

# Liste der noch zu erledigenden Punkte

dieses kapitel sollte aussagekräftig sein, da projektive geometrie im titel vorkommt	7
wie schreibt man Achsen-ausgerichtet?	9
wie viele punkte braucht man zum definieren von Ellipsen	9
den zweiten teil überarbeiten hier	14
Modell wortwiederholungen	15
siehe später, dazu mehr später????	19
warum gerade dieses muster? warum ist das so gut? wie berechnet man das muster?	
welche eigenschaften hat es?	19
Warum die extrinscieln nicht? kamera position gegeben dann ??	19
Vergleich ohne und mit Kalibrierung	19
Umbenennen	19
Formulierung	20
hier passendes Wort einfügen	20
Komma-Setzung	20
hier schon erläutern oder erst in späteren kapiteln?	25
LSQ irgendwo erklären	29





WESTFÄLISCHE  
WILHELMS-UNIVERSITÄT  
MÜNSTER

# 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



# **Zusammenfassung**

In der vorliegenden Arbeit



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Theoretische Grundlagen</b>	<b>3</b>
2.1	Kegel . . . . .	3
2.2	Kamerakalibrierung und projektive Geometrie . . . . .	7
2.3	Ellipsen . . . . .	9
2.3.1	allgemein . . . . .	9
2.3.2	Abstand: Punkt zu Ellipse . . . . .	12
2.4	RANSAC . . . . .	15
2.5	Hough-Transformation für Linien . . . . .	16
2.6	Canny . . . . .	16
2.7	Blobdetection . . . . .	16
<b>3</b>	<b>Methodik</b>	<b>17</b>
3.1	Kalibrierungsmuster . . . . .	17
3.1.1	Anzahl der Samples . . . . .	19
3.2	Intrinsische Kamerakalibrierung . . . . .	19
3.3	Detektion der Samples . . . . .	19
3.4	Ellipsen-Detektion . . . . .	20
3.5	Zuordnung der Punkte . . . . .	23
3.6	Weltkoordinaten bestimmen . . . . .	24
3.7	Entfaltung . . . . .	25
3.7.1	Vorwärtsentfaltung . . . . .	25
3.7.2	Rückwärtsentfaltung . . . . .	26
<b>4</b>	<b>Implementierung</b>	<b>27</b>
<b>5</b>	<b>Analyse</b>	<b>29</b>
<b>6</b>	<b>Fazit und Ausblick</b>	<b>35</b>
	<b>Abbildungsverzeichnis</b>	<b>37</b>
	<b>Tabellenverzeichnis</b>	<b>39</b>





# 1 Einleitung

einleitung

hier setup mit bild



## 2 Theoretische Grundlagen

Links Koordinatensystem...

### 2.1 Kegel

#### 2.1.1 Definition (Kegel)

kegel  $S$  bezeichnet die Seitenhöhe und ist definiert durch  $S = \sqrt{H^2 + R^2}$   $S \geq R$   
Dreiecksungleichung

In der weiteren Arbeit betrachten wir nur gerade Kreiskegel

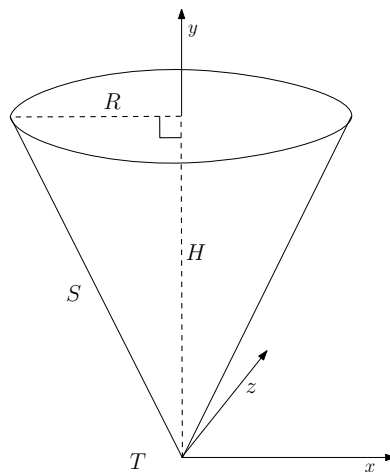


Abbildung 2.1: Gerader Kreiskegel

Ein Kegel mit Spitze  $T(0, 0, 0)$ , Radius  $R$  und Höhe  $H$  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  $\theta \in [0, 2\pi)$

### 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). Das Stück von Grundfläche zur Schnittfläche nennen wir Kegelstumpf. Die Differenz zum eigentlichen Kegel wird als Ergänzungskegel bezeichnet.

$H, R, S$  bleiben die Angaben des gesamten Kegels. Hinzu kommen  $h, r, s$  als Angaben 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).

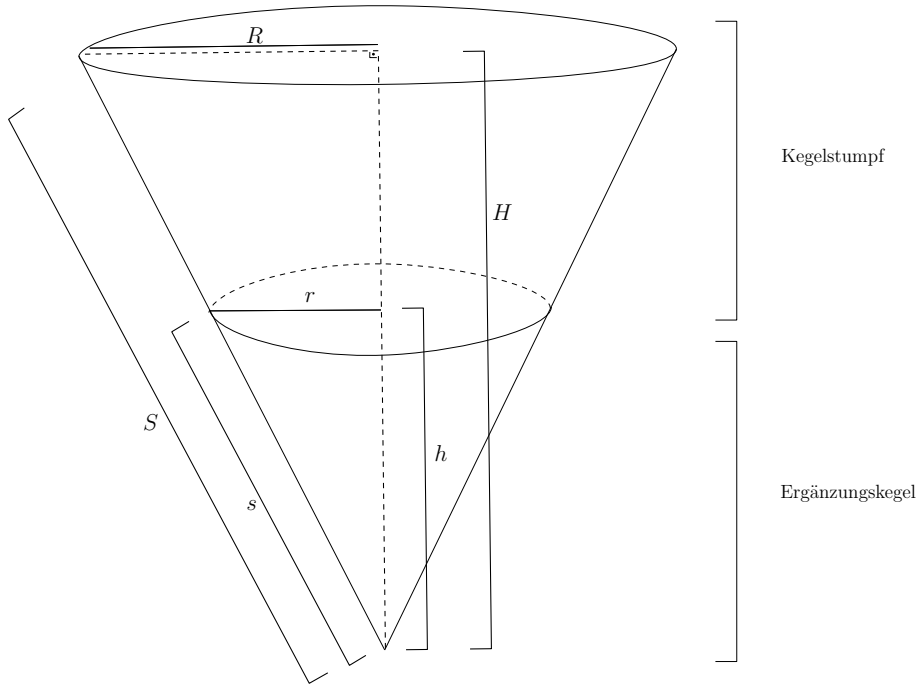


Abbildung 2.2: Kegelstumpf und Ergänzungskegel

Analog zum Kreiskegel definieren wir einen Kegelstumpf durch folgende Parametrisierung:

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

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

Die Mantelfläche des Kegelstumpfes aus Abbildung 2.4 kann dann parametrisch beschrieben werden als:

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

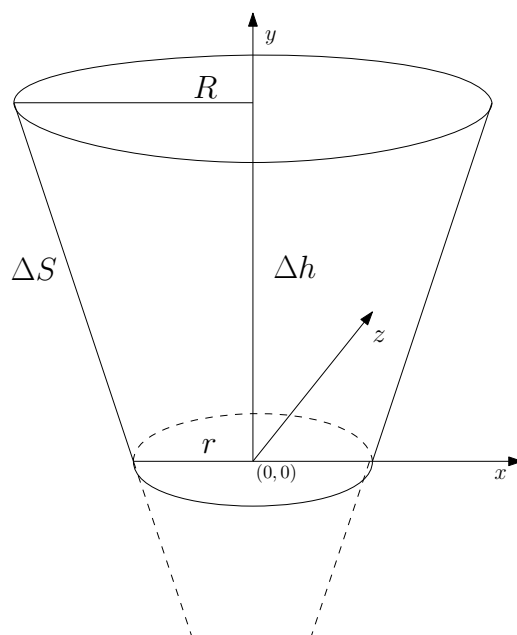


Abbildung 2.3: Kegelstumpf

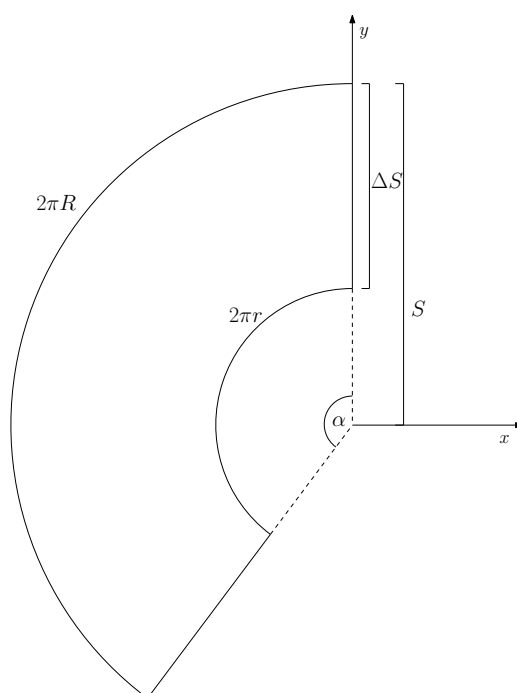


Abbildung 2.4: Kegelmantelfläche

mit  $u \in [0, \Delta H]$  und  $\phi \in [0, \alpha] \subseteq [0, 2\pi]$  mit  $\alpha S = 2\pi R \implies \alpha = 2\pi \frac{R}{S}$

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:

Seien ein Punkt  $C(x, y, z)$  auf der Oberfläche des Kegelstumpfs gegeben. Wir wissen aus der parametrischen Form 2.2, dass  $C$  die Form

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

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

Aus der  $y$ -Koordinate kann man als die Höhe ablesen und somit den Radius in der Mantelfläche als lineare Interpolation zwischen  $s$  und  $S$  bestimmen (siehe Abbildung 2.5). Wir definieren uns hierfür eine Hilfsfunktion

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

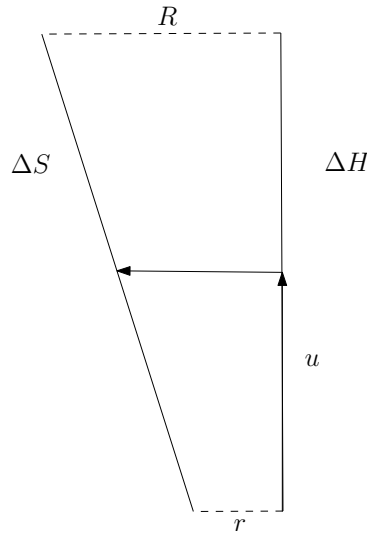


Abbildung 2.5: Abbildung der Kegelstumpfhöhe auf die Seitenhöhe

Da  $R, r, \Delta H$  und nun auch die Höhe bekannt sind, kann man den Winkel  $\theta$  im Kegelstumpf einfach ausrechnen. Anschließend muss dieser noch mit  $\frac{R}{S}$  multipliziert werden um ihn auf  $[0, \alpha]$  zu skalieren (siehe 2.3). Auch hierfür definieren wir eine Hilfsfunktion:

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

, wobei wir  $\operatorname{atan2}$  benutzen um den Winkel im richtigen Quadranten, also in  $[0, 2\pi)$ , bestimmen zu können.

