Linien gekennzeichnet¹. Auch ein inhomogener Hintergrund, erschwert die Ermittlung eines Mittelpunkts. Befinden sich Objekte im Hintergrund, werden durch die Kantendetektion auch deren Kanten ermittelt. Die Hough-Transformation detektiert nun auch außerhalb des Kalibrierungsmusters Linien. Die Schnittpunkte mit solchen Linien können beliebig stark von dem eigentlichen Mittelpunkt abweichen. Würde man an dieser Stelle beispielsweise den Mittelwert aller Schnittpunkte berechnen, könnten solche Ausreißer die Lokalisierung des Mittelpunkts erheblich verschlechtern.

Um also möglichst robust einen Kandidaten auszuwählen, wird zuerst der Median der x -Koordinaten der Schnittpunkte und dann der Median der y -Koordinaten bestimmt. Die erhaltenen Koordinaten bilden den Schnittpunkt, wie in Abbildung 3.6 beispielhaft zu sehen.

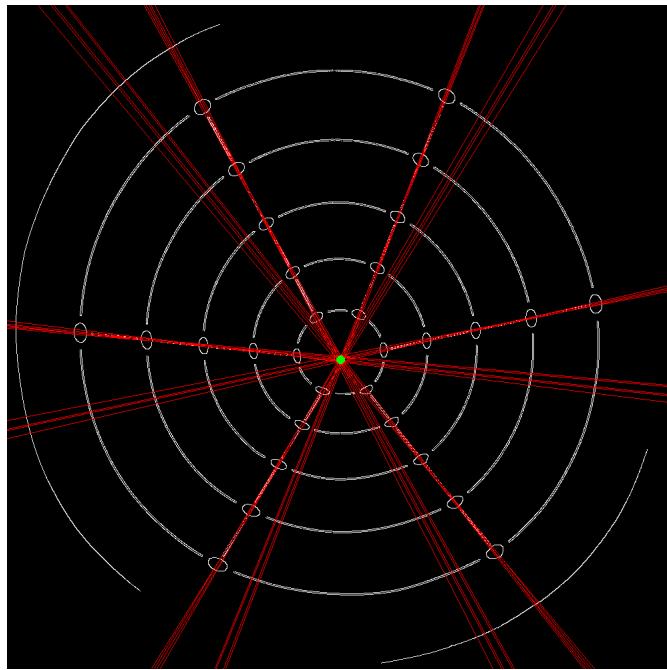


Abbildung 3.6.: Hough-Transformation zur Linien-Detektion (rot) und bestimmter Schnittpunkt (grün)

Von diesem Schnittpunkt aus werden nun, in einer vorher definierte Anzahl, gleichmäßig, in alle Richtungen Strahlen ausgesendet. Trifft ein Strahl ein weißes Pixel, wird dessen Position gekennzeichnet, trifft er den Rand des Bildes, wird er ignoriert. In Abbildung 3.7(a) sind beispielhaft die getroffenen weißen Pixel und der zugehörige Aussendepunkt eingezeichnet.

Mit Hilfe der Positionen der weißen Pixel, schätzen wir anschließend mittels RANSAC (siehe Kapitel 2.6) eine Ellipse.

Es wird konkret wiederholt für sechs zufällig ausgewählte Punkte das lineare Gleichungssystem

$$\begin{pmatrix} x_1^2 & y_1^2 & x_1y_1 & x_1 & y_1 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

¹Stellt man sich ein relativ breites Liniensegment vor, so gibt es einmal den Übergang vom Hintergrund auf die Linie, sowie den Übergang von der Linie wieder auf den Hintergrund

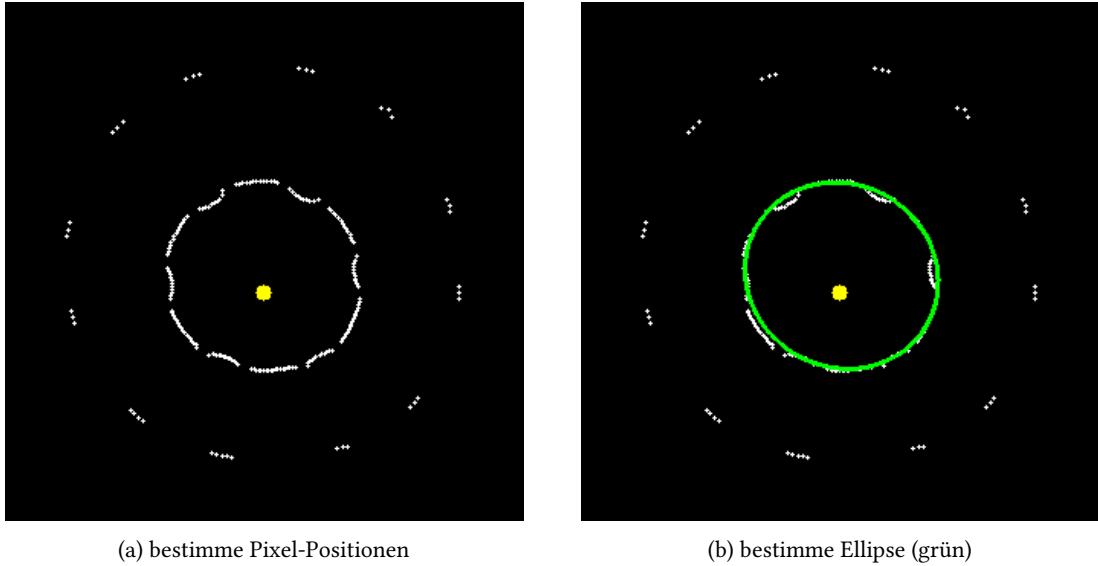


Abbildung 3.7.: Ellipsendetektion: bestimme Pixel-Positionen (weiß), Aussendepunkt (gelb)

gelöst, was auf der Gleichung 2.19 aus Kapitel 2.7 basiert. Wir benutzen hier die Polynomschreibweise einer Ellipse, da diese im Gegensatz zu der Form (x_0, y_0, a, b, ϕ) linear in den Parametern (a, b, c, d, e, f) ist. Das Gleichungssystem ist also ein lineares Gleichungssystem und kann mit herkömmlichen Methoden der Numerik gelöst werden. Ein Nachteil dieses Ansatzes ist jedoch, dass nicht alle Werte für (a, b, c, d, e, f) eine nicht entartete reelle Ellipse definieren. Nach dem Lösen muss demnach geprüft werden, ob es sich tatsächlich um eine Ellipse handelt und diese dann mittels Hauptachsentransformation (siehe Kapitel 2.7) in die Ellipsenform (x_0, y_0, a, b, ϕ) umgewandelt werden.

Um die, für RANSAC benötigte, Distanz zu berechnen, wird das Verfahren aus Kapitel 2.7.2 genutzt, was die exakte euklidische Distanz eines Punktes zu einer Ellipse approxmiert. Ein Verfahren wie das Verfahren der kleinsten Quadrate anstelle von RANSAC funktioniert hier nicht, da die weißen Pixel bezüglich einer zu bestimmenden Ellipse, ausreißerbehaftet sind. Wird zum Beispiel auf Grund schlechter Lichtverhältnisse eine Kreislinie nicht deutlich aufgenommen, kann es in dem Kantenbild dazu kommen, dass Kreislinien nicht zusammenhängend sind und folglich treffen die ausgesendeten Strahlen teilweise die nächst äußere Kreislinie. Diese Problem ist in Abbildung 3.8 zu sehen. Man sieht, dass die Anzahl der Pixel, auf der Ellipse, dessen Parameter wir schätzen wollen, im Vergleich zur Abbildung 3.7 stark abgenommen hat. Die Strahlen treffen vermehrt auf die nächste äußere Ellipse. Die Ausreißeranzahl nimmt stark zu.

Da die Laufzeit nicht im Vordergrund steht, kann eine großzügige Schätzung des Fehleranteils von $\epsilon = 0.4$ mit einer gewünschten Wahrscheinlichkeit $p = 0.9999$ gewählt werden, was zu einer Mindestanzahl an Iterationen von circa 200 führt (siehe Kapitel 2.6). Die letztendlich bestimmten Ellipsen sind beispielhaft in Abbildung 3.9 zu sehen.

3.4.2. Analytical Deformable Templates

Als Alternative zu RANSAC können die Ellipsen auch mittels *Analytical Deformable Templates* (siehe Kapitel 2.8) detektiert werden. Wir betrachten dazu folgende Funktion, deren Nullstellen wie in Kapitel 2.7, eine um θ gedrehte Ellipse, mit Zentrum (x_0, y_0) und Haupt- und Nebenachse a und b beschreiben.

$$G(x, y) = \frac{((x - x_0) \cos \theta + (y - y_0) \sin \theta)^2}{a^2} + \frac{((x - x_0) \sin \theta - (y - y_0) \cos \theta)^2}{b^2} - 1$$

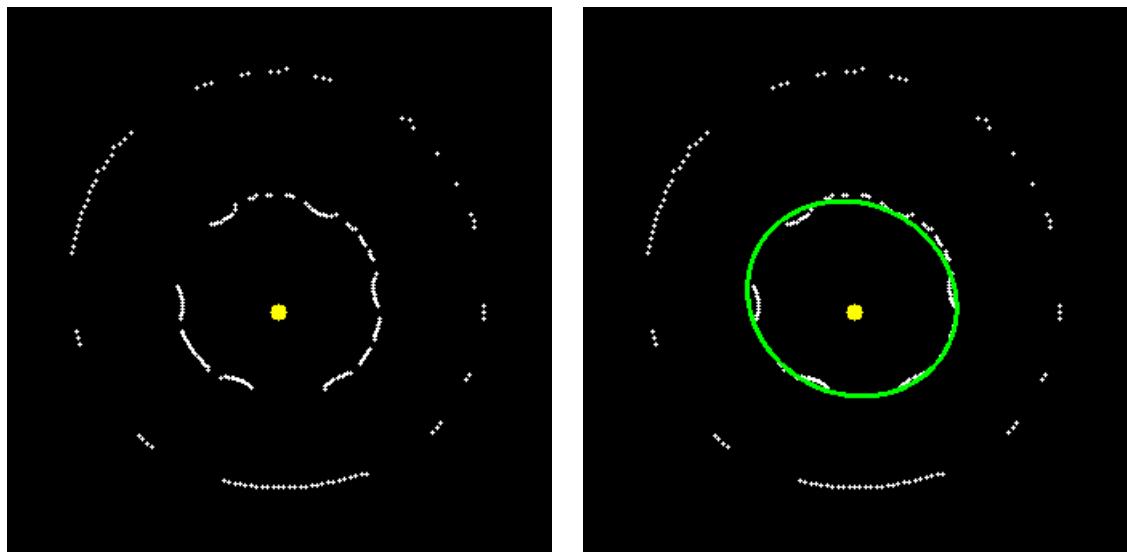


Abbildung 3.8.: Ellipsendetektion bei Ausreißern

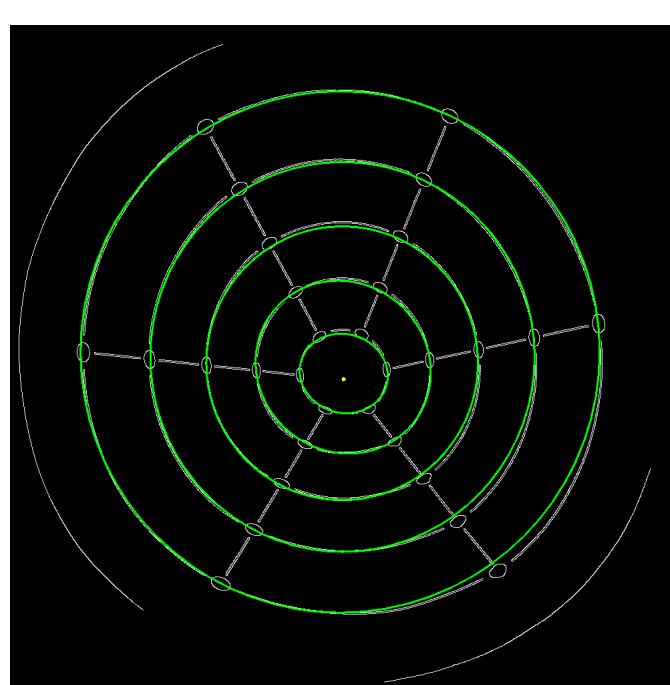


Abbildung 3.9.: detektierte Ellipsen

Wir konstruieren eine Energiefunktion

$$E = E_M + E_A + E_S,$$

die sich zusammensetzt aus einem Term E_M , der die Kantenstärke auf dem Ellipsenrand maximiert, sowie einem Term E_A der die Winkelähnlichkeit zwischen Ellipsennormale und Gradientenrichtung des Bildes

maximiert. Darüber hinaus fügen wir einen Term E_S hinzu, der die Ellipse schrumpfen lässt.

