

---

# Automatisiertes Auffinden und Auffüllen von Scanschatten in 3D-Laserscans

Florian OTTE, Johannes ZIMMERMANN, Jochen SPRICKERHOF, Kai LINGEMANN,  
Thomas WIEMANN und Joachim HERTZBERG

## Zusammenfassung

Die Aufgabe, gleichzeitig eine Karte zu generieren und sich in ihr zu lokalisieren, zählt zu den fundamentalen Problemen der Robotik. Die Forschung auf dem Gebiet der Robotik hat in den letzten Jahren robuste Lösungen für dieses SLAM-Problem (*Simultaneous Localization and Mapping*) geliefert. Für den autonomen Einsatz mit mobilen Robotern fehlt jedoch ein weiteres Modul: Eine Strategie zur effizienten Exploration - also die Entscheidung bei gegebener partieller Karte und Explorations-Historie, zu welcher Stelle der Roboter als nächstes fahren soll, um die Karte optimal zu erweitern bzw. Scanschatten aufzufüllen.

Wir beschreiben unterschiedliche Möglichkeiten, Scanschatten in den Laserdaten zu finden und zu charakterisieren. Ebenso ermitteln wir Posen, von denen die Scanschatten eingesehen werden können, planen Pfade dorthin und integrieren die neuen Informationen in die bestehende 3D-Karte.

## 1 Einleitung

Präzise Umgebungsrepräsentationen lassen sich mit Hilfe von 3D-Laserscans erstellen. Zur vollständigen Erfassung einer Szene sind in der Regel mehrere Scans erforderlich, die von verschiedenen Posen aus aufgenommen werden müssen. Die automatische Registrierung basierend auf einer fehlerbehafteten Poseschätzung stellt heute kaum noch ein Problem dar. Ein weitgehend ungelöstes Problem bleibt allerdings, basierend auf bereits aufgenommenen Scans, automatisiert weitere Scanposen zu berechnen, die einen maximalen Informationsgewinn gewährleisten. Bereits in relativ strukturierten Szenen können einzelne Objekte im Messfeld des Laserscanners beträchtliche nicht einsehbare Bereiche (Schatten) erzeugen. Für dynamische Anwendungen, wie sie zum Beispiel in der Robotik bei der autonomen Erkundung eines unbekanntes Gebietes auftreten, gilt das um so mehr, da hier eine Fahrt durch noch nicht explorierte Bereiche ansteht, in denen Hindernisse wie Möbel, Türen und dergleichen, vorhanden sein können, die den Roboter daran hindern, alles zu erfassen.

Je komplexer eine aufzunehmende Umgebung ist, desto komplexer wird die Struktur der entstehenden Schattenwürfe. Es ist darum von grundlegender Bedeutung, die Unwissenheit explizit mit zu modellieren und in die Anwendungen zu integrieren. Unser Ansatz zum Auffinden und Auffüllen von Scanschatten hat seinen Ausgangspunkt genau an dieser Stelle und versucht zunächst, das Unbekannte zu orten: Ein Laserscan muss dahingehend analysiert werden, *ob* und *wo* Regionen vorhanden sind, von denen angenommen werden

muss, dass sie verdeckt sind. Zur Lösung der oben genannten Probleme müssen die vorliegenden Daten in geeignete Datenstrukturen überführt werden, die es erlauben, verdeckte Gebiete zu erkennen und plausible Posen für weitere Scans mit idealerweise optimalem Informationsgewinn zu berechnen.

Im Folgenden werden wir zwischen Schattenraum und Schattenfläche unterscheiden. Die *Schattenfläche* ist eine durch Lichtquelle und Gegenstand erzeugte Projektion auf der eigenen Oberfläche oder der eines anderen Gegenstandes und ist demnach zur realistischen Darstellung (Schattierung) besonders in der Computergrafik relevant. Hingegen ist der *Schattenraum* ein Raum, der von der Lichtquelle aus gesehen hinter einem Gegenstand entsteht und begrenzt wird durch seine Rückseite und das ungedämpfte Licht der Lichtquelle. Der Schattenraum wird für die weitere Untersuchung der relevante Schatten sein: In einem Laserscan ist der Scanner die einzige Lichtquelle, er leuchtet und betrachtet gleichzeitig. Es gibt im Laserscan keine Schattenflächen, die ein Objekt sichtbar auf andere wirft, sondern nur Leerstellen hinter Hindernissen. Auch gehört die Rückseite eines betrachteten Objekts nicht unmittelbar zum Scanschatten, da über die Rückseite überhaupt nichts bekannt ist. Vielmehr interessiert der blinde Fleck hinter der registrierten Frontseite: Der Raum, der von der Scanposition aus durch das Hindernis versperrt wird. Somit ergibt sich unsere Definition:

