

Diplomarbeit

**„Architekturvisualisierung mittels 3D-Programmen -
fotorealistische Darstellung von Bauobjekten.“**

Ausgeführt zum Zweck der Erlangung des akademischen Grades

Dipl. Ing. (FH) Telekommunikation und Medien

am Fachhochschul-Diplomstudiengang Telekommunikation und Medien St. Pölten

unter der Erstbetreuung von

Mag. Peter Adametz

Zweitbegutachtung von

Dipl. Ing. Hannes Raffaseder

ausgeführt von

Beate Mungenast

tm0210038081

Wien, am 11. 08.2006

Unterschrift:

Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Diplomarbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.

- ich dieses Diplomarbeitsthema bisher weder im Inland, noch im Ausland einem Begutachter/einer Begutachterin zur Beurteilung, oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Diese Arbeit stimmt mit der, von den Begutachtern beurteilten, Arbeit überein.

Ort, Datum

Unterschrift

Zusammenfassung

Diese Arbeit beschäftigt sich mit der fotorealistischen Darstellung von Architekturmodellen. Besonderes Augenmerk wird dabei auf Beleuchtung, Materialien, Kamera und Rendertechniken gelegt. Die einzelnen Techniken werden mit Beispielen aus der Architekturvisualisierung ergänzt.

Auch der allgemeine Arbeitsablauf der Erstellung eines 3D-Modells wird behandelt.

Darüber hinaus werden verschiedenen Visualisierungsformen näher beschrieben. Des Weiteren wird ein Einblick in die Geschichte von dreidimensionalen analogen Architekturmodellen in dieser Arbeit gegeben, waren sie doch lange Zeit die einzige Möglichkeit ein Bauobjekt vor dem Bau dreidimensional darzustellen.

Abstract

This thesis is about the photorealistic presentation of achitectural models. Lighting, materials, camera and render techniques are of special importance. Examples of architectural visualisation complete the techniques discussed. This paper describes the general procedure of the realisation of a 3D-model and different forms of visualisation.

Further more, it gives an insight into the history of three-dimensional, analogical architectural models, as they used to be the only possibility to present a building-project three-dimensionally before construction.

Inhaltsverzeichnis

I.	Inhaltsverzeichnis	4
II.	Einleitung	7
II 1	Definition des Erkenntnisgegenstandes	8
II 2	Problembenennung	8
II 3	Ziel und Aufbau der Arbeit	9
II 4	Forschungsleitende Fragestellung.....	9
II 5	Hypothesen.....	10
III.	Entwicklung, Geschichte	11
IV.	Programme	13
IV 1	CAD und CAAD	13
IV 1.1	2D CAD	14
IV 1.2	Objektorientiertes CAD CAAD	15
	Graphisoft ArchiCAD	15
IV 2	3D-Software.....	17
IV 3	Austauschformate, Kompatibilitätsproblematik.....	18
IV 3.1	Verwendung von alten Dateiformaten	18
	Probleme bei der Konvertierung.....	19
IV 3.2	Verwendung eines speziellen Programms zum Datenaustausch ..	21
IV 3.3	Eigens programmierte interne Software	21
V.	Architekturvisualisierung	22
V 1	Visualisierungsformen	22
V 1.1	2D [zweidimensional]	23
	Rendering.....	23
	Fotomontage	24
V 1.2	3D [dreidimensional]	25
	[analoges] Architekturmodell	25
	3D-Plot	25
V 1.3	4D [vierdimensional]	26
	Animation	26
V 1.4	4Di [vierdimensional, interaktiv]	27
	QuickTime VR [Virtual Reality]	27
	Echtzeitgrafik.....	28
V 2	Arbeitsablauf.....	29
V 2.1	Ideenfindung	29
V 2.2	Pläne 3D Modell	31
V 2.3	Nachbearbeitung	33