Konkret sehen die Terme wie folgt aus

$$\begin{aligned} E_M &= -\alpha \frac{1}{n} \sum_{i=0}^{n-1} I_M(p_i) \\ E_A &= \beta \frac{1}{n} \sum_{i=0}^{n-1} \left(I_O(p_i) - \text{atan2}\left(\frac{\partial G}{\partial y}(p_i), \frac{\partial G}{\partial x}(p_i)\right)\right)^2 \\ E_S &= \gamma \frac{1}{n} \sum_{i=0}^{n-2} (p_i - p_{i+1})^2, \end{aligned} \quad (3.1)$$

wobei n die Anzahl der Punkte auf dem Ellipsenrand sind. p_i ist definiert als der i -te Punkt auf der Ellipse (siehe parametrische Darstellung einer Ellipse). Man wählt dabei als Startwert eine Ellipse, deren Zentrum dem Bildzentrum entspricht und deren Hauptachse und Nebenachse möglichst groß sind und bestimmt dann numerisch ein Minimum der Funktion. Der Term E_M wird minimal, wenn entlang der Ellipse die Kantenstärke groß ist. Der Term E_A wird minimal wenn die Winkel der Normalenvektoren ähnlich zu denen der Kanten sind und E_S wird minimal, wenn die Distanz aufeinanderfolgender Ellipsepunkte klein wird, also der Ellipsenumfang klein wird. Der Term lässt die Ellipse also schrumpfen. α , β und γ steuern hierbei den Einfluss der einzelnen Terme. Das Minimum der Funktion sollte die äußerste Ellipse im Kalibrierungsmuster sein. Anschließend wird diese Ellipse vom Bild entfernt und man sucht wiederholt Minima der Funktion bis alle Ellipse detektiert wurden.

3.5. Zuordnung der Punkte

Nach der Bestimmung der Ellipsen muss jede Sample-Position der zugehörigen Kreislinie, sowie dem zugehörigen Liniensegment zugeordnet werden, um seine Position auf dem Kegel bestimmen zu können. Zunächst wird für jeden Punkt diejenige Kreislinie ausgewählt, dessen zugehörige Ellipse die kürzeste Distanz zu ihm hat (siehe Abbildung 3.11(a)).

Um nun die Samples auch ihren Liniensegmenten zuzuordnen, wird zunächst der Mittelpunkt der Samples auf der innersten Ellipse bestimmt. Anschließend werden die Samples auf der innersten Ellipse nach dem Winkel der Verbindungslinien zwischen Sample und Mittelpunkt mit der x -Achse sortiert. Die restlichen Samples können nicht nach dem gleichen Schema sortiert werden, da der bestimmte Mittelpunkt nicht der genaue Schnittpunkt aller Liniensegmente ist. Das Problem ist in Abbildung 3.10(a) am Beispiel eines Liniensegments für zwei Samples illustriert. Das bestimmte Zentrum C liegt nicht auf der Verlängerung des Liniensegments. Die Winkel der Verbindungslinien zum Zentrum α_1 , sowie α_2 unterscheiden sich. Bei einer Sortierung nach Winkel ist nicht mehr gewährleistet, dass die Samples zu einer Verbindungslinie gehören, da die Winkel der Samples eines Liniensegments beliebig voneinander abweichen können. Damit dieses Verfahren funktioniert müsste die Situation aus Abbildung 3.10(b) für alle Liniensegmente gelten, das heißt der Winkel (hier: β) ist für alle Verbindungslinien identisch.

Stattdessen wird für jedes Sample auf den darauffolgenden Ellipsen das Sample auf der vorherigen Ellipse mit der kürzesten Distanz bestimmt. Die Samples können nun entsprechend sortiert werden. Die zugeordneten Liniensegmente sind exemplarisch in Abbildung 3.11(b) zu sehen. Darüber hinaus lässt sich hier erkennen, dass das Liniensegment mit dem kleinsten Winkel zur x -Achse das erste Liniensegment (rot) ist. Das Liniensegment wird später unsere Bezugspunkt zur Entzerrung sein, also diejenige Kante, an der der Kegel entrollt wird.

Mit Hilfe dieser Zuordnung schätzen wir die Ellipsen erneut. Die Samples dienen als Messpunkte und wir nutzen das Verfahren der kleinsten Quadrate, da wir eine optimale Lösung für alle Samples anstreben.

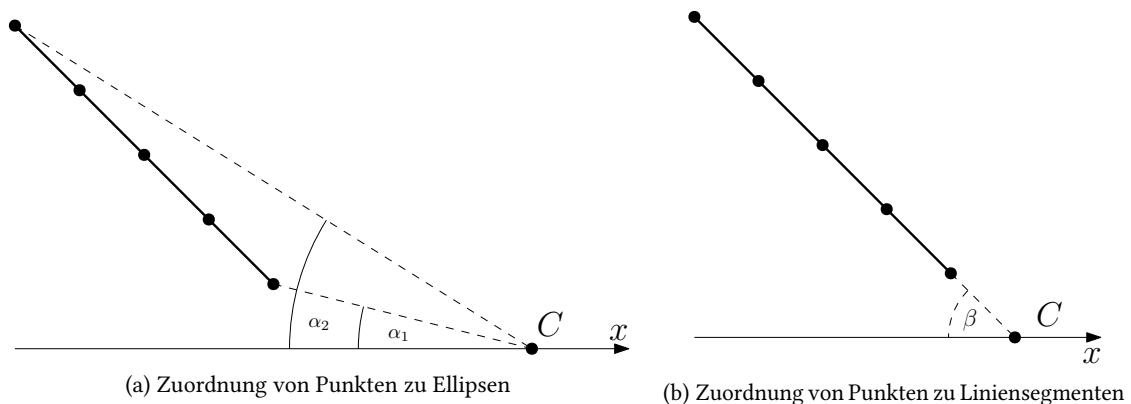
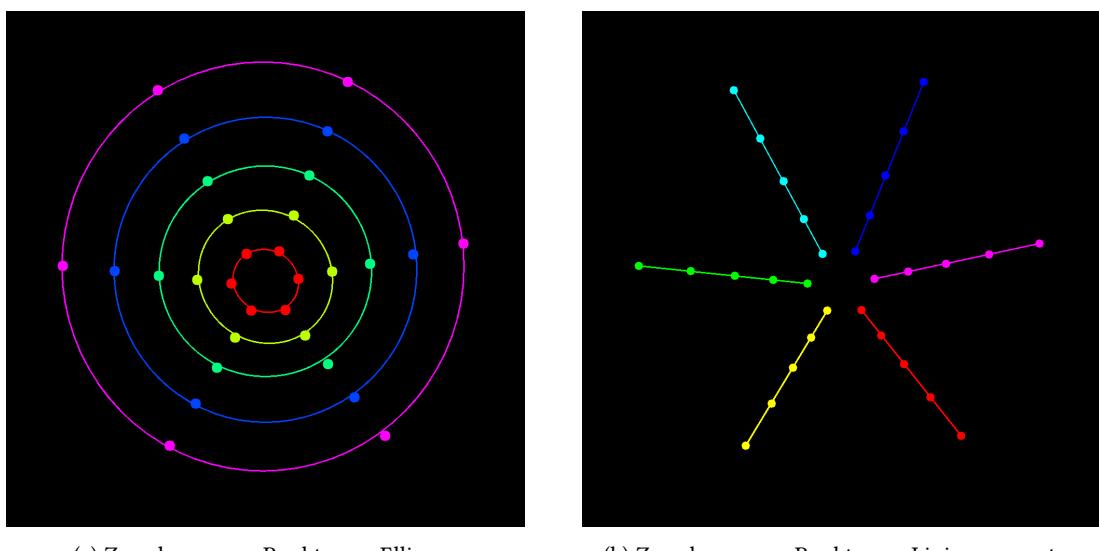


Abbildung 3.10.: Zuordnung von Punkten zu Ellipsen (links) und Liniensegmenten (rechts)

Abbildung 3.11.: Zuordnung von Punkten zu Ellipsen (links) und Liniensegmenten (rechts) farblich sortiert:
1. rot, 2. gelb, 3. grün, 4. blau, 5. magenta

3.6. Weltkoordinaten bestimmen

Wir können nun jedem Sample eine Kreislinie, sowie ein Liniensegment zuordnen. Damit wir Korrespondenzen zwischen den Samples und dem Kegel herstellen können, müssen wir im nächsten Schritt jedem Sample dreidimensionale Koordinaten auf der Kegeloberfläche zuweisen.

Mit Hilfe der Parametrisierung des Kegelstumpfs können wir die 3D-Koordinaten eines Samples folgendermaßen angeben:

Ohne Beschränkung der Allgemeinheit, seien die Ellipsen $i = 0, \dots, n - 1$ aufsteigend nach ihrer „Größe“² sortiert. Außerdem seien die Liniensemente $j = 0, \dots, m - 1$ aufsteigend nach Winkel mit der x -Achse, wie in Kapitel 3.5 beschrieben, sortiert. Eine Sample kann also eindeutig durch ein Tupel $(i, j) \in [0, n - 1] \times [0, m - 1]$ identifiziert werden und (x_{ij}, y_{ij}, z_{ij}) bezeichne seine Koordinaten im Weltkoordinatensystem.

²Etwas formaler, könnte man die Ellipsen hier nach ihrem Flächeninhalt sortieren. Für Ellipsen $E_0(x_0, y_0, a_0, b_0, \theta_0)$ und $E_1(x_1, y_1, a_1, b_1, \theta_1)$ gilt $E_0 \leq E_1$ g.d.w. $\pi \cdot a_0 \cdot b_0 \leq \pi \cdot a_1 \cdot b_1$

Analog zur parametrischen Darstellung von Kegelstümpfen (Gleichung 2.2) in Kapitel 2.1 ergibt sich:

$$x_{ij} = r_i \cos\theta_j$$

$$y_{ij} = h_i$$

$$z_{ij} = r_i \sin\theta_j$$

$\forall (i, j) \in [0, n - 1] \times [0, m - 1]$ mit

$$r_i = r + \frac{i}{n - 1} \cdot (R - r) \quad \forall i \in [0, n - 1]$$

$$h_i = \frac{i}{n - 1} \cdot \Delta H \quad \forall i \in [0, n - 1]$$

$$\theta_j = \frac{j}{m - 1} \cdot 2\pi \quad \forall j \in [0, m - 1]$$

3.7. Entfaltung

Die eigentliche Entfaltung des Kegels kann mit zwei unterschiedlichen Ansätzen realisiert werden. Die erste Möglichkeit ist die *Vorwärtsentfaltung*. Hierbei wird für jedes Pixel auf dem Kegelbild eine 3D-Koordinate durch geeignete Interpolation bestimmt und dann auf die Mantelfläche abgebildet. Beim zweiten Ansatz, der *Rückwärtsentfaltung*, wird ein Punkt von der Mantelfläche zurück auf den Kegel abgebildet und von dort mit einer Projektionsmatrix auf die Bildebene projiziert und dann interpoliert.

3.7.1. Vorwärtsentfaltung

Bei der *Vorwärtsentfaltung* muss wie oben erwähnt zu jedem Pixel die zugehörige 3D Koordinate im Weltkoordinatensystem berechnet werden. Da bisher jedoch nur die Positionen der Samples bekannt sind muss hier entsprechend interpoliert werden.

Zunächst betrachten wir diejenigen Pixel, die sich weder auf einer Kreislinie, noch auf einem Liniensegment befinden. Es gibt zu einem Pixel P also immer vier Sample-Nachbarn (b_l, b_r, t_r, t_l). Diese Situation ist in Abbildung 3.12 illustriert.

Nachdem die vier Nachbarn bestimmt wurden, können wir im ersten Schritt die Abstände d_1 und d_2 von Pixel P zu inneren Ellipse E_b , respektive äußeren Ellipse E_t berechnen. Mithilfe dieser Abständen kann nun eine *Interpolationsellipse* E_1 definiert werden als

$$E_{int} = \left(\frac{d_1}{d_1 + d_2} \right) \cdot E_t + \left(\frac{d_2}{d_1 + d_2} \right) E_b,$$

wobei eine Multiplikation mit einem Skalar alle Charakteristika einer Ellipse skaliert. Der Drehwinkel θ wird hierbei bei 2π umgebrochen. Eine Addition geschieht elementweise. Im nächsten Schritt wird der Schnittpunkt L mit dem Liniensegment $\overline{b_l t_l}$, sowie der Schnittpunkt R mit dem Liniensegment $\overline{b_r t_r}$ bestimmt.

Da sich t_l, L und b_l nun auf einem gemeinsamen Liniensegment befinden, kann bezüglich der Weltkoordinaten linear interpoliert werden. Analoges gilt für t_r, R und b_r . Die interpolierte Weltkoordinaten von L , sowie R bezeichnen wir als L_W und R_W

Die drei Punkte (L, P, R) befinden sich auf der Interpolationsellipse und somit können die Winkel (ϕ_L, ϕ_P, ϕ_R) bezüglich des gemeinsamen Ellipsenkoordinatensystems bestimmt werden.

