

# Robuste Ausgleichung einer Kugel mithilfe des Gauß-Helmert-Modells und des RANSAC-Algorithmus

Die Ausgleichung einer Kugel in einer Punktwolke kann mit einer entsprechenden Parametrisierung in einem Gauß-Helmert-Modell erfolgen (siehe Abschnitt 1). Da die Punktwolke im Anwendungsfall nicht nur die Kugeloberfläche, sondern auch andere Objekte (Untersatz und Stativ) wiedergibt und auch Ausreißer beinhaltet, wäre die ausgeglichene Kugel unter Berücksichtigung der gesamten Punktwolke verfälscht. Mithilfe des RANSAC-Algorithmus können, die Punkte, die die Kugeloberfläche repräsentieren, automatisiert detektiert werden und eine unverfälschte Kugel ausgeglichen werden (siehe Abschnitt 2).

## 1 Ausgleichung einer Kugel mithilfe des Gauß-Helmert-Modells

Zur Ausgleichung einer Kugel wird eine entsprechende Parametrisierung, die die Beobachtungen  $\mathbf{l}$  (Koordinaten der Punkte auf der Kugeloberfläche; siehe Gleichung 2) und die Parameter  $\mathbf{x}$  (Koordinaten des Mittelpunktes und Radius der Kugel; siehe Gleichung 3) in einen Zusammenhang bringt, benötigt.

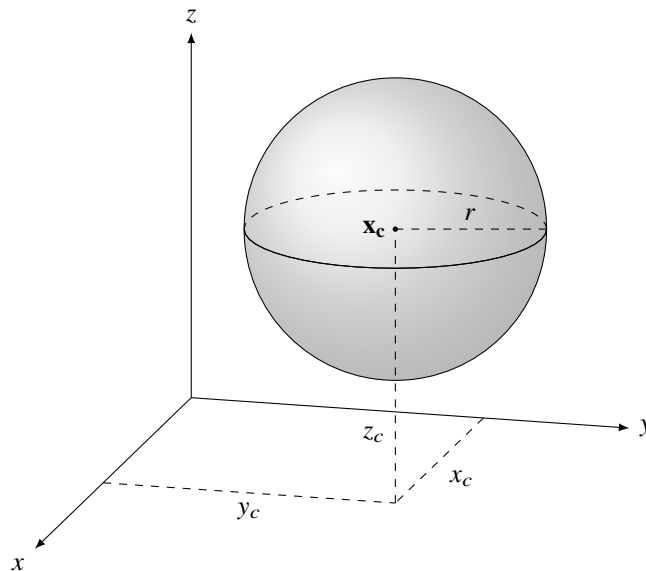


Abbildung 1: Parametrisierung einer Kugel

Für fehlerfreie Beobachtungen  $\mathbf{l}$  lässt sich folgender Zusammenhang festhalten.

$$f_i(\mathbf{l}, \mathbf{x}) = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2} - r = 0 \quad \text{mit} \quad i = 1, \dots, n \quad (1)$$

Beobachtungen  $\mathbf{l}$ :

$$\mathbf{l} = [x_1 \quad y_1 \quad z_1 \quad \dots \quad x_n \quad y_n \quad z_n]^T \quad (2)$$

Parameter  $\mathbf{x}$ :

$$\mathbf{x} = [x_c \quad y_c \quad z_c \quad r]^T \quad (3)$$

Unter der Annahme normalverteilter, zufälliger Messabweichungen  $\mathbf{l} \sim N(\mathbf{0}, \Sigma_{\mathbf{l}})$  mit der Kovarianzmatrix der Beobachtungen  $\Sigma_{\mathbf{l}}$  können die Parameter  $\mathbf{x}$  anhand der Beobachtungen  $\mathbf{l}$  im Sinne einer Kleinsten-Quadrate-Ausgleichung, bei der die Summe der quadratischen Verbesserungen  $\mathbf{v}^T \Sigma_{\mathbf{l}}^{-1} \mathbf{v}$  minimiert wird, geschätzt werden. Die Kovarianzmatrix der Beobachtungen  $\Sigma_{\mathbf{l}}$  kann bei gleichgewichteten, unkorrelierten Beobachtungen als Einheitsmatrix angenommen werden. Da im vorliegenden funktionalen Modell (siehe Gleichungen 4 und 5) die Beobachtungen  $\mathbf{l}$  nicht als Funktion der Parameter  $\mathbf{x}$  ausgedrückt werden können, wird das Gauß-Helmert-Modell

