

# Entwicklung und Implementierung von Algorithmen zur Merkmalsdetektion in 3D-Laserscanpunktvolken nach der Multiskalenstrategie

EBERHARD GÜLCH<sup>1</sup>, PAUL RAWIEL<sup>2</sup> & JÜRGEN PILZ<sup>3</sup>

*Zusammenfassung: Zur Erfassung von Objekten erzeugen 3D-Laserscanner Punktvolken. Um aus diesen Punktvolken Informationen zu gewinnen, werden insbesondere für die merkmalsbasierte Registrierung und die automatische Objekterkennung Merkmale benötigt. Diese Arbeit stellt in Matlab implementierte Algorithmen zur automatisierten Detektion von 3D-Linien, Eckpunkten und Ebenen vor. Die Multiskalenstrategie wird verwendet, um charakteristische von nicht charakteristischen Merkmalen zu trennen.*

*Die Detektion von Linien stützt sich auf den Edge Focusing Algorithmus. Dieser Algorithmus detektiert in größter Auflösung Kanten, die dann in immer feineren Auflösungen genauer lokalisiert werden. Das Edge-Linking fasst die Kantenpixel zu Pixelketten zusammen, durch die der Ramer-Douglas-Peucker Algorithmus 2D-Liniensegmente legt. Um die 2D-Linien zurück in den 3D-Raum zu transformieren, werden für jede 2D-Linie die entsprechenden 3D-Punkte ausgewählt. Es folgt die Berechnung einer orthogonalen Regressionsgerade durch diese 3D-Punkte. Ein Eckpunkt kann aus dem Schnittpunkt von 3D-Linien abgeleitet werden, falls diese sich in einem Winkel von etwa 90° schneiden.*

*Ebene Bereiche der Punktvolke lassen sich durch Analyse der Koordinatenunterschiede finden. Gebiete, in denen keine starken Sprünge vorliegen, werden mittels Region Labeling detektiert. In diese Punkte wird eine Ebene eingepasst. Die äußeren Linien einer Delaunay-Triangulation stellen die begrenzende Hülle jeder Ebene dar.*

*Die detektierten Merkmale werden auf Laufdauer, Quantität und Qualität untersucht und mit den Ergebnissen von Faro SCENE verglichen.*

## 1 Einleitung

Zur detaillierten Erfassung dreidimensionaler Szenen stellt das terrestrische 3D-Laserscanning ein geeignetes Aufnahmeverfahren mit großem Potenzial dar. Mögliche Anwendungsgebiete sind z.B. die Bestandsaufnahme von Gebäuden und Innenräumen sowie die Dokumentation von historischen Objekten, das Reverse Engineering, der Maschinen- und Anlagenbau und das Facility Management (DVW e.V., 2005). Im Gegensatz zur Tachymetrie werden die bedeutsamen Objekte nicht punktbasiert gemessen, sondern die gesamte Umgebung flächenhaft

---

<sup>1</sup> Prof. Dr. Eberhard Gülch, HFT Stuttgart, Schellingstr. 24, 70174 Stuttgart; E-Mail: [eberhard.guelch@hft-stuttgart.de](mailto:eberhard.guelch@hft-stuttgart.de)

<sup>2</sup> Prof. Dr. Paul Rawiel, HFT Stuttgart, Schellingstr. 24, 70174 Stuttgart; E-Mail: [paul.rawiel@hft-stuttgart.de](mailto:paul.rawiel@hft-stuttgart.de)

<sup>3</sup> MEng. Jürgen Pilz, HFT Stuttgart, Schellingstr. 24, 70174 Stuttgart

abgetastet. Aus der Menge mehrerer Millionen nicht zusammenhängender Punkte gilt es Informationen über die Form der gescannten Objekte zu gewinnen.

Das Ziel dieser Arbeit ist die Detektion von Merkmalen wie Linien, Eckpunkte und Ebenen, die einen höheren Informationsgehalt besitzen als eine unstrukturierte Menge von Punkten. Merkmale werden vor allem für die merkmalsbasierte Registrierung und die automatische Objekterkennung benötigt. Dazu gibt es bereits einige erste Entwicklungen auf dem Markt. Zur Objektrekonstruktion bietet die Software Cyclone von Leica (LEICA GEOSYSTEMS, 2012) schon seit längerem halbautomatische Werkzeuge an. Die Software Faro SCENE (FARO TECHNOLOGIES INC., 2012) ermöglicht mittlerweile eine vollautomatische Registrierung rein anhand von Merkmalen von bestehenden Objekten. Nach der automatischen Suche solcher Merkmale in jeder Punktwolke werden diese in Netzwerkgraphen geordnet und in einer Korrespondenzsuche zur Deckung gebracht. Auch das bei der Intergeo 2011 erstmals gezeigte Scantra-Modul (SCALYPSO, 2012) von Scalypso und Technet führt eine merkmalsbasierte Registrierung durch. Allerdings werden hier ausschließlich Ebenen genutzt.