V 3	Materialien Texturen	35
V 3.1	Surface Shading Techniques Schattierungsmethoden	36
	Faceted Shading Polygonal Shading.....	36
	Smooth Shading	37
	Specular Shading	37
V 3.2	Material Surface Shaders	38
V 3.3	Reflexion	39
	Umgebungsreflexion Ambient Reflection.....	39
	Diffuse Reflexion	40
	Spiegelreflexion Specular Reflection	40
V 3.4	Selbstillumination	41
V 3.5	Materialfarbe	41
V 3.6	Transparenz	42
V 3.7	Image Mapping Texturierung	43
	Erstellen einer Image Map.....	43
	Projektionsmethoden.....	44
	Color Maps	45
	Transparency Maps.....	46
	Reflection Maps.....	48
	Environment Maps	49
	Bump Maps Relief-Maps	51
	Displacement Maps	52
	Zweidimensionale Prozedur Maps	52
	Dreidimensionale Prozedur Maps	53
V 4	Licht	53
V 4.1	Lichtquellen	54
	Ambient Light Umgebungslicht.....	54
	Omni Light Point Light Punktlicht.....	55
	Spot Light Scheinwerfer	56
	Infinite Light Directional Light	56
	Area Light Flächenlichter	57
	Linear Light.....	58
	Volumenlicht Volume Light	58
V 4.2	Lichteigenschaften	59
	Lichtfarbe und Intensität	59
	Lichtabnahme und Fall-Off	60
	Licht-Einfallswinkel	61
	Schein Glow Linseneffekte	61
	Globale und Lokale Lichtquellen	62
	Schatten	63
V 4.3	Lichtfunktionen	64
	Hauptlicht Führungslicht [Key Light]	64
	Fülllicht	64
V 4.4	Image Based Lighting [IBL]	65

V 5	Kamera	66
V 5.1	Allgemein	66
	Kameralinsen, Brennweite.....	66
	Bildformat	67
	Kamerastandpunkt POV [Point of View], POI [Point of Interest].....	67
	Unschärfe Tiefenschärfe DOF [Depth of field]	68
V 5.2	Kameraanimation	69
	Änderung der Kameraposition.....	69
	Änderung der Kameraorientierung	69
	Pfadanimation	70
V 6	„Zubehör“	71
V 6.1	Billboards	72
V 6.2	RCP [Real Photorealistic Content]	72
V 6.3	Crowd Simulation	73
V 6.4	Möbel, Fahrzeuge	74
V 6.5	Pflanzengeneratoren	75
V 7	Rendertechniken.....	78
V 7.1	Scanline Rendering	78
V 7.2	Z-Buffer	79
V 7.3	Raytracing	79
V 7.4	globale Beleuchtungsverfahren	82
V 7.5	Radiosity	82
V 7.6	Photon Mapping	84
V 7.7	Image Based Rendering	85
V 7.8	Render-Software Renderer	86
VI.	Zusammenfassung	87
VI 1	Beantwortung der forschungsleitenden Fragestellung	87
VI 2	Überprüfung der Hypothesen	89
VI 3	Reflexion und Ausblick	91
VII.	Anhang	94
VII 1	Glossar	94
VII 2	Literaturverzeichnis.....	99
VII 2.1	Buch Zeitschrift	99
VII 2.2	Internet	102
VII 3	Abkürzungsverzeichnis.....	104
VII 4	Softwareverzeichnis.....	105
VII 5	Abbildungsverzeichnis	106

I. Einleitung

“Das Architekturmodell ist ein Medium der Darstellung. Es bündelt ästhetische und technische Informationen, die zuvor auf anderen Ebenen festgelegt wurden, und übersetzt sie in eine plastische Form.“

[Evers|1994|S.10]

Seit der Frührenaissance werden Bauobjekte mit Hilfe von dreidimensionalen Modellen, damals noch in Form von analogen Architekturmodellen aus Holz, Gips und ähnlichen Materialien, visualisiert.

Der Architekt hat dadurch die Möglichkeit sein Werk in einer, auch für Nicht-Fachleute verständlichen Form, zu präsentieren was die Kommunikation zwischen Auftraggeber und Architekt erheblich verbessern kann.

In den 80er Jahren begann der Einsatz von CAD [Computer Aided Design | Computer Aided Drafting] -Software in Architekturbüros. Das händische Zeichnen von Plänen wurde durch das Zeichnen am Computer abgelöst.

