

Schriftlicher Beleg zur Übung

# Random Sample Consensus (RANSAC)

im Modul

*G 05/GIT 02 Aktuelle Verfahren der  
photogrammetrischen Geodatenakquisition*

Autoren: Gianluca Höhnke & Simon Schäffler

Betreuung: Dr.-Ing. Ferdinand Maiwald

Modulverantwortlicher: Prof. Dr. habil. Hans-Gerd Maas

## 1. Praktische Aufgaben

### 1.1 Klassische Parameterschätzung nach Methode der kleinsten Quadrate

Abbildung 1: MKQ mit Schwellenwert 1000

### 3.2 Robuste Parameterschätzung mit RANSAC

Abbildung 3: RANSAC Durchführung

### 3.3 Einfluss der Streuparameter auf den RANSAC Algorithmus

Abbildung 4: RANSAC mit Teilmengenzahl = 4

Abbildung 5: RANSAC mit Fehlertoleranz = 0,01

## 2. Theoretische Aufgaben

### 2.1 Funktionsweise des RANSAC-Algorithmus

### 2.2 Vorteile von RANSAC in Kombination mit bekannten Ausgleichungsansätzen

### 2.3 Anwendungsbeispiele und typische Fehlerquellen des RANSAC-Algorithmus

#### 2.3.1 Relative Orientierung

#### 2.3.2 Räumlichen Rückwärtsschnitt

#### 2.3.3. Bündelblockausgleichung

### 2.4 RANSAC-Algorithmus für einen hyperbolischen Paraboloid in einer 3-D-Punktmenge

# 1. Praktische Aufgaben

## 1.1 Klassische Parameterschätzung nach Methode der kleinsten Quadrate

Führen Sie zunächst eine MKQ-Ausgleichung ohne Ausreißerdetektion durch. Wählen Sie dazu für den Schwellenwert zur Eliminierung großer Beobachtungen einen sehr großen Wert. Analysieren Sie das Ergebnis der Berechnung!

Als Schwellenwert wurde 1000 ausgewählt. Damit wurde eine Ebene erzeugt, die aber nicht das offensichtliche Punkt-Kluster abdecken konnte (vgl. Abb. 1).

