

# Automatisches Texture Mapping von 3D-Modellen in Architektur und Archäologie

DIRK STALLMANN<sup>1</sup>, THOMAS P. KERSTEN<sup>2</sup>

*Zusammenfassung: Detaillierte, vollständige und genaue 3D-Modelle mit foto-realistischen Texturen werden heute auch zunehmend für zahlreiche Anwendungen in der Architektur und in der Archäologie nachgefragt. Da eine manuelle Texturierung eines 3D-Modells durch digitale Bilddaten mit Softwarepaketen wie Cinema 4D, 3D Studio oder Maya heute immer noch einen aufwendigen Arbeitsweg darstellt, sind Verfahren zur automatischen Texturierung von 3D-Modellen sehr gefragt. In diesem Beitrag werden zwei automatische Verfahren vorgestellt, mit denen einerseits mit Hilfe von Webservices 3D-Oberflächenmodelle mit Texturen hergestellt und andererseits bereits existierende 3D-Modelle mit tmapper, einer in der Programmiersprache C++ entwickelten Software auf Grundlage des Multi-Layer 3D Image (ML3DImage) Algorithmus, texturiert werden. Die Untersuchungen anhand von Beispieldaten zeigen, dass Selbstverdeckungen im 3D-Modell vorhanden sind und die Sichtbarkeitsanalyse im ML3DImage Algorithmus nicht ausreicht, um akzeptable Ergebnisse der automatischen Texturierung zu erzielen. Zur Lösung des Sichtbarkeitsproblems und der korrekten perspektivischen Entzerrung der verwendeten Texturen soll zukünftig der Point Cloud Painter Algorithmus mit dem Z-Puffer-Verfahren, ein bekanntes Verfahren aus der Computergraphik, kombiniert werden.*

## 1 Einleitung

Die Visualisierung von 3D-Objekten gewinnt zunehmend an Bedeutung, da sowohl die 3D-Aufnahmetechnologien wie Photogrammetrie und Laserscanning effizienter, als auch die Anwendung und Entwicklung von computergestützten Animationstechniken und die Computertechnologie immer leistungsfähiger werden. In der 3D-Computergraphik existieren mittlerweile zahlreiche Verfahren, um dreidimensionale Objekte möglichst als foto-realistisches Bild des erstellten Modelles zu darzustellen. Das abzubildende Objekt liegt als 3D-Oberflächenmodell in Form einer Dreieckszerlegung und/oder in Form geometrischer Primitive (Geraden, Ebenen, Zylinder, etc.) vor (LUHMANN 2010). Die Überlagerung eines Objekts mit einer foto-realistischen oder synthetischen (künstlichen) Textur zur Darstellung dreidimensionaler Räume/Objekte nennt man Texturierung (engl. Texture Mapping). Dabei ist ein Texturbild ein digitales Bild (Foto), dessen Muster wie bei einer Diaprojektion dem Objekt (dem einzelnen Dreieck) überlagert bzw. drauf projiziert wird. So eine Texturierung ermöglicht je nach Auflösung der verwendeten Texturen auch weniger detailreiche 3D-Modelle realistisch darzustellen (BOLLMANN & KOCH, 2002).

Die Texturierung kann dabei manuell oder automatisch erfolgen (KADA 2007). Bei der manuellen Texturierung werden die Texturen mit Softwarepaketen wie Maxon Cinema 4D, Autodesk 3Ds Max oder Maya durch den Anwender interaktiv auf die Oberflächenelemente bzw. Dreiecke gelegt. Dagegen erfolgt eine automatische Texturierung eines 3D-Modells, in dem die natürliche Oberflächentextur eines räumlich orientierten Messbildes überlagert und der Bezug zu dem Ausgabebild über die photogrammetrischen Abbildungsgleichungen her-

<sup>1</sup> New Class Consulting Ltd., 8A Chanion Street, 8027 Paphos, Cyprus, E-Mail: dsv2a001@googlemail.com

<sup>2</sup> HafenCity Universität Hamburg, Labor für Photogrammetrie & Laserscanning, Hebebrandstrasse 1, 22297 Hamburg, E-Mail: Thomas.Kersten@hcu-hamburg.de

gestellt wird. Die automatische Methode setzt voraus, dass die innere und äußere Orientierung aller verwendeten Bilder bekannt ist und dass die Geometrie der 3D-Objekte durch unregelmäßige räumliche Dreiecksnetze dargestellt wird. Um höchste geometrische Präzision für das Texture Mapping der 3D-Modelle zu erreichen, werden die mathematischen Kameramodelle aus der photogrammetrischen Bündelausgleichung auch für die Texturierung verwendet.