*Ein Scanschatten ist der Schattenraum hinter einem gemessenen Hindernis, der aus einer zusammenhängenden Menge von nicht einsehbaren Punkten im Raum besteht.*

Dieses Papier beschreibt Verfahren, mit denen sich effektiv nicht einsehbare Gebiete innerhalb eines Laserscans auffinden lassen. Basierend auf den so gewonnenen Informationen werden in einem zweiten Schritt diejenigen Scanposen berechnet, welche die vorher nicht eingesehenen Gebiete optimal erfassen.

## 2 Wissenschaftlicher Hintergrund

Scanschatten (engl. *scan shadow*, *blind spot*, *occlusion*) werden schnell akut, wenn nicht einsehbare Bereiche zum Anlass von Unfällen werden. Für Maschinen, Autos oder Werkzeug ist das Erkennen von blinden Flecken beim Einsatz wichtig, um Arbeitsunfälle zu vermeiden (TEIZER et al. 2010). Eine allgemeine, geometrische Beschreibung und mathematische Modellierung von Schattenflächen, Lichtquellen und Verdeckungen liefern KNILL (1997) und SHAFER (1983). Weiterhin tritt in der Literatur zum Thema häufig das Erkennen von Scanschatten in hochauflösenden Luftaufnahmen auf. Soll aus ihnen zum Beispiel der genaue Verlauf der Erdoberfläche extrahiert werden, verdecken viele Objekte wie Bäume oder Häuser den Boden. Sie müssen erkannt und extrahiert werden (ZHAN 2005).

Die Frage, welches die nächste Scanpose mit optimalem Informationsgewinn ist, ist eng mit den Kunstgalerie-Problemen verwandt, bekannt aus der algorithmischen Geometrie. Dabei geht es darum, in einem einfach zusammenhängenden polygonalen Gebiet (der „Kunstgalerie“) eine möglichst kleine Zahl von Wachleuten mit Rundumsicht so zu platzieren, dass das gesamte Gebiet des Polygons eingesehen werden kann (O'ROURKE 1987, CHATAL 1975). Für Anwendung in der mobilen Robotik sind vor allem Online-

Varianten dieser Probleme interessant, bei denen die Struktur der Umgebung zunächst unbekannt ist, so dass diese sukzessive mit einer möglichst geringen Zahl an Laserscans vollständig erfasst wird.

Kunstgalerie-Probleme sind allerdings bereits für einfache, praxisrelevante Fragestellungen NP-vollständig (z.B. wenn der Blickwinkel der Wachleute beschränkt ist); d. h. für realistische Anwendungsszenarien auf einen mobilen Roboter kommen solche Ansätze aufgrund der hohen algorithmischen Komplexität nicht in Frage. Daher werden in der mobilen Robotik in der Regel spezielle, auf die vorhandene Umgebungsgeometrie angepasste Ansätze zur Scanposeplanung angewendet. Eine relativ aufwändige Strategie ist in ALBERS et al. (1999) beschrieben. Weitere relevante Ansätze lassen sich aus ICKING et al. (2003) und HOFFMANN et al. (2001) entnehmen.

Unsere Arbeit zielt nicht darauf ab, eine mathematisch beweisbar optimale Lösung des vorgestellten Problems zu finden. Vielmehr präsentieren wir eine leicht zu implementierende heuristische Explorationsstrategie. Anhand der durchgeführten Experimente wird gezeigt, dass sich mit diesem Ansatz plausible Posen für weitere Scans zur optimalen Erfassung der Umgebung berechnen lassen. Die mit unserem Verfahren berechneten Scanposen werden autonom angefahren und die aufgenommenen Scans werden anschließend automatisch registriert und in die Umgebungsrepräsentation integriert.