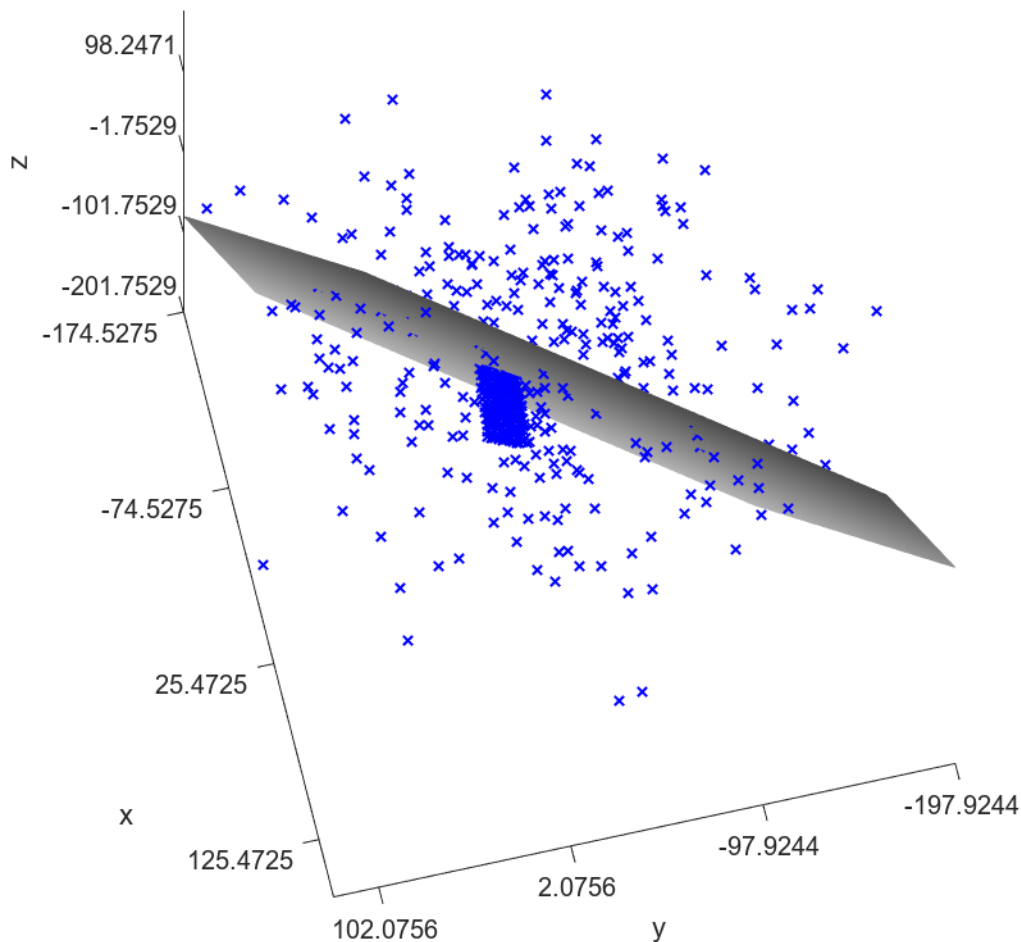


Abbildung 1: MKQ mit Schwellenwert 1000

Legen Sie anschließend einen sinnvollen Schwellenwert zur Eliminierung großer Beobachtungen fest. Begründen Sie Ihre Wahl!

Der Schwellenwert wurde auf 40 gesetzt. Der Wertebereich der Achsen liegt im Bereich von etwa 200 bis -200. Damit stellt die Zahl 40 ein 100stel des Wertebereichs einer Achse dar.

Wiederholen Sie die MKQ-Anpassung unter Ausschluss des jeweils größten bzw. letztlich aller Ausreißer für weitere Berechnungsschritte. Was stellen Sie bei der Ergebnisanalyse fest?

Nach fünf Iterationen hat die MKQ-Anpassung mit einem Schwellenwert von 40 und dem Ausschluss der Ausreißer sein Optimum erreicht. Weitere Iterationen führten zu keiner Veränderung der Ebene.