Durch die ständige Weiterentwicklung von Soft- und Hardware ist es heute möglich auch fotorealistische Architekturmodelle zu erstellen. Auch das Generieren von Virtual Reality Modellen, durch die sich der Betrachter schon vor dem Errichten bewegen kann, ist möglich.

Architekturmodelle zu einem unverzichtbaren Bestandteil von fast jedem größeren Bauvorhaben entwickelt.

I|1 Definition des Erkenntnisgegenstandes

Der Bereich der Architekturvisualisierung umfasst verschiedenste Visualisierungsformen wie einfache Skizzen, Bilder, aber auch Animationen bis hin zu Virtual-Reality-Anwendungen.

Was eingesetzt wird, hängt einerseits vom Zweck, den die Visualisierung erfüllen soll, andererseits natürlich vom verfügbaren Budget ab.

Je nach Art der Darstellung variiert die Komplexität des Modells. Wenn zum Beispiel ein Bild der Fassade eines Gebäudes generiert werden soll, sind Innenräume und Rückseite des Bauwerks völlig unwichtig und müssen somit auch nicht modelliert werden, dafür können aber Pflanzen und Menschen, die im Bild zu sehen sind, sehr wichtig für den Gesamteindruck sein.

Geht es aber darum, ein 3D-Modell zu erstellen, aus dem in weiterer Folge ein analoges Modell mittels 3D-Plotter generiert werden soll, muss das Computermodell exakt sein. Dafür müssen keine Personen, Bäume oder Fahrzeuge platziert werden.

Manchmal existiert zur Zeit der Entstehung des 3D-Modells auch nur eine Vision, die es dann ohne genaue Vorgaben umzusetzen gilt.

I|2 Problembenennung

Spezialsoftware für Architekturplanung wie *Graphisoft ArchiCAD*¹ entspricht im Bereich der fotorealistischen Darstellung meist nicht den Anforderungen der Architekten. Stattdessen müssen herkömmliche 3D-Programme angewandt werden, da diese über leistungsstarke *Renderers*² verfügen, mit welchen sich aufwendige Rendereffekte berechnen lassen.

Um die mittels 3D-Programmen erzeugten architektonischen Modelle möglichst fotorealistisch aussehen zu lassen, müssen aufwendige Techniken in Anspruch

¹ siehe III|1.2 Objektorientiertes CAD|CAAD

² siehe IV|7.8 Render-Software | Renderer

genommen werden, die eventuell nicht dem Know-how des Architekten entsprechen.

Die Genauigkeit des Modells hängt vom Einsatzgebiet ab.

I|3 Ziel und Aufbau der Arbeit

Diese Arbeit soll die wichtigsten Aspekte der fotorealistischen Visualisierung von Bauobjekten mit 3D-Programmen aufzeigen.

Kapitel 3 beschäftigt sich mit der Geschichte von Architekturmodellen.

In Kapitel 4 werden die wichtigsten in der Architektur verwendeten Software-Gruppen näher erläutert und der Austausch von Dateien zwischen diesen behandelt.

In Kapitel 5 wird der allgemeine Arbeitsablauf einer Visualisierung, die verschiedenen Visualisierungsformen, Materialien, Beleuchtung, Kameraführung, „Zubehör“ zu den eigentlichen Modellen und Rendertechniken beschrieben und mit Beispielen aus der Architekturvisualisierung ergänzt.

I|4 Forschungsleitende Fragestellung

Welche Lösungen gibt es zum Datenaustausch zwischen den Programmen

Was ist im Bereich der Architekturvisualisierung besonders zu beachten?

In welchen Phasen der Planung wird Architekturvisualisierung mittels 3D-Programmen angewandt?

Wie schaut der Arbeitsablauf im Allgemeinen aus?

Welche Render-Techniken kommen der Architekturvisualisierung zugute?

Von welchen Funktionen der 3D-Programme profitiert dieser Bereich besonders?

I|5 Hypothesen

Es ergeben sich folgende Hypothesen:

Der Datenaustausch zwischen den einzelnen verwendeten Programmen ist kompliziert.