### 3 Auffinden von Scanschatten

Die vom Laserscanner gelieferten 3D-Punktwolken werden zunächst in eine zweidimensionale Rasterkarte überführt, indem alle Messwerte innerhalb eines benutzerdefinierten Volumens diskretisiert auf die Grundebene projiziert werden. Die folgenden Abschnitte beschreiben drei Methoden zum Auffinden von Scanschatten: Mittels eines Raytracing-Ansatzes werden zunächst noch nicht eingesehene Gebiete im Scan detektiert. Anschließend werden die vorhandenen Hindernisse in eine Liniendarstellung überführt, in der so genannte „Sprungkanten“ gesucht werden. In einem dritten Schritt wird eine polygonale Darstellung der Schattengebiete ermittelt.

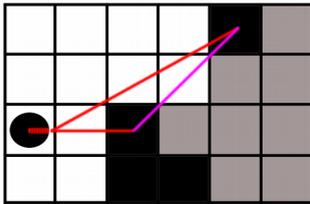
#### 3.1 Raytracing

In der von uns verwendeten Karte können die Zellen die Werte *frei*, *belegt* oder *unbekannt* einnehmen. Basierend auf den aufgenommenen Laserscans werden alle Zellen, über denen sich ein oder mehrere Messpunkte befinden, als *belegt* markiert. Mittels Raytracing werden alle Zellen, die sich zwischen der Position des Laserscanners und den ermittelten Hindernissen befinden, als *frei* markiert, da sich in diesem Gebiet sicher keine Objekte befinden. Alle danach noch als *unbekannt* markierten Zellen innerhalb der Scanner-Reichweite waren vom Laserscanner nicht einsehbar, sind also Scanschatten.

#### 3.2 Jumpspot

Nahe Objekte verdecken die Sicht auf fernere Objekte. Die dadurch entstehenden Scanschatten lassen sich durch Sprünge in den gemessenen Entfernungen an den Kanten dieser Objekte erkennen. Ziel der zweiten Methode ist es, solche Übergänge zwischen

Objekten zu finden, da die Wahrscheinlichkeit groß ist, dass sie auf einen Scanschatten hindeuten, in dem sich weitere Objekte befinden können, die noch erkundet werden müssen. Dazu wird für jede als Hindernis markierte Zelle bestimmt, ob sich in ihrer Umgebung weitere als Hindernis markierte Zellen befinden. Überschreitet die Entfernung zwischen einem Hindernispunkt zum nächsten einen festgelegten Schwellwert, ist davon auszugehen, dass ein Objekt ein anderes verdeckt hat. Solche Sprungkanten sind daher Übergänge zu vorher verdeckten Gebieten.



**Abb. 1:** Ergebnis der Methode Jumpspot.

Der Jumpspot-Algorithmus sortiert zunächst alle gefundenen Hindernisse nach ihrer Entfernung radial um die Scanposition herum. So können jeweils in der sortierten Folge zwei aufeinanderfolgende Hindernispunkte bezüglich ihrer Entfernung miteinander verglichen werden. Überschreitet der Abstand zwischen zwei Punkten einen Schwellwert, kann die Linie zwischen diesen Punkten als Grenze des Scanschattens angesehen werden (siehe Abb. 1). Auf diese Art lassen sich die Konturen der vorliegenden Schatten teilweise rekonstruieren.

### 3.3 Polygonfill

Nachdem mit Raytracing eine undifferenzierte Menge aller Schattenzellen gefunden wurde und mit Jumpspot einzelne Punkte, die einen gewissen Abstand zueinander haben, ausgemacht wurden, sollen diese nun zu Polygonen verbunden werden. Gehen wir die Liste aller von Jumpspot gefundenen Sprungstellen durch und fügen sie nach und nach zu Eckpunkten eines Polygons hinzu, müssen zwei Kriterien erfüllt sein, damit ein für unsere Zwecke geeignetes Polygon entsteht:

- Validität. Zum einen muss das Polygon aus den Punkten tatsächlich konstruierbar sein. Der Einfachheit halber fordern wir, dass das Polygon nicht überschlagen darf, das heißt, seine Kanten dürfen sich nur an den Eckpunkten berühren.
- Schattenpolygon. Ebenfalls muss ein Polygon Zellen im Inneren tragen, die als unbekannt (durch Raytracing) eingestuft wurden, da es sonst keinen Schatten begrenzen würde (siehe Abb. 2).

Polygonfill findet nun alle maximal zusammenhängenden Polygone in der Jumpspot Liste, welche beide Kriterien erfüllen. Damit wird die anfängliche grobe Schattenmenge in kleine, getrennte Gebiete zerlegt, die einzelnen Schatten entsprechen und so eine genauere Parametrierung und Bestimmung der Schattengröße für das weitere Vorgehen ermöglichen.