(Allgemeinfall der Ausgleichsrechnung), welches eine Verallgemeinerung des Gauß-Markov-Modells darstellt, verwendet.

$$f_i(\mathbf{l} + \mathbf{v}, \mathbf{x}) = \sqrt{(x_i + v_{x_i} - x_c)^2 + (y_i + v_{y_i} - y_c)^2 + (z_i + v_{z_i} - z_c)^2} - r = 0 \quad \text{mit} \quad i = 1, \dots, n \quad (4)$$

Verbesserungen  $\mathbf{v}$ :

$$\mathbf{v} = [v_{x_1} \quad \dots \quad v_{x_n} \quad v_{y_1} \quad \dots \quad v_{y_n} \quad v_{z_1} \quad \dots \quad v_{z_n}]^T \quad (5)$$

Da der in Gleichung 4 gegebene funktionale Zusammenhang nicht linear ist, erfolgt eine Linearisierung an der Stelle  $\mathbf{v}_0$  und  $\mathbf{x}_0$  mithilfe einer Taylorreihenentwicklung bis zum linearen Term, um die Verbesserungen  $\mathbf{v}$  und die Zuschläge  $\Delta \mathbf{x}$  iterativ zu berechnen. Die Verbesserungen  $\mathbf{v}_0$  und die Parameter  $\mathbf{x}_0$  sind jeweils die ausgeglichenen Werte der vorangegangenen Iteration. In der ersten Iteration sind die Verbesserungen  $\mathbf{v}_0 = \mathbf{0}$  und die Parameter  $\mathbf{x}_0$  Näherungswerte, die heuristisch ermittelt werden müssen. Das linearisierte, funktionale Modell kann mit der Designmatrix  $\mathbf{A}$  (siehe Gleichung 9) und der Bedingungsmatrix  $\mathbf{B}$  (siehe Gleichung 10) angeschrieben werden.

$$\mathbf{A}(\mathbf{x} - \mathbf{x}_0) + \mathbf{B}(\mathbf{v} - \mathbf{v}_0) + f(\mathbf{l} + \mathbf{v}_0, \mathbf{x}_0) = \mathbf{0} \quad (6)$$

Durch Definition der Widersprüche  $\mathbf{w}$  (siehe Gleichung 8) und Einsetzen in Gleichung 6 erhält man folgende Form.

$$\mathbf{A}\Delta \mathbf{x} + \mathbf{B}\mathbf{v} + \mathbf{w} = \mathbf{0} \quad \text{mit} \quad \Delta \mathbf{x} = \mathbf{x} - \mathbf{x}_0 \quad (7)$$

Widersprüche  $\mathbf{w}$ :

$$\mathbf{w} = -\mathbf{B}\mathbf{v}_0 + f(\mathbf{l} + \mathbf{v}_0, \mathbf{x}_0) = \begin{bmatrix} \frac{(x_1 + v_{x_1} - x_c)(x_1 - x_c) + (y_1 + v_{y_1} - y_c)(y_1 - y_c) + (z_1 + v_{z_1} - z_c)(z_1 - z_c) - r \cdot r_1}{r_1} \\ \vdots \\ \frac{(x_n + v_{x_n} - x_c)(x_n - x_c) + (y_n + v_{y_n} - y_c)(y_n - y_c) + (z_n + v_{z_n} - z_c)(z_n - z_c) - r \cdot r_n}{r_n} \end{bmatrix} \quad (8)$$