Analog zu oben werden die 3D-Koordinaten von P als lineare Interpolation zwischen den bestimmten 3D-Koordinaten von L und R bestimmt. Als Interpolationsfaktor benutzen wir $\frac{\phi_L - \phi_P}{\phi_L - \phi_R}$. Wir erhalten für die Weltkoordinaten P_W des Pixels P :

$$P_W = \left(\frac{\phi_L - \phi_P}{\phi_L - \phi_R} \right) R_W + \left(1 - \frac{\phi_L - \phi_P}{\phi_L - \phi_R} \right) L_W$$

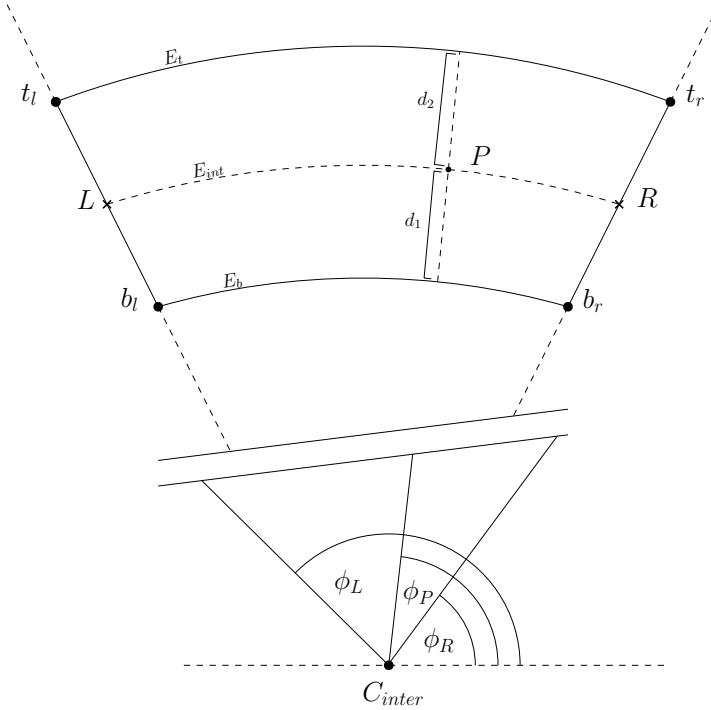


Abbildung 3.12.: Interpolation der 3D-Koordinaten mittels vier Sample-Nachbarn

Analog werden die auf Liniensegmenten befindlichen Punkte einfach linear interpoliert. Die Punkte, die sich auf Kreislinien befinden werden über Winkel interpoliert.

Jedes Pixel hat nun 3D-Koordinaten im Kegel erhalten. Mit Hilfe der konstruierten Abbildung 2.6 aus Kapitel 2.1, können wir nun die interpolierten Punkte auf die Mantelfläche abbilden und erhalten schließlich das entzerrte Bild (siehe Abbildung 3.13)

3.7.2. Rückwärtsentfaltung

Bei der Rückwärtsentfaltung gehen wir von dem entzerrten Bild aus, dessen geometrische Eigenschaften aus der parametrischen Form der Mantelfläche des Kegels bekannt sind und versuchen rückwärts Pixelkandidaten im Ursprungsbild zu bestimmen.

Wir müssen zunächst für einen Punkt auf dem entzerrten Bild dreidimensionale Koordinaten auf der Kegeloberfläche berechnen. Wir nutzen dafür die konstruierte Umkehrabbildung 2.9 aus Kapitel 2.1. Im nächsten Schritt müssen wir die erhaltenen Kegelkoordinaten auf das Ursprungsbild abbilden, um die Pixelkandidaten bestimmen zu können.

Wir konstruieren dafür eine Projektionsmatrix. Wie in Kapitel 2.3 nutzen wir die Punktkorrespondenzen der Samples aus, um mittels *Direct Linear Transformation* eine Projektionsmatrix zu bestimmen. Wir erhalten

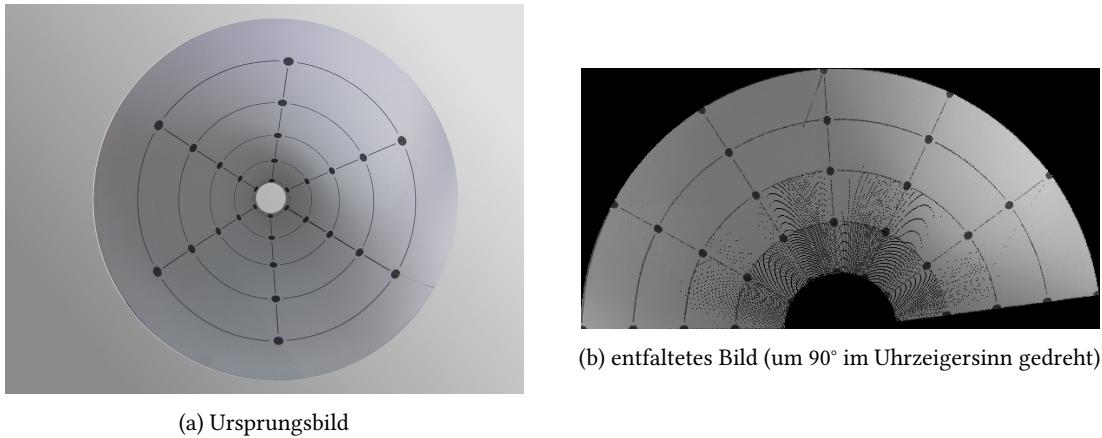


Abbildung 3.13.: Vorwärtsentfaltung

somit die gewünschte Abbildung, die wir auf jede 3D-Koordinate anwenden, wodurch wir Bildkoordinaten erhalten. Im Allgemeinen, sind diese Koordinaten nicht ganzzahlig und wir müssen entsprechen interpolieren.

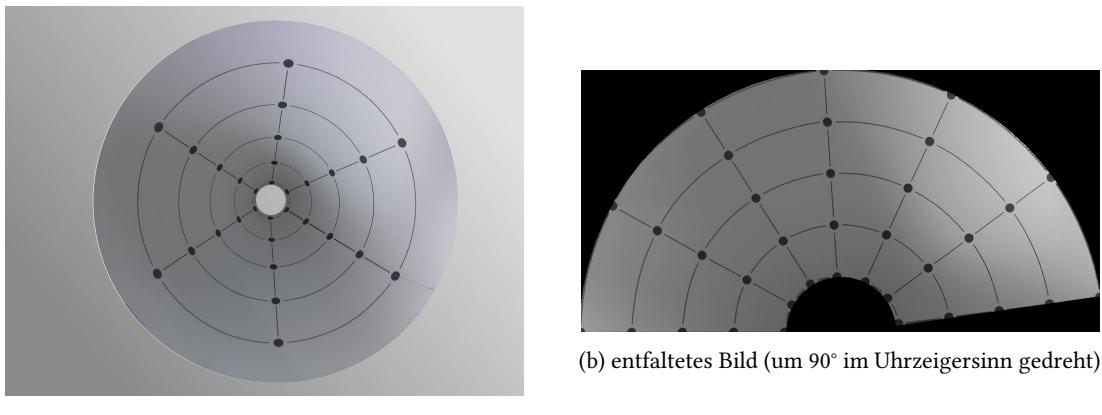


Abbildung 3.14.: Rückwärtsentfaltung

4. Implementierung

Zwecks Vereinfachung des Kalibrierungsprozesses wurde ein Assistent in Form einer graphischen Benutzeroberfläche implementiert.

Im ersten Schritt des Assistenten wird dabei optional, wie in Abbildung 4.1 zu sehen, eine intrinsische Kamerakalibrierung durchgeführt.

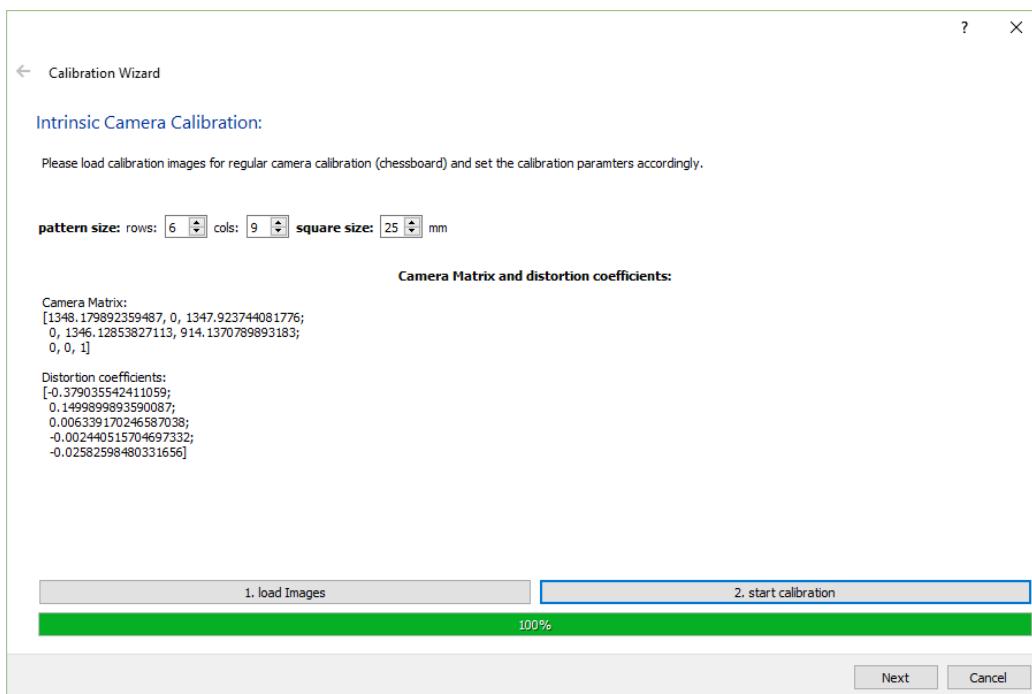


Abbildung 4.1.: Kalibrierungsassistent: intrinsische Kamerakalibrierung

Anschließend wird die eigentliche Kegelkalibrierung durchgeführt. Dabei wird zunächst das Kalibrierungsbild geladen und gegebenenfalls nach einer stattgefundenen intrinsischen Kalibrierung entzerrt. Es werden nun die Sample-Positionen detektiert und der Nutzer hat die Möglichkeit zu überprüfen, ob alle Positionen korrekt detektiert wurden und andernfalls fehlerhafte Punkte zu entfernen und / oder Punkte hinzuzufügen. Im Anschluss werden die Ellipsen und Liniensegmente bestimmt und Punktcorrespondenzen hergestellt (siehe Abbildung 4.2).

Im letzten Schritt kann zwischen beiden Entfaltungsverfahren gewählt werden. Anschließend können die Einstellungen in eine XML-Datei exportiert werden, in der, neben der Kamera-Matrix und Verzerrungskoeffizienten, auch die zwei Abbildungsmatrizen des ausgewählten Verfahrens gespeichert werden.

Die Abbildungsmatrizen sind dabei wie folgt aufgebaut. Als *dst* bezeichnen wir das entfaltete Bild. *src* ist das Ursprungsbild.