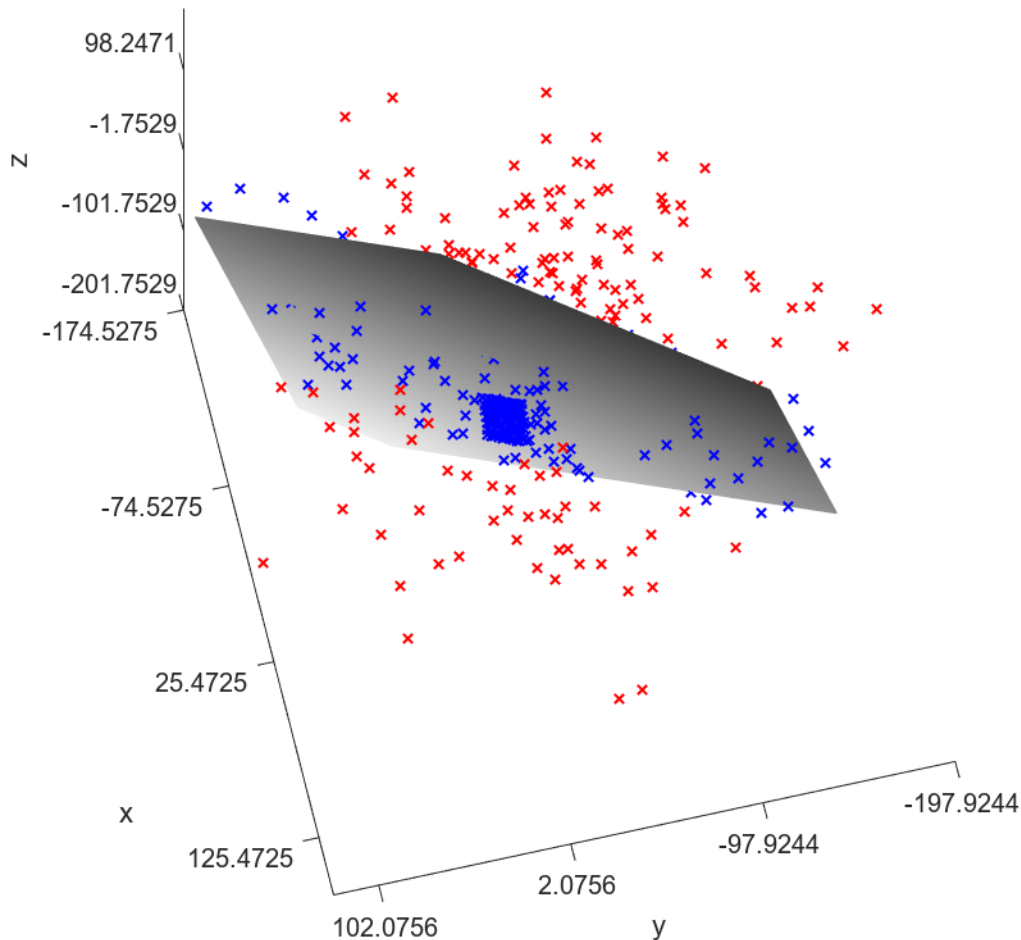


Abbildung 2: MKQ mit Schwellenwert 40 nach fünf Iterationen

Allgemein „drehte“ sich die Ebene von Iteration zu Iteration näher an das Cluster heran, konnte aber an keiner Stelle dieses mit sich selbst schneiden. Die MKQ-Ausgleichung konnte die Ebene verbessern, sie ist jedoch noch nicht für das Punkt-Cluster geeignet.

### 3.2 Robuste Parameterschätzung mit RANSAC

Berechnen Sie, wie viele Teilmengen maximal gezogen werden müssen, um mindestens ein richtiges Ergebnis zu finden! Stellen Sie sinnvolle Annahmen für fehlende Angaben auf (Wahrscheinlichkeit, dass der Algorithmus nicht die korrekte Lösung findet, Wahrscheinlichkeit, dass eine zufällig ausgewählte Beobachtung korrekt ist).

Für die Berechnung der zu ziehenden Teilmengen kann  $n_{samples}$  berechnet werden. Weiter kann anhand der Aufgabenstellung  $n_{good} = 144$ , &  $n_{total} = 444$  abgelesen werden. Zusätzlich wird nach einer Ebene gesucht, die die Punkte umfasst. So kann für den Parameter  $n_{min} = 3$  gewählt werden, da 3 Punkte für das Erstellen einer Ebene benötigt werden.

Als Irrtumswahrscheinlichkeit, oder  $P_{fail}$  wurde 1%, wie auch in der Vorlesung vorgeschlagen, verwendet. Dementsprechend ist die Wahrscheinlichkeit, dass eine zufällig ausgewählte Beobachtung korrekt ist, 99%. Zur Berechnung der  $n_{samples}$  wird die vereinfachte Formel (3.2.1) verwendet.

$n_{samples} = \frac{\log(P_{fail})}{\log\left(1 - \left(\frac{n_{good}}{n_{total}}\right)^{n_{min}}\right)}$	<b>(3.2.1)</b>
---	----------------

Mit den entsprechenden Werten (3.2.2) eingesetzt ergibt sich als Endergebnis 132,67, welches auf 133 Iterationen aufgerundet wird.

$n_{\text{samples}} = \frac{\log(0,01)}{\log(1 - (\frac{144}{444})^3)}$	<b>(3.2.2)</b>
---	----------------

Entscheiden Sie sich für eine sinnvolle Fehlertoleranz. Begründen Sie Ihre Wahl!

Die Fehlertoleranz kann anhand der Iterationen k, der Wahrscheinlichkeit eines richtigen Ergebnisses sowie der Gesamtzahl an Punkten berechnet werden (3.2.3). Als Wert für die Fehlertoleranz ergibt sich demnach 1.

$1 - p = (1 - w^n)^k$ $1 - 0,99 = (1 - w^{444})^{133}$ $w = 0,9908$	<b>(3.2.3)</b>
---	----------------

Legen Sie fest, wie viele Beobachtungen zum Modell passen müssen, damit die Modellparameter akzeptiert werden! Begründen Sie kurz Ihre Wahl!

Es wurde bereits vorab festgestellt, dass  $n_{\text{good}} = 144$  ist. Darauf basierend wurde eine Abweichung von  $\pm 5\%$  an Punkten gewählt.

Führen Sie die robuste Parameterschätzung mit RANSAC durch!