Mit diesen beiden Hilfsfunktionen und 2.3 ergibt sich insgesamt:

$$\Psi: [r, R] \times [0, \Delta H] \times [r, R] \rightarrow [s, S] \times [s, S]$$

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} -\Sigma(y) \sin \Phi(x, y, z) \\ \Sigma(y) \cos \Phi(x, y, z) \end{pmatrix} \quad (2.5)$$

Analog lässt die sich Umkehrabbildung konstruieren:

Sein ein Punkt  $L(x, y)$  auf der Mantelfläche des Kegelstumpfs gegeben. Aus der parametrischen Form 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  $\phi$  mit inverser Skalierung also analog durch folgende Hilfsfunktion bestimmen:

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

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

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

Insgesamt ergibt sich:

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

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} \left( r + \frac{H(x, y)}{\Delta H} (R - r) \right) \cos (\Theta(x, y)) \\ H(x, y) \\ \left( r + \frac{H(x, y)}{\Delta H} (R - r) \right) \sin (\Theta(x, y)) \end{pmatrix} \quad (2.6)$$

## 2.2 Kamerakalibrierung und projektive Geometrie

### 2.2.1 Definition

Kamerakalibrierung ist ein Verfahren, bei dem die interne Kamerageometrie und optischen Eigenschaften (intrinsische Parameter) und/oder die 3D-Position und Orientierung der Bildebene relativ zu einem Weltkoordinatensystem (extrinsische Parameter)

dieses kapitel sollte aussagekräftig sein, da projektive geometrie im titel vorkommt

bestimmt werden [Tsa87].

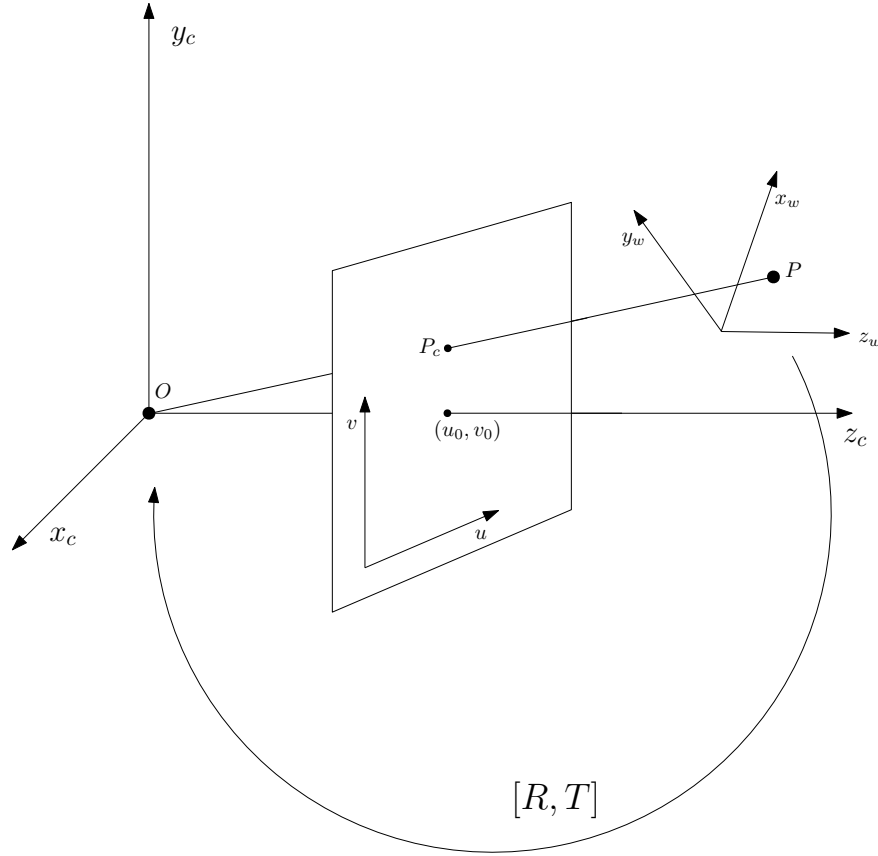


Abbildung 2.6: Projektion eines Punktes  $P$  im Weltkoordinatensystem  $(x_w, y_w, z_w)$  auf die Bildebene im Kamerakoordinatensystem  $(x_c, y_c, z_c)$

Lochkamera bild, erklären warum man eine virtuelle Bildebene for die Kamera macht, damit sich das bild nicht dreht. die optischen eigenschaften ändern sich nicht.

Kamerakalibrierung wird benötigt, um eine Beziehung zwischen Punkten in 3D im Weltkoordinatensystem und den Punkten auf der Bildebene herzustellen. Genauer suchen wir eine Abbildung

$$\begin{pmatrix} wu \\ wv \\ w \end{pmatrix} = \begin{pmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}, \quad (2.7)$$

die einen Punkt  $P = (x, y, z)$  in homogenen Koordinaten  $\tilde{P} = (x, y, z, 1)$  auf einen Punkt  $C = (u, v)$  auf die Bildebene abbildet.

Schritt einzeln erklären Zunächst wird das Weltkoordinatensystem mit einer Rotation und einer Translation in das Kamerakoordinatensystem überführt.



Selfcalibrating vs bla bla

erst mal außen vor. erst nach dem rest Zu den intrinsischen Parametern gehören neben Brennweite  $f$  der Kamera, die Bildmitte (auch Bildmittelpunkt)  $(u_0, v_0)$ , metrische Pixelgrößen  $s_x$  und  $s_y$ , sowie Verzerrungskoeffizienten. Mit Hilfe der intrinsischen und extrinsischen Parameter lässt sich eine

wofür braucht man das? billige linsen, verzerrungen, 3d rekonstruktionen. abbildung von 3d auf 2d, lochkamera, intrinsisch, extrinsisch Lochkamera

Linsenverzerrungen

projektionsmatrix (homogene Koordinaten????) SVD, QR, LSQ?

## 2.3 Ellipsen

### 2.3.1 allgemein

Ellipsen sind deshalb für diese Arbeit interessant, als dass sie als perspektivische Verzerrung von Kreisen entstehen. Nimmt man das Kalibrierungsmuster nicht perfekt im Lot auf, entstehen zwangsläufig Ellipsen.

#### 2.3.1 Definition (Ellipse)

Ellipsen entstehen durch Kegelschnitt und so bla bla. im weiteren bla bla meinen wir mit Hauptachsen immer die Semihauptachsen

In ihrer einfachsten Form liegt die Ellipse im Zentrum des Koordinatensystems und ihre Haupt- und Nebenachse  $a$  und  $b$  sind Achsen-ausgerichtet, das heißt ihre Hauptachse liegt auf der X-Achse und ihre Nebenachse auf der Y-Achse. Sie kann dann in der impliziten Form

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (2.8)$$

beschrieben werden.

Befindet sich die Ellipse nicht im Ursprung so muss eine Verschiebung beziehungsweise bei einer Rotation ein Drehwinkel (siehe Abbildung 2.7) ergänzt werden.

$$\frac{(x - x_0)^2}{a^2} + \frac{(y - y_0)^2}{b^2} = 1 \quad (2.9)$$

$$\frac{((x - x_0) \cos \theta + (y - y_0) \sin \theta)^2}{a^2} + \frac{((x - x_0) \sin \theta - (y - y_0) \cos \theta)^2}{b^2} = 1 \quad (2.10)$$

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

wie schreibt man Achsen-ausgerichtet?

wie viele punkte braucht man zum definieren von Ellipsen

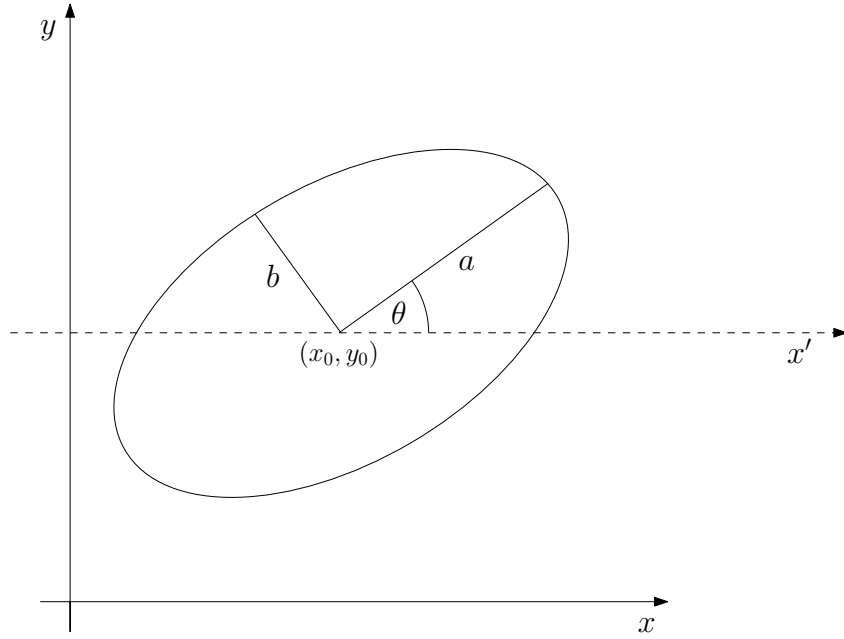


Abbildung 2.7: Ellipse mit Zentrum  $(x_0, y_0)$ , Hauptachse  $a$ , Nebenachse  $b$ , sowie Drehwinkel  $\theta$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x_0 + a \cos \phi \cos \theta - b \sin \phi \sin \theta \\ y_0 + a \cos \phi \sin \theta + b \sin \phi \cos \theta \end{pmatrix} \quad (2.11)$$

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

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

$$ax^2 + by^2 + cxy + dx + ey + f = 0 \quad \text{mit} \quad c^2 - 4ab < 0 \quad (2.12)$$

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 2.10 und 2.12 sind äquivalent, falls die Ellipse nicht degeneriert ist (ohne Beweis). Da wir die Umformung von 2.10 nach 2.12 später brauchen, wird sie hier einmal exemplarisch vorgeführt.