Bei der Implementierung der Algorithmen zur Detektion von 3D-Linien, Eckpunkten und Ebenen in Matlab wird als zentrales Element die Multiskalenstrategie (LINDEBERG, 2009) verfolgt, um die Menge an Merkmalen steuern zu können und charakteristische von nicht charakteristischen Merkmalen zu trennen. Dabei kommt die Image Processing Toolbox zum Einsatz (THE MATHWORKS, 2012)

Der Workflow stellt sich wie folgt dar:

- Erstellung eines Bildes aus einer 3D-Punktwolke
- Liniendetektion in 3D-Punktwolken
  - Einpassung von 2D-Linien
  - Transformation der 2D-Linien in den 3D-Raum
- Detektion von Eckpunkten
- Ebenendetektion

## **2 Erstellung eines Bildes aus einer 3D-Punktwolke**

Die 3D-Punkte, deren  $x$ -,  $y$ - und  $z$ -Koordinaten in einer Textdatei gespeichert sind werden zunächst eingelesen. Um in den Punktwolken nach Linienelementen suchen zu können, wird sie verebnet. Da die Punktwolke eines terrestrischen Raster-scanners vorliegt, kann sie auf ein Bild projiziert werden. Ein Pixel entspricht dabei einem Satz Polarrichtungen. Die Winkel haben eine konstante Schrittweite. Aus den Farb- oder Intensitätswerten der Punkte wird ein Intensitätsbild berechnet. Ein Distanzbild stellt die Entfernung zum Scanner dar. Da einige wenige Punkte in kurzer oder in großer Entfernung den Grauwertbereich ungleichmäßig verzerren, wird beim Distanzbild eine Histogrammebnung angewandt. In diesen Bildern kann mit

Bildverarbeitungsmethoden nach Kanten gesucht werden. Hierzu werden die Bilder vorab mit einem Medianfilter bearbeitet, um kleinere „Löcher“ im Bild zu eliminieren, die dadurch entstehen, dass nicht bei jeder Polarkoordinate des Laserscanners ein Signal empfangen wird.



Abb. 2: Intensitätsbild

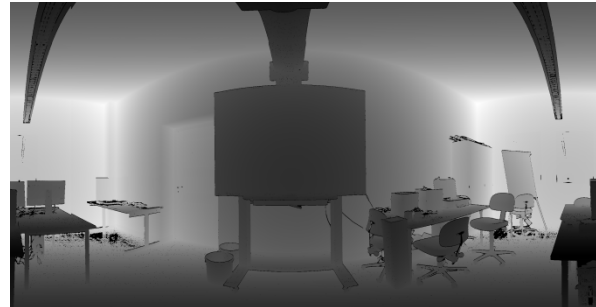


Abb. 1: Distanzbild

### 3 Liniendetektion in 3D-Punktwolken

Die Kantensuche erfolgt mittels Edge Focusing (BERGHOLM, 1987). Der Algorithmus verfolgt die Multiskalenstrategie und detektiert in grober Auflösung nur charakteristische Kanten. Die Multiskalenstrategie orientiert sich an dem menschlichen Sehen. Objekte erscheinen unterschiedlich, je nachdem in welcher „Skala“ man sie betrachtet. In dieser Arbeit wird der Skalenraum durch eine gewichtete Bildglättung nach Gauß gebildet. Die Filterung steuert, ob die Feinstrukturen eines Bildes erkennbar sind oder nur noch die groben Eigenschaften eine Rolle spielen. Die entdeckten Kanten werden dann in immer feineren Auflösungen örtlich genauer lokalisiert. Da vorab zumeist nicht bekannt ist, welche Skalierung von Bedeutung ist, berechnet und analysiert man eine Reihe von Skalierungsebenen. Die größte und feinste Auflösung muss dabei jeweils vorgegeben werden.

Die Abbildung 3 zeigt die Fassade eines Gebäudes in drei verschiedenen Auflösungen und die Ergebnisse eines Canny-Kanten-Operators ( (CANNY, 1986), (TOENNIES, 2005)), angewandt auf diese Bilder. Man kann hier gut erkennen, dass in der höchsten Auflösung sehr viele Kanten vorliegen, die allerdings auch teilweise durch Rauschen entstehen. Je stärker man das Bild glättet, desto mehr verschwinden die kleineren Details und der Kantenoperator detektiert nur noch gröbere Kanten.