In diesem Beitrag werden neben der manuellen Texturierung (Kap. 2) automatische Verfahren vorgestellt, mit denen einerseits texturierte 3D-Oberflächenmodelle durch Verfahren der Computer Vision und Photogrammetrie mit Webservices hergestellt (Kap. 3) und andererseits bereits existierende 3D-Modelle durch orientierte Bilddaten einer kalibrierten Kamera mit *tmapper*, einer in der Programmiersprache C++ entwickelten Software auf Grundlage des Multi-Layer 3D Image (ML3DImage) Algorithmus, texturiert werden (Kap. 4). Ergebnisse und Probleme der durch *tmapper* automatisch texturierten Modelle werden in Kap. 5 vorgestellt.

## 2 Manuelle Texturierung von 3D-Modellen

Die manuelle Texturierung von 3D-Modellen (z.B. CAD-Volumenmodelle) ist eine zeit- und kostenintensive Arbeit, die allerdings qualitativ sehr hochwertige, ansehnliche Resultate liefert, wenn professionelle Softwarepakete wie Autodesk 3ds Max, Autodesk Maya oder Maxon Cinema 4D eingesetzt werden (Abb. 1). Als Ergebnisse können foto-realistische Ansichten, Animationen oder Videos ausgegeben werden, in dem aus dem virtuellen 3D-Modell viele perspektivische 2D-Ansichtsbilder entlang eines vordefinierten Kamerapfades berechnet werden. Die Generierung eines Bildes aus der 3D-Szene nennt man in der Computergraphik *rendern*. Beim *Rendern* erfolgt aus Sicht der virtuellen Kamera eine Berechnung der sichtbaren Objekte und des Aussehens der Objektflächen unter Verwendung von Materialeigenschaften und Lichtquellen. Der Grad der Bildqualität und der physikalischen Korrektheit beeinflusst die Rechenzeit für das *Rendern*. Abb. 1 zeigt Ergebnisse der manuellen Texturierung von CAD-Modellen mit 3ds, Maya und Cinema 4D. Vor der Texturierung wird das Objektteil, das eine Textur erhalten soll, ausgewählt und ggf. noch weiter unterteilt, um je nach Auflösung die Texturen eindeutig zuweisen zu können. Aus den verfügbaren Fotos wird die entsprechende Textur mit einer Bildbearbeitungssoftware ausgeschnitten und entzerrt, bevor es auf den Objektteil projiziert wird.



Abb. 1: v.l.n.r. Niendorfer Kirche als CAD-Modell und als Visualisierung mit 3ds Max, Kaiserdom Königslutter (mit Maya) und Leuchtturm Neuwerk (mit Cinema 4D)

Eine weitere Möglichkeit zur Texturierung von 3D-Modellen bietet u.a. die Software *Geomagic*. Dabei werden in dem vermaschten 3D-Oberflächenmodell und auch in dem Bild, das als Textur verwendet wird, interaktiv identische Punkte gemessen, über die anschließend die Textur auf das Objekt transformiert wird. In Abb. 2 ist das texturierte 3D-Gebäudemodell der Festung Kristiansten in Trondheim dargestellt. Das Gebäude wurde mit einem terrestrischen

Laserscanner Riegl VZ-400 gescannt und anschließend wurden die registrierten Scans zu einem 3D-Modell (bestehend aus 2.954.935 Dreiecken) vermascht und texturiert.



Abb. 2: Verschiedene Ansichten des Kristiansten Festungsgebäudes in Trondheim – Aufnahme durch Riegl VZ-400, Dreiecksvermaschung und Texturierung mit Geomagic

### 3 Automatische Erstellung von 3D-Modellen mit Texturen

Seit einigen Jahren gibt es eine große Anzahl von günstigen und leistungsfähigen Kameras (8-20 MPixel) auf dem Markt, die mit entsprechenden Algorithmen wie Structure from Motion oder Dense Image Matching effizient dichte 3D-Punktwolken und 3D-Modelle aus Bildsequenzen erzeugen. Diese Systeme werden erfolgreich für die Aufnahme und Rekonstruktion verschieden großer Objekte eingesetzt und sie stellen heute bereits Alternativen zu klassischen Messsystemen und Auswertemethoden dar. KERSTEN et al. (2012) stellen praktische Untersuchungen vor, bei denen Ergebnisse von dem Softwarepaket Bundler/PMVS2 (open source Software von der Universität von Washington) und dem Autodesk Webservice Photofly/123D Catch Beta mit Ergebnissen vom terrestrischen Laserscanning verglichen wurden.