Designmatrix  $\mathbf{A}$ :

$$\mathbf{A} = \left. \frac{\partial f(\mathbf{l} + \mathbf{v}, \mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{l} + \mathbf{v}_0, \mathbf{x}_0} = \begin{bmatrix} \frac{\partial f_1}{\partial x_c} & \frac{\partial f_1}{\partial y_c} & \frac{\partial f_1}{\partial z_c} & \frac{\partial f_1}{\partial r} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial f_n}{\partial x_c} & \frac{\partial f_n}{\partial y_c} & \frac{\partial f_n}{\partial z_c} & \frac{\partial f_n}{\partial r} \end{bmatrix} = \begin{bmatrix} -\frac{x_1 + v_{x_1} - x_c}{r_1} & -\frac{y_1 + v_{y_1} - y_c}{r_1} & -\frac{z_1 + v_{z_1} - z_c}{r_1} & -1 \\ \vdots & \vdots & \vdots & \vdots \\ -\frac{x_n + v_{x_n} - x_c}{r_n} & -\frac{y_n + v_{y_n} - y_c}{r_n} & -\frac{z_n + v_{z_n} - z_c}{r_n} & -1 \end{bmatrix} \quad (9)$$

Bedingungsmatrix  $\mathbf{B}$ :

$$\mathbf{B} = \left. \frac{\partial f(\mathbf{l} + \mathbf{v}, \mathbf{x})}{\partial \mathbf{v}} \right|_{\mathbf{l} + \mathbf{v}_0, \mathbf{x}_0} = \begin{bmatrix} \frac{\partial f_1}{\partial v_{x_1}} & \dots & \frac{\partial f_1}{\partial v_{x_n}} & \frac{\partial f_1}{\partial v_{y_1}} & \dots & \frac{\partial f_1}{\partial v_{y_n}} & \frac{\partial f_1}{\partial v_{z_1}} & \dots & \frac{\partial f_1}{\partial v_{z_n}} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial v_{x_1}} & \dots & \frac{\partial f_n}{\partial v_{x_n}} & \frac{\partial f_n}{\partial v_{y_1}} & \dots & \frac{\partial f_n}{\partial v_{y_n}} & \frac{\partial f_n}{\partial v_{z_1}} & \dots & \frac{\partial f_n}{\partial v_{z_n}} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{x_1 + v_{x_1} - x_c}{r_1} & 0 & \dots & 0 & \frac{y_1 + v_{y_1} - y_c}{r_1} & 0 & \dots & 0 & \frac{z_1 + v_{z_1} - z_c}{r_1} & 0 & \dots & 0 \\ 0 & \ddots & \vdots & 0 & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & \ddots & 0 & \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \dots & 0 & \frac{x_n + v_{x_n} - x_c}{r_n} & 0 & \dots & 0 & \frac{y_n + v_{y_n} - y_c}{r_n} & 0 & \dots & 0 & \frac{z_n + v_{z_n} - z_c}{r_n} \end{bmatrix} \quad (10)$$

Hilfsvariable  $r_i$ :

$$r_i = \sqrt{(x_i + v_{x_i} - x_c)^2 + (y_i + v_{y_i} - y_c)^2 + (z_i + v_{z_i} - z_c)^2} \quad (11)$$

Die ursprüngliche Minimumsforderung für  $\mathbf{v}^T \Sigma_{\Pi}^{-1} \mathbf{v}$  muss um die Nebenbedingung aus Gleichung 7 erweitert werden. Das resultierende Problem kann als Extremwertaufgabe mit Nebenbedingungen verstanden werden und unter Verwendung der Lagrange-Multiplikatoren  $\lambda$ , die formal durch die Korrelaten  $\mathbf{k}$  ( $2\mathbf{k} = \lambda$ ) ersetzt werden, folgende zu minimierende Funktion angeschrieben werden.

$$f(\mathbf{l} + \mathbf{v}, \mathbf{x}) = \mathbf{v}^T \Sigma_{\Pi}^{-1} \mathbf{v} - 2\mathbf{k}^T (\mathbf{A}\Delta \mathbf{x} + \mathbf{B}\mathbf{v} + \mathbf{w}) \stackrel{!}{=} \min. \quad (12)$$

Das Minimum erhält man, indem man die partiellen Ableitungen nach  $\mathbf{v}$ ,  $\mathbf{x}$  und  $\mathbf{k}$  der obigen Funktion gleich Null setzt.