Zunächst einmal fällt auf, dass der gemischten 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 2.12 kann umgeformt werden zu:

$$\underbrace{\begin{pmatrix} x & y \end{pmatrix}}_{=:u^T} \underbrace{\begin{pmatrix} a & \frac{c}{2} \\ \frac{c}{2} & b \end{pmatrix}}_{=:M} \underbrace{\begin{pmatrix} x \\ y \end{pmatrix}}_{=:u} + \underbrace{\begin{pmatrix} d & e \end{pmatrix}}_{=:u} \begin{pmatrix} x \\ y \end{pmatrix} + f = 0$$

$$\Leftrightarrow u^T M u + \begin{pmatrix} d & e \end{pmatrix} u + f = 0$$

Der gemischte Term wird alleine durch  $M = \begin{pmatrix} a & \frac{c}{2} \\ \frac{c}{2} & b \end{pmatrix}$  bestimmt. Da die Matrix  $M$  symmetrisch ist, ist sie orthogonal diagonalisierbar. Des Weiteren hat  $M$  zwei von null verschiedene Eigenwerte, denn

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

ist nur dann gleich null, wenn  $c^2 - 4ab = 0$ , was ein Widerspruch zur Annahme in 2.12 ist.  $M$  ist somit eine von null verschiedene Determinante und somit vollen Rang, hat also zwei von null verschiedene Eigenwerte. Insbesondere gibt es also zwei Eigenvektoren von  $M$ , die zueinander orthogonal sind.

Es gilt  $M = S^T D S$ , 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:

$$\begin{aligned} u^T (S^T D S) u + \begin{pmatrix} d & e \end{pmatrix} \underbrace{(S^T S)}_{=: \mathbb{1}} u + f &= 0 \\ \Leftrightarrow (Su)^T D (Su) + \begin{pmatrix} d & e \end{pmatrix} S^T (Su) + f &= 0 \\ \Leftrightarrow v^T D v + \begin{pmatrix} d & e \end{pmatrix} S^T v + f &= 0 \end{aligned} \tag{2.13}$$

Man kann leicht nachrechnen, dass der gemischte Teil somit eliminiert wurde. Durch Anwenden der Transformation  $S$  wurde  $u$  also in das Koordinatensystem, in dem die Ellipse Achsen- ausgerichtet ist, transformiert.

Eine Rotationsmatrix mit Rotationswinkel  $\theta$  ist definiert durch:

$$R = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \tag{2.14}$$

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

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

Multipliziert man nun 2.13 aus ergibt sich:

$$\begin{aligned}
 & \lambda_1 v_1^2 + \lambda_2 v_2^2 + \underbrace{(d \ e) S^T}_{=:(d', e')} v + f = 0 \\
 \Leftrightarrow & \lambda_1 v_1^2 + \lambda_2 v_2^2 + d' v_1 + e' v_2 + f = 0 \\
 \Leftrightarrow & (\lambda_1 v_1^2 + d' v_1) + (\lambda_2 v_2^2 + e' v_2) + f = 0 \\
 \Leftrightarrow & (\lambda_1 v_1^2 + d' v_1) + \left( \frac{d'^2}{4\lambda_1} - \frac{d'^2}{4\lambda_1} \right) + (\lambda_2 v_2^2 + e' v_2) + \left( \frac{e'^2}{4\lambda_2} - \frac{e'^2}{4\lambda_2} \right) + f = 0 \quad (2.15) \\
 \Leftrightarrow & \left[ \lambda_1 \left( v_1^2 + \frac{2d'}{2\lambda_1} v_1 + \frac{d'^2}{4\lambda_1^2} \right) - \frac{d'^2}{4\lambda_1} \right] + \left[ \lambda_2 \left( v_2^2 + \frac{2e'}{2\lambda_2} v_2 + \frac{e'^2}{4\lambda_2^2} \right) - \frac{e'^2}{4\lambda_2} \right] + f = 0 \\
 \Leftrightarrow & \lambda_1 \underbrace{\left( v_1 + \frac{d'}{2\lambda_1} v_1 \right)^2}_{=-x'_0} + \lambda_2 \underbrace{\left( v_2 + \frac{e'}{2\lambda_2} v_2 \right)^2}_{=-y'_0} - \underbrace{\left( \frac{d'^2}{4\lambda_1} + \frac{e'^2}{4\lambda_2} - f \right)}_{=: \sigma} = 0,
 \end{aligned}$$

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

Das Zentrum der transformierten Ellipse kann nun aus 2.15 einfach abgelesen werden. Um das Zentrum der eigentlichen Ellipse zu bestimmen, muss mit der inversen Rotation  $S^T$  multipliziert werden:

$$(x_0, y_0)^T = S^T (x'_0, y'_0)^T$$

Obige Gleichung lässt sich anschließend weiter vereinfachen:

$$\begin{aligned}
 & \lambda_1 (v_1 - x'_0)^2 + \lambda_2 (v_2 - y'_0)^2 = \sigma \\
 \Leftrightarrow & \frac{\lambda_1}{\sigma} (v_1 - x'_0)^2 + \frac{\lambda_2}{\sigma} (v_2 - y'_0)^2 = 1 \quad (2.16)
 \end{aligned}$$

wobei  $\sigma \neq 0$ , wenn die Ellipse nicht zum Punkt entartet ist [Law72]. Vergleicht man nun 2.16 mit 2.9 so sieht man das

$$\begin{aligned}
 & \frac{\lambda_1}{\sigma} = \frac{1}{a^2} \quad \text{und} \quad \frac{\lambda_2}{\sigma} = \frac{1}{b^2} \\
 \Leftrightarrow & \sqrt{\frac{\sigma}{\lambda_1}} = a \quad \text{und} \quad \sqrt{\frac{\sigma}{\lambda_2}} = b \quad (2.17)
 \end{aligned}$$

Es gilt wie erwartet  $a \geq b$ , da  $\lambda_1 \leq \lambda_2$

### 2.3.2 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 Achsen-ausgerichtet sind und darüber hinaus nur Punkte im ersten Quadranten. Ansonsten wird die Ellipse in den Ursprung verschoben und um ihren entgegengesetzten Drehwinkel rotiert. Da die Ellipse dann bezüglich der X- und Y- Achse

symmetrisch ist, kann der Punkt einfach durch Spiegelung in den richtigen Quadranten transformiert werden. Der Abstand ändert sich dadurch nicht.

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

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

- Die Ellipse kann stets durch die implizite Gleichung

$$\frac{x_0^2}{a^2} + \frac{x_1^2}{b^2} = 1$$

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

$$F(\theta) = |\mathcal{E}(\theta) - Q|^2. \quad (2.18)$$

Die Ableitung von  $F$

$$F'(\theta) = 2 (\mathcal{E}(\theta) - Q) \cdot \mathcal{E}'(\theta). \quad (2.19)$$

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 2.8).

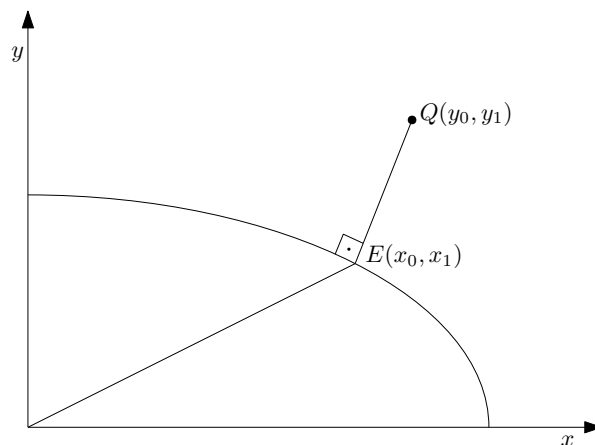


Abbildung 2.8: Ellipsenausschnitt im ersten Quadranten mit Abfragepunkt  $Q$  und eingezeichneter kürzester Distanz zur Ellipse

Betrachten wir also die Funktion:

$$G(x_0, x_1) = \frac{x_0^2}{a^2} + \frac{x_1^2}{b^2} - 1. \quad (2.20)$$

$(x_0, x_1)$  ist genau dann ein Punkt auf der Ellipse, wenn  $G(x_0, x_1) = 0$ . Der Gradient von  $G$  in  $(x_0, x_1)$  ist ein Normalenvektor auf der Ellipse, somit auch der halbe Gradient  $\nabla G(x_0, x_1)/2$ . Der Vektor von  $E$  zu  $Q$  muss dieselbe Richtung haben. Es gilt somit:

$$(y_0, y_1) - (x_0, x_1) = t \frac{\nabla G(x_0, x_1)}{2} = t \left( \frac{x_0}{a^2}, \frac{x_1}{b^2} \right) \quad (2.21)$$

für ein  $t \in \mathbb{R}$ .

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

$$y_0 = x_0 \left( 1 + \frac{t}{a^2} \right), \quad y_1 = x_1 \left( 1 + \frac{t}{b^2} \right) \quad (2.22)$$

$$x_0 = \frac{a^2 y_0}{t + a^2}, \quad x_1 = \frac{b^2 y_1}{t + b^2} \quad (2.23)$$

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  $X$ -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 kann gezeigt werden, dass der Punkt auf der Ellipse mit der kürzesten Distanz zu  $(y_0, 0)$  für  $0 < y_0 < (a^2 - b^2)/a^2$ ,  $E = (x_0, x_1)$  ist mit

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  $d = |y_0 - a|$  Einfach nur es kann gezeigt werden, dass bla?