Dabei wurde festgestellt, dass die geometrischen Differenzen zwischen den bild- und distanzbasierten Systemen bei der Modellierung von Oberflächenmodellen nicht sehr groß sind. Allerdings konnte man auch schlussfolgern, dass die bild-basierten Systeme bei großen Objekten wie z.B. Gebäuden Probleme hatten, geometrisch korrekte Ergebnisse zu erzielen. Ansprechend gute visuelle 3D-Modelle konnten mit dem Webservice von Autodesk erzielt werden, wenn die Objekte rundliche Formen aufweisen und nicht sehr groß sind. Die Skalierung dieser generierten 3D-Modelle erfolgt über einen Maßstab, der bei der Aufnahme in den Objektraum platziert wird, oder über Passpunkte. Abb. 3 zeigt texturierte Oberflächenmodelle, die mit Photofly/123D Catch Beta generiert wurden.

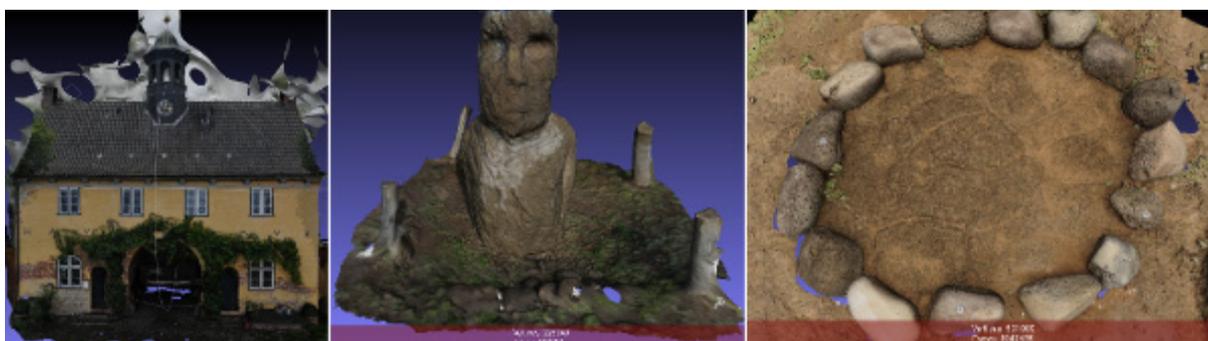


Abb. 3: Texturierte 3D-Oberflächenmodelle – Rückfassade vom Torhaus in Jersbek (links), Moai Vaihu (Mitte) und eine Schildkröte als Petroglyphe auf der Osterinsel (rechts)

## 4 Automatische Texturierung von 3D-Modellen

Für die automatische Texturierung verschieden großer 3D-Modelle als Anwendungen in Architektur und Archäologie wird die Software tmapper in der Programmiersprache C++ entwickelt. Grundlage dieser Softwareentwicklung sind die folgenden drei Methoden, die von ABDELHAFIZ (2009) in seiner Dissertation für die automatische Texturierung vorgeschlagen wurden:

1. Der Point Cloud Painter (PCP) für die Detektion von Fremdverdeckung durch Vergleich der Farben in korrespondierenden Bildregionen,
2. der Multi-Layer 3D Image (ML3DImage) Algorithmus für die automatische Texturierung und
3. der Photo Occlusion Finder (POF) als Kombination der beiden vorher erwähnten Methoden.

Für die Anwendung der oben aufgeführten Algorithmen müssen folgende Voraussetzungen erfüllt sein:

- Das 3D-Modell muss als ein unregelmäßiges Dreiecksnetz vorliegen.
- Die innere und äußere Orientierung der zu verwendenden Bilder muss mit ausreichender Genauigkeit bekannt sein.
- Das 3D-Modell und die Orientierung der dazugehörigen Bilder müssen sich auf dasselbe Koordinatensystem beziehen.

Abb. 4 zeigt das vermaschte 3D-Oberflächenmodell vom Almaqah-Tempel in Sirwah (Republik Jemen). Der Tempel wurde im Februar 2007 mit dem terrestrischen Laserscanner Trimble GS100 und der digitalen Spiegelreflexkamera Fujifilm Fine Pix S2 Pro (28 mm Objektiv) aufgenommen, in Geomagic modelliert (LINDSTAEDT et al. 2008) und anschließend durch den Multi-Layer 3D Image (ML3DImage) Algorithmus automatisch texturiert (ABDELHAFIZ (2009)).