$$\frac{\partial f(\mathbf{l} + \mathbf{v}, \mathbf{x})}{\partial \mathbf{v}} = 2\mathbf{v}^T \Sigma_{\parallel}^{-1} - 2\mathbf{k}^T \mathbf{B} \stackrel{!}{=} \mathbf{0} \Rightarrow \mathbf{v} = \Sigma_{\parallel} \mathbf{B}^T \mathbf{k} \quad (13)$$

$$\frac{\partial f(\mathbf{l} + \mathbf{v}, \mathbf{x})}{\partial \mathbf{x}} = -2\mathbf{k}^T \mathbf{A} \stackrel{!}{=} \mathbf{0} \Rightarrow \mathbf{A}^T \mathbf{k} = \mathbf{0} \quad (14)$$

$$\frac{\partial f(\mathbf{l} + \mathbf{v}, \mathbf{x})}{\partial \mathbf{k}} = -2(\mathbf{A}\Delta\mathbf{x} + \mathbf{B}\mathbf{v} + \mathbf{w}) \stackrel{!}{=} \mathbf{0} \Rightarrow \mathbf{A}\Delta\mathbf{x} + \mathbf{B}\mathbf{v} + \mathbf{w} = \mathbf{0} \quad (15)$$

Durch Einsetzen der Gleichung 13 in 15 erhält man folgende Gleichung.

$$\mathbf{A}\Delta\mathbf{x} + \mathbf{B}\Sigma_{\parallel} \mathbf{B}^T \mathbf{k} + \mathbf{w} = \mathbf{0} \Rightarrow \mathbf{A}\Delta\mathbf{x} + \mathbf{B}\Sigma_{\parallel} \mathbf{B}^T \mathbf{k} = -\mathbf{w} \quad (16)$$

Aus den Gleichungen 16 und 14 lässt sich folgendes Gleichungssystem formulieren.

$$\begin{bmatrix} \mathbf{B}\Sigma_{\parallel} \mathbf{B}^T & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{k} \\ \Delta\mathbf{x} \end{bmatrix} = \begin{bmatrix} -\mathbf{w} \\ \mathbf{0} \end{bmatrix} \quad (17)$$

Durch Inversion ist dieses Gleichungssystem direkt lösbar und kann zur Bestimmung der Zuschläge der Parameter  $\Delta\mathbf{x}$  verwendet werden.

$$\begin{bmatrix} \mathbf{B}\Sigma_{\parallel} \mathbf{B}^T & \mathbf{A} \\ \mathbf{A}^T & \mathbf{0} \end{bmatrix}^{-1} \begin{bmatrix} -\mathbf{w} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathbf{k} \\ \Delta\mathbf{x} \end{bmatrix} \quad (18)$$

Alternativ kann zur Bestimmung der Zuschläge  $\Delta\mathbf{x}$  die Gleichung 16 wie folgt umgeformt werden.

$$\mathbf{k} = -(\mathbf{B}\Sigma_{\parallel} \mathbf{B}^T)^{-1}(\mathbf{A}\Delta\mathbf{x} + \mathbf{w}) \Rightarrow \mathbf{A}^T \mathbf{k} = -\mathbf{A}^T (\mathbf{B}\Sigma_{\parallel} \mathbf{B}^T)^{-1}(\mathbf{A}\Delta\mathbf{x} + \mathbf{w}) \quad (19)$$

Durch Einsetzen der Gleichung 14 in 19 und anschließender Umformung gelangt man zu folgendem Ausdruck zur Bestimmung der Zuschläge  $\Delta\mathbf{x}$ :

$$\mathbf{0} = -\mathbf{A}^T (\mathbf{B}\Sigma_{\parallel} \mathbf{B}^T)^{-1}(\mathbf{A}\Delta\mathbf{x} + \mathbf{w}) \Rightarrow \Delta\mathbf{x} = -(\mathbf{A}^T (\mathbf{B}\Sigma_{\parallel} \mathbf{B}^T)^{-1} \mathbf{A})^{-1} \mathbf{A}^T (\mathbf{B}\Sigma_{\parallel} \mathbf{B}^T)^{-1} \mathbf{w} \quad (20)$$