Beleuchtung, Materialien und Kameraperspektiven sind für die fotorealistische Darstellung von Architekturmodellen besonders wichtig.

Es gibt noch keinen standardisierten Arbeitsablauf im Bereich der digitalen 3D-Architekturvisualisierung.

Architekturvisualisierung mit 3D-Programmen wird hauptsächlich zur Präsentation des fertig geplanten Bauobjekts verwendet.

Digitale 3D-Modelle werden analoge Architekturmodelle vollständig ersetzen.

II. Entwicklung, Geschichte

Das Architekturmodell ist eine originäre Erfindung der Frührenaissance, was durch zahlreiche dokumentierte Modelle aus der Zeit zwischen 1350 und 1500 belegbar ist.

Es gibt zwar auch Modelle, die früher entstanden sind, doch dienten diese nicht zur Veranschaulichung von Bauvorhaben sondern als Grabbeigaben oder Weihgeschenke.

Die ersten architektonischen, nachweisbaren Gesamtmodelle sind ein Maßstabsmodell des Doms in Florenz [1367] und eines für das S. Petronio in Bologna [1390]. Diese unterscheiden sich jedoch noch sehr von unserer heutigen Vorstellung von Architekturmodellen. Das Modell von S. Petronio beispielsweise wurde aus normal großen Ziegeln im Maßstab 1:12 gefertigt. Es war 15 m lang und 3,5 m hoch, der Bau dauerte zwei Jahre.

Ende des 13. Jahrhunderts taucht das erst Mal das Wort „modello“ als korrekte Bezeichnung für ein architektonisches Modell in den Mailänder Dombauakten auf. Bei dieser Miniaturdarstellung handelte es sich bereits um ein transportables Holzmodell, das farbig bemalt wurde, um ihm das Aussehen eines kleinen Bauwerks zu geben.

[vgl. Evers|1994|S.10-12]

Da die Anzahl von Nicht-Fachleuten auf entscheidenden Posten der Baubehörde zunahm, veränderten sich die Planungs- und Darstellungsmittel und der künstlerische Wettbewerb wurde als wichtigster Entscheidungsprozess eingeführt. In der Folge ging die technische Leitung der Bauhütten von den Klerikern auf die Zünfte über und diese wollten sich durch Wettbewerbe die besten Ergebnisse schon im Vorfeld sichern.

Als primäres Entscheidungs- und Darstellungsmittel wurde das Modell gewählt, da es auch für Nicht- Fachleute leicht verständlich war.

Auch die Architekten erkannten schnell den Vorteil von Miniaturmodellen als Präsentationsmittel, da sie auf Auftraggeber viel überzeugender wirkten als Zeichnungen.

Grundsätzlich kann aber kein Bauobjekt ohne Architekturzeichnung bzw. Plan gebaut werden, ein Modell ist aber im Gegensatz zur Zeichnung kein notwendiges Element für den Erfolg der Errichtung eines Gebäudes.

Andererseits haben sich Architekturmodelle zu einem unverzichtbaren Bestandteil von fast jedem größeren Bauvorhaben entwickelt.

[vgl. Evers|1994|S.10-12]



Abb. III – 1 | Modell eines Hauses mit Garten, Theben, 209-1998 v. Chr.



Abb. III - 2 | Holzmodell des Doms von Pavia, Cristoforo Rocchi und Giovan Pietro Gugazza, ca. 1490

III. Programme

Für das Zeichnen von Plänen werden seit den 80er Jahren CAD-Programme verwendet. Später wurde auch CAD-Software speziell für die Anforderungen der Architekturplanung, so genannte CAAD [Computer Aided Architectural Design] - Programme, entwickelt.

Im Bereich der fotorealistischen Visualisierung dominiert jedoch 3D-Software, da sie mehr Möglichkeiten zur realistischen Darstellung von Bauobjekten bietet. In den folgenden Kapiteln werden diese Softwaregruppen genauer beschreiben.

III|1 CAD und CAAD

Architektonische Entwürfe und Pläne werden seit langem mittels CAD und CAAD Software erstellt. Deshalb sollen im folgenden Kapitel kurz die Entstehung und die verschiedenen Ausprägungen solcher Programme erläutert werden.