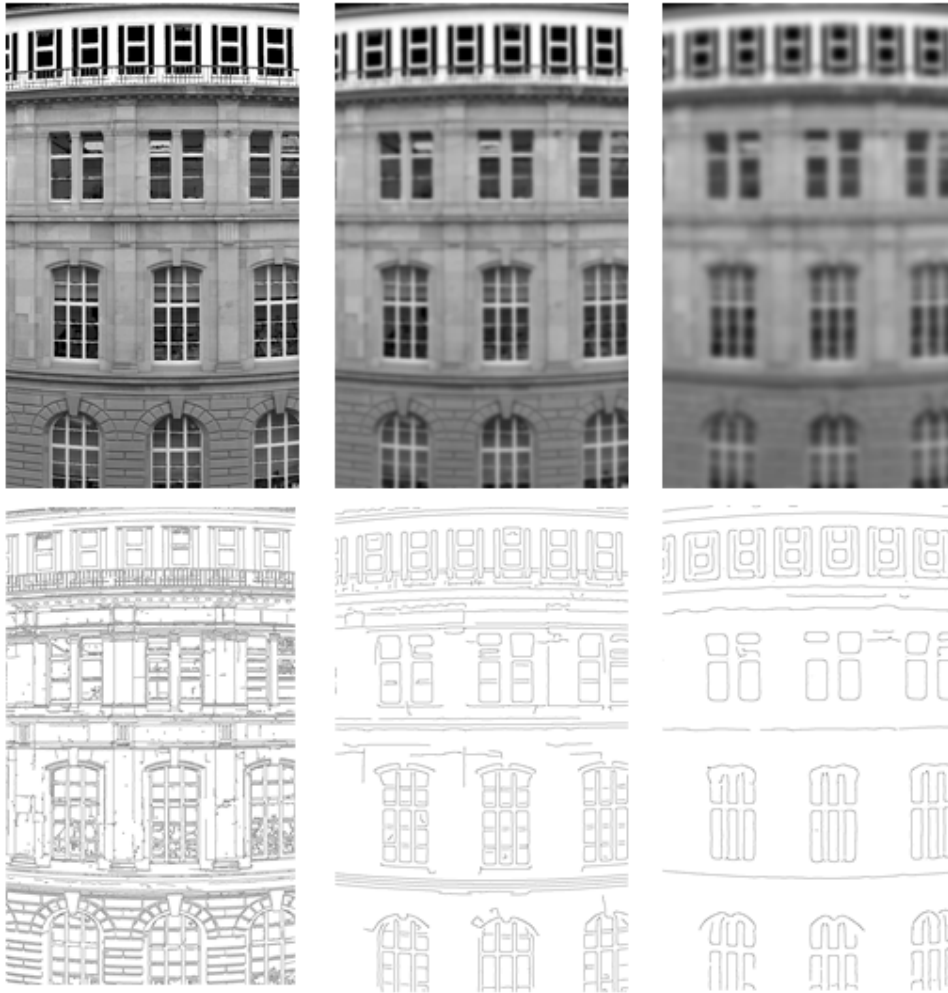


Abb. 3: Ergebnisse des Canny-Operators bei verschiedenen Auflösungen

Da Kanten bei der Detektion in hoher Auflösung mitunter auch Rauschen und unnötige Details enthalten, allerdings bei größerer Auflösung örtlich immer ungenauer werden, wird in dieser Arbeit das Edge Focusing verwendet, welches die Multiskalenstrategie verfolgt. Dieser Algorithmus verfolgt Kanten von grober zu feiner Auflösung und detektiert somit ausschließlich signifikante Kanten mit hoher örtlicher Genauigkeit ( (BERGHOLM, 1987), (ZHANG & DYKES, 1997) und (SHENG, YANG, McREYNOLDS, ZHANG, GAGNON, & SÉVIGNY, 1999)).