Alternative Schreibweise mit der Normalgleichungsmatrix  $\mathbf{N}$ :

$$\Delta\mathbf{x} = -\mathbf{N}^{-1} \mathbf{A}^T (\mathbf{B}\Sigma_{\parallel} \mathbf{B}^T)^{-1} \mathbf{w} \quad \text{mit} \quad \mathbf{N} = \mathbf{A}^T (\mathbf{B}\Sigma_{\parallel} \mathbf{B}^T)^{-1} \mathbf{A} \quad (21)$$

Die ausgeglichenen Parameter  $\mathbf{x}$  ergeben sich durch Addition der Näherungswerte  $\mathbf{x}_0$  und der Zuschläge  $\Delta\mathbf{x}$  und dienen wiederum als Näherungswerte  $\mathbf{x}_0$  für die nächste Iteration:

$$\mathbf{x} = \mathbf{x}_0 + \Delta\mathbf{x} \quad (22)$$

Anhand der Gleichung 19 (links) lassen sich die Korrelaten  $\mathbf{k}$  und daraus mit Gleichung 13 (rechts) die Verbesserungen  $\mathbf{v}$ , welche in der nächsten Iteration als Näherungswerte  $\mathbf{v}_0$  verwendet werden, bestimmen. Die Standardabweichung der Gewichtseinheit  $\sigma_0$  kann wie folgt bestimmt werden.

$$\sigma_0 = \sqrt{\frac{\mathbf{v}^T \Sigma_{\parallel}^{-1} \mathbf{v}}{n - m}} \quad \text{mit} \quad \begin{array}{ll} n & \dots \text{Anzahl der Beobachtungen (Punkte)} \\ m & \dots \text{Anzahl der Parameter} \end{array} \quad (23)$$

Die Genauigkeit der ausgeglichenen Parameter  $\mathbf{x}$  ergibt sich aus der Kovarianzmatrix  $\Sigma_{\mathbf{xx}}$ .

$$\Sigma_{\mathbf{xx}} = \sigma_0^2 \mathbf{N} = \sigma_0^2 \begin{bmatrix} \sigma_{x_c}^2 & \sigma_{x_c} \sigma_{y_c} & \sigma_{x_c} \sigma_{z_c} & \sigma_{x_c} \sigma_r \\ \sigma_{x_c} \sigma_{y_c} & \sigma_{y_c}^2 & \sigma_{y_c} \sigma_{z_c} & \sigma_{y_c} \sigma_r \\ \sigma_{x_c} \sigma_{z_c} & \sigma_{y_c} \sigma_{z_c} & \sigma_{z_c}^2 & \sigma_{z_c} \sigma_r \\ \sigma_{x_c} \sigma_r & \sigma_{y_c} \sigma_r & \sigma_{z_c} \sigma_r & \sigma_r^2 \end{bmatrix} \quad (24)$$

Nach jeder Iteration sollte überprüft werden, ob der Ausgleich zu einem Minimum konvergiert ist. Dies kann durch Vergleich der Zuschläge der Parameters  $\Delta\mathbf{x}$  mit einem geeigneten Schwellenwert  $\delta$  erfolgen. Nachdem Konvergenz erreicht wurde, sollte eine Hauptprobe durchgeführt werden. Dies kann durch Einsetzen der ausgeglichenen Parameter  $\mathbf{x}$  und der ausgeglichenen Verbesserungen  $\mathbf{v}$  in den grundlegenden funktionalen Zusammenhang (Gleichung 4) erfolgen.