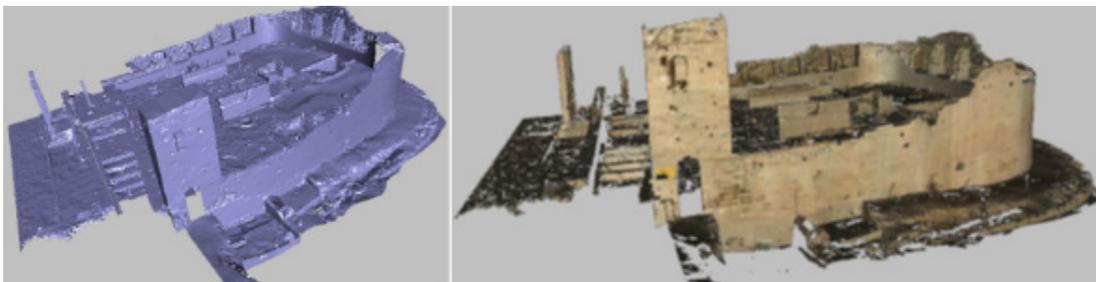


Abb. 4: Vermaschtes 3D-Oberflächenmodell vom Almaqah-Tempel in Sirwah/Jemen (links) – automatisch texturiertes 3D-Modell (rechts) (ABDELHAFIZ 2009)

Als Datengrundlage dient das Dreiecksnetz des jeweiligen Objektes, welches entweder von einer Oberflächenvermaschung (3D-Triangulation) oder von einer 3D-CAD-Konstruktion (Flächen- oder Volumenmodell) kommt. Die 3D-Aufnahmemethoden sind dafür in erster Linie terrestrisches Laserscanning, Streifenprojektion (für kleinere Objekte) und/oder photogrammetrische Verfahren, die separat oder in Kombination eingesetzt werden können. In eher seltenen Fällen werden auch Totalstationen als 3D-Erfassungsmethode für größere und einfache Objekte eingesetzt. Die Oberflächenvermaschung erfolgt durch die Triangulation von 3D-Punktwolken, die wiederum durch Messungen eines Laserscanners oder Dense Image Matching in Bilddaten (siehe auch KERSTEN et al. 2012) generiert werden (siehe Abb. 5a links). Objekte können auch aus 3D-Punktwolken oder durch diskrete photogrammetrische Punktmessungen in Stereobildern oder in mehreren Bildern im CAD konstruiert werden (KERSTEN 2006). Aus dem verwendeten CAD-Konstruktionsmodell wird ein Dreiecksnetz

abgeleitet, in dem die Daten vom AutoCAD-Format DWG z.B. durch die Software 3ds Max in ein anderes Format konvertiert werden. Das resultierende Dreiecksnetz besteht aus unterschiedlich großen Dreiecken, die auch partiell verdeckt sein können (Abb. 5a).

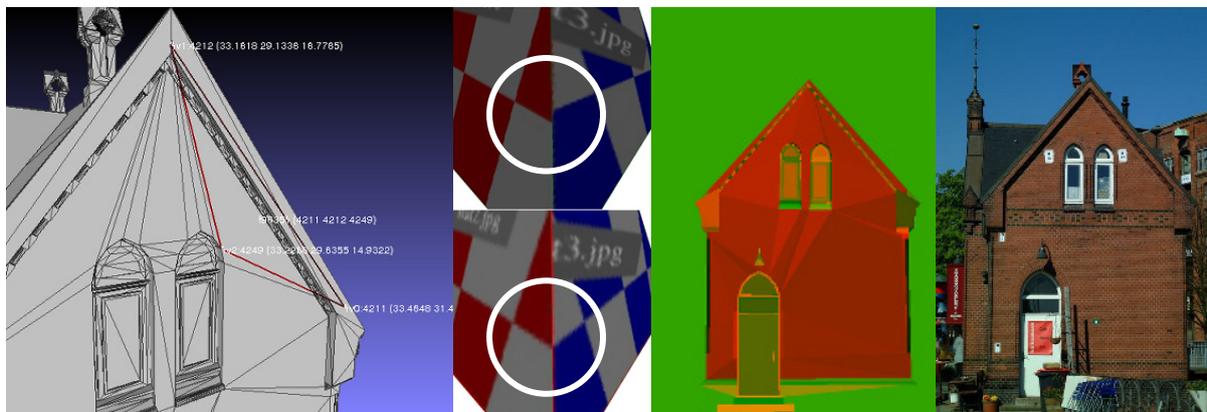


Abb. 5: a) Dreiecksnetz aus einem CAD-Modell (links); b) Affine Entzerrung (oben) und projektive (korrekte) Entzerrung (unten) synthetischer Daten im Vergleich; c) Bestimmung der Selbstverdeckung durch Strahlverfolgung (Mitte) und korrespondierendes Originalbild (rechts)