Der Edge Focusing Algorithmus arbeitet mit einem Grauwertbild und benötigt die Standardabweichung der Gaußfunktion für die größte und feinste Auflösung als Übergabeparameter. Die feinste Auflösung  $\sigma_{\text{End}}$  legt man zumeist auf den Wert 1 oder 1.5 fest, um eine hohe Positionsgenauigkeit zu erreichen. Die größte Auflösung  $\sigma_{\text{Start}}$  wird manuell oder automatisch anhand der Punktemenge ausgewählt. Dabei wird die empirisch entwickelte Formel genutzt, bei der die Anzahl der vorliegenden Punkte  $k_{\text{AnzPunkte}}$  einfließt:

$$\sigma_{\text{Start}} = \text{round}(\sqrt{k_{\text{AnzPunkte}} \cdot 3}) + 2$$

Wie (Bergholm, 1987) zeigt, ist die Formänderung von Kanten  $\leq 2 \cdot \Delta\sigma$ . Eine Schrittweite von  $\Delta\sigma=0.5$  ist sinnvoll, da sich so eine Kante pro Iteration um maximal ein Pixel verschieben kann. Das Bild wird beginnend bei  $\sigma_{\text{Start}}$  in x- und y-Richtung nach Gauß geglättet und anschließend wird der Gradient bestimmt. Danach werden Kantenpixel nach dem Canny-Operator bestimmt. Allerdings erfolgt die Filterung nur bei der ersten Iteration im gesamten Bild. In den weiteren Iterationen findet die Suche jeweils nur in einer kleinen Umgebung im Bereich der Kanten aus der vorherigen Iteration statt (region-of-interest filtering). Der Edge Focusing Algorithmus berechnet somit beginnend bei  $\sigma_{\text{Start}}$  ein Kantenbild nach Canny. Anschließend wird  $\sigma$  um 0.5 reduziert und im Bereich der gefundenen Kanten werden erneut die Schritte ähnlich zum Canny-Operator durchgeführt. Diese Iteration erfolgt so lange bis  $\sigma < \sigma_{\text{End}}$ .

### 3.1 Einpassung von 2D-Linien

Um anschließend, nach Vereinigung der beiden Bilder, aus detektierten Kantenpixeln Linienelemente zu generieren, werden die bekannten Verfahren des Edge-Linking (TOENNIES, 2005) und der Linieneinpassung mittels dem Ramer-Douglas-Peucker-Algorithmus (RAMER, 1972) verwendet. Das Edge-Linking speichert alle zusammenhängenden Pixel (8-Nachbarschaft) eines binären Bildes in eine Liste ab. Dabei kann ein Schwellwert für die minimale Anzahl von zusammenhängenden Pixeln berücksichtigt werden. Dieser beträgt 11 Pixel, da eine Regression einer 3D-Linie erst ab 11 Stützpunkten erfolgt. Mit dem Ramer-Douglas-Peucker-Algorithmus werden anhand dieser Pixelketten 2D-Linien-Segmente erzeugt. Die Funktion zur Erzeugung der Liniensegmente erwartet dabei eine Toleranz, wie weit die Pixel von der Linie entfernt sein dürfen. Die Toleranz liegt zumeist bei zwei Pixeln. Bei nur einem Pixel werden Linien zu oft unterbrochen. Bei einer höheren Toleranz entfernen sich die Linien zu weit vom eigentlichen Objekt. Die Start- und Endpixel einer Kette werden mit einer Linie verbunden. Falls die Distanz des am weitesten von der Linie entfernten Pixels größer als die Toleranz ist, wird die Linie an dieser Stelle geteilt. Dieses Verfahren wird solange wiederholt, bis kein Pixel mehr außerhalb der Toleranz liegt.

### 3.2 Transformation der 2D-Linien in den 3D-Raum

Für jede 2D-Linie gilt es die entsprechenden 3D-Punkte zu finden. Da ein Pixel einem Messwert und somit einem 3D-Punkt entspricht, wäre die einfachste Möglichkeit, die Pixel direkt auf der 2D-Linie zu verwenden. Diese Linien liegen jedoch sowohl auf dem verdeckten Objekt als auch auf dem verdeckenden Objekt. Fehlen Punkte an räumlichen Kanten, kann eine Linie auch auf gar nicht vorhandenen 3D-Punkten verlaufen. Die Zuordnung von 3D-Punkten aus 2D-Linien kann also nicht direkt erfolgen. Es wird ein Verfahren eingesetzt, das zu jedem Liniapixel den passenden 3D-Punkt sucht. Orthogonal zur Linienrichtung wird für jedes Pixel ein  $\pm 5$  Pixel langes Profil im originalen Distanzbild untersucht. Durch die Wahl von 5 Pixeln wird sichergestellt, dass das Profil die tatsächliche Kante beinhaltet, jedoch nicht allzu lange ist. Bei einer räumlichen Kante sollte das Profil eine Rampe aufweisen und bei einer Intensitätskante