**Implementierung in MATLAB:**

- Initialisierung: Bestimmung von Näherungswerten für die Kugelparameter  $\mathbf{x}_0$  und Definition der Verbesserungen  $\mathbf{v}_0 = \mathbf{0}$
- while-Schleife für  $k$  Iterationen der Ausgleichung
  - Aufstellung der Designmatrix  $\mathbf{A}$ , der Bedingungsmatrix  $\mathbf{B}$  und der Widersprüche  $\mathbf{w}$
  - Aufstellung der Normalgleichungsmatrix  $\mathbf{N}$
  - if-Abfrage, ob die Normalgleichungsmatrix  $\mathbf{N}$  invertierbar ist ( $\text{abs}(\det(\mathbf{N})) > \delta \rightarrow \text{invertierbar}$ )
    - ja  $\rightarrow$  Bestimmung der Zuschläge der Kugelparameter  $\Delta\mathbf{x}$ , der ausgeglichenen Kugelparameter  $\mathbf{x}$ , der Korrelaten  $\mathbf{k}$ , der Verbesserungen  $\mathbf{v}$ , der Standardabweichung der Gewichtseinheit  $\sigma_0$  und der Kovarianzmatrix der ausgeglichenen Kugelparameter  $\Sigma_{\mathbf{xx}}$
    - nein  $\rightarrow$  Abbruch der Iteration mit **break**  $\rightarrow$  keine gültige Lösung
  - Aktualisierung der Näherungswerte der Kugelparameter  $\mathbf{x}_0 = \mathbf{x}$  und Verbesserungen  $\mathbf{v}_0 = \mathbf{v}$  für die nächste Iteration
  - Abbruch der while-Schleife mit **break**, falls Konvergenz erreicht wurde ( $\max(\text{abs}(\Delta\mathbf{x})) < \delta \rightarrow \text{Hauptprobe } (\max(\text{abs}(f_i(\mathbf{l} + \mathbf{v}, \mathbf{x}))) < \delta) \rightarrow \text{gültige Lösung}$

**Anmerkung:** Die Ausgleichung einer Kugel mithilfe des Gauß-Helmert-Modells soll als Funktion implementiert werden, wobei die Anzahl der Beobachtungen (Punkte) variieren kann, da die Ausgleichung für den RANSAC-Algorithmus mit genau 4 und auch mit mehr Beobachtungen durchgeführt wird.

## 2 RANSAC-Algorithmus zur robusten Ausgleichung einer Kugel

Der RANSAC-Algorithmus (*RANdom SAMple Consensus*) ermöglicht die robuste Schätzung von Parametern, d.h. wenn die vorliegenden Beobachtungen nicht frei von Ausreißern ist und/oder nicht alle Beobachtungen das modellierte Phänomen beschreiben (*inlier* und *outlier*). Das Prinzip des Algorithmus beruht darauf, dass in einem iterativen Ablauf jeweils eine zufällige Teilmenge (*sample*) der vorliegenden Menge an Beobachtungen ausgewählt wird und daraus ein Modell (Kugelparameter) geschätzt wird. Die Teilmenge setzt sich jeweils aus einer minimalen Anzahl an Beobachtungen  $s$  zusammen, um die Parameter des Modells eindeutig bestimmen zu können. Anschließend werden jene Beobachtungen aus der ursprünglichen Menge ermittelt, welche zu diesem Modell passen (*inlier*), und mit diesen das Modell neu geschätzt. Nur bei einem Modell, das aus einer Teilmenge von Beobachtungen, die auch das zu modellierende Phänomen beschreiben, geschätzt wurde, werden auch viele Beobachtungen zu diesem Modell passen. Nach der Iteration ist jenes Modell mit den meisten *inlier* die optimale Lösung.

### Zufällige Teilmenge zur Schätzung einer Kugel:

Mit 4 Punkten, die auf der Kugeloberfläche und nicht in einer Ebene liegen, ist die Kugel eindeutig bestimmt. Aus diesem Grund muss in jeder Iteration eine zufällige Teilmenge der Beobachtungen (Punkte) im Umfang  $s = 4$  gewählt werden und nicht in einer Ebene liegen, um ein Modell schätzen zu können.

### Bestimmung der *inlier* und *outlier*:

Zur Bestimmung der *inlier* und *outlier* wird für jede Beobachtungen die verbleibende Abweichung  $d_{i,\perp}$  im Bezug auf das geschätzte Modell bestimmt und mit einem Schwellenwert  $t$  verglichen. Die Abweichungen  $d_{i,\perp}$  können für jeden Punkt  $i$  mit den ursprünglichen Beobachtungen  $\mathbf{l}$  und den ausgeglichenen Parametern  $\mathbf{x}$  bestimmt werden.

$$d_{i,\perp} = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2} - r \quad (25)$$

Der Schwellenwert  $t$  zur Bestimmung der *inlier* und *outlier* sollte an die Messgenauigkeit der Punkte  $\sigma$  angepasst werden und kann mit der Herstellerangabe angesetzt werden.