Wenn  $x_1 \neq 0$  kann gezeigt werden, dass für die Distanz gilt:

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

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 2.22 lässt sich folgende Einschränkung für  $t$  herleiten:

$$\begin{aligned} 0 < y_0 &= x_0 \left( 1 + \frac{t}{a^2} \right) \\ \Leftrightarrow -1 \cdot a^2 &< t. \end{aligned} \quad (2.24)$$

Analog ergibt sich mit  $y_1: -b^2 < t$ . Da  $a \geq b$  gilt, reicht es, sich nur die zweite Ungleichung anzuschauen, dass sie die erste impliziert. Setzt man nun 2.23 in 2.20 ein, erhält man:

$$F(t) = \left( \frac{ay_0}{t + a^2} \right)^2 + \left( \frac{by_1}{t + b^2} \right)^2 - 1 \quad (2.25)$$

Man kann nun zeigen, dass diese Funktion auf dem gesamten Intervall  $[-b^2, \infty)$  monoton fällt und links gekrümmt ist. 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.

## 2.4 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, nutzt RANSAC nur eine Teilmenge der Messreihe.

RANSAC wählt aus der Menge der Messdaten wiederholt zufällig die minimale Anzahl Messdaten aus, die nötig sind um das Modell eindeutig zu beschreiben und prüft 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 bei, charakterisiert. 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.

Modell  
wortwieder-  
holungen

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:

$$\frac{\log(1 - p)}{\log(1 - (1 - \epsilon)^k)} \quad (2.26)$$

## 2.5 Hough-Transformation für Linien

## 2.6 Canny

Der *Canny*-Algorithmus[Can86] ist ein Verfahren zur Kantendetektion. Das  
Kantendetektion (canny sobel)

## 2.7 Blobdetection

??



## 3 Methodik

In diesem Kapitel stellen wir ein Verfahren vor, dass ermöglicht anhand eines Bildes (siehe Einleitung).

Zunächst gehen wir auf das verwendete Kalibrierungsmuster ein, woraufhin die genaue Vorgehensweise zur Entfaltung erläutert wird.

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 links-händigen Weltkoordinatensystem an der Position  $(0, 0, 0)$  (siehe Abbildung 2.3) befindet. Durch diese Einschränkung gehen jegliche absolute Größenverhältnisse verloren. Es ist also nicht mehr möglich die Größe einer Larve zu bestimmen.

Wir wollen eine Beziehung zwischen Bildpunkte und Kegelpunkten herstellen. wo ist z achse?

### 3.1 Kalibrierungsmuster

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 realitätsgetreu wiedergegeben werden.

Wir haben uns für ein Muster entschieden, dass in äquidistanten Abständen  $\Delta R$ , 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 zu sehen, 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 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 nun ein Punkt auf dem Kegel im Weltkoordinatensystem zugeordnet werden. Da ein Kegel beliebig um die  $Y$  Achse rotiert werden, kann ist diese Abbildung zunächst nicht eindeutig. Dazu nehmen wir an, dass das Liniensegment mit dem kleinsten Winkel zur  $X$ -Achse mit dem Kegelwinkel  $\theta = 0$  korrespondiert (siehe 2.2).

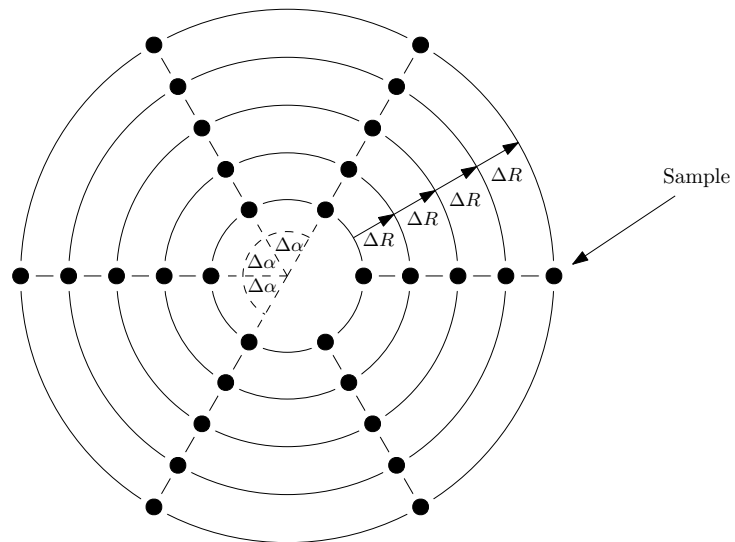


Abbildung 3.1: Kalibrierungsmuster von oben mit  $n = 5, m = 6$

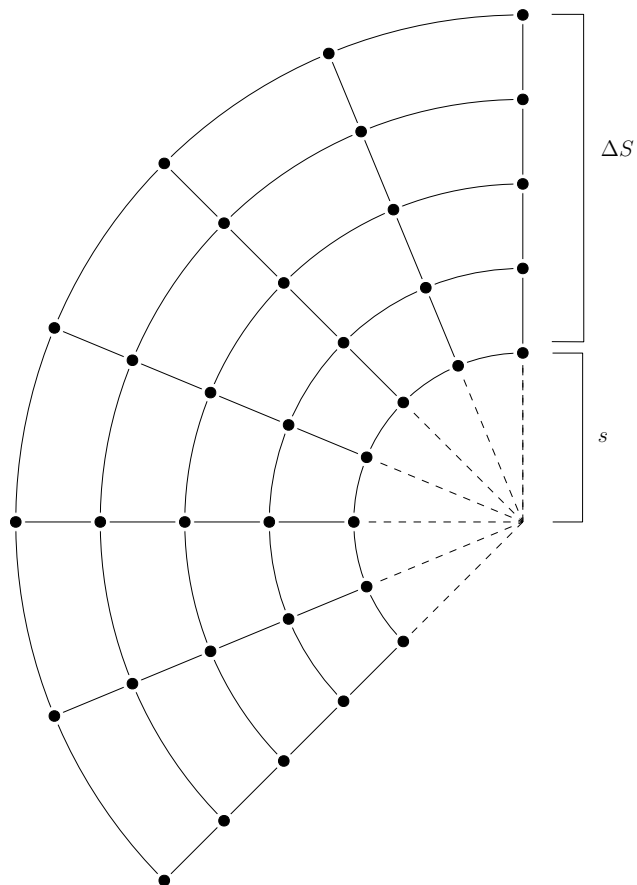


Abbildung 3.2: Kalibrierungsmuster entfaltet mit  $n = 5, m = 6$

Das Muster ist also

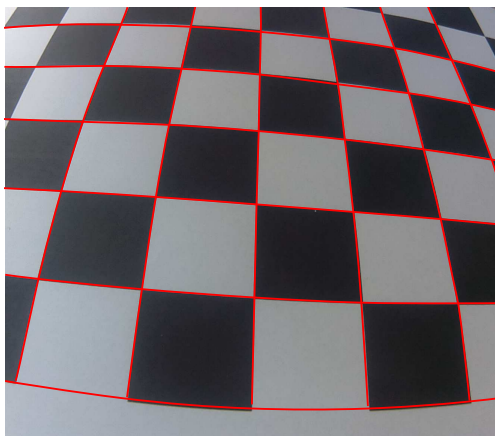
### 3.1.1 Anzahl der Samples

Die Anzahl der Samples sollte groß genug sein, um möglichst viel 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 klein wird, was eine Detektion erschwert. Des Weiteren sollte noch ein möglichst großer Teil der Kreislinie zu sehen bleiben, da diese für die Ellipsendetektion benötigt werden.

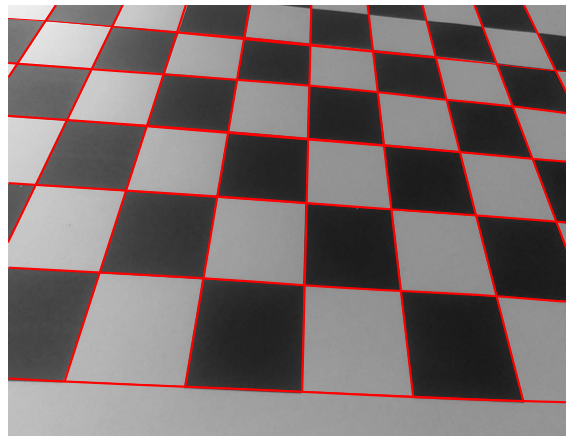
Bilder von Kegel mit Muster drin?

## 3.2 Intrinsische Kamerakalibrierung

Bedingt durch die Wahl einer Weitwinkelkamera, enthält die Linse der Kamera eine starke tonnenförmige (nach außen gewölbte) Verzerrung. Ohne eine intrinsische Kamerakalibrierung würde. Es werden nur die intrinsischen Parameter benötigt.



(a) vor Kalibrierung



(b) nach Kalibrierung und Entzerrung

Abbildung 3.3: Kamerakalibrierung

## 3.3 Detektion der Samples

Nach der Kamerakalibrierung und entsprechender Entzerrung werden die Bildkoordinaten der Samples bestimmt. Dazu wird ein Blob-Detektor 2.7 benutzt. Um die Samples

siehe später, dazu mehr später????

warum gerade dieses Muster? warum ist das so gut? wie berechnet man das Muster? welche Eigenschaften hat es?

Warum die extrinsischen nicht? Kamera position gegeben dann ??

Vergleich ohne und mit Kalibrierung

Umbenennen

korrekt von den Kreislinien, sowie den Liniensegmenten trennen zu können, ist es wichtig dass die Umgebung eines Samples frei ist. .

Filtert Farbe (schwarz), kleine Flächen werden verworfen, Kreisförmigkeit, Konvexität

### 3.4 Ellipsen-Detektion

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.

Zunächst werden die Kanten mit Hilfe von Canny (2.6) detektiert (siehe Abbildung 3.4).