sollten keine Sprünge vorhanden sein. Der Algorithmus benötigt einen Schwellwert  $S$ , der aussagt, ob ein Distanzwertunterschied wesentlich ist oder nicht. Dieser wird empirisch festgelegt

$$S = \sqrt{k_{MedianDichte} / k_{MedianDistanz}} \cdot 3$$

und liegt bei den verwendeten Datensätzen zwischen 0.1 m und 0.5 m. Zu Beginn wird geprüft, ob die Distanzen des Profils  $d$  eine Intensitätskante bilden. Ist dies der Fall, existiert kein merklicher Distanzunterschied und es wird der Punkt in der Mitte des Profils und somit direkt auf der Linie ausgewählt. Es folgt die Berechnung einer Regressionsgerade durch diese 3D-Punkte (NIEMEIER, 2008), (LUHMANN, 2010). Allerdings zeigt sich, dass Ausreißerpunkte nicht komplett auszuschließen sind. Da diese Punkte einen großen Einfluss auf die Regression nach der Methode der kleinsten Quadrate haben, ist dieser Schätzer hier nicht optimal. In (JÄGER, MÜLLER, SALER, & SCHWÄBLE, 2003) und (NIEMEIER, 2008) werden Ausgleichungsverfahren mit robusten Schätzern vorgestellt, die von groben Fehlern weniger stark beeinflusst werden. Allerdings werden diese nur im Gauß-Markov-Modell beschrieben. Das Prinzip lässt sich aber auf eine Ausgleichung mit Bedingungen übertragen.

## 4 Detektion von Eckpunkten

Eckpunkte haben den großen Vorteil, dass sie meist nur in sehr begrenzter Menge vorliegen und eine stark richtungsunabhängige Position besitzen. Eckpunkte werden in der Bildverarbeitung häufig mit dem Harris-Operator (BURGER & BURGE, 2006) detektiert. In 3D-Punktwolken wird die Detektion von Eckpunkten jedoch nicht mittels Bildverarbeitung gelöst, sondern mit Hilfe der 3D-Linien und bekannten Methoden der linearen Algebra. Ein Eckpunkt in einer 3D-Punktwolke definiert sich hier als Schnittpunkt von mindestens zwei senkrecht zueinander stehenden Geraden. Man nutzt dabei die Geraden direkt nach der Regression und nicht die verketteten Geraden. Es werden dazu die Schnittwinkel  $\alpha$  zwischen allen möglichen Geraden berechnet indem es aus den Richtungen  $r$  aller Geraden das Skalarprodukt bildet. Liegt der Schnittwinkel im Intervall  $90^\circ \pm 2.5^\circ$ , so werden die betreffenden Geraden als rechtwinklig angenommen. Da sich zwei Geraden im Raum allerdings meist nicht exakt schneiden, wird ein Verfahren zur Berechnung des Schnittpunktes zweier windschiefer Geraden nach (LUHMANN, 2010) implementiert.

## 5 Ebenendetektion

Ebenen stellen ein von Linien und Eckpunkten unabhängiges Merkmal dar. Sie charakterisieren sich durch eine Menge von Punkten, die in etwa auf einer Ebene liegen. Aufgrund der großen Anzahl an Punkten auf einer ebenen Fläche ist die Genauigkeit der Normalenrichtung einer

Ebene sehr hoch. Gerade in Innenräumen haben Ebenen eine starke Aussagekraft, da sie Böden, Decken und Wände beschreiben.

Der erste Schritt besteht in der Segmentierung der Punktwolke in Bereiche, in denen die Punkte nahezu auf einer Ebene liegen. Das Verfahren der Projektion wird auch bei der Detektion von Ebenen angewandt. Allerdings werden nun die xyz-Koordinaten in ein „Bild“ mit drei Bildebenen projiziert. Ähnlich zu dem in der Bildverarbeitung bekannten Laplace-Operator (JÄHNE, 2005) findet die Suche nach Sprüngen in jeder Bildebene und somit in jeder Koordinatenrichtung statt. Fasst man alle drei Bildebenen zusammen, erhält man ein Bild mit der Information über die räumlichen Sprünge. Nach (TOENNIES, 2005) erweisen sich die Schwellwertsegmentierung und das darauffolgende Region Labeling als geeignete Methoden zur Detektion von homogenen Regionen.

Durch die 3D-Punkte einer Region wird eine Regressionsebene berechnet. Bei der Regression ist es wichtig, dass sie robust gegenüber Ausreißern ist, da Ausreißerpunkte, ähnlich wie bei der Detektion von Linien nicht auszuschließen sind. Die Regression einer Ebene ist leicht von der Regression einer Linie ableitbar. Die äußeren Linien einer Delaunay-Triangulation stellen die begrenzende Hülle jeder Ebene dar. Zu große Dreiecke werden dabei vernachlässigt, um eine konkave Hülle zu bestimmen (THE MATHWORKS, 2010).