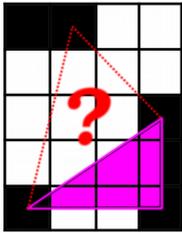


Abb. 2: Eine mögliche Erweiterung für das Schattenpolygon.

## 4 Auffüllen von Scanschatten

Nachdem das Auffinden von Scanschatten skizziert wurde, muss zunächst darauf eingegangen werden, was die notwendigen Schritte sind, um ein Auffüllen zu ermöglichen. An erster Stelle steht die Berechnung einer Pose, von der aus ein maximaler Informationsgewinn möglich ist. Dieser Informationsgewinn kann verschieden definiert werden, wie die nächsten Abschnitten näher erläutern. Der nächste Schritt ist es, einen Pfad zu dieser Pose zu konstruieren, der alle Randbedingungen erfüllt. Die finalen Schritte erfolgen dann durch Aufnahme eines neuen Scans, sowie durch Registrieren der gewonnenen Daten. An dieser Stelle soll ausschließlich auf die Berechnung einer Pose eingegangen werden. Pfadplanung, sowie Scanregistrierung werden in der Literatur bereits ausführlich behandelt, siehe z.B. LAVALLE (2006), BESL (1992).

### 4.1 Auffüllen mehrerer kleiner Scanschatten

Häufig enthält ein Laserscan viele kleine Scanschatten. Eine optimale Pose ist nun diejenige, die die meisten Schatten einsehbar macht. Der Informationsgewinn ist demnach nicht über die auffüllbaren Bereiche definiert, sondern über die Menge der Einzelbereiche. An dieser Stelle soll also eine Pose bestimmt werden, die möglichst viele Schatten einsehbar macht.

Dafür werden die gefundenen Eckpunkte der Jumpspot-Methode genutzt. Aus den Eckpunkten errechnet sich leicht das arithmetische Mittel, um welches ein Kreis mit einem festen Durchmesser gelegt wird. Alle Punkte innerhalb der Kreise sind vorerst mögliche Scanpositionen. Damit mehrere Schatten einsehbar werden, ist es sinnvoll, die Überlappungen zwischen diesen Kreisen zu suchen. Um hinterher sinnvolle Überlappungen zwischen diesen möglichen Posen finden zu können, werden zusätzliche Informationen zu jedem Punkt innerhalb der Kreise abgelegt. Diese enthalten den Mittelpunkt des Schattens und zwei Vektoren, die jeweils auf einen der Randpunkte des Schattens zeigen. Mit diesen zusätzlichen Informationen lassen sich nun Überlappungen zwischen verschiedenen Kreisen finden, die die möglichen Einschränkungen bezüglich des Scanwinkels beachten.

Es werden dann die Posen vorgemerkt, von denen aus mehr als ein Schatten sichtbar ist. Da diese Menge von Posen bei vielen kleinen Schatten beliebig groß werden kann, wird sie weiter ausgedünnt. Ein Kriterium ist hierbei, dass ein Rand mit einer Mindestbreite sichtbar bleibt um hinterher das Einfügen in eine vorhandene Repräsentation zu erleichtern. Die letztendlich optimale Pose ist jene, welche die meisten Schatten sichtbar macht. Sollte es

keine Pose geben, so wird beliebig diejenige Pose gewählt, die den Schatten mit dem breitesten Rand abdeckt.

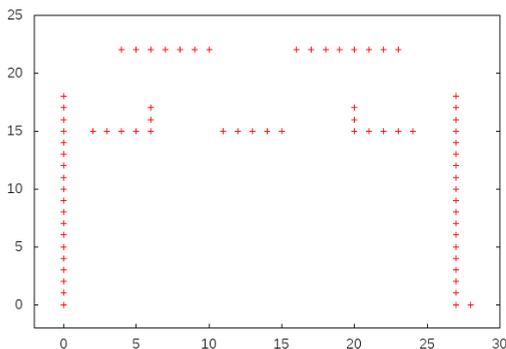
## 4.2 Berechnung einer Scanpose aus Polygonvertices

In dem Fall, dass ein Grundriss der Umgebung genügt oder nur eine grobe Repräsentation benötigt wird, ist die sinnvollste Pose diejenige, die den größten Bereich einsehbar macht. Dieser Bereich wird von der Polygonfill-Methode zur Verfügung gestellt.