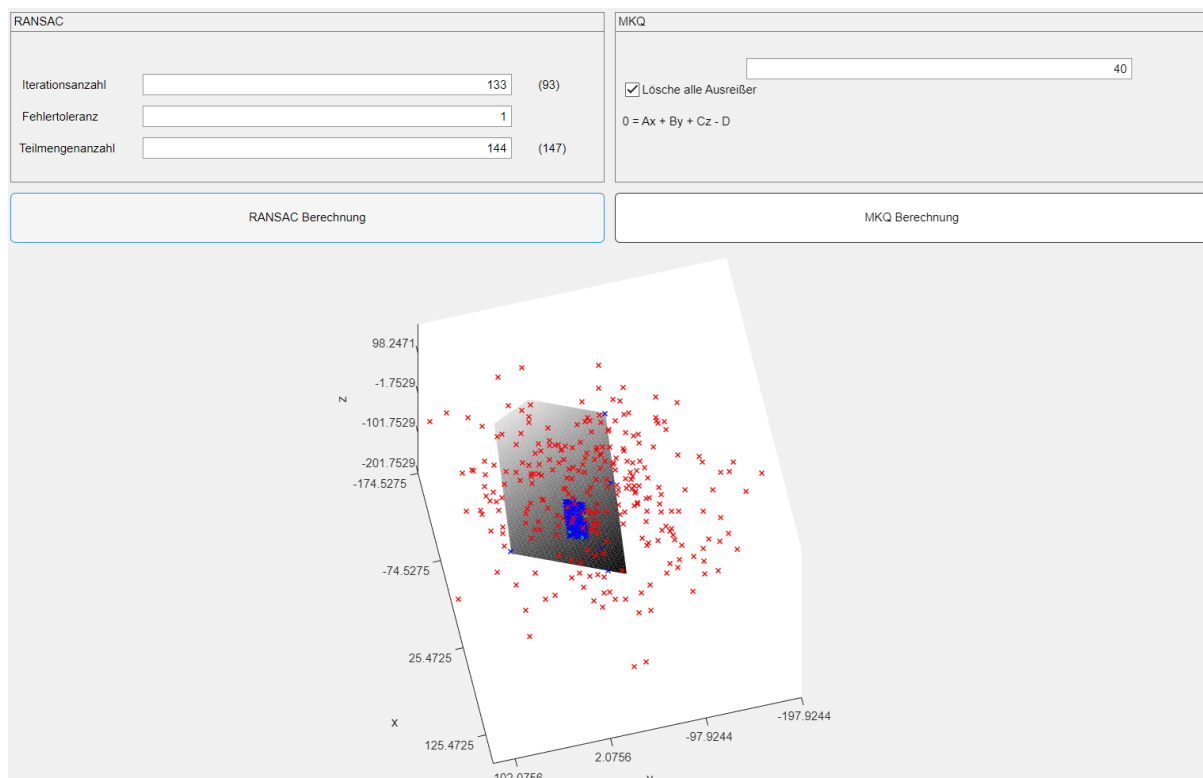


Abbildung 3: RANSAC Durchführung

Führen Sie mit den resultierenden Beobachtungen anschließend eine Kleinste-Quadrate-Ausgleichung durch! Was stellen Sie bei der Ergebnisanalyse fest?

Eine MKQ-Ausgleichung hat in diesem Beispiel und unter den aufgeführten Parametern zu keiner wahrnehmbaren Veränderung der Ebene geführt. Entsprechend ist die durch RANSAC erstellte Ebene ebenfalls aus Perspektive der MKQ-Ausgleichung „optimal“.

### 3.3 Einfluss der Streuparameter auf den RANSAC Algorithmus

Setzen Sie die minimale Teilmengenzahl auf 4. Wiederholen Sie die Berechnung und erläutern Sie das Ergebnis!

Wenn die Teilmengenzahl auf 4 (vgl. Abb. 4) gesetzt wird, erzeugt der RANSAC-Algorithmus eine Ebene, die versucht, mindestens 4 Punkte zu umfassen. Bei insgesamt 444 möglichen Punkten kann diese Ebene sehr einfach außerhalb der gewünschten Punktwolke liegen, da früher ein passendes Ergebnis gefunden wird, da bei der initialen Erstellung der Ebene die drei Ausgangspunkte bereits in dieser liegen und so nur ein weiterer Punkt benötigt wird.