$$\begin{array}{ll} d_{i,\perp}^2 < t^2 & \dots \text{ inlier} \\ d_{i,\perp}^2 \geq t^2 & \dots \text{ outlier} \end{array} \quad \text{mit } t = \sigma \quad (26)$$

**Anzahl der Iterationen  $N$ :**

Da der RANSAC-Algorithmus in jeder Iteration eine zufällige Teilmenge aus sämtlichen Beobachtungen auswählt und nicht sämtliche Kombinationen berücksichtigt werden können, kann nur mit einer Wahrscheinlichkeit  $p$  die optimale Lösung gefunden werden. Die nötige Anzahl der Iterationen  $N$  kann mit dem Verhältnis  $\epsilon$  zwischen der Anzahl der *outlier*  $n_{outlier}$  und Anzahl sämtlicher Beobachtungen (Punkte)  $n$  bestimmt werden.

$$N = \frac{\log(1-p)}{\log(1-(1-\epsilon)^s)} \quad \text{mit} \quad \epsilon = \frac{n_{outliers}}{n} \quad (27)$$

Da im Anwendungsfall zu Beginn die Anzahl der *outlier* nicht bekannt ist, wird die Anzahl der nötigen Iterationen  $N$  adaptiv ermittelt. Zu Beginn wird die Anzahl der Iterationen  $N = \infty$  angesetzt und bei jeder Iteration mit einer Lösung mit der derzeit geringsten Anzahl an *outlier* mit der obigen Gleichung die nötige Anzahl an Iterationen  $N$  aktualisiert. Da der berechnete Wert von  $N$  im Allgemeinen keine Ganzzahl ist, muss dieser aufgerundet werden. Im Anwendungsfall kann eine Wahrscheinlichkeit  $p = 99\%$  verwendet werden.

**Implementierung in MATLAB:**

- Datengrundlage: Punktwolke aus Punkten auf der Kugeloberfläche (*inlier*) und anderen Punkten (*outlier*) im ASCII-Format (Spalten:  $x_i$ ,  $y_i$  und  $z_i$ )
- Einlesen der Punktwolke mit `importdata()`
- Reduktion der Punktwolke auf z.B. 1000 Punkte mit `randsample()`
- Bestimmung der Näherungswerte für die Kugelparameter z.B. durch Mittelwertbildung
- Initialisierung: Anzahl der Iterationen  $N = \infty$
- **while**-Schleife für  $N$  Iterationen des RANSAC-Algorithmus
  - Zufällige Auswahl einer Teilmenge der Punkte im Umfang  $s = 4$  (*sample*) mit `randsample()`
  - Bestimmung der Kugelparameter mithilfe des Gauß-Helmert-Modells durch Aufruf der selbst implementierten Funktion `gauss_helmert_model()` mit der Teilmenge aus 4 Punkten
  - **if**-Abfrage, ob eine Lösung bestimmt werden konnte (`abs(det(N)) > δ`)
    - ja → gültige Iteration
    - nein → keine gültige Iteration (zählt nicht als Iteration)
  - Bestimmung der Anzahl der *inlier* durch Vergleich der verbleibenden Abweichungen  $d_{i,\perp}^2$  mit dem Schwellenwert  $t^2$
  - **if**-Abfrage, ob die vorliegende Lösung die meisten *inlier* vorweist
    - ja → Bestimmung der Kugelparameter mithilfe des Gauß-Helmert-Modells durch Aufruf der selbst implementierten Funktion `gauss_helmert_model()` mit sämtlichen *inlier*
    - nein → Aktualisierung der nötigen Iterationen  $N$  des RANSAC-Algorithmus
  - Abbruch der **while**-Schleife mit **break** nach  $N$  Iterationen
- Die optimalen Kugelparameter sind jene, die mit den meisten *inlier* bestimmt wurden

**Literaturverzeichnis**

- [1] W. SCHWARZ (Hrsg.): *Ingenieurgeodäsie: Handbuch der Geodäsie, herausgegeben von Willi Freeden und Reiner Rummel*. Springer Berlin Heidelberg, 2017
- [2] R. HARTLEY & A. ZISSERMAN: *Multiple View Geometry in Computer Vision*. Second Edition, Cambridge University Press, 2004