Als weitere Eingabedaten für tmapper dienen neben dem Dreiecksnetz des Objektes die digitalen Bilddaten (JPG-Format) und die Orientierungsparameter der Bilder. Die Orientierungsparameter werden üblicherweise in einer Bündelausgleichung mit Selbstkalibrierung bestimmt. Die photogrammetrischen Abbildungsmodelle der Bündelblockausgleichungsprogramme Pictran-B (Technet GmbH) und PHIDIAS (PHOCAD GmbH) sind im Programm tmapper implementiert, d.h. die Ergebnisse der Programme können unmittelbar übernommen werden.

Der im Programm tmapper implementierte ML3DImage-Algorithmus führt die Texturierung in zwei Schritten durch. Im ersten Schritt wird die Selbstverdeckung von Objektteilen bzw. von Dreiecken bestimmt (Sichtbarkeitsanalyse der Selbstverdeckung). Dazu werden die Eckpunkte (Knoten) des Dreiecksnetzes mit der bekannten Bildorientierung anhand der Kollinearitätsgleichungen in das Bild projiziert. Dadurch werden Strahlen im Raum definiert, während die Dreiecksflächen ihrerseits Ebenen im Raum definieren. Der Schnitt eines Strahls mit einer Ebene wird berechnet. Liegt der Schnittpunkt innerhalb eines Dreiecks und näher am Projektionszentrum als der zu überprüfende Eckpunkt, wird der Eckpunkt vom Objekt selbst verdeckt und ist im Bild nicht sichtbar. Andernfalls ist der Punkt sichtbar.

Und im zweiten Schritt erfolgt die Auswahl der Textur nach der Sichtbarkeit der Eckpunkte jedes einzelnen Dreiecks. Dabei wird einem Dreieck nur dann eine Textur zugewiesen, wenn deren drei Eckpunkte in mindestens einem Bild sichtbar sind. Sind jedoch die drei Eckpunkte in mehreren Bildern sichtbar, so wird das Bild verwendet, indem das Dreieck mit dem größten Flächeninhalt abgebildet wird, d.h. es wird das Bild mit der besten Auflösung verwendet. Die Textur wird dem Bild unmittelbar entnommen, d.h. bei der ML3DImage-Methode findet keine strenge Entzerrung (projektive Abbildung), sondern nur eine affine Abbildung statt. Sind die Dreiecksflächen wie bei Laserscanning-Daten relativ klein, können die auftretenden Verzerrungen vernachlässigt werden. Der Effekt wirkt sich allerdings signifikant bei größeren Dreiecken und Bildneigungen aus. Abb. 5b (oben und unten) zeigt die Auswirkung auf die Texturierung am simulierten Modell einer Pyramide. Die Schachbrettmuster auf den Seitenflächen der Pyramide sind im linken Bild etwas versetzt, dagegen stoßen die Muster im rechten korrekt aufeinander.

Die Berechnung der Selbstverdeckung benötigt die meiste Rechenzeit (>99% der gesamten Berechnungszeit). Um die rechenintensiven Prozesse zu beschleunigen, soll in Zukunft die Graphics Processing Unit (GPU) verwendet werden. Denn die dafür notwendigen geometrischen Operationen sind direkt in der GPU-Hardware implementiert. Als Programmierschnittstelle könnte dafür OpenGL (<http://www.opengl.org>) oder Direct3D verwendet werden. Zur Sichtbarkeitsanalyse wird in der Computergraphik oft das Z-Puffer-Verfahren oder die Strahlverfolgung (engl. ray tracing) verwendet. Das Z-Puffer-Verfahren ist bereits in der GPU. Die GPU verfügt über viele und schnell parallel arbeitende Geometrie-Recheneinheiten (Geometry-Shader), die diese Aufgabe übernehmen.

Die Strahlverfolgung ist ein alternatives Verfahren und wird in der Bildsynthese verwendet. Hier wird auf eine hohe Bildqualität und eine physikalische Korrektheit Wert gelegt. Da die Strahlverfolgung recht einfach implementiert werden kann, wurden erste Versuche in tmapper damit ausgeführt. Das Ergebnis der Sichtbarkeitsanalyse mittels Z-Puffer-Verfahren oder Strahlverfolgung ist ein Bild mit der gleichen Größe und Orientierung des Originalbildes (siehe Abb. 5c rechts). Jedes Pixel gibt an, ob und welches Dreieck in diesem Pixel sichtbar ist. Zur visuellen Kontrolle und zur Speicherung wurde das Sichtbarkeitsbild in ein RGB-Farbbild umgerechnet, wobei jedem einzelnen Dreieck genau eine RGB-Farbe zugeordnet wird.