Der Begriff CAD tauchte erstmals in den 50er Jahren auf, damals gab es zwar noch keine Ausgabe mittels eines Displays. Die Idee, eine Kathodenstrahlröhre an einen Computer anzuschließen, stand aber schon im Raum.

Außerdem war passive Computergrafik, also die Möglichkeit einer grafischen Darstellung durch einen Drucker, schon möglich. Fortschritte in der interaktiven Computergrafik waren aufgrund der beschränkten Rechnerleistung vorläufig nur minimal zu verzeichnen.

1962|63 wurde erstmals ein interaktives Grafikprogramm vorgestellt: Ian E. Sutherlands Sketchpad, das am MIT auf einem Lincoln TX-2 lief. Sketchpad war das erste Programm mit einer grafischen Benutzeroberfläche [GUI – Graphical User Interface]. Es nutzte bereits einen Röhrenbildschirm und wurde mit dem kurz davor erfundenen Light Pen bedient. Eine weitere Besonderheit war die Verwendung von Objekten und Instanzen.

Ein Jahr später wurde ein experimentelles, interaktives Entwurfssystem namens COPLANNER von Souder und Clarc präsentiert.

Ein weiteres Programm, das in den 60er Jahren entstand, war URBAN5, entwickelt unter der Leitung von Nicholas Negroponte.

In den 70er Jahren begann der Einsatz von CAD in großen, kapitalkräftigen Firmen. Der Begriff CAAD wurde besonders durch die Veröffentlichung des Buches „Computer-Aided-Architectural-Design“ von William J. Mitchell geprägt.

In den 80ern fanden CAD|CAAD Systeme durch die enormen Kostenreduktionen auch in kleinen und mittleren Architekturbüros Verwendung.

Der Einsatz von CAD-Software hatte grundsätzliche Änderungen in der Arbeitsweise der Architekturbranche zur Folge.

[vgl. AmSchö|2003|S.3 | StreWei|1996|S.16-18 | www|wiki|sketchpad]

III|1.1 2D CAD

Bei 2D CAD-Programmen werden die Daten am Computer zweidimensional dargestellt. Im Prinzip handelt es sich dabei um ein digitales Zeichenblatt. Zu den 2D CAD-Programmen zählt z.B. Autodesk AutoCAD. Obwohl diese mittlerweile auch durch 3D-Funktionen erweitert wurden, wird oft nur die 2D-Funktionalität genutzt, da diese Programme von vorn herein nicht für 3D-Anwendungen konzipiert waren. Daher sind diese Systeme noch nicht richtig ausgereift und die 3D-Konstruktion sehr aufwendig und kompliziert.

[vgl. www|2Dcad]

III|1.2 Objektorientiertes CAD|CAAD

„Computer Aided Architectural Design [CAAD] bedeutet (...) die maschinelle Unterstützung des architektonischen Denkens.“

[IEMAR|2003|S.20]

In den Bereich der objektorientierten CAD Programme fallen Graphisoft ArchiCAD, das mittlerweile aufgelassene Autodesk Architectural Desktop, Autodesk Revit, Nemetschek Allplan und Computerworks VectorWorks.

Beim klassischen CAD bestehen Elemente wie Wände etc. aus vielen einzelnen Punkten, Linien und Flächen, beim objektorientierten CAD wird ein Objekt als einzelnes Element behandelt. Es kann als ganzes gedreht, skaliert, kopiert, etc. werden. Es werden also dreidimensionale Gebäudekomponenten simuliert und das Modell entsteht von vorn herein in einer 3D-Ansicht, aus der dann erst in weiterer Folge 2D-Pläne erstellt werden können.

Richtig eingesetzt ist objektorientierte CAD-Software auch in der Lage, aus dem 3D-Modell Mengendaten wie z.B. die Anzahl der benötigten Fenster eines Gebäudes zu kalkulieren.

Als Beispiel für ein CAAD Programm soll im Folgenden Graphisoft ArchiCAD näher beschrieben werden:

Graphisoft ArchiCAD

Graphisoft ArchiCAD ist seit 1984 auf dem Markt, die Software war damals allerdings nur für Apple-Rechner verfügbar. Mittlerweile ist sie eine der wenigen CAAD-Anwendungen, die sowohl für Mac OS als auch für Windows erhältlich ist.