Es ist möglich, dass eigentlich zusammengehörige Ebenen durch Abschattungen, Rauschen oder durch den Lückenschluss bei 360°-Aufnahmen voneinander getrennt detektiert werden. Deshalb ist ein Nachverarbeitungsschritt notwendig. Es werden alle möglichen Schnittwinkel  $\alpha$  zwischen den Normalenvektoren berechnet. Man führt wieder eine Toleranz  $\delta$  von z.B. 5° ein. Ist  $|\alpha_{ij}| < \delta/2$ , sind die Normalenvektoren  $i$  und  $j$  als parallel zueinander zu betrachten. Anschließend überprüft man, wie weit der Schwerpunkt  $P_j$  der Ebene  $j$  von der Ebene  $i$  entfernt ist. Ist die Distanz  $d_{ij}$  kleiner als der Schwellwert  $D$ , gehören die Ebenen zusammen.

Zuletzt wird noch der Abstand der Hüllpunkte beider Ebenen verglichen. Gibt es zwei Umrandungspunkte der Ebenen  $i$  und  $j$ , deren euklidische Distanz zueinander geringer ist als der Schwellwert  $l$ , werden die Ebenen vereinigt. Dazu wird eine neue Regressionsebene durch beide Punktwolken gelegt.

## 6 Vergleich der detektierten Merkmale mit Faro SCENE

Die in Matlab programmierten Algorithmen werden mit der Software SCENE von Faro an realen Datensätzen verglichen. Bei der Laufzeit weist SCENE klare Vorteile auf. Die Anzahl an Linien ist in der eigenen Software zwar um einiges geringer als bei SCENE, die Kantensuche nach der Multiskalenstrategie zeigt hier aber deutliche Vorteile, da sie keine Linien von geringer Bedeutung oder gar durch Rauschen entstandene Linien detektiert. Die Anzahl an Eckpunkten ist

in der eigenen Software größer, dafür werden aber im Gegensatz zu SCENE keine wichtigen Eckpunkte übersehen. Die Menge an Ebenen lässt sich durch eine minimale Anzahl an Punkten oder eine minimale Fläche kontrollieren.

Im Vergleich zu SCENE weicht die Position der Eckpunkte im Mittel um einige Millimeter ab. Die Ebenen können jedoch als identisch angesehen werden. Die relative Abweichung identischer Eckpunkte in zwei unterschiedlichen Punktwolken nach einer Ähnlichkeitstransformation beträgt im Mittel einen halben Zentimeter. Die mittels Tachymetrie bestimmte 8 m lange Strecke zwischen zwei Eckpunkten im Raum weicht um +2.4 mm von dem Ergebnis der eigenen Software ab und um -2.4 mm von Leica Cyclone.