## 5 Ergebnisse der automatischen Texturierung

Der in tmapper implementierte Multi-Layer 3D Image (ML3DImage) Algorithmus wurde anhand folgender verschiedener Datensätze (3D-Modelle), deren technische Spezifikationen in Tabelle 1 aufgeführt sind, getestet.

3D-Modell	# Punkte	# Dreiecke	# Bilder	Kamera
Inschriftenstein, Sirwah (Jemen)	26840	53120	12	Fujifilm S2
Kristiansten Festung, Trondheim	150646	298137	24	Nikon D70
Fleetschlösschen, Hamburg-HafenCity	17371	37219	9	Fujifilm S2

Tab. 1: Technische Spezifikationen der verwendeten Testdaten (3D-Modelle)

Die 3D-Modelle des sabäischen Inschriftensteines des Almaqah-Tempels in Sirwah (Republik Jemen) und der Kristiansten Festung in Trondheim (Norwegen) wurden aus terrestrischen Laserscanning-Daten durch Dreiecksvermaschung in Geomagic abgeleitet. Der Inschriftenstein wurde im Februar 2007 mit dem terrestrischen Laserscanner Trimble GS100 und der digitalen Spiegelreflexkamera Fujifilm Fine Pix S2 Pro (28 mm Objektiv) aufgenommen (LINDSTAEDT et al. 2008), während das Festungsgebäude in Trondheim im September 2010 mit dem Laserscanner Riegl VZ-400 und der digitalen Kamera Nikon D70 (28 mm Objektiv) erfasst wurde. Dagegen wurde von dem Fleetschlösschen, das in der Hamburger HafenCity steht, auf Grundlage von Laserscanning-Daten (Trimble GS100) und photogrammetrischen Aufnahmen (Kamera Fujifilm S2 mit 28 mm Objektiv) durch kombinierte Auswertung und Modellierung ein 3D-CAD-Modell im Rahmen eines studentischen Projektes im Sommer 2006 erstellt. Abb. 6 zeigt die zwölf für die automatische Texturierung verwendeten Bilder des Inschriftensteins, während in Abb. 7 das texturierte 3D-Modell dargestellt ist, in dem jedoch noch eine Lücke aufgrund eines fehlenden Fotos vorhanden ist.



Abb. 6: Übersicht der verwendeten Bilder für die automatische Texturierung des sabäischen Inschriftensteins im Almaqah-Tempel von Sirwah (Jemen)



Abb. 7: Ergebnis der automatischen Texturierung für das 3D-Modell des sabäischen Inschriftensteins im Almaqah-Tempel von Sirwah (Jemen)

Die Bestimmung der Bildorientierungs- und der Kamerakalibrierungsparameter erfolgte mit der photogrammetrischen Auswertesoftware PICTRAN (Kristiansten Festung und Fleetschlösschen) und mit Phidias (Inschriftenstein). Da für das geometrische Kameramodell die zusätzlichen Parameter der Selbstkalibrierung bei der Ausgleichung berücksichtigt werden, wird eine recht hohe geometrische Genauigkeit der Orientierungsparameter durch die Bündelausgleichung erreicht. Wenn das texturierte Modell noch über signalisierte Passpunkte verfügt, dann kann die Genauigkeit der automatischen Texturierung durch Überlagerung des texturierten Modells mit den Passpunkten überprüft werden. Die Textur des Passpunktes muss mit dem Passpunkt selbst, hier dargestellt als transparente Kugel, möglichst gut zusammenfallen (siehe Abb. 8).



Abb. 8: Überlagerung des texturierten 3D-Modells mit Passpunkten zur geometrischen Qualitätskontrolle

Ein problematischer Aspekt wird in Abb. 11a aufgezeigt: Am Rand bzw. an Kanten des Modells kann es durch den Hintergrund, hier der blaue Himmel, zu Farbsäumen kommen, die sich momentan automatisch nicht korrigieren lassen. In diesem Fall müsste der Anwender zur Steigerung der radiometrischen Qualität des Modells manuelle Korrekturen vornehmen.

Abb. 9 zeigt die 24 für die automatische Texturierung verwendeten Bilder des Festungsbauwerks in Trondheim, während in Abb. 10 zwei Ansichten des texturierten 3D-Modells abgebildet sind.