Um einen schnellen Überblick über mögliche Posen zu bekommen, wird ein Kreis um das gesamte Polygon gelegt. Diese Menge wird weiter reduziert, indem alle Posen im unbekanntem Bereich entfernt werden. Die verbleibende Menge ist häufig noch recht groß. Aus diesem Grund werden alle Posen entfernt, die weniger als 90% des Schattens sichtbar machen. Die verbleibende Liste wird dann nach Sichtbarkeit sortiert und es wird die Pose gewählt, die den breitesten Rand, der nicht zum Scanschatten gehört, einschließt. Diese Forderung ist äquivalent zu der Forderung, dass der Öffnungswinkel zwischen der Pose und den beiden nächstgelegenen Eckpunkte des Polygons minimal ist.

## 5 Experimente

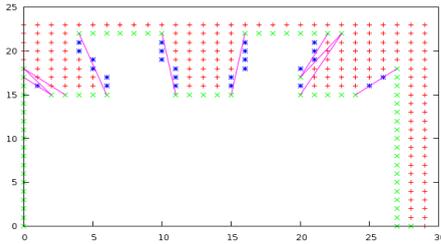
Die zuvor beschriebenen Methoden wurden mit verschiedenen Datensätzen evaluiert. In diesem Abschnitt soll beispielhaft ein Datensatz ausgewertet werden. Die konstruierte Szene besteht aus einem leeren Raum, in dem drei Hindernisse Schatten werfen. Abb. 3 zeigt die reduzierte und diskretisierte Punktwolke. Es erfolgt ein 8 cm dicker Schnitt durch die 3D-Punktwolke auf einer Höhe von 5 cm. Die sichtbaren Hindernisse sind drei umgedrehte Tische. Dieses Raster ist nun die Grundlage für die Methoden des Auffindens.



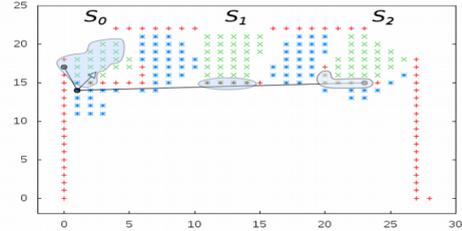
**Abb. 3:** Ausgangsszene mit drei Hindernissen.

In Abb. 4 sind die Ergebnisse der Methoden Raytracing und Jumpspot dargestellt. Die roten Zellen markieren hierbei den von Raytracing gefundenen Schatten. Obwohl die Schatten vollständig gefunden werden, wird deutlich, dass diese Methode die Schatten nicht genau charakterisiert. Sämtliche unbekanntem Bereiche werden gefunden, aber sie müssen nicht den auffällbaren Bereichen entsprechen. Eine bessere Übersicht bringt hier Jumpspot, das die Schattenränder klassifiziert. Ein Einblick ist nur durch diese gefundenen Ränder möglich, und somit ist dieses Ergebnis im Folgenden sinnvoller nutzbar. Abb. 5 zeigt alle

gefundenen Überlappungen der ersten Methode zur Berechnung einer Pose. Diese Überlappungen sind blau dargestellt. Die letztlich gewählte Pose ist in Schwarz hervorgehoben. Die sichtbaren Schattenzellen sind blau überlagert. Es ist offensichtlich, dass diese Pose zwar alle Schatten sichtbar macht, aber diese nur unzureichend einsehbar werden. Weitere Kriterien sind an dieser Stelle notwendig, um die Wahl der Pose zu verbessern.