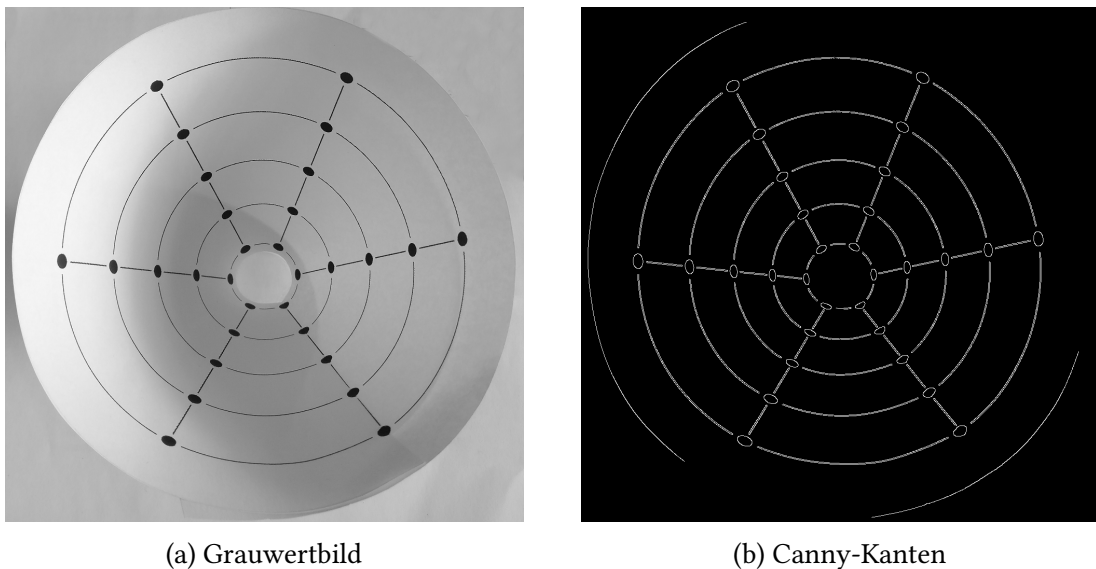


Abbildung 3.4: Canny-Kantendetektion auf Grauwertbild

Anschließend versuchen wir möglichst genau das Zentrum der innersten Ellipsen zu bestimmen. Wir benutzen dafür Hough-Transformationen, um Linien im Canny-Bild zu . Es werden anschließend Schnittpunkte aller Liniensegmente bestimmt. Bedingt durch Ungenauigkeiten beim Ausschneiden und Zusammenlegen im Kegel, schneiden sich nicht alle Liniensegmente in einem Punkt. Darüber hinaus werden, auf Grund der Liniendicke auf dem Kalibrierungsmuster, durch Canny viele Linien doppelt erkannt. Auch ein inhomogener Hintergrund, erschwert die Schnittpunktsbestimmung. 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 (siehe Abbildung 3.5).

Von diesem Schnittpunkt aus werden, 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 3.6(a) sind die getroffenen weißen Pixel und der zugehörige Aussendepunkt eingezeichnet.

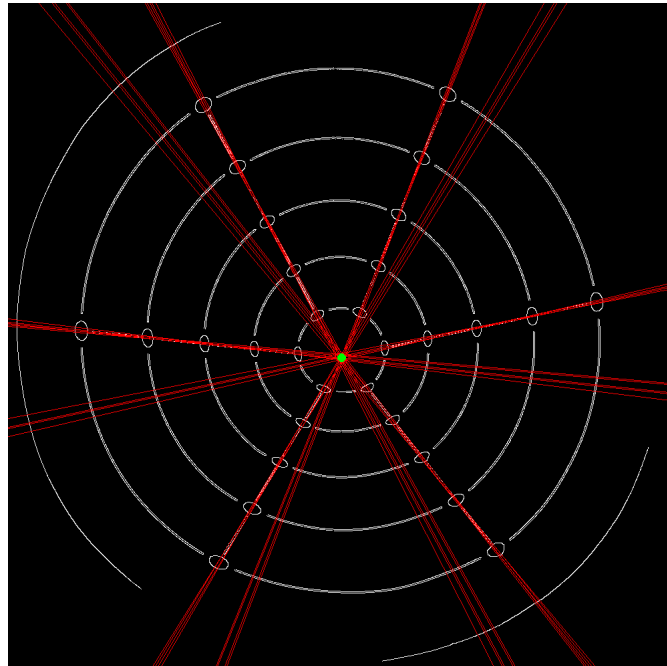
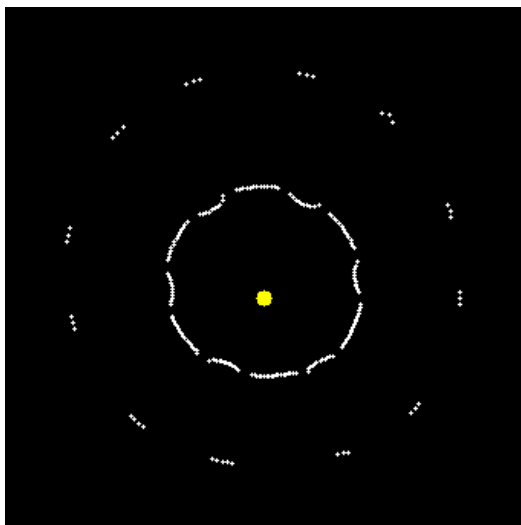
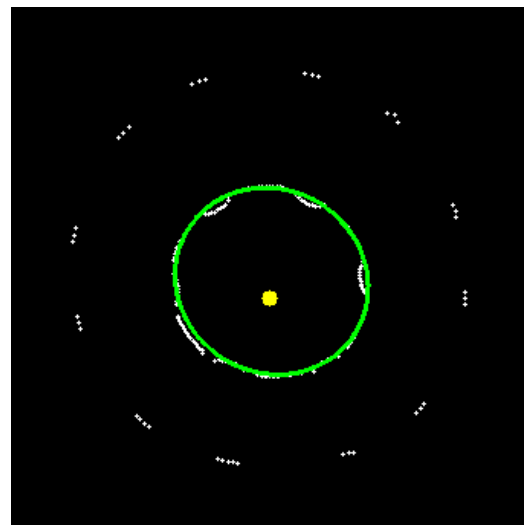


Abbildung 3.5: Hough-Transformation zur Linien-Detektion (in rot gekennzeichnet) und bestimmter Schnittpunkt (in grün)



(a) bestimme Pixel-Positionen

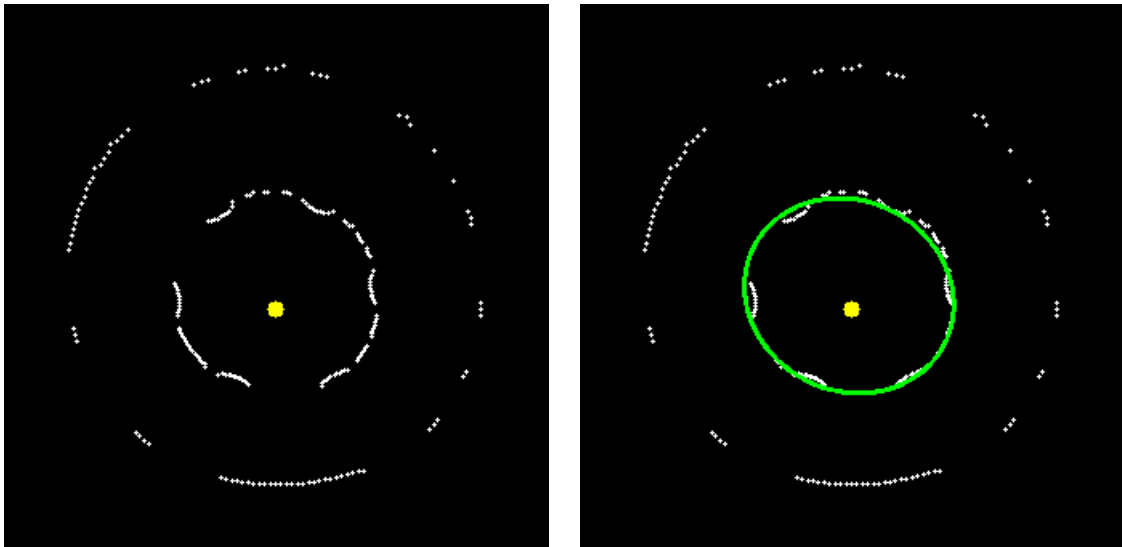


(b) bestimme Ellipse (grün)

Abbildung 3.6: Ellipsendetektion: bestimme Pixel-Positionen (weiß), Aussendepunkt (gelb)

Mit Hilfe der Positionen der weißen Pixel, wird anschließend durch RANSAC (siehe Kapitel 2.4) eine Ellipse geschätzt. Um die, für RANSAC benötigte, Distanz zu berechnen, wird das Verfahren aus Kapitel 2.3.2 genutzt, was die exakte euklidische Distanz eines Punktes zu einer Ellipse bestimmt. Ein Verfahren wie das Verfahren der kleinsten Quadra-

ten funktioniert hier nicht, dass die weißen Pixel bezüglich einer zu bestimmen Ellipse, Ausreißer behaftet sind. Wird zum Beispiel auf Grund schlechter Lichtverhältnisse eine Kreislinie nicht deutlich aufgenommen, kann es in dem Kantenbild (Abbildung 3.4) zu „Löchern“ in den Kreislinien und folglich treffen die ausgesendeten Strahlen die nächst äußere Kreislinie (siehe Abbildung 3.9). 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 2.4). Die letztendlich bestimmen Ellipsen sind beispielhaft in Abbildung 3.8 zu sehen.