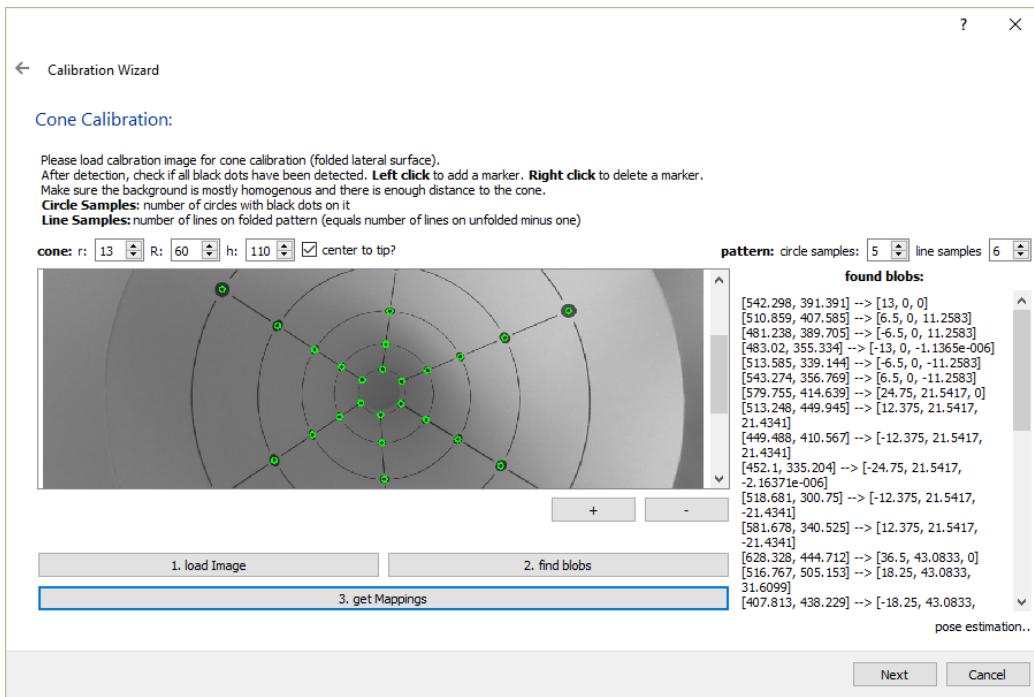


Abbildung 4.2.: Kalibrierungsassistent: Kegelkalibrierung

Bei der Vowärtsentfaltung gilt:

$$dst(map_x(x, y), map_y(x, y)) = src(x, y),$$

wobei map_x und map_y die Abbildungsmatrizen sind und die gleiche Größe wie das Ursprungsbild haben. Für ein Pixel (x, y) auf dem Ursprungsbild src geben die Matrizen map_x , sowie map_y an, auf welche Position ein das Pixel auf dem entzerrten Bild abgebildet wird. $map_x(x, y)$ bestimmt dabei die x -Koordinate, $map_y(x, y)$ die y -Koordinate. Da wir bei der Entfaltung die Größe des Ergebnisbildes benötigen, und wir diese bei der Erstellung der Abbildungsmatrizen berechnet haben, sind Breite und Höhe, in $map_x(0, 0)$, respektive $map_y(0, 0)$ kodiert. Für das Pixel an der Position $(0, 0)$ auf dem Ursprungsbild ist anschließend keine Auswertung der Matrizen mehr möglich. Dieses Pixel ist jedoch ohnehin nicht Teil des Kalibrierungsmusters.

Bei der Rückwärtsentfaltung gilt:

$$dst(x, y) = src((map_x(x, y), map_y(x, y)),$$

wobei map_x und map_y wieder die Abbildungsmatrizen sind, die analog zur Vowärtsentfaltung aufgebaut sind. Der grundlegende Unterschied ist hier, dass die wir von einem Pixel auf dem entzerrten Bild auf eine Position auf dem Ursprungsbild schließen. Die Abbildungsmatrizen haben hier die gleiche Größe wie das entfaltete Bild. Eine Kodierung wie bei der Vowärtsentfaltung ist hier also nicht notwendig.

5. Analyse

Im folgenden Kapitel vergleichen wir beide, in Kapitel 3, vorgestellten Verfahren zur Entzerrung. Anschließend gehen wir auf verschiedene Einflussfaktoren auf die Qualität und Präzision der Entzerrung ein, woraufhin wir die Laufzeiten analysieren. Im letzten Teil des Kapitels evaluieren wir beide Ansätze zur Ellipsendetektion.

Die Laufzeitanalysen wurden dabei auf einem **Raspberry Pi 2 Model B** mit folgenden Eigenschaften

- 900MHz quad-core ARM Cortex-A7 CPU
- 1GB RAM
- GCC-4.9.2
- OpenCV 2.4.13

durchgeführt.

5.1. Vergleich Vorwärtsentfaltung und Rückwärtsentfaltung

Untersucht man die interpolierten Weltkoordinaten bei der Vorwärtsentfaltung, stellt man fest, dass die Oberfläche des Kegels, wie in Abbildung 5.1 zu sehen, eckig scheint. Zur Veranschaulichung wird hierbei nur ein Bruchteil der interpolierten Werte dargestellt. Zwischen den Samples auf einer Kreislinie sollten die interpolierte Positionen nach außen gewölbt sein, da ein Kegel rund ist. Da wir jedoch interpolieren und nicht extrapoliieren, gehen die Werte nie über die zur Interpolation genutzten Werte hinaus. Konkreter kann eine interpolierte Position keine größeren oder kleineren x - und y -Koordinaten erhalten, als die der genutzten Samples. Diese wäre jedoch notwendig, sodass die Rundung des Kegels erhalten bleibt.

Darüber hinaus zeigt sich ein weiteres Problem nach der Entfaltung. Die Entfaltung geschieht über die Abbildung 2.6, die in Kapitel 2.1 konstruiert wurde. Damit im entfalteten Bild nicht 1mm einem Pixel entspricht, müssen die erhaltenen Werte skaliert werden. Wir wollen dabei entsprechend einer gewünschten Auflösung der entzerrten Seitenhöhe skalieren. Auch schon bei kleineren Skalierungen entstehen auffällige Risse im entfalteten Bild, die in Abbildung 5.2 exemplarisch zu sehen sind. Grund dafür ist, dass jedes Pixel aus dem Ursprungsbild auf eine 2D-Koordinate der Mantelfläche abgebildet wird. Auch nach einer Skalierung handelt es sich bei diesen Werten im Allgemeinen nicht um ganzzahlige Werte. Es entstehen Rundungsfehler. Auch wenn Rundungsfehler nicht entstanden, fehlte es auf Grund der starken Stauchung, in den inneren, kleineren Regionen (siehe Abbildung 5.2 in rot) des Ursprungsbild an genügend Bildinformation. Solche Risse, beziehungsweise nicht erreichte Pixel auf dem entfalteten Bild, bezeichnen wir als Defekte.

Da wir von dem Ursprungsbild aus auf das entfaltete Bild abbilden, ist außerdem eine Interpolation auf dem Ursprungsbild nicht möglich, da wir vorher nicht wissen, wo Defekte entstehen. Man müsste also entweder auf dem resultierendem Bild interpolieren (siehe Kapitel 6), oder zu gegebenen Defekten über die Umkehrabbildung auf Pixelkandidaten im Ursprungsbild schließen und dann entsprechend interpolieren. Es bietet sich dann jedoch an, direkt die Umkehrabbildung zu nutzen, was die Motivation hinter der Rückwärtsentfaltung ist.

Wir möchten untersuchen wie sich die Anzahl der Defekte bei einer Änderung der Ausgabeauflösung verhält. Diese Beziehung lässt sich in Abbildung 5.3 ablesen.

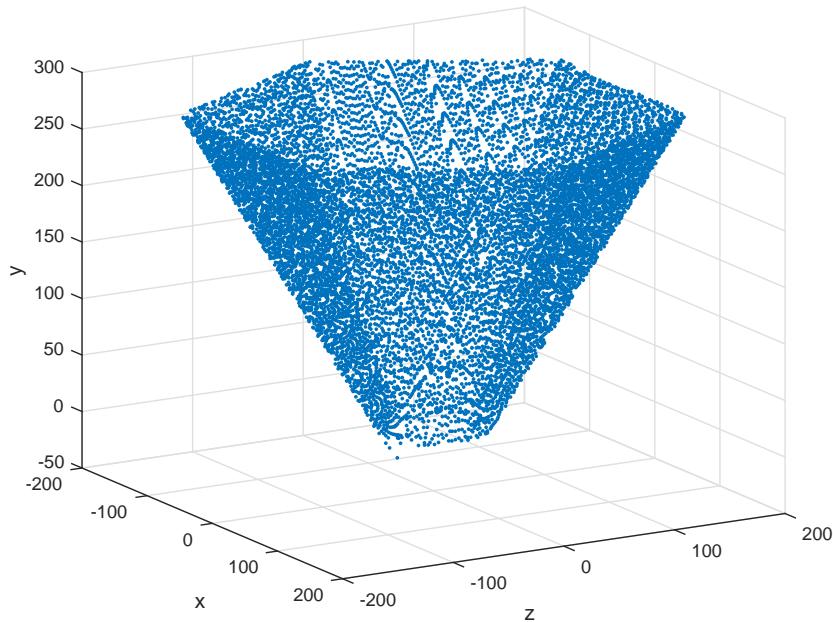


Abbildung 5.1.: interpolierter Kegel

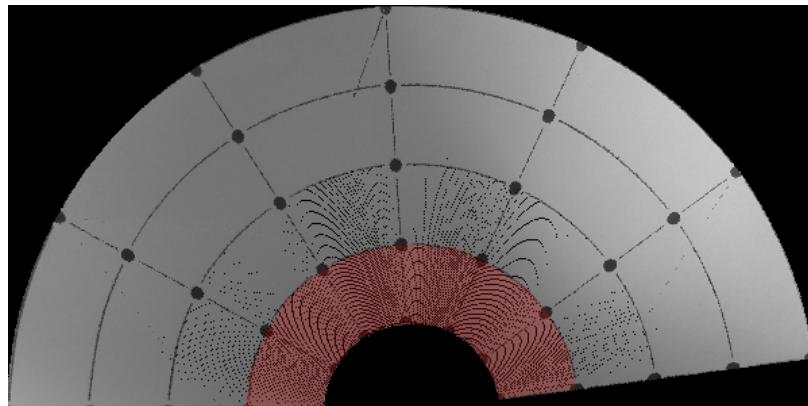


Abbildung 5.2.: Vorwärtsentfaltung

Wie erwartet verhält sich die Anzahl der Defekte quadratisch zur gewählten Ausgabeauflösung. Der Informationsgehalt (Pixelanzahl) des Ursprungsbildes bleibt trotz erhöhter Ausgabeauflösung konstant. Wie in Abbildung 5.4 zu sehen, verhält sich die Ausgabeauflösung quadratisch zur Gesamtzahl der Pixel für ein konstantes α . Für die Breite b des Ausgabebildes gilt $b = res_S$. Die Höhe h setzt sich wie folgt zusammen: $h = res_S + res_S \cdot \cos(2\pi - \alpha) = res_S \cdot (1 + \cos(2\pi - \alpha)) = res_S \cdot (1 + \cos \alpha)$. Dazu betrachten wir das rechtwinklige Dreieck in der rechten oberen Ecke der Abbildung (in magenta). Die Hypotenuse des Dreiecks beträgt res_S und mit $\beta = 2\pi - \alpha$ ergibt sich für die Ankathete eine Länge von $res_S \cdot \cos(2\pi - \alpha)$. Die Gesamtzahl der Pixel beträgt demnach $b \cdot h = res_S^2(1 + \cos(\alpha))$.

Obwohl die Laufzeit der Kalibrierung nicht im Vordergrund steht, vergleichen wir die Laufzeit beider Verfahren bei verschiedenen Auflösungen der Seitenhöhe. Da der Großteil des Kalibrierungsprozesses bei beiden

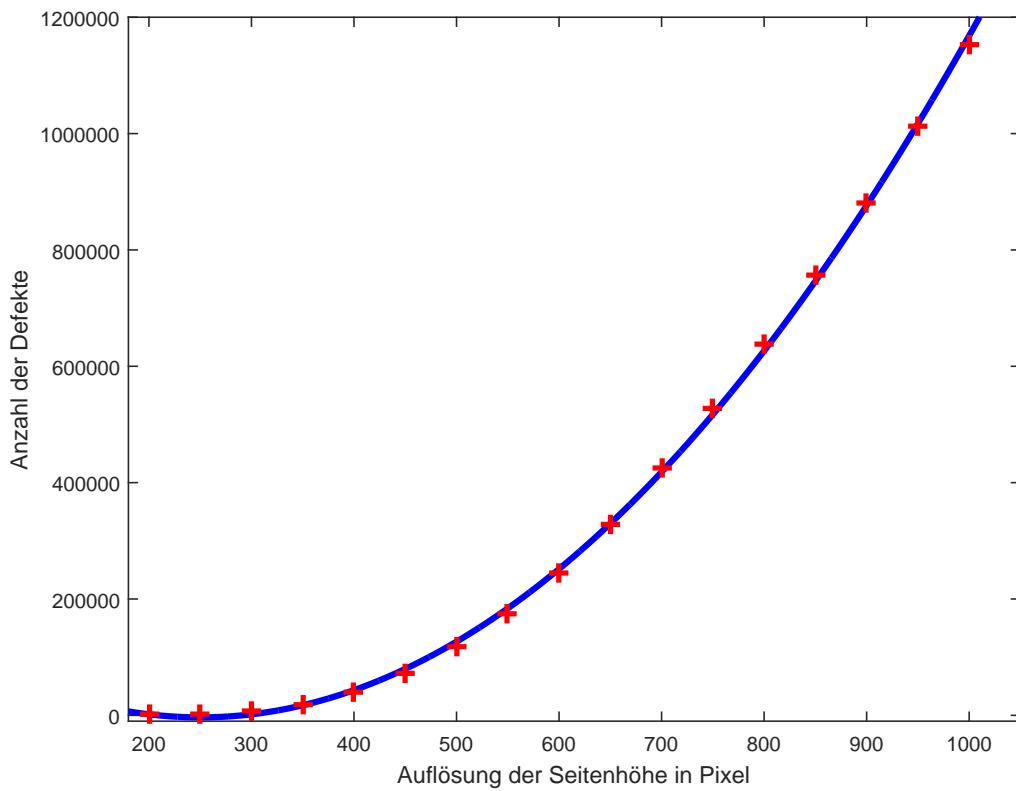


Abbildung 5.3.: Einfluss der Ausgabeauflösung auf die Anzahl der Defekte

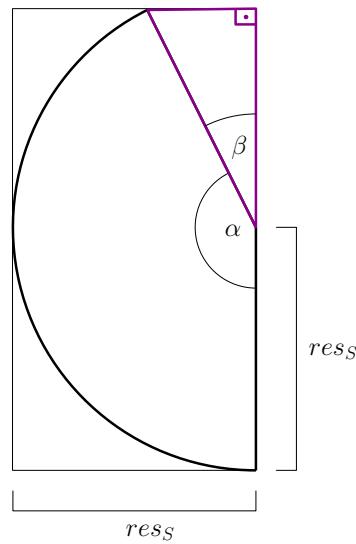


Abbildung 5.4.: Größe des Ausgabebildes bei einer Seitenhöhe von $ress_S$

Verfahren identisch ist, vergleichen wir nun den Teil, der sich unterscheidet. Das heißt wir messen bei der

Vorwärtsentfaltung die Laufzeit für die Interpolation der Weltkoordinaten und die anschließende Abbildung auf die Mantelfläche, und bei der Rückwärtsentfaltung die Laufzeit für die Abbildung auf die Kegelkoordinaten, die Bestimmung der Projektionsmatrix und die anschließende Abbildung auf die Bildkoordinaten, inklusive Interpolation.