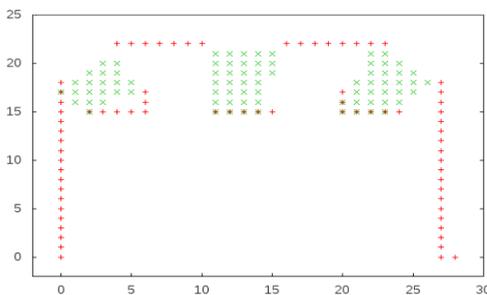


**Abb. 4:** Ergebnisse der Raytracing- und der Jumpspot-Methode.

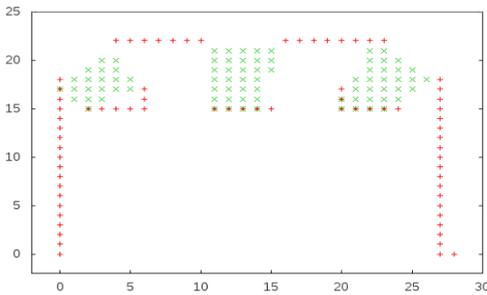


**Abb. 5:** Gefundene Überlappungen und mögliche Scanposen.

Abb. 6 zeigt die Ergebnisse der Polygonfill-Methode; die grünen Punkte gehören zu den entsprechenden Polygonen. Diese Methode klassifiziert die Schatten am sinnvollsten, es werden nur Zellen gefunden, die auch tatsächlich den gesuchten Schatten entsprechen. Abb. 7 zeigt an dieser Stelle das Ergebnis der zweiten Methode zum Auffüllen. Alle blau markierten Punkte sind hierbei Posen, die es ermöglichen, ein Polygon vollständig aufzufüllen. Die letztlich gewählte Pose ist auch an dieser Stelle schwarz markiert, zusätzlich ist die Blickrichtung und die beiden äußersten Zellen markiert, die sichtbar sind. Diese Zellen spannen mit der Pose den minimalen Öffnungswinkel des neuen Scans auf, der benötigt wird, um das Polygon aufzufüllen. Alle anderen markierten Posen erlauben, eines der Polygone aufzufüllen. An dieser Stelle wird aber auch sichtbar, dass mehrere Schatten einsehbar sind und gleichzeitig ein ganzes Polygon auffüllbar ist.



**Abb. 6:** Gefundene Schattenbereiche der Methode Polygonfill.



**Abb. 7:** Berechnete neue Scanpose.

## 6 Ausblick

Die beschriebenen Heuristiken können naturgemäß nur einen kleinen Teil des Problems lösen und viele Details bleiben dabei unberücksichtigt. Interessant wäre es zum Beispiel, die Algorithmen auf Datensätze anzuwenden, die mit anderen Robotern und Laserscannern aufgenommen wurden. Außerdem wird bei diesem Ansatz der Scanwinkel des Laserscanners explizit modelliert. Aktuelle 3D-Laserscanner sind aber in der Lage, einen Bereich von  $360^\circ$  zu erfassen. Ebenso gehen wir davon aus, dass der Roboter stehen muss, um einen Laserscan aufzunehmen. Ein Roboter, der während der Fahrt scannt, hat viel weitergehende Möglichkeiten. So kann er, während er zur anvisierten Pose fährt, schon einen Teil des Scanschattens einsehen und dadurch seine endgültige Pose weiter optimieren. Beide Aspekte bedeuten aber auch eine Steigerung der Komplexität, da mehr Daten in kürzerer Zeit verarbeitet werden müssen. In zukünftigen Arbeiten werden wir diese Probleme angehen.

## Literatur

- Albers S., Kursawe K., Schuierer S. (1999): *Exploring Unknown Environments with Obstacles*, Proc. 10th ACM-SIAM Sympos. Discrete Algorithms, S. 842-843
- Besl P.J., McKay H.D. (1992): *A method for registration of 3-D shapes*, IEEE Transactions on Pattern Analysis and Machine Intelligence
- Chatal V. (1975): *A Combinatorial Theorem in Plane Geometry*, J. Combin. Theory Ser. B, 18, S. 39-41
- Hoffmann F., Icking C., Klein R., Kriegel K. (2001): *The Polygon Exploration Problem*, SIAM J. Comp. 31, S. 577-600
- Icking C., Klein R., Ma L. (2003): *How to Look Around a Corner*, Proc. 5th Can. Conf. on 3-D Digital Imaging and Modeling
- Knill D.C., Mamassian P., Kersten D. (1997): *Geometry of shadows*, J. Optical Society of America A 14, Nr. 12, S. 3216-3232

- LaValle, S.M. (2006): *Planning Algorithms*. New York (Cambridge University Press)
- O'Rourke J. (1987): *Art Gallery Theorems and Algorithms*, Int. Series of Monographs on Computer Science, Oxford University Press
- Shafer S.A., Kanade T. (1983): *Using shadows in finding surface orientations*, Computer Vision, Graphics and Image Processing 22, Nr. 1, S. 145-176
- Teizer J., Allread B., Mantripragada U. (2010): *Automating the blind spot measurement of constructing equipment*, Automation in Construction 19, S. 491-501
- Zhan Q., Shi W., Xiao Y. (2005): *Quantitative analysis of shadow effects in high-resolution images of urban areas*, Intl. Archives of Photogrammetry and Remote Sensing 36