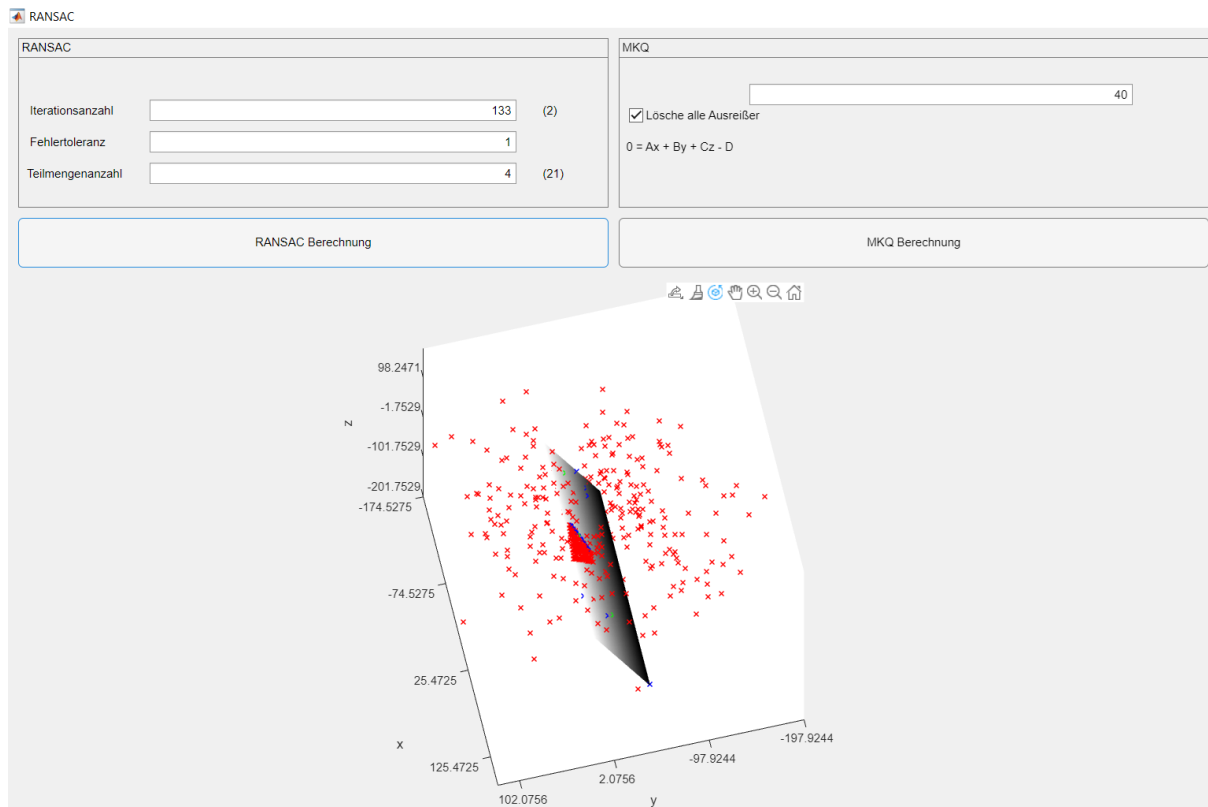


Abbildung 4: RANSAC mit Teilmengenzahl = 4

*Was passiert, wenn die Fehlertoleranz viel zu groß oder viel zu klein gewählt wird? Begründen Sie!*

Die Wahl der Größe der Fehlertoleranz ist sehr signifikant für das finale Ergebnis des Algorithmus. Sehr vereinfacht ausgedrückt bestimmt die Fehlertoleranz, wie viel Abweichung von der gespannten Ebene für die Bestimmung der passenden Punkte verwendet wird.

Bei einer zu kleinen Fehlertoleranz (vgl. Abb. 5) wird versucht, eine (fast) perfekte Ebene zu berechnen, die die Punkte beinhaltet. Entsprechend ist die Anzahl an gefundenen Punkten sehr klein.