(a) bestimme Pixel-Positionen

(b) bestimme Ellipse (grün)

Abbildung 3.7: Ellipsendetection bei Ausreißern

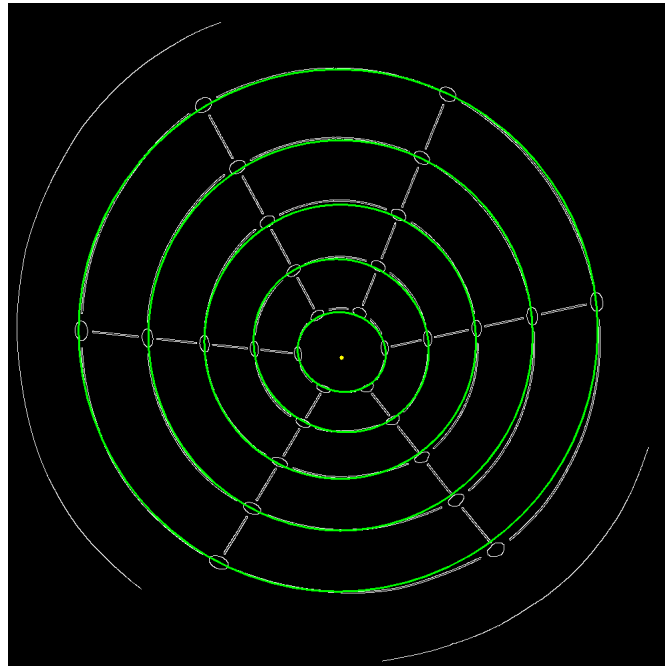


Abbildung 3.8: Ellipsen

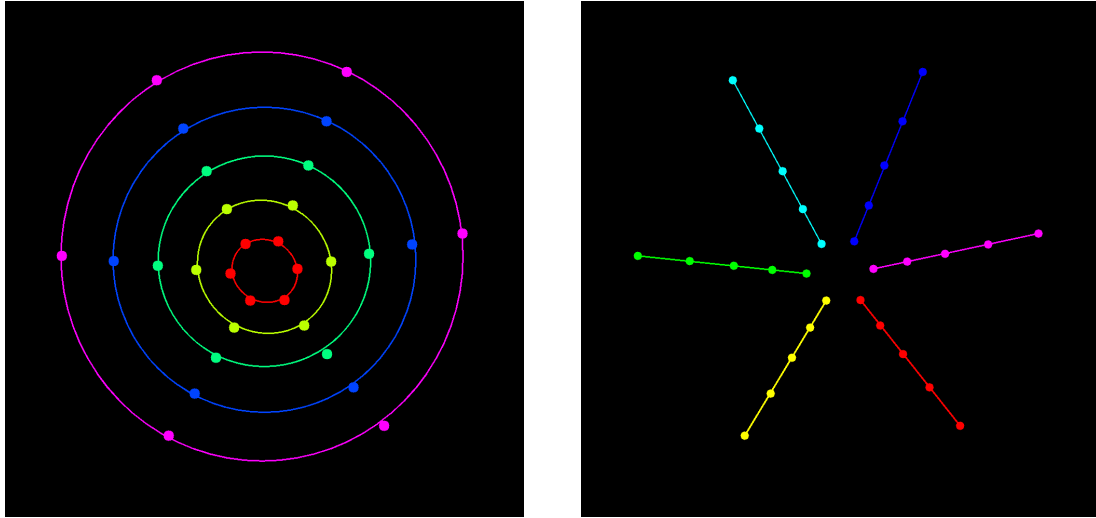
### 3.5 Zuordnung der Punkte

Nach der Bestimmung der Ellipsen muss jede Sample-Positionen der zugehörigen Kreislinie, sowie Liniensegment zugeordnet werden, um seine Position auf dem Kegel bestimmen zu können. Zunächst wird für jeden Punkte diejenige Kreislinie ausgewählt, dessen zugehörige Ellipse die kürzeste Distanz zu ihm hat (siehe Abbildung 3.9(a)).

Mit Hilfe dieser Zuordnung können die Ellipsen aus Kapitel 3.4 erneut geschätzt werden. Diesmal wird das Verfahren der kleinsten Quadrate genutzt, da nur die ausreißerfreien Samples als Messdaten dienen und wir eine optimale Lösung für alle Samples anstreben.

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 Ellipsen 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. Der Winkel zwischen der Samples auf einem Liniensegment und des bestimmen Mittelpunkts ist also nicht 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 3.9(b) zu sehen.



(a) Zuordnung von Punkten zu Ellipsen

(b) Zuordnung von Punkten zu Liniensegmenten

Abbildung 3.9: Zuordnung von Punkten zu Ellipsen (links) und Liniensegmenten (rechts)

### 3.6 Weltkoordinaten bestimmen

Aus dem vorherigen Kapitel wissen wir nun für jedes Sample den Ort auf dem Kegel. Wir können die 3D-Koordinaten also folgendermaßen angeben:

Ohne Beschränkung der Allgemeinheit, seien die Ellipsen  $i = 0, \dots, n - 1$  aufsteigend nach ihrer „Größe“<sup>1</sup> sortiert, so wie es das Verfahren in 3.4 beschreibt. Außerdem seien die Liniensegmente  $j = 0, \dots, m - 1$  aufsteigend nach Winkel mit der  $X$ -Achse, wie in 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.

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

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

$$y_{ij} = h_i$$

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

<sup>1</sup>Etwas 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.  $\pi \cdot a_0 \cdot b_0 \leq \pi \cdot a_1 \cdot b_1$



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

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

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

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

## 3.7 Entfaltung

Die eigentliche Entfaltung des Kegels, mit zwei unterschiedlichen Ansätzen realisiert werden. Die erste Möglichkeit ist die *Vorwärtse entfaltung*. 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 *Rückwärtse entfaltung* wird ein Punkt von der Mantelfläche zurück auf den Kegel abgebildet und von dort mit einer Projektionsmatrix auf die Bildebene projiziert und dann interpoliert.

Im folgenden wird genauer auf beide Verfahren eingegangen, sowie deren Probleme erläutert.

hier schon  
erläutern  
oder erst  
in späteren  
kapiteln?

### 3.7.1 Vorwärtse entfaltung

Bei der *Vorwärtse entfaltung* 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

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 ( $bl, br, tr, tl$ ). Diese Situation ist in Abbildung 3.10 illustriert.

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

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

Wo eine Multiplikation mit einem Skalar alle Charakteristika einer Ellipse skaliert<sup>2</sup>, und eine Addition elementweise addiert. Im nächsten Schritt wird der Schnittpunkt  $L$  mit dem Liniensegment  $\overline{bltl}$ , sowie der Schnittpunkt  $R$  mit dem Liniensegment  $\overline{brtr}$  bestimmt.

Da sich  $tl, L$  und  $bl$  nun auf einem gemeinsamen Liniensegment befinden, kann bezüglich der Weltkoordinaten linear interpoliert werden. Analoges gilt für  $R$ .

Die drei Punkte  $(L, P, R)$  befinden sich auf der Interpolationsellipse und somit können die Winkel  $(\phi_L, \phi_P, \phi_R)$  bezüglich des Ellipsenkoordinatensystems bestimmt werden.

<sup>2</sup>der Drehwinkel  $\theta$  wird hierbei anschließend  $\bmod 2\pi$  gerechnet

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

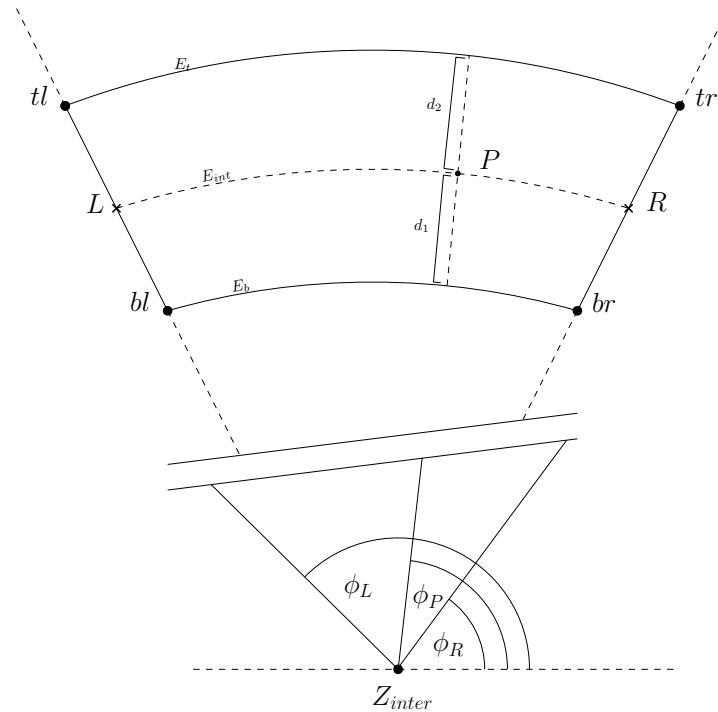


Abbildung 3.10: Interpolation

Probleme: bei der skizze geht ein punkt auf der kreislinie  $E_t$  mit der y-kordinate niemals über die y koordinate von  $tl$  und  $tr$  hinaus.

bei der transformaiton entstehen unvermeidbare löcher.

### 3.7.2 Rückwärtseinfaltung

Projektionsmatrix bestimmen

# 4 Implementierung

GUI mit bildern



# 5 Analyse

analyse?

projektionsfehler bei dem einen anzahl löcher bei dem anderen  
projektionsfehler ohne kalib ist wesentlich größer, da die implizite intrinsische kalib durch die svd keine verzerrung entfernt

kann man untersuchen wie sich die intrinsische kalibrierung auswirkt?

datenerhebung oder so was? laufzeit?

robustheit ransac ellipse?

robustheit reverse warping? bestimmung der keypoints?

laufzeitdurchschnitt pro bild auf raspberry von 1000 bildern bei verschiedenen auflösungen

Raspberry Pi 2 Model B:

- 900MHz ARM Cortex-A7 CPU
- 1GB RAM
- GCC-4.??? Release
- Opencv???, Qt????