Wie in Abbildung 5.5 zu sehen, ist der Aufwand bei der Vorwärtsentfaltung weitestgehend konstant. Dies da dadurch zu begründen, dass sich die Eingabeauflösung nicht ändert. Der Interpolationsaufwand bleibt gleich und es wird anschließend die gleiche Anzahl Pixel auf die Mantelfläche abgebildet. Nur der Ort der Projektion wird entsprechend der Ausgabeauflösung skaliert. Bei der Rückwärtsentfaltung hingegen, wächst der Interpolationsaufwand quadratisch. Wir gehen dort von dem Ausgabebild aus, das quadratisch mit der Auflösung wächst und bilden dort jeden Punkt auf das Ursprungsbild ab. Je mehr Pixel auf dem Ausgabebild, desto häufiger muss interpoliert werden. Bei einer Auflösung der Seitenhöhe von 850 Pixeln ist die Laufzeit bei beiden Verfahren gleich.

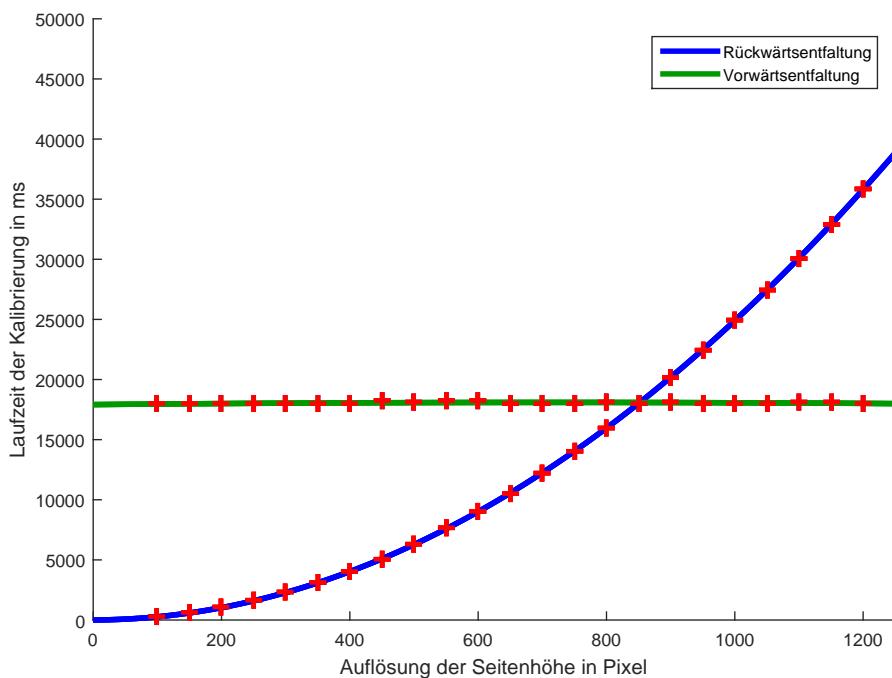


Abbildung 5.5.: Laufzeitvergleich zwischen Vorwärtsentfaltung (blau) und Rückwärtsentfaltung (rot)

Als Reprojektionsfehler einer Abbildung wird die Distanz zwischen einem gemessenen Punkt und einem korrespondierendem (durch die Abbildung) projizierten Punkt bezeichnet.

Im Falle der Rückwärtsentfaltung sind die gemessenen Punkte die Bildpositionen der Samples im Ursprungsbild. Da die Geometrie des Kegels bekannt ist, wissen wir wo sich die Samples auf dem entfalteten Bild befinden müssen (siehe Parametrisierung der Mantelfläche 2.3 in Kapitel 2). Wir können nun diese Positionen mit Hilfe der Abbildung zur Entfaltung und der Projektionsmatrix zurück auf das Ursprungsbild abbilden. Diese Punkte sind die projizierten Punkte. Da die Abbildung von der Mantelfläche zur Kegeloberfläche exakt ist, ist der Reprojektionsfehler der Rückwärtsentfaltung alleine durch die Projektionsmatrix bestimmt.

Bei der Vorwärtsentfaltung sind die gemessenen Punkte gegeben durch die bekannten Sample-Positionen auf der Mantelfläche. Die projizierten Punkte erhält man, nach der Abbildung der detektierten Sample-Positionen

des Ursprungsbild auf die Mantelfläche. Der Reprojektionsfehler ist also alleine durch die Genaugkeit der Sample-Detektion definiert und somit bei diesem Verfahren immer nahe null. Obwohl das Endergebnis, bedingt durch die Defekte, bei der Vorwärtsentfaltung optisch schlechter ist, ist der Reprojektionsfehler also bei der Vorwärtsentfaltung immer kleiner. Als Vergleich zwischen den beiden Verfahren eignet sich der Reprojektionsfehler also nicht.

Auf Grund der fehlenden Defekte bei der Rückwärtsentfaltung und der besseren Laufzeit bei Auflösungen unter 850 Pixeln, beziehen sich alle folgenden Auswertungen nur noch auf die Rückwärtsentfaltung.

5.2. Einfluss der intrinsischen Kalibrierung

Ein wichtiger Einflussfaktor auf die Qualität der Entzerrung ist die intrinsische Kamerakalibrierung, die vor der eigentlichen Kegelkalibrierung stattfindet. Ihre Hauptaufgabe besteht darin, die Linsenverzerrungen der Kamera herauszurechnen.

Wir messen den Einfluss der Kamerakalibrierung mit Hilfe des Reprojektionsfehlers. Dazu betrachten wir fünf verschiedene Bilder, die ein Mal mit und ein Mal ohne intrinsische Kalibrierung entzerrt werden. Bei beiden Gruppen wird anschließend jeweils der Reprojektionsfehler pro Sample bestimmt und verglichen. Die Ergebnisse sind in Abbildung 5.6 zu sehen. In der linken Abbildung sind hierbei die Fehler im kalibrierten Fall zu erkennen, im Linken im Unkalibrierten. In der Abbildung ist dabei ein Kreuz an der Stelle (u, v) , falls die Abweichung des projizierten Punktes in x -Richtung u , sowie in y -Richtung v beträgt.

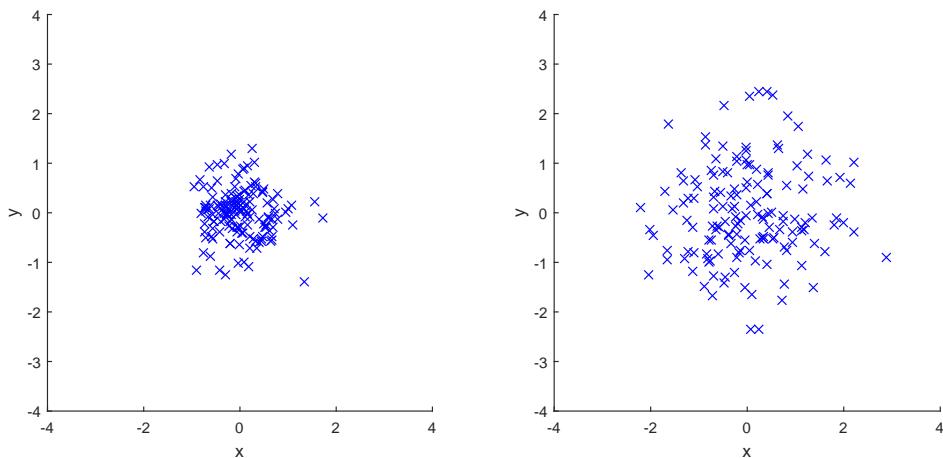


Abbildung 5.6.: Einfluss der intrinsischen Kalibrierung auf den Reprojektionsfehler

Es ist klar zu erkennen, dass der Reprojektionsfehler bei den Bildern ohne intrinsische Kamerakalibrierung wesentlich größer ist. Der starke Einfluss kommt unter Anderem daher, dass wir eine Weitwinkelkamera mit starker tonnenförmiger Verzerrung eingesetzt haben. Ohne eine Modellierung der Linsenverzerrungen weichen die Abstände zwischen den Sample-Positionen stark von der Realität ab. Die Projektionsmatrix wird mit fehlerhaften Daten bestimmt. Die eigentlichen Sample-Positionen auf dem entzerrten Bild, werden also nicht korrekt auf die Sample-Positionen im Ursprungsbild abgebildet.

5.3. Einfluss der Rotation der Kamera

Um den Einfluss der Rotation der Kamera untersuchen zu können, wurde der Kegel mit Kalibrierungsmuster gerendert, da die Kameraposition dann exakt bekannt ist und äußere Faktoren wie Lichtverhältnisse und inhomogene Hintergründe kontrolliert werden können. In Abbildung ?? ist bei zwei unterschiedlichen Winkeln der gerenderte Kegel zu sehen.

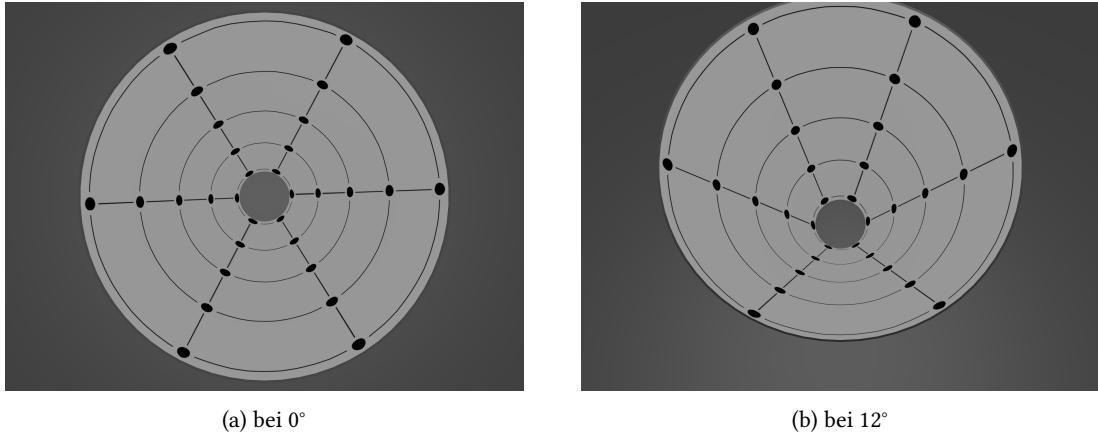


Abbildung 5.7.: gerendeter Kegel mit Kalibrierungsmuster in Blender aus verschiedenen Blickrichtungen

Es werden anschließend Bilder erzeugt, in denen in 1° Schritten die Kamera von 0° bis 12° um die x -Achse rotiert wird. Für jedes dieser Bilder wird anschließend eine Rückwärtsentfaltung durchgeführt und der durchschnittliche Reprojektionsfehler ermittelt. Abbildung 5.8 zeigt, dass der Reprojektionsfehler relativ rotationsinvariant ist bei einem durchschnittlichen . Ab einem Winkel von 13° , werden die Samples teilweise vom Blob-Detektor nicht mehr erkannt (bei 13° und 14° werden zwei nicht erkannt, bei 15° und 16° jeweils vier) und mussten per Hand markiert werden. Ab einem Winkel von 16° sind die Abstände zwischen den Kreislinien so klein, dass die Ellipsendetektion fehlschlägt.