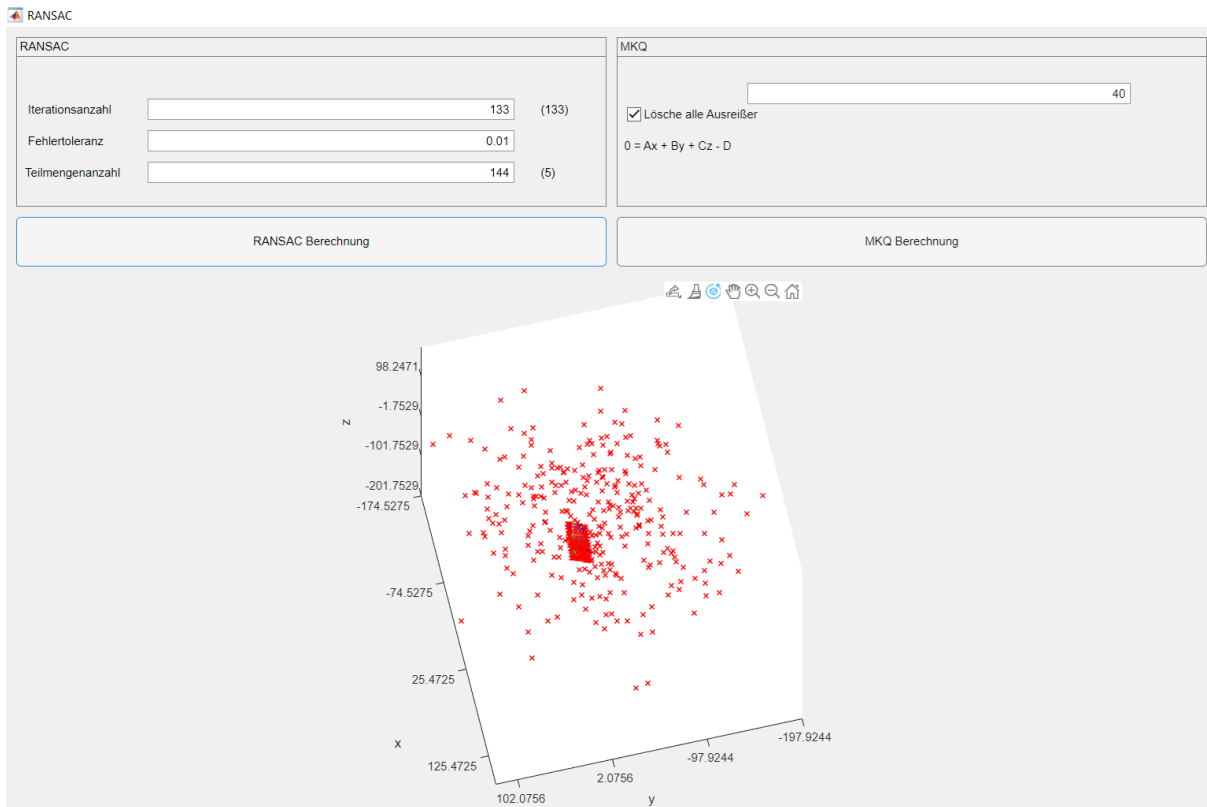


Abbildung 5: RANSAC mit Fehlertoleranz = 0,01

Wird die Fehlertoleranz zu hoch gesetzt, ähnelt das Ergebnis des RANSAC-Algorithmus eher dem des MKQ-Ausgleichs. Sehr viel mehr Punkte werden als passende Punkte interpretiert und somit wird eine Ebene berechnet, die deutlich mehr Punkte umfasst als gewünscht ist.