Abb. 9: Übersicht der 24 verwendeten Bilder für die automatische Texturierung des 3D-Modells der Kristiansten Festung



Abb. 10: Zwei Ansichten vom Ergebnis der automatischen Texturierung für das 3D-Modell der Kristiansten Festung

In Abb. 10 erkennt man folgende Probleme der automatischen Texturierung: a) die Texturen des Gebäudes wurden radiometrisch nicht angepasst, da diese Anwendung noch nicht in tmapper implementiert ist, b) einige Dreiecke haben keine Textur (siehe auch Abb. 11b), c) Dachbereiche können nicht texturiert werden, wenn keine Bilddaten aufgrund der Aufnahmeconfiguration vorliegen, und d) Objekte im Vordergrund werden ohne Sichtbarkeitsanalyse der Fremdverdeckungen auf das modellierte Objekt texturiert (siehe Abb. 10 links, rechte Gebäudeseite unten), wenn sie nicht vorher manuell retuschiert werden wie in Abb. 2.

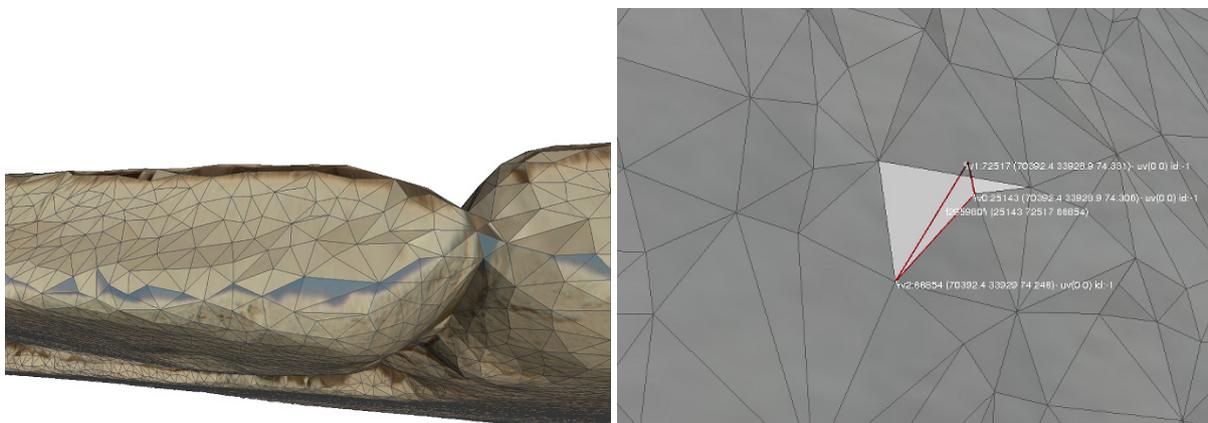


Abb. 11: Problemfälle: a) Farbsäume beim 3D-Modell des Inschriftensteins (links) und b) Dreiecke ohne Textur beim 3D-Modell der Kristiansten Festung (rechts)

Dreiecke ohne Textur stellen ein generelles Problem des ML3DImage Algorithmus dar, denn bei dieser Methode wird Dreiecken nur eine Textur zugewiesen, wenn mindestens alle drei Eckpunkte sichtbar sind. Da Dreiecke aus Laserscanning-Daten allgemein relativ klein sind und ein zusammenhängendes Dreiecksnetz bilden, werden die meisten Dreiecke texturiert (siehe Abb. 10). Wie auch in Abb. 11b jedoch zu sehen ist, wurden einige Dreiecke an dem Gebäude der Festung Kristiansten nicht automatisch texturiert, weil sie partiell verdeckt sind. Da die aus der CAD-Konstruktion abgeleiteten 3D-Modelle recht große Dreiecke aufweisen können, stellt sich das Problem der unvollständigen Texturierung hier momentan deutlicher. Abb. 12 zeigt das Fleetschlösschen nach der automatischen Texturierung. Viele Dreiecke wurden nicht texturiert, da in der Sichtbarkeitsanalyse eine Selbstverdeckung nachgewiesen

wurde, obwohl ein großer Teil der Dreiecksfläche noch sichtbar ist (siehe Abb. 13 links). Zukünftig wird eine Sichtbarkeitsanalyse mittels Z-Puffer-Verfahren oder Strahlverfolgung das Problem lösen können. Eine automatische so genannte „quick& dirty“ Texturierung ohne Sichtbarkeitsanalyse generiert eine im ersten Blick visuell ansprechende Lösung, jedoch mit einigen Fehlern bei der Platzierung der Texturen und in der Geometrie.