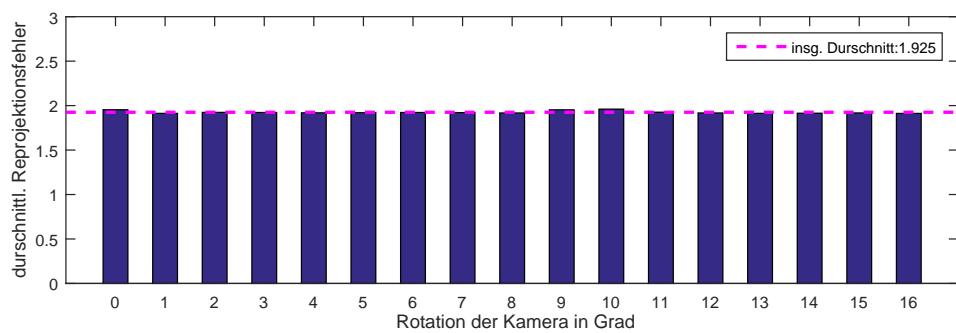


Abbildung 5.8.: Einfluss der Rotation der Kamera

5.4. Laufzeit der Entfaltung

Durch den Kalibrierungsprozess erhalten wir zwei Abbildungsmatrizen (siehe Kapitel 4), mit denen wir anschließend eine Reihe von Bildern entzerren möchten. Wir untersuchen dabei die Laufzeit pro gewählter

Auflösung, in 50er Schritten, und ermitteln jeweils die Laufzeit gemittelt über 200 Bilder. Untersucht wird die Laufzeit mit bikubischer, sowie mit (bi-) linearer Interpolation.

Im Gegensatz zu linearer Interpolation, werden bei bikubischer Interpolation , 16 Nachbarn statt vier betrachtet und mittels (stückweise stetiger) Polynome interpoliert. Der Aufwand bei der bikubischen Interpolation ist wesentlich größer, es entstehen aber weniger Artefakte [Key81].

Wie in Abbildung 5.9 zu sehen, wächst die Laufzeit bei beiden Verfahren quadratisch mit der Auflösung der Seitenhöhe. Die Laufzeit des Verfahrens mittels bikubischer Interpolation wächst dabei um einen Faktor von 1.35 schneller, als das Verfahren mittels linearer Interpolation.

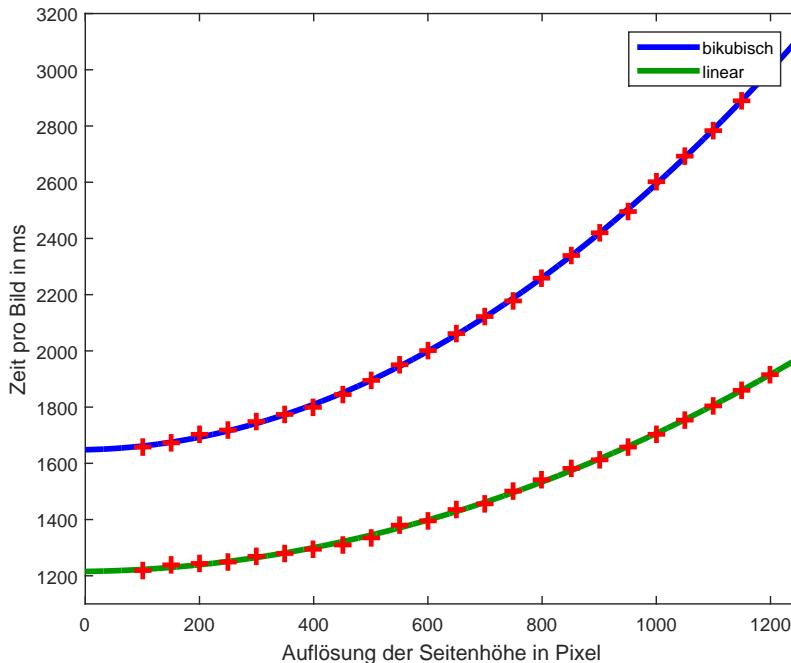


Abbildung 5.9.: Einfluss der Ausgabeauflösung auf die Laufzeit der Entfaltung

5.5. Evaluierung der Ellipsendetektion mittels Deformable Templates

Das Problem bei der Ellipsendetektion mittels Deformable Templates ist, dass die Energiefunktion, auch nach starker Glättung des Kanten- und Kantenrichtungsbilds, schnell in kleine lokale Minima läuft, obwohl ein wesentlich stärkeres Minimum in näherer Umgebung wäre.

In Abbildung 5.10 sieht man beispielhaft einen Ausschnitt der Energiefunktion, wobei in diesem Fall zwecks Veranschaulichung nur die Haupt- und Nebenachse der Ellipse variable sind. Das Zentrum bleibt das Zentrum des Bildes und der Winkel θ ist konstant null. Trotz dieser Einschränkungen lässt sich gut erkennen, dass circa bei $a = b = 350$ ein starkes Minimum liegt. Es gibt jedoch neben diesem Minimum in näherer Umgebung kleine irrelevante Minima, wie beispielsweise in (350, 260). Solche „Becken“ (in rot markiert) erschweren die Detektion erheblich.

Darüber hinaus muss für ein robustes und effizientes Optimierungsverfahren der Gradient und im besten Fall sogar die Hesse-Matrix der zu minimierenden Funktion zur Verfügung gestellt werden. Jede Ableitung des Kanten- und Kantenorientierungsbilds bringt Unstetigkeitsstellen mit sich. Mit einer zu starke Glättung, geht jedoch viel Informationsverlust einher.

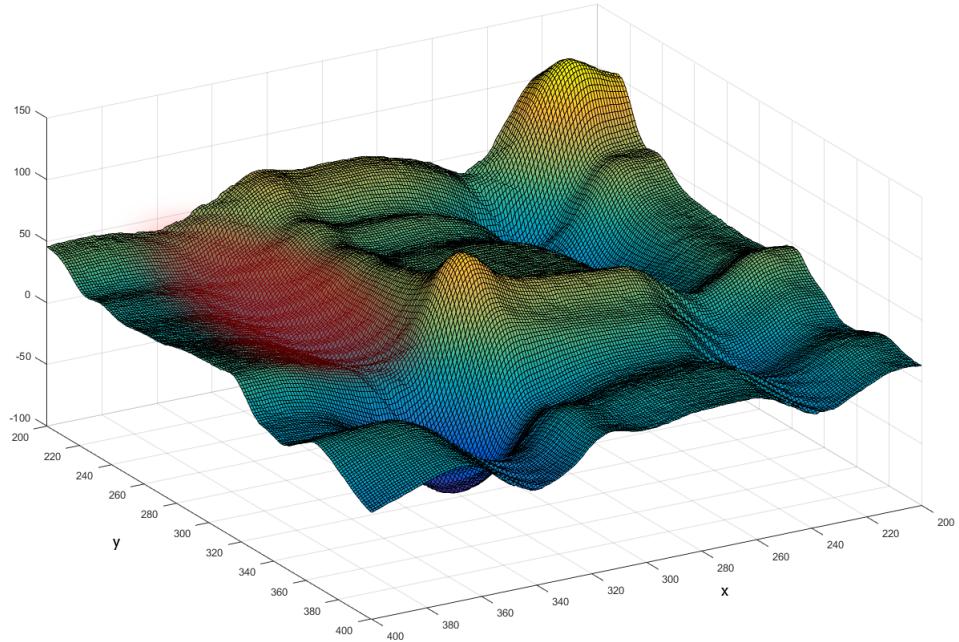


Abbildung 5.10.: Deformable Templates: Ausschnitt der Energiefunktion für variable Haupt- und Nebenachse

5.6. Evaluierung der Ellipsendetektion mittels RANSAC

Die robuste Ellipsendetektion ist der wichtigste Schritt bei der Entzerrung. Bei beiden Verfahren werden die bestimmten Ellipsen genutzt um Korrespondenzen zwischen den Sample-Positionen und Punkten auf dem Kegel im Weltkoordinatensystem herzustellen. Bei der Vorwärtsentfaltung werden die Ellipsen darüber hinaus benötigt, um für die Pixel geeignete 3D-Koordinaten interpolieren zu können. Es ist also von großer Bedeutung wie gut die Ellipsendetektion mittels RANSAC funktioniert.

Wir betrachten zunächst, wie sich eine Gleichverteilung von Ausreißern auf die Schätzung der Ellipse auswirkt. Wie in Abbildung 5.11(a) erzeugen wird dazu zunächst eine Ellipse und bestimmen 500 gleichverteilte Punkte auf der Ellipse. Anschließend erzeugen wir auf dem gesamten Bild gleichverteilt zu einem gegebenen Fehleranteil Ausreißer. Wir inkrementieren den Fehleranteil schrittweise von 0% bis 60% und vergleichen die Schätzungen durch Ransac und der Methoden der kleinsten Quadrate mit der eigentlichen Ellipse. Als Fehlermaß benutzen wir den Dice-Koeffizienten [Dic45]:

$$D = \frac{2|A \cap B|}{|A| + |B|} \in [0, 1]$$

für zwei Mengen A und B mit einem Wert von eins für eine perfekte Übereinstimmung ($A = B$) und einem Wert von null, falls $A \cap B = \emptyset$. Wir betrachten hierbei die Flächen zweier Ellipsen beziehungsweise die Schnittfläche. Da die Berechnung der Schnittfläche zweier Ellipsen analytisch sehr aufwändig ist [Ebe08], berechnen wir stattdessen jeweils die Menge der gemeinsamen Pixel zweier auf ein Bild gezeichneter Ellipsen.

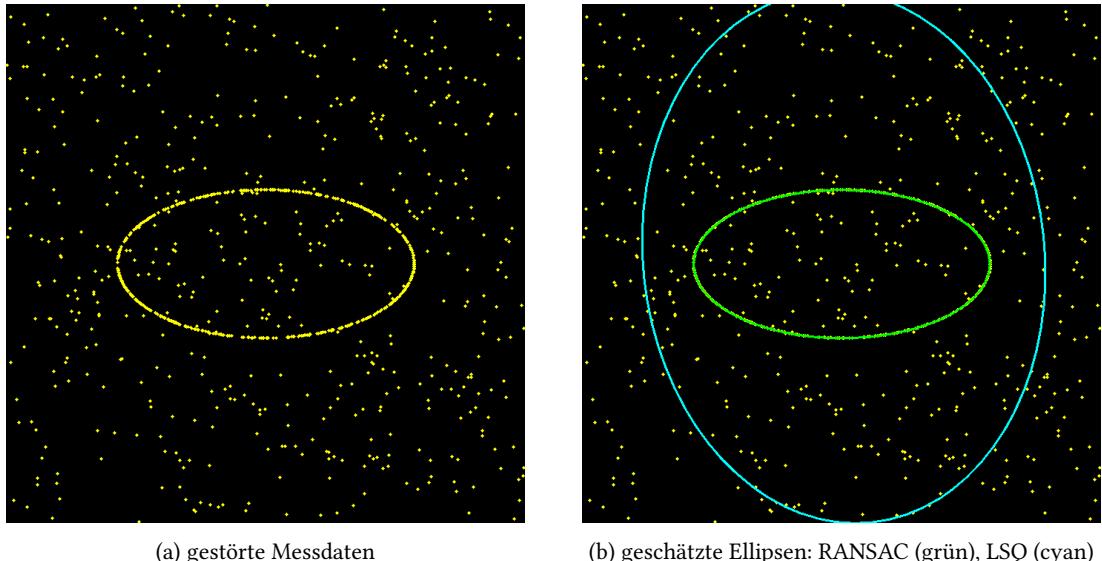


Abbildung 5.11.: Vergleich RANSAC und LSQ bei gleichverteilten Ausreißern $\epsilon = 0.5, p = 0.99$

Die Anzahl der Iterationen für RANSAC kann in jedem Schritt mit Hilfe der Formel aus Kapitel 2.6 berechnet werden.

In Abbildung 5.12 sieht man, dass der Dice-Koeffizient bei RANSAC konstant nahezu bei eins liegt. Das heißt, dass mittels RANSAC auch bei einem Fehleranteil von 60% immer noch mit sehr großer Genauigkeit die Ellipse korrekt geschätzt wird. Bei der Methode der kleinsten Quadrate weicht die geschätzte Ellipse jedoch schon bei einem kleinen relativen Fehleranteil von 10% stark von der korrekten Ellipse ab.

In unserem Verfahren ist eine Gleichverteilung der Ausreißer unwahrscheinlich. Viel wahrscheinlicher ist es, dass eine der nächst äußeren Ellipsen frühzeitig sichtbar wird (siehe Kapitel 3.4). Dieser Fall ist in Abbildung 5.13 illustriert. Wir wollen nun untersuchen, wie stark solche Ellipsen das korrekte Ergebnis beeinflussen. Wir gehen dabei analog zum dem Verfahren zur Untersuchung von gleichverteilten Ausreißern vor. Ein Fehleranteil von 50% bedeutet hier, dass auf der inneren Ellipse genauso viele Punkten sind, wie auf der äußeren.