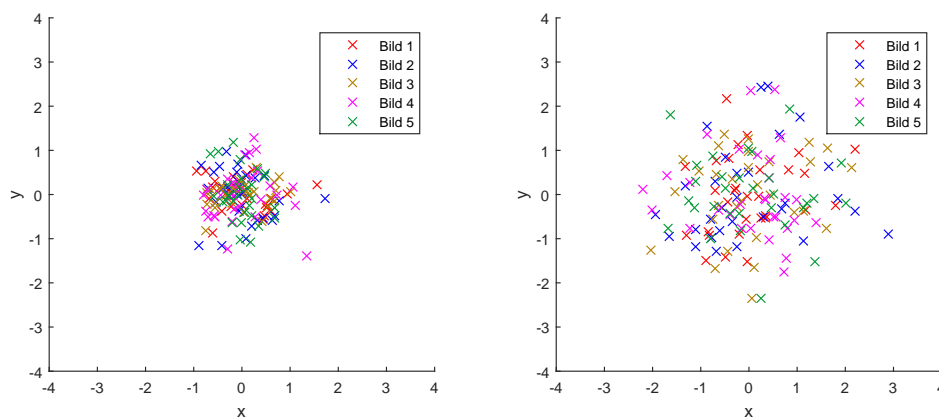


Abbildung 5.1: Einfluss der intrinsischen Kalibrierung

reprojection fehler unter verschiedenen Sichtwinkeln, Lichtverhältnissen?

LSQ irgend-  
wo erklären

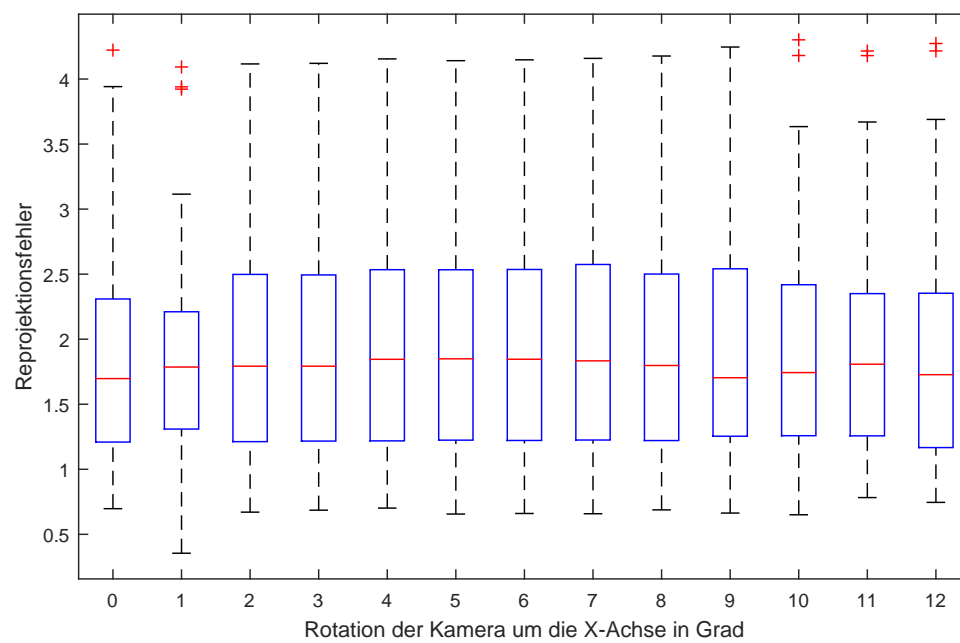


Abbildung 5.2: Einfluss der Rotation der Kamera

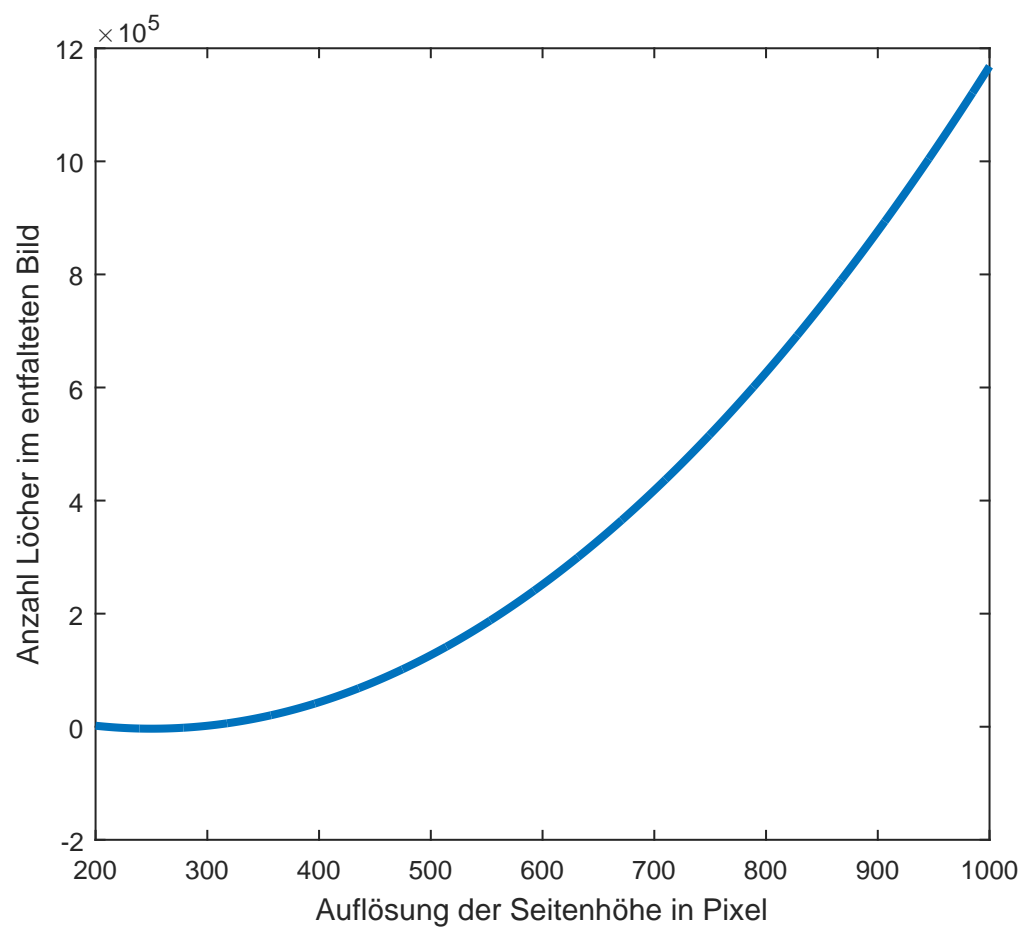
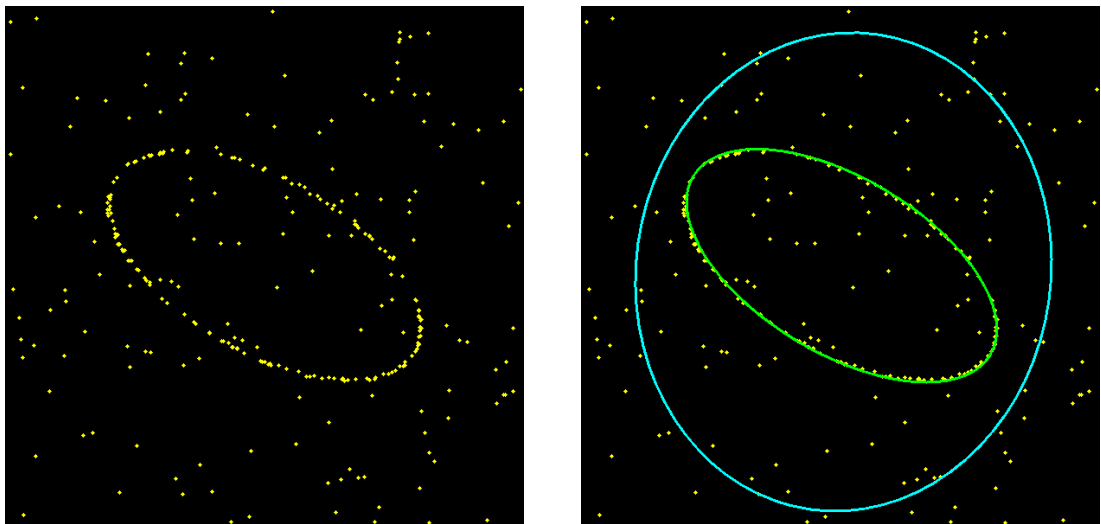


Abbildung 5.3: Einfluss der intrinischen Kalibrierung



(a) gestörte Messdaten

(b) detekt. Ellipsen: RANSAC (grün), LSQ (cyan)

Abbildung 5.4: Vergleich RANSAC und LSQ bei gleichverteilten Ausreißern  $\epsilon = 0.5, p = 0.99$