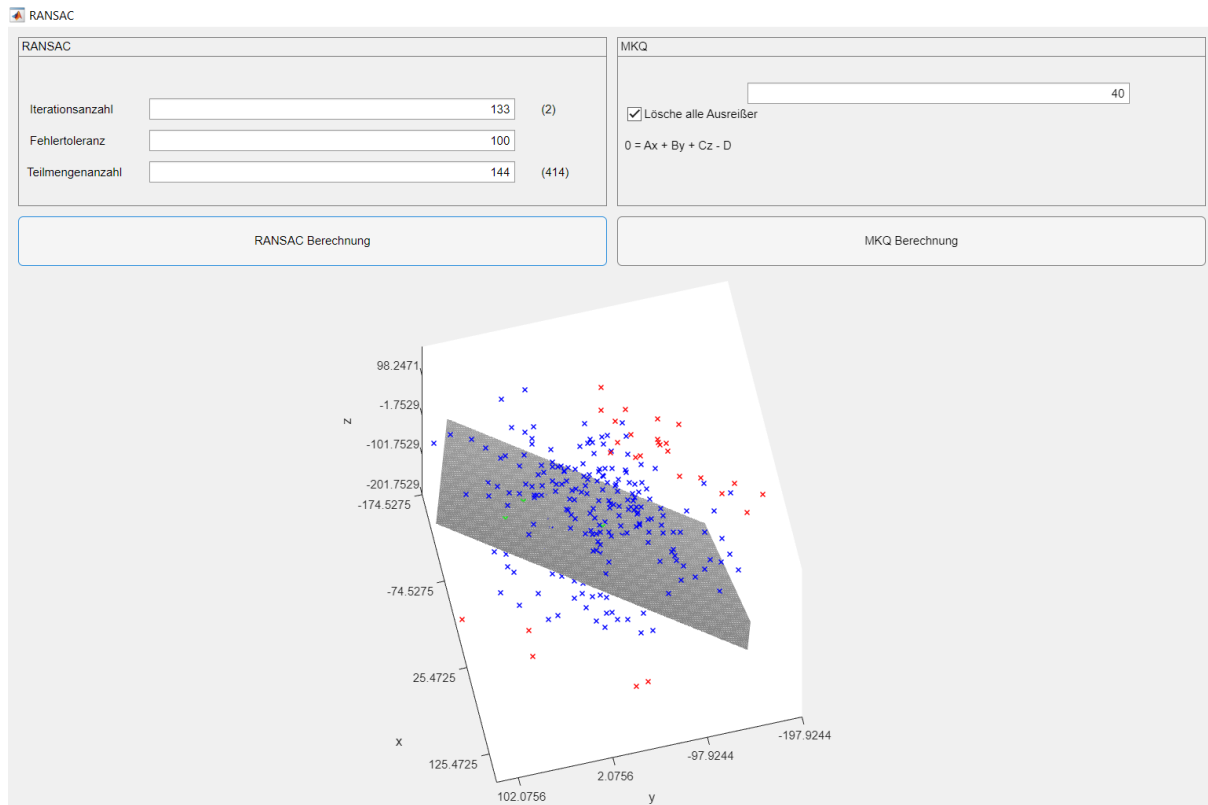


Abbildung 6: RANSAC mit Fehlertoleranz = 100

## 2. Theoretische Aufgaben

### 2.1 Funktionsweise des RANSAC-Algorithmus

Der Random-Sample-Consensus-Algorithmus (RANSAC) ist ein Verfahren zur iterativen Schätzung der Parameter eines mathematischen Modells. Dafür werden eine minimal für das Modell benötigte Auswahl an zufälligen Beobachtungen als wahr angenommen und anhand derer die Parameter des Modells bestimmt. Anschließend wird getestet, ob die restlichen Beobachtungen ebenfalls mit diesen Parametern (inklusive einer zuvor bestimmten Toleranz) zustandekommen können. Dieser Vorgang wird für andere zufällig ausgewählte Startbeobachtungen und ihre Parameter wiederholt. Die Beobachtungen, die das Modell mit der höchsten Anzahl an Beobachtungen innerhalb der Toleranzgrenze bestimmen, werden als richtig angenommen. Abschließend wird mit diesen so bestimmten Beobachtungen die mathematische Funktion des Modells mit Hilfe einer Ausgleichsmethode wie die der Kleinsten Quadrate bestimmt.

### 2.2 Vorteile von RANSAC in Kombination mit bekannten Ausgleichsansätzen

Der Nachteil von bisher bekannten Ausgleichsansätzen zur Bestimmung eines mathematischen Modells anhand von Messwerten liegt vor allem darin, dass angenommen wird, die Fehler seien normalverteilt und es keine groben Ausreißer gäbe. Mit modernen automatischen Messmethoden bei der eine Vielzahl an Beobachtungen aufgenommen werden, steigt auch die Anzahl an groben Fehlern und Ausreißern. Bekannte Verfahren wie die Methode der kleinsten Quadrate ziehen Ausreißer im Verhältnis zu ihrer Größe mit in das Modell ein und beeinflussen es daher überproportional, wenn sie nicht normalverteilt sind.

## 2.3 Anwendungsbeispiele und typische Fehlerquellen des RANSAC-Algorithmus

### 2.3.1 Relative Orientierung

Eine typische Anwendung des RANSAC-Algorithmus in der Computer Vision ist bei der Berechnung der relativen Orientierung von Bildpaaren anhand von merkmalsbasierter Zuordnung. Mit dem Aufkommen von Algorithmen zur automatischen Findung von markanten Punkten in Bildern zum Image Matching, erhöht sich auch die Anzahl von falschen Passpunkten im Gegensatz zur interaktiven Passpunkt-Findung. Bei Verfahren wie dem Moravec-Operator, dem Harris-Interest-Operator oder dem Förstner-Operator ist daher eine Aussortierung von fehlerhaften Punkten vor dem Matching nötig.

RANSAC kann hierbei helfen, indem mit fünf Punkten, welche mindestens benötigt werden, die Parameter der relativen Orientierung berechnet werden und dann ermittelt wird, wie viele der restlichen Passpunkte ebenfalls zu dieser berechneten Orientierung passen. Dies geschieht über die Berechnung der Kennlinien dieser Punkte, was anhand der bestimmten relativen Orientierung möglich ist. Es werden dann die Punkte bestimmt, bei denen die relative Orientierung eine maximale Anzahl an Schnitten der Kernlinien mit dem im zweiten Bild durch den Interest-Operator bestimmten Punkten besteht. Gerade durch repetitiven Muster können die Kontrastwerte in mehreren gefundenen Punkten sehr ähnlich sein und die Gefahr von Fehlzusammenordnungen steigen.