In Abbildung 5.14 sieht man, dass bei der Schätzung mittels RANSAC die Dice-Koeffizienten bis circa 50% nahezu bei eins liegt. Bei 50% ist jedoch ein Sprung auf einen Wert von circa 0.6. Dies liegt daran, dass ab 50% auf der äußeren Ellipse mehr Punkte liegen, als auf der inneren. Es ist also wahrscheinlicher bei einer zufälligen Auswahl, Punkte der äußeren Ellipse zu erhalten. Die Koeffizienten der Schätzung mittels der Methode der kleinsten Quadrate verhalten sich ähnlich wie im Fall der gleichverteilten Ausreißer. Würde man den Ausreißeranteil weiter erhöhen, würde der Dice-Koeffizient hier gegen den gleichen Wert wie bei RANSAC konvergieren. Der Wert ist durch die Entfernung der äußeren Ellipse zur inneren Ellipse definiert. Erhöht man diesen, verringert sich erreichte Dice-Koeffizient.

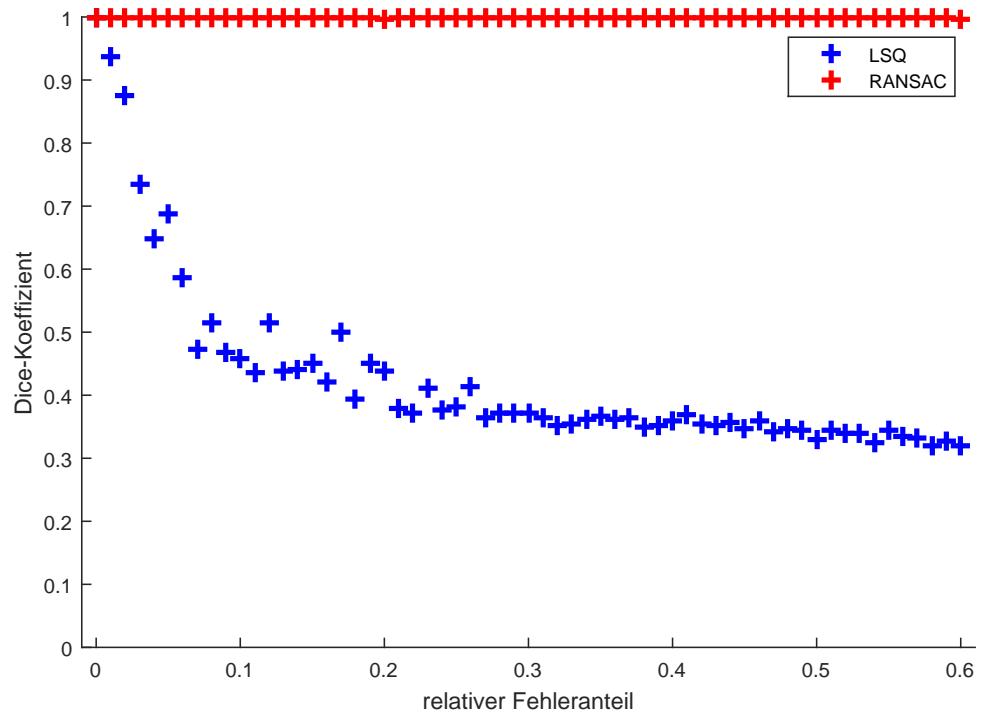


Abbildung 5.12.: Dice-Koeffizienten bei einer Schätzung mittels RANSAC und der Methode der kleinsten Quadrate (LSQ) bei gleichverteilten Ausreißern

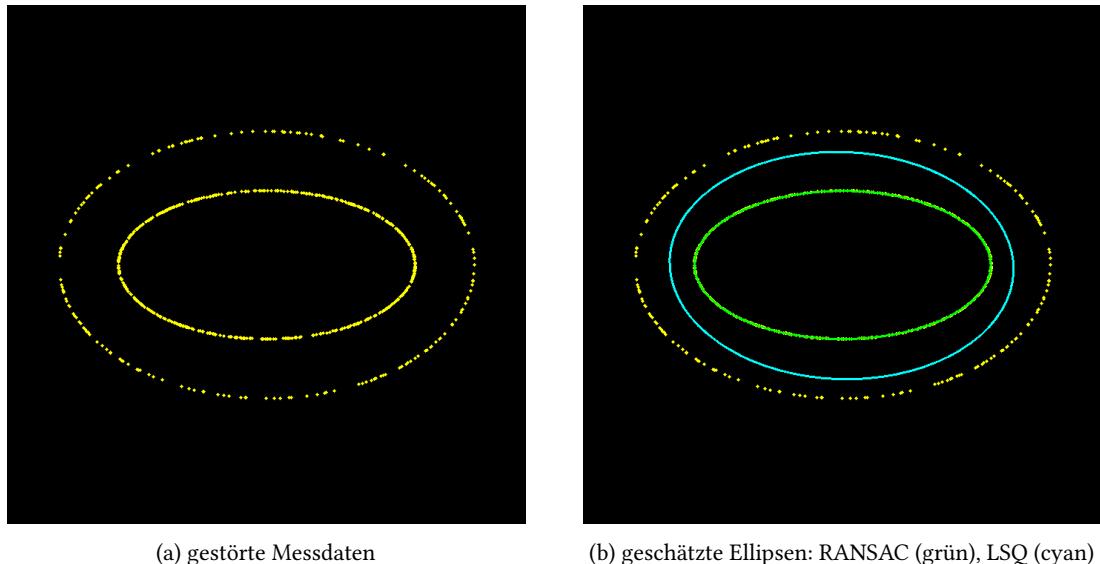


Abbildung 5.13.: Vergleich RANSAC und LSQ bei bei Ausreißern auf einer äußeren Ellipse mit $p = 0.99$ und $\epsilon = 0.33$

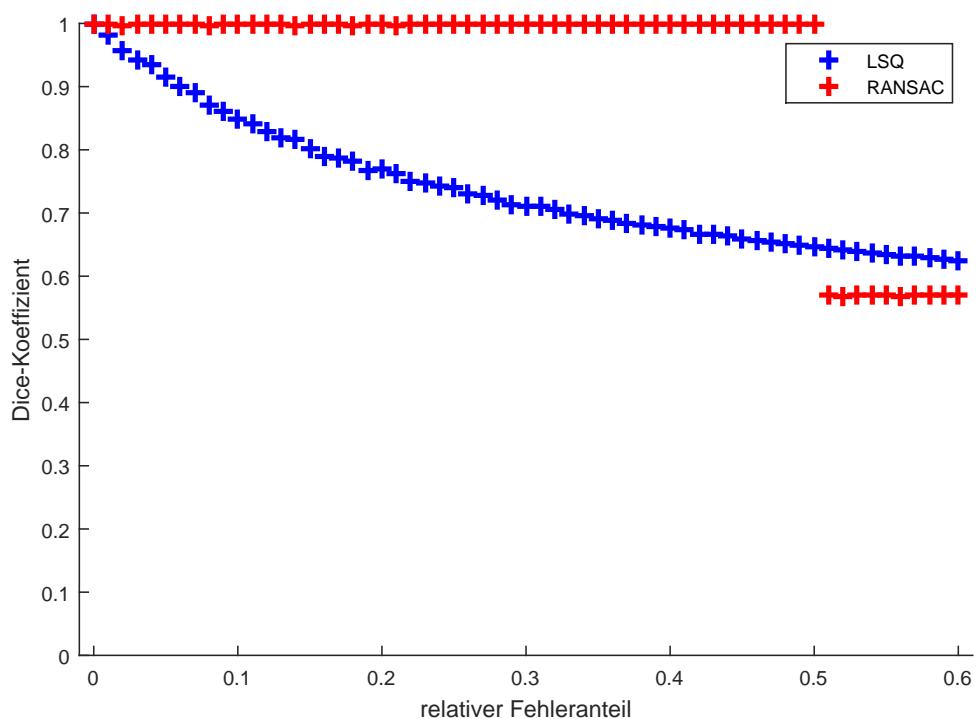


Abbildung 5.14.: Dice-Koeffizienten bei einer Schätzung mittels RANSAC und der Methode der kleinsten Quadrate (LSQ) bei Ausreißern auf einer äußeren Ellipse

6. Fazit und Ausblick

In der vorliegenden Arbeit haben wir zwei Verfahren vorgestellt mit denen Kegeloberflächen entzerrt werden können. Bla bla hier werden einige Verbesserungen vorgeschlagen...

6.1. Parallelisierung

Da der Kalibrierungsprozess nur ein Mal vor der eigentlichen Entzerrung durchgeführt wird, bietet sich eine Parallelisierung hier deshalb nur bedingt an. Dennoch könnte man hier besonders bei den komplexen Aufgaben, wie der Liniendetektion [Hav+14] und anschließenden Schnittpunktberechnung, sowie der Ellipsendetektion auf parallele Methoden zurückgreifen. Bei der Ellipsendetektion mittels RANSAC werden für eine vorher definierte Anzahl an Iterationen, wiederholt Punkte ausgewählt mit denen anschließend eine Ellipse geschätzt wird. Es bietet sich hier an, mehrere Gruppen von Punkten parallel zu verarbeiten.

Von größerer Bedeutung ist eine Laufzeitverbesserung der Entzerrung. Nach der Bestimmung der Abbildungsmatrizen werden, nach dem in Kapitel 4 beschriebenen Schema, eine Reihe von Bildern entzerrt. Um die aufgenommenen Bilder nicht zwischenspeichern zu müssen, wäre eine echtzeitfähige Entfaltung wünschenswert. Die Bilder könnten somit direkt entzerrt gespeichert werden. Um die Laufzeit der Entfaltung zu verbessern könnte man hier beispielsweise parallel mehrere Bildblöcke gleichzeitig verarbeiten.

6.2. Verbesserung der Linien-Detektion

Die Linien-Detektion basiert auf Hough-Transformationen und funktioniert nur mit relativ homogenen Hintergründen. Darüber hinaus werden durch den aktuellen Ansatz sehr viele Linien detektiert, wobei anschließend die Schnittpunkte zwischen all diesen Kandidaten bestimmt werden müssen. Ein besserer Ansatz wäre die Gradientenrichtung des Kantenbilds miteinzubeziehen. Dies reduziert die Anzahl falscher Votes und verbessert darüber hinaus die Laufzeit [OC76].

6.3. Verbesserung der Vorwärtsentfaltung

Das Hauptproblem der Vorwärtsentfaltung sind die Defekte auf dem entfalteten Bild. Man könnte mit Hilfe einer Delaunay-Triangulierung versuchen, die Defekte zu beheben. Die auf die Mantelfläche abgebildeten Punkte werden bei diesem Verfahren nicht gerundet. Anschließend werden die Punkte trianguliert. Es entstehen Dreiecksnetze, wie beispielhaft in Abbildung 6.1 zu sehen sind. Zum Zwecke der Veranschaulichung wurde hierbei nur ein Teil der Daten benutzt. Mit Hilfe dieser Dreiecke wird nun das Ergebnisbild abgetastet, das heißt man untersucht für jedes Pixel auf dem entzerrten Bild in welchem Dreieck es sich befindet und bestimmt dann die Farbe des Pixels mittels baryzentrischer Interpolation aus den drei umgebenden Punkten. Die Delaunay-Triangulierung ist jedoch relativ aufwendig mit einer Laufzeit von $O(p \log p)$ für p Punkte [SD95]. Sie richtet sich bei einer konstanten Eingabeauflösung also nach der Gesamtanzahl der Pixel im Kalibrierungskegel. Anschließend muss für jedes Pixel auf dem Ausgabebild eine Interpolation durchgeführt werden. Die Laufzeit der Vorwärtsentfaltung wird somit quadratisch mit einer großen additiven Konstante bedingt durch die Triangulierung.