Abb. 12: Ergebnis der automatischen Texturierung für das 3D-Modell des Fleetschlösschens in Hamburg-HafenCity – Problem: Dreiecke ohne Textur bei Verwendung der Sichtbarkeitsanalyse vor der automatischen Texturierung



Abb. 13: Unvollständige Texturierung (links) und automatische „quick& dirty“ Texturierung mit Fehlern beim 3D-Modell des Fleetschlösschens (rechts)

## 6 Fazit und Ausblick

In diesem Beitrag wird das Potential der manuellen und automatischen Texturierung von 3D-Modellen anhand von verschiedenen Beispielen vorgestellt. Durch pixel-basierte Matching-Verfahren in digitalen Bilddaten können heute bereits vollautomatisch texturierte 3D-Oberflächenmodelle generiert werden, die bei Objekten kleinerer bis mittlerer Größe die Qualität von Laserscanning-Daten erreichen können (KERSTEN et al. 2012). Mit dem Programm tmapper wird eine Software zur automatischen Texturierung von 3D-Modellen (Drei-

ecksvermaschung und CAD) entwickelt, die den Multi-Layer 3D Image (ML3DImage) Algorithmus verwendet. Dabei zeigte sich jedoch die Einschränkung, dass die Texturierung mit der ML3DImage-Methode nur für Dreiecksnetze aus Laserscanning-Daten mit möglichst kleinen Dreiecken brauchbar ist. Die Methode ist für 3D-Modelle aus CAD-Anwendungen ungeeignet, da die Bestimmung der Selbstverdeckungen nur auf den Eckpunkten der Dreiecke basiert und somit eine große Anzahl von Dreiecken ohne Textur bleibt. Eine Lösung für dieses Problem wäre die Bestimmung von Selbstverdeckungen für jedes Pixel im Bildraum durch die Z-Puffer-Methode oder die Strahlverfolgung. Da beide Verfahren sehr rechenintensiv sind, bietet es sich an, die Berechnungen auf dem Graphikprozessor durchführen zu lassen. Um außerdem das Problem der Fremdverdeckungen bei der pixel-basierten Sichtbarkeitsanalyse zu lösen, ist es naheliegend, die beiden Methoden mit dem Point-Cloud-Painter-Algorithmus (PCP) zu verbinden.

Momentan kann jedoch nur die manuelle Texturierung von 3D-Modellen für Anwendungen in der Architektur und Archäologie den hohen geometrischen, radiometrischen und visuellen Ansprüchen gerecht werden.

## 7 Literaturverzeichnis

- ABDELHAFIZ, A., 2009: Integrating Digital Photogrammetry and Terrestrial Laser Scanning. DGK Reihe C, Heft Nr. 631.
- BOLLMANN, J. & KOCH, W.G., (Hrsg.) 2002: Lexikon der Kartographie und Geoinformatik in zwei Bänden. Bd. 2, Karto bis Z, Spektrum Akademischer Verlag GmbH, Heidelberg, Berlin, 455 S.
- KADA, M., 2007: Zur maßstabsabhängigen Erzeugung von 3D-Stadtmodellen. Dissertation, Universität Stuttgart.
- KERSTEN, TH., 2006: Kombination und Vergleich von digitaler Photogrammetrie und terrestrischem Laserscanning für Architekturanwendungen. Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V., Band 15, Hrsg. E. Seyfert, 26. Wissenschaftlich-Technische Jahrestagung der DGPF, 11.-13. Sept. 2006 in Berlin, pp. 247-256.
- KERSTEN, TH., LINDSTAEDT, M., MECHELKE, K. & ZOBEL, K., 2012: Automatische 3D-Objektrekonstruktion aus unstrukturierten digitalen Bilddaten für Anwendungen in Architektur, Denkmalpflege und Archäologie. Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V., Band 21, Hrsg. E. Seyfert, 32. Wissenschaftlich-Technische Jahrestagung der DGPF, 14.-17. März 2012 in Potsdam.
- LINDSTAEDT, M., KERSTEN, TH., MECHELKE, K., GÖTTING, M. & HEIDEN, R., 2008: Virtuelles 3D-Modell der antiken Tempelanlage in Sirwah/Jemen zur archäologischen Objektdokumentation durch terrestrisches Laserscanning und Photogrammetrie. Publikationen der Deutschen Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation e.V., Band 17, Hrsg. E. Seyfert, 28. Wissenschaftlich-Technische Jahrestagung der DGPF, 23.-25. April 2008 in Oldenburg, pp. 59-68.
- LUHMANN, T., 2010: Nahbereichsphotogrammetrie – Grundlagen, Methoden und Anwendungen. 3., völlig neu bearbeitete und erweiterte Auflage, Wichmann, VDE Verlag GmbH, Berlin und Offenbach, 674 Seiten.