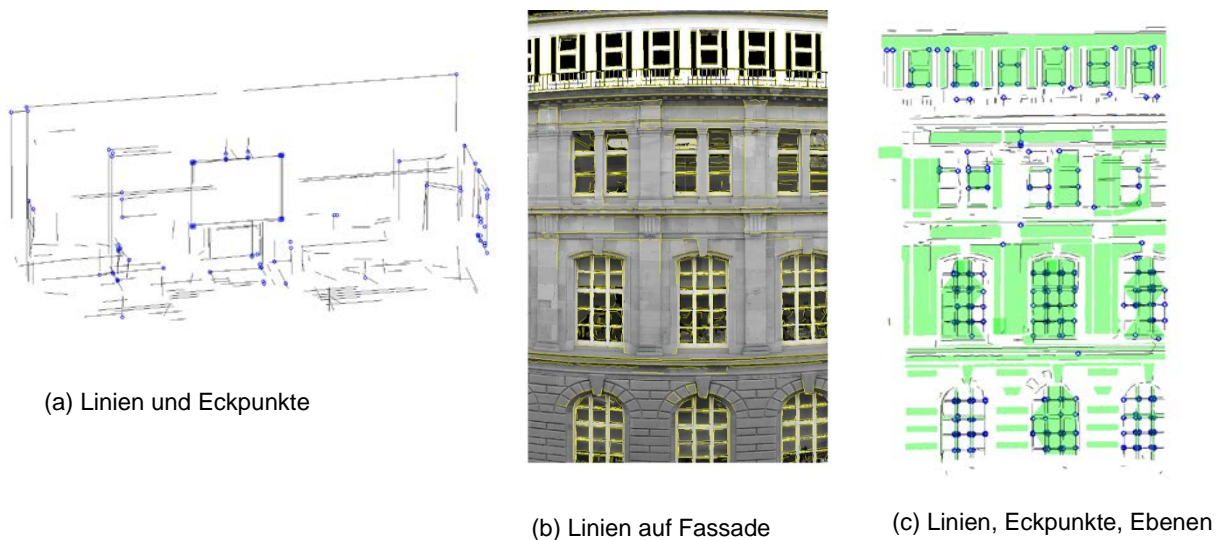


Abb. 4: Detektierte Merkmale des erstellten Programms

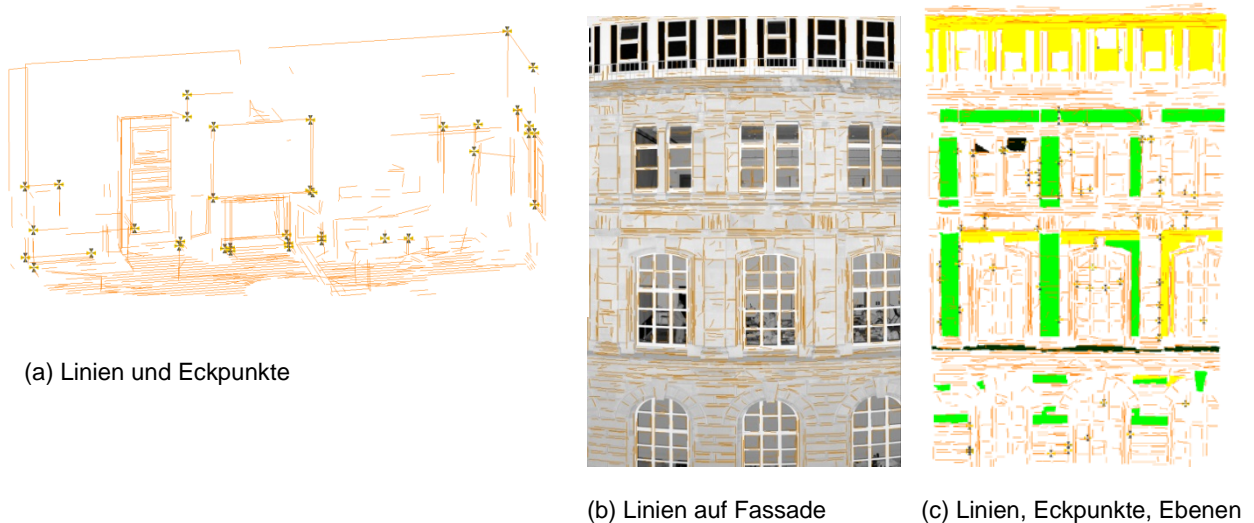


Abb. 5: Detektierte Merkmale von Faro SCENE



## 7 Fazit und Ausblick

Die Detektion von Merkmalen ist eine zunehmend bedeutende Aufgabe im Bereich des Laserscannings. Für eine automatische Objekterkennung oder Registrierung werden genaue und robuste Merkmale benötigt. Mit Hilfe der implementierten Algorithmen ist es möglich in einer terrestrischen 3D-Punktwolke Linien, Eckpunkte und Ebenen zu detektieren. Somit können Informationen automatisiert aus einer Menge von Millionen einzelner, unabhängiger Punkte gewonnen werden, die dann zur Registrierung und Modellbildung weiterverwendet werden können.

Das für die Kantendetektion implementierte Edge Focusing hat sich als ein geeignetes Werkzeug zur Suche von Kantenpixeln herausgestellt. Im Gegensatz zu anderen Kantenoperatoren, wie der häufig genutzte Canny-Operator, verfolgt das Edge Focusing eine Multiskalenstrategie. Dadurch ist es möglich, nur Kanten zu detektieren, die über mehrere Skalen hinweg persistent sind. Der Vergleich zu Faro SCENE zeigt, dass aus diesem Grund eine weitaus geringere Anzahl an Linien detektiert wird, ohne dass dabei relevante Linien fehlen. Die Menge an Eckpunkten ist wiederum größer im Vergleich zu SCENE. Dafür fehlen jedoch auch keine wichtigen Eckpunkte.

Die Anzahl an detektierten Ebenen lässt sich durch einen Schwellwert beeinflussen. Die durchgeführten Untersuchungen zeigen, dass die Merkmale auf einige Millimeter genau detektiert werden können.