### 2.3.2 Räumlichen Rückwärtsschnitt

Der Räumliche Rückwärtsschnitt ist ein Verfahren, die äußere Orientierung eines Luftbildes zu bestimmen, von welchem die Position von Bodenpunkten und deren innere räumliche Orientierungen bekannt sind. Für eine eindeutige Bestimmung braucht es mindestens drei dieser Punkte. Für eine höhere Genauigkeit werden aber in der Regel mehr Punkte miteinbezogen. Die üblichen Algorithmen zur automatischen Erfassung von Passpunkten bringen dabei eine hohe Anzahl an Ausreißern mit, welche eliminiert werden müssen, bevor eine Ausgleichung stattfinden kann. Dafür eignet sich RANSAC gut.

Zuerst wird mit der minimal benötigten Anzahl an Punkten, also drei, durch den räumlichen Rückwärtsschnitt eine Ebene bestimmt. Die dabei definierten Orientierungsparameter werden nun dazu verwendet, die restlichen Passpunkte in den 3D-Raum zu projizieren. Eine vorher bestimmte Toleranzgrenze entscheidet, ob diese Punkte bei einer anschließenden Ausgleichung mit einbezogen werden sollen oder nicht. Dieses Verfahren wird anschließend iterativ für jeweils eine Auswahl von drei Punkten durchgeführt. Die Kombination von drei Punkten bzw. deren Orientierungsparameter, welche die höchste Anzahl an Passpunkten im Toleranzbereich der Ebene verzeichnet, wird für die anschließende Ausgleichung nach Methode der Kleinsten-Quadrate verwendet. Ein mögliches Problem könnte dabei sein, dass die drei Punkte mehr oder weniger auf einer Linie liegen und dadurch die Ebene nicht ausreichend genau definiert werden kann. Außerdem müssen die Punkte weit genug auseinander liegen, um keine zu große Varianz der Bildorientierung zuzulassen. Für beide Problematiken können Vorab-Überprüfungen implementiert werden.

### 2.3.3. Bündelblockausgleichung

Bei der Bündelblockausgleichung werden viele sich überlappende Bilder über sehr viele Verknüpfungspunkten in jedem Bild über deren Kameraperspektiven miteinander verrechnet. Diese rechenaufwändige Verfahren involviert eine überaus große Anzahl an unbekanntem Parametern. Daher wäre ein Ansatz, den RANSAC Algorithmus dafür zu verwenden, Ausreißer in den Verknüpfungspunkten zu eliminieren, nicht sinnvoll. Die Bündelblockausgleichung für jeweils eine minimal nötige Anzahl an Verknüpfungspunkten, welche schon sehr hoch ist, dauert im Normalfall jeweils einige Sekunden. Nun für jede mögliche Auswahl an Passpunkten ein Ergebnis zu berechnen, wäre verarbeitungstechnisch viel zu aufwändig - selbst bei einer vorher an die Erfolgswahrscheinlichkeit gekoppelten, verfrühten Abbruchregelung. Eine Lösung für das Problem kann das Partitionieren sein und den RANSAC für die relative Orientierung wie vorher schon beschrieben als Bildpaar oder Triple zu rechnen. Nach erfolgreicher Detektion der Ausreißer kann die Bündelblockausgleichung im Gesamten ohne diese gerechnet werden.

## 2.4 RANSAC-Algorithmus für ein hyperbolisches Paraboloid in einer 3-D-Punktmenge

Ein hyperbolisches Paraboloid wird beschrieben durch die Gleichung  $0 = a_{11}x^2 + a_{22}y^2 + a_{33}z^2 + 2a_{12}xy + 2a_{13}xz + 2a_{23}yz + a_1x + a_2y + a_3z + \beta$ . In einem 3D-Punktraum wird von einem Teil der Punkte auf dieser Fläche. Die Frage ist nun, ob RANSAC dazu eingesetzt werden kann, die richtigen Punkte innerhalb eines Toleranzbereiches herauszufinden und Ausreißer zu eliminieren. Das Problem dabei ist allerdings, dass in der Gleichung, welche die Fläche bestimmt, zu viele Unbekannte sind, welche bestimmt werden müssen. Bei zusätzlichen Unbekannten steigt die Anzahl an möglichen Kombinationen exponentiell an und somit auch der Rechenaufwand von RANSAC. Dieser steigt so groß, dass RANSAC hierbei keine sinnvolle Anwendung darstellt. Es gibt aber einen neuen Ansatz, bei dem RANSAC zusammen mit anderen Methoden zur Unterstützung verwendet wird.<sup>1</sup>

---

<sup>1</sup> Birdal, Tolga et al. "Generic Primitive Detection in Point Clouds Using Novel Minimal Quadric Fits." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42 (2019): 1333-1347.