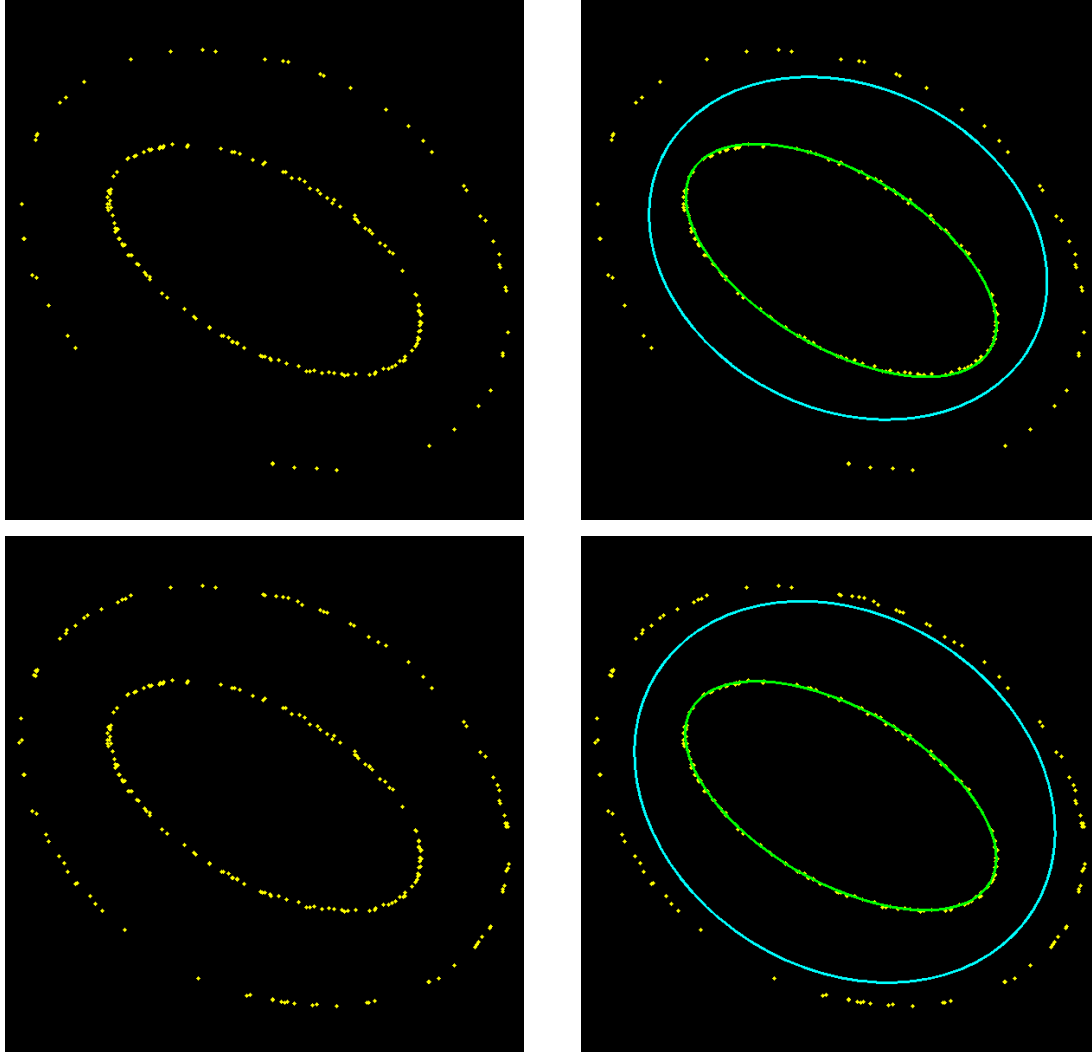


Abbildung 5.5: Vergleich RANSAC und LSQ bei Schattenellipsen mit  $p = 0.99$  und  $\epsilon = 0.25$  (oben),  $\epsilon = 0.4$  unten, links gestörte Messdaten, rechts detektierte Ellipsen RANSAC (grün), LSQ (cyan)

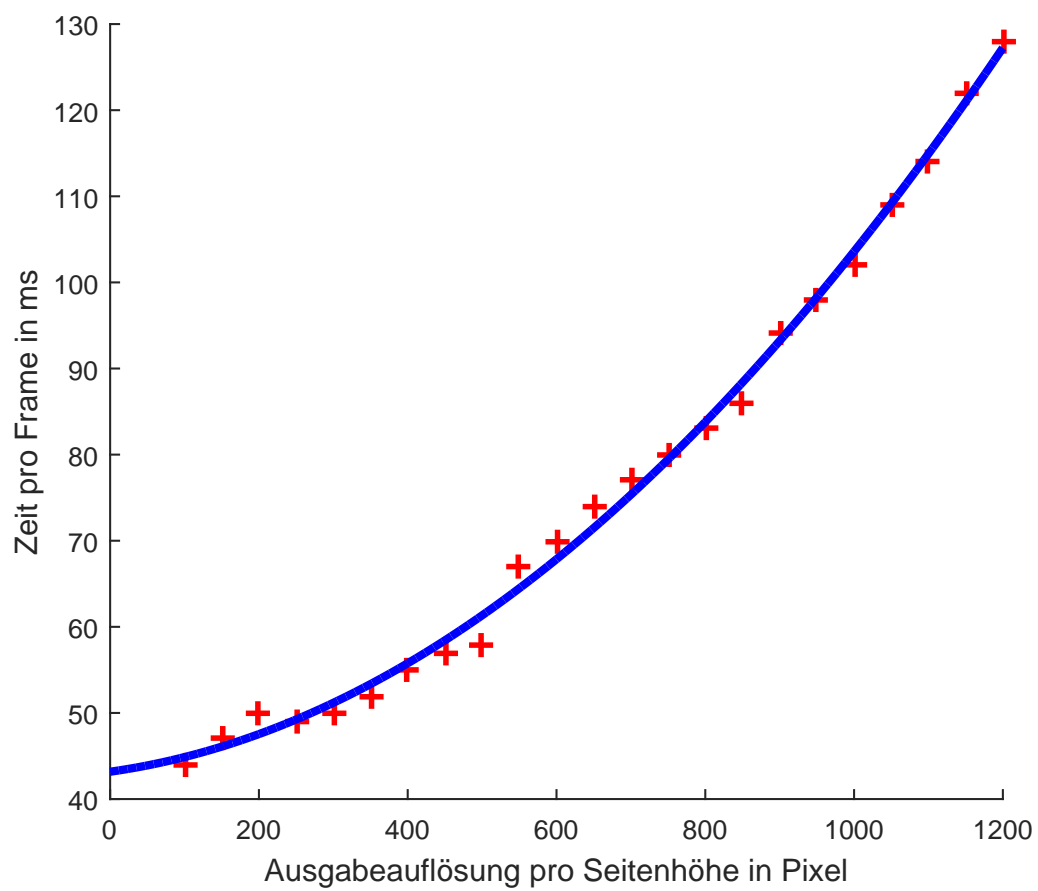


Abbildung 5.6: Einfluss der intrinsichen Kalibrierung

## 6 Fazit und Ausblick

Parallelisierung?

delaunay? lohnt sich eigentlich nicht, weil reverse sehr gute ergebnisse

deformable templates als alternative zu ransac für ellipsen? probleme mit lokalen minima

deformable templates

evtl noch am ende delaunay

iteratives verfahren zum minimieren des reprojection fehlers ausprobieren



# Abbildungsverzeichnis

2.1	Gerader Kreiskegel . . . . .	3
2.2	Kegelstumpf und Ergänzungskegel . . . . .	4
2.3	Kegelstumpf . . . . .	5
2.4	Kegelmantelfläche . . . . .	5
2.5	Abbildung der Kegelstumpfhöhe auf die Seitenhöhe . . . . .	6
2.6	Projektion eines Punktes $P$ im Weltkoordinatensystem $(x_w, y_w, z_w)$ auf die Bildebene im Kamerakoordinatensystem $(x_c, y_c, z_c)$ . . . . .	8
2.7	Ellipse mit Zentrum $(x_0, y_0)$ , Hauptachse $a$ , Nebenachse $b$ , sowie Drehwinkel $\theta$ . . . . .	10
2.8	Ellipsenausschnitt im ersten Quadranten mit Abfragepunkt $Q$ und eingezeichneter kürzester Distanz zur Ellipse . . . . .	13
3.1	Kalibrierungsmuster von oben mit $n = 5, m = 6$ . . . . .	18
3.2	Kalibrierungsmuster entfaltet mit $n = 5, m = 6$ . . . . .	18
3.3	Kamerakalibrierung . . . . .	19
3.4	Canny-Kantendetektion auf Grauwertbild . . . . .	20
3.5	Hough-Transformation zur Linien-Detektion (in rot gekennzeichnet) und bestimmter Schnittpunkt (in grün) . . . . .	21
3.6	Ellipsendetektion: bestimme Pixel-Positionen (weiß), Aussendepunkt (gelb)	21
3.7	Ellipsendetektion bei Ausreißern . . . . .	22
3.8	Ellipsen . . . . .	23
3.9	Zuordnung von Punkten zu Ellipsen (links) und Liniensegmenten (rechts)	24
3.10	Interpolation . . . . .	26
5.1	Einfluss der intrinsischen Kalibrierung . . . . .	29
5.2	Einfluss der Rotation der Kamera . . . . .	30
5.3	Einfluss der intrinsischen Kalibrierung . . . . .	31
5.4	Vergleich RANSAC und LSQ bei gleichverteilten Ausreißern $\epsilon = 0.5, p = 0.99$	32
5.5	Vergleich RANSAC und LSQ bei Schattenellipsen mit $p = 0.99$ und $\epsilon = 0.25$ (oben), $\epsilon = 0.4$ unten, links gestörte Messdaten, rechts detektierte Ellipsen RANSAC (grün), LSQ (cyan) . . . . .	33



# **Tabellenverzeichnis**





# Literatur

- [Can86] John Canny. „A Computational Approach to Edge Detection“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6 (1986), S. 679–698. ISSN: 01628828. DOI: 10.1109/TPAMI.1986.4767851 (siehe S. 16).
- [Ebe13] David Eberly. „Distance from a Point to an Ellipse, an Ellipsoid, or a Hyperellipsoid“. In: (2013), S. 1–13 (siehe S. 12).
- [FB81] Martin a Fischler und Robert C Bolles. „Random Sample Consensus: A Paradigm for Model Fitting with Applicatlons to Image Analysis and Automated Cartography“. In: *Communications of the ACM* 24.6 (1981), S. 381–395. ISSN: 0001-0782. DOI: 10.1145/358669.358692. URL: <http://dx.doi.org/10.1145/358669.358692> (siehe S. 15).
- [HS97] Janne Heikkilä und Olli Silvén. „A Four-step Camera Calibration Procedure with Implicit Image Correction.“ In: *Cvpr* (1997), S. 1106–1112. ISSN: 1063-6919. DOI: 10.1109/CVPR.1997.609468. URL: <http://dx.doi.org/10.1109/CVPR.1997.609468> <http://publication/doi/10.1109/CVPR.1997.609468>.
- [Law72] J D Lawrence. *A Catalog of Special Plane Curves*. Dover Publications, 1972, S. 62–63 (siehe S. 12).
- [Tsa87] Roger Y. Tsai. „A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses“. In: *IEEE Journal on Robotics and Automation* 3.4 (1987), S. 323–344. ISSN: 08824967. DOI: 10.1109/JRA.1987.1087109 (siehe S. 8).



# 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, 3. 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, 3. September 2016