Die Software Faro SCENE ermöglicht im Gegensatz zu Leica Cyclone eine Registrierung rein anhand von automatisch detektierten und natürlichen Merkmalen. Die hier vorgestellten vorbereitenden Arbeiten zur merkmalsbasierten Registrierung können rechenaufwendige Algorithmen wie Teile des ICP-Algorithmus (RUSINKIEWICZ & LEVOY, 2001) ersetzen oder zumindest gute Näherungswerte liefern (DVW e.V., 2007).

Weitergehende Ansätze zur Registrierung rein anhand von Ebenen können mit der programmierten Software implementiert und getestet werden.

## 8 Literaturverzeichnis

- ALSHAWA, M. (2007). ICL: Iterative closest line. In *ISPRS 2nd Summer School*. Ljubljana, Slowenien.
- BERGHOLM, F. (1987). Edge Focusing. *IEEE Trans. Pattern Anal. Mach. Intell.*
- BURGER, W., & BURGE, M. J. (2006). *Digitale Bildverarbeitung - Eine Einführung mit Java und ImageJ* (2. Ausg.). Berlin Heidelberg: Springer Verlag.
- CANNY, J. F. (1986). A Computational Approach to Edge Detection. *IEEE Trans. Pattern Anal. Mach. Intell.*
- DVW e.V. (2005). *Terrestrisches Laserscanning (TLS) - Ein geodätisches Messverfahren mit Zukunft*. Augsburg: Wißner-Verlag.

- DVW e.V. (2007). *Terrestrisches Laserscanning (TLS 2007) - Ein Messverfahren erobert den Raum*. Augsburg: Wißner-Verlag.
- FARO Technologies Inc. (2012). *3D-Koordinatenmesstechnik von FARO - Messarme - Laserscanner - Lasertracker*. Abgerufen am 2012. April 2012 von <http://www.faro.com/>
- JÄGER, R., MÜLLER, T., SALER, H., & SCHWÄBLE, R. (2003). *Klassische und robuste Ausgleichungsverfahren*. Heidelberg: Herbert Wichmann Verlag.
- JÄHNE, B. (2005). *Digitale {B}ildverarbeitung* (6. Ausg.). Berlin Heidelberg: Springer Verlag.
- LEICA GEOSYSTEMS. (2012). *Leica Cyclone*. Abgerufen am 15. Juni 2012 von [http://www.leica-geosystems.de/de/Leica-Cyclone\\_6515.htm](http://www.leica-geosystems.de/de/Leica-Cyclone_6515.htm)
- LINDEBERG, T. (2009). Scale-space. In B. Wah (Hrsg.), *Encyclopedia of Computer Science and Engineering*. Hoboken, New Jersey: John Wiley and Sons.
- LUHMANN, T. (2010). *Nahbereichsphotogrammetrie* (3. Ausg.). Berlin, Offenbach: Herbert Wichmann Verlag.
- NIEMEIER, W. (2008). *Ausgleichsrechnung* (2. Ausg.). Berlin: de Gruyter.
- RAMER, U. (1972). An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*.
- RUSINKIEWICZ, S., & LEVOY, M. (2001). Efficient Variants of the ICP Algorithm. In *Proceedings Third International Conference on 3-D Digital Imaging and Modeling* (S. 145-152). Quebec.
- SCALYPSO. (2012). *Das Scantra Modul*. Abgerufen am 15. Juni 2012 von <http://www.scalypso.de/scantra-modul.phtml>
- SHENG, Y., YANG, X., McREYNOLDS, D., ZHANG, Z., GAGNON, L., & SÉVIGNY, L. (1999). Real-world multisensor image alignment using edge focusing and Hausdorff distances. *Proceedings of the Conference Sensor Fusion: Architectures, Algorithms and Applications III*.
- THE MATHWORKS, I. (2010). *Thread Subject: 2D Interpolation with Missing Data*. Abgerufen am 18. Mai 2012 von [http://www.mathworks.de/matlabcentral/newsreader/view\\_thread/286773](http://www.mathworks.de/matlabcentral/newsreader/view_thread/286773)
- THE MATHWORKS, I. (2012). *MATLAB Documentation Center - MathWorks Deutschland*. Abgerufen am 4.. April 2012 von <http://www.mathworks.de/help/>
- TOENNIES, K. D. (2005). *Grundlagen der Bildverarbeitung*. München: Pearson Studium.
- ZHANG, X., & DYKES, S. G. (1997). Distributed edge detection: issues and implementations. *IEEE Computational Science & Engineering, Spring Issue*.