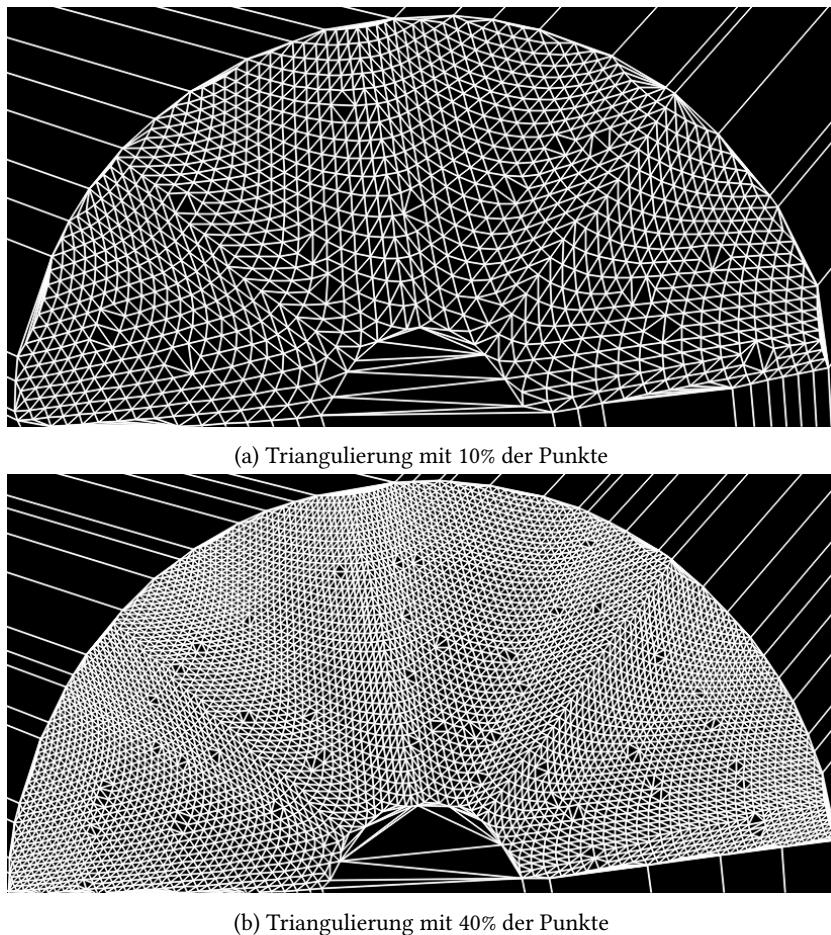


Abbildung 6.1.: Delaunay-Triangulierung

6.4. Verbesserung der Rückwärtsentfaltung

Aktuell wird die Projektionsmatrix bei der Rückwärtsentfaltung durch *Direct Linear Transformation* bestimmt. Dieses Verfahren minimiert jedoch nicht den Reprojektionsfehler. Besser wäre hier deshalb ein iteratives Verfahren, wie der Levenberg–Marquardt-Algorithmus (konkret für Projektionsmatrizen beschrieben in [HZ03]). Im Gegensatz dazu könnte auch ein RANSAC-Ansatz verwendet werden. Statt eine optimale Lösung für alle detektierte Samples zu bestimmen, berechnet man wiederholt für sechs Punkte¹ eine Projektionsmatrix und untersucht dann den Reprojektionsfehler für alle anderen Punkte und wählt schließlich die Projektionsmatrix mit dem größten Consensus Set aus.

¹sechs Punkte sind mindestens notwendig um eine Projektionsmatrix bestimmen zu können, da es elf unbekannte gibt (siehe Kapitel 2.3).

6.5. Verbesserung der Ellipsendetektion mittels Deformable Templates

Die Minimierung der konstruierten Energiefunktion wird stark durch irrelevante lokale Minima erschwert. Ein Umformulieren der Energiefunktion könnte hier Abhilfe schaffen. Darüber hinaus muss untersucht werden, wie die Einflussparameter α , β und γ eingestellt werden müssen, um zuverlässige Ergebnisse zu liefern.

A. Inhalt der CD

Auf der beiliegenden CD befinden sich folgende Ordner:

1. Blender-Dateien (Kegel mit Muster, beide Versuchsaufbauten)
2. CAD-Dateien (parametrisierbares Kalibrierungsmuster)
3. Bilder des Kegels (generierte Bilder, Bilder mit Weitwinkelkamera und normaler Kamera)
4. Daten (CSV-Dateien der Ergebnisse aus Kapitel 5)
5. Latex (Ausarbeitung)
6. Matlab (Deformable Templates und Skripte zum Zeichnen der Ergebnisse)
7. Qt (GUI- und Konsolenanwendung)

Abbildungsverzeichnis

1.1.	beide Ansätze im Vergleich	3
2.1.	Gerader Kreiskegel	4
2.2.	Kegelstumpf und Ergänzungskegel	5
2.3.	Kegelstumpf	6
2.4.	Kegelmantelfläche	6
2.5.	Abbildung der Kegelstumpfhöhe auf die Seitenhöhe	7
2.6.	Lochkameramodell	10
2.7.	Linsenverzerrungen	12
2.8.	Ellipse mit Brennpunkten f_1, f_2 , Zentrum C , Hauptachse a , Nebenachse b und Scheitelpunkten V_1/V_3 und V_2/V_4	14
2.9.	Ellipse mit Zentrum (x_0, y_0) , Hauptachse a , Nebenachse b , sowie Drehwinkel θ	15
2.10.	Ellipsausschnitt im ersten Quadranten mit Abfragepunkt Q und eingezeichneter kürzester Distanz zum Ellipsenpunkt E . Die Ellipse wird beschrieben durch \mathcal{E}	18
3.1.	Kalibrierungsmuster von oben mit $n = 5, m = 6$	22
3.2.	Kalibrierungsmuster entfaltet mit $n = 5, m = 6$	22
3.3.	Kamerakalibrierung	23
3.4.	Detektion der Samples	24
3.5.	Canny-Kantendetektion	24
3.6.	Hough-Transformation zur Linien-Detektion (rot) und bestimmter Schnittpunkt (grün)	25
3.7.	Ellipsendetektion: bestimme Pixel-Positionen (weiß), Aussendepunkt (gelb)	26
3.8.	Ellipsendetektion bei Ausreißern	27
3.9.	detektierte Ellipsen	27
3.10.	Zuordnung von Punkten zu Ellipsen (links) und Liniensegmenten (rechts)	29
3.11.	Zuordnung von Punkten zu Ellipsen (links) und Liniensegmenten (rechts) farblich sortiert: 1. rot, 2. gelb, 3. grün, 4. blau, 5. magenta	29
3.12.	Interpolation der 3D-Koordinaten mittels vier Sample-Nachbarn	31
3.13.	Vorwärtsentfaltung	32
3.14.	Rückwärtsentfaltung	32
4.1.	Kalibrierungsassistent: intrinsische Kamerakalibrierung	33
4.2.	Kalibrierungsassistent: Kegelkalibrierung	34
5.1.	interpolierter Kegel	36
5.2.	Vorwärtsentfaltung	36
5.3.	Einfluss der Ausgabeauflösung auf die Anzahl der Defekte	37
5.4.	Größe des Ausgabebildes bei einer Seitenhöhe von $ress$	37
5.5.	Laufzeitvergleich zwischen Vorwärtsentfaltung (blau) und Rückwärtsentfaltung (rot)	38
5.6.	Einfluss der intrinsischen Kalibrierung auf den Reprojektionsfehler	39
5.7.	gerendeter Kegel mit Kalibrierungsmuster in Blender aus verschiedenen Blickrichtungen	40
5.8.	Einfluss der Rotation der Kamera	40
5.9.	Einfluss der Ausgabeauflösung auf die Laufzeit der Entfaltung	41
5.10.	Deformable Templates: Ausschnitt der Energiefunktion für variable Haupt- und Nebenachse	42

5.11. Vergleich RANSAC und LSQ bei gleichverteilten Ausreißern $\epsilon = 0.5, p = 0.99$	43
5.12. Dice-Koeffizienten bei einer Schätzung mittels RANSAC und der Methode der kleinsten Quadrate (LSQ) bei gleichverteilten Ausreißern	44
5.13. Vergleich RANSAC und LSQ bei bei Ausreißern auf einer äußeren Ellipse mit $p = 0.99$ und $\epsilon = 0.33$	44
5.14. Dice-Koeffizienten bei einer Schätzung mittels RANSAC und der Methode der kleinsten Quadrate (LSQ) bei Ausreißern auf einer äußeren Ellipse	45
6.1. Delaunay-Triangulierung	47

Literatur

- [Bos06] Siegfried Bosch. *Lineare Algebra*. 2006. ISBN: 3-540-29884-3. DOI: 10.1007/3-540-29885-1 (siehe S. 16).
- [Can86] J Canny. „A Computational Approach to Edge Detection“. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 8.6 (1986), S. 679–698. DOI: 10.1109/TPAMI.1986.4767851 (siehe S. 13, 24).
- [Dic45] Lee Raymond Dice. „Measures of the Amount of Ecologic Association Between Species“. In: *Ecological Society of America* 26.3 (1945), S. 297–302 (siehe S. 42).
- [Ebe08] David Eberly. „The Area of Intersecting Ellipses“. In: (2008) (siehe S. 42).
- [Ebe13] David Eberly. „Distance from a Point to an Ellipse, an Ellipsoid, or a Hyperellipsoid“. In: (2013), S. 1–13 (siehe S. 17, 20).
- [FB81] Martin A. Fischler und Robert C. Bolles. „Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography“. In: *Commun. ACM* 24.6 (1981), S. 381–395. DOI: 10.1145/358669.358692 (siehe S. 13, 14).
- [Hav+14] Jiří Havel u. a. „Real-time detection of lines using parallel coordinates and CUDA“. In: *Journal of Real-Time Image Processing* 9.1 (2014), S. 205–216. DOI: 10.1007/s11554-012-0303-4 (siehe S. 46).
- [HS97] Janne Heikkilä und Olli Silvén. „A Four-step Camera Calibration Procedure with Implicit Image Correction“. In: *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*. CVPR '97. Washington, DC, USA: IEEE Computer Society, 1997, S. 1106–. ISBN: 0-8186-7822-4 (siehe S. 11, 12).
- [HZ03] Richard Hartley und Andrew Zisserman. *Multiple View Geometry in Computer Vision*. 2. Aufl. New York, NY, USA: Cambridge University Press, 2003. ISBN: 0521540518 (siehe S. 47).
- [Key81] R. Keys. „Cubic convolution interpolation for digital image processing“. In: *Acoustics, Speech and Signal Processing, IEEE Transactions on* 29.6 (1981), S. 1153–1160. DOI: 10.1109/TASSP.1981.1163711 (siehe S. 41).
- [Law72] J D Lawrence. *A Catalog of Special Plane Curves*. Dover Publications, 1972, S. 62–63. ISBN: 978048602882 (siehe S. 15, 17).
- [Lin93] Tony Lindeberg. „Detecting Salient Blob-like Image Structures and Their Scales with a Scale-space Primal Sketch: A Method for Focus-of-attention“. In: *International Journal of Computer Vision* 11.3 (1993), S. 283–318. DOI: 10.1007/BF01469346 (siehe S. 12).
- [OC76] F. O'Gorman und M. B. Clowes. „Finding Picture Edges Through Collinearity of Feature Points“. In: *IEEE Transactions on Computers* 25.4 (1976), S. 449–456. DOI: 10.1109/TC.1976.1674627 (siehe S. 46).
- [SD95] Peter Su und Robert L. Scot Drysdale. „A Comparison of Sequential Delaunay Triangulation Algorithms“. In: *Proceedings of the Eleventh Annual Symposium on Computational Geometry*. SCG '95. Vancouver, British Columbia, Canada: ACM, 1995, S. 61–70. ISBN: 0-89791-724-3. DOI: 10.1145/220279.220286 (siehe S. 46).
- [Sto07] Burlisch Stoer. *Numerische Mathematik 1*. Springer, 2007, S. 249–253 (siehe S. 9).
- [Sto11] Burlisch Stoer. *Numerische Mathematik 2*. Springer, 2011, S. 21–23 (siehe S. 10).
- [Tsa87] Roger Y. Tsai. „Radiometry“. In: USA: Jones und Bartlett Publishers, Inc., 1987. Kap. A Versatile Camera Calibration Technique for High-accuracy 3D Machine Vision Metrology Using Off-the-shelf TV Cameras and Lenses, S. 323–344. ISBN: 0-86720-294-7 (siehe S. 11).
- [YHC92] Alan L. Yuille, Peter W. Hallinan und David S. Cohen. „Feature Extraction from Faces Using Deformable Templates“. In: *Int. J. Comput. Vision* 8.2 (1992), S. 99–111. DOI: 10.1007/BF00127169 (siehe S. 20).

- [Zha00] Zhengyou Zhang. „A Flexible New Technique for Camera Calibration“. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 22.11 (2000), S. 1330–1334. DOI: [10.1109/34.888718](https://doi.org/10.1109/34.888718) (siehe S. 12).

Plagiatserklärung

Hiermit versichere ich, dass die vorliegende Arbeit über

Entzerrung von Kegeloberflächen aus einer Einkameraansicht basierend auf projektiver Geometrie

selbstständig verfasst worden ist, dass keine anderen Quellen und Hilfsmittel als die angegebenen benutzt worden sind und dass die Stellen der Arbeit, die anderen Werken – auch elektronischen Medien – dem Wortlaut oder Sinn nach entnommen wurden, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht worden sind.

Lars Haalck, Münster, 24. September 2016

Ich erkläre mich mit einem Abgleich der Arbeit mit anderen Texten zwecks Auffindung von Übereinstimmungen sowie mit einer zu diesem Zweck vorzunehmenden Speicherung der Arbeit in eine Datenbank einverstanden.

Lars Haalck, Münster, 24. September 2016