Der dxf|dwg-Datenaustausch funktioniert in ArchiCAD sehr gut, dafür fehlt seit Version 8 der obj – Export, der in den vorhergehenden Versionen vorbildlich war.

[vgl. KuHa|03|2005|S.45 | KuHa|06|2005|S.57 | KuHa|2004|S.40]

Ein spezielles Feature von ArchiCAD ist die GDL [Geometric Description Language] – Objekttechnologie. Mit dieser auf Basic basierenden Scriptsprache ist es möglich, Objekte zu erstellen und diese so oft wie gewünscht im gleichen und auch in anderen Projekten zu verwenden.

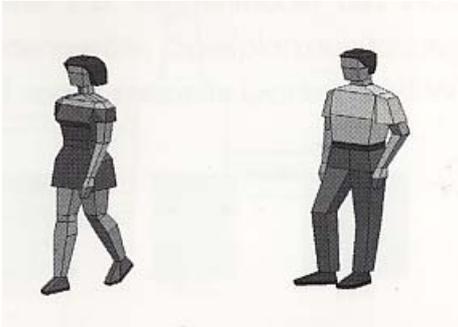


Abb. IV-1 | GDL-Frau, GDL-Mann

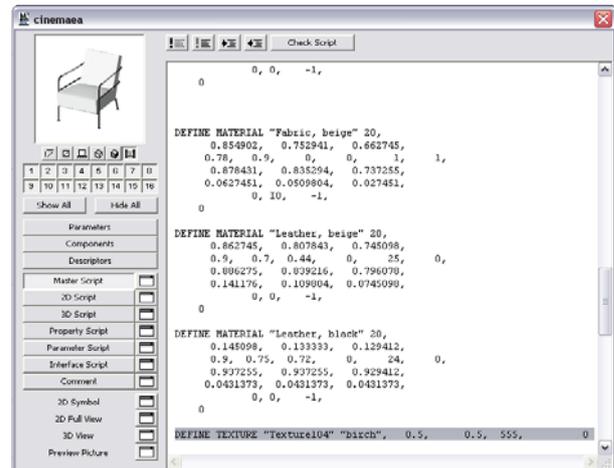


Abb. IV-2 | GDL- Objekt

Mittlerweile gibt es unzählige Objektbibliotheken. Die so genannte Basisbibliothek ist im Lieferumfang der ArchiCAD Lizenz enthalten und verfügt über ungefähr 1000 Objekte.

Darüber hinaus hat der Benutzer auch die Möglichkeit, sich von verschiedenen Möbelherstellern [z.B. Alvar Aalto], teilweise sogar gratis, GDL Bibliotheken herunter zu laden, die Modelle von existierenden Möbelstücken enthalten.

Neben Möbelstücken sind z.B. auch Bäume, Menschen, Tiere, Vorhänge etc. als GDL Bibliotheken erhältlich.

[vgl. MartPet|2002|S.43-51]

III|2 3D-Software

„(...)allzu gut lassen sich mit ihr wesentliche Aspekte der Architektur visualisieren: Raum, Farbe, Materialität, Licht und Schatten. Die beiden letzten nehmen aus meiner Sicht eine Sonderstellung ein, weil sie in besonderem Maße den Einsatz von 3D-Software rechtfertigen(...)“

[Sonder|2006|S. 6]

Der Einsatz von leistungsfähiger 3D-Software in der Architekturvisualisierung hat die Wettbewerbsszene völlig verändert. Bestanden früher Einreichungen oft nur aus wenigen Zeichnungen, so haben heute Projekte, die nicht spektakulär präsentiert werden, kaum Chancen prämiert zu werden.

Auch wenn in einigen Fällen aufwendige Visualisierungen nicht wirklich notwendig wären, müssen sich Architekten dem zeitgemäßen graphischen Standard unterwerfen, wenn sie mit der Konkurrenz mithalten wollen.

[vgl. AmsSchö|2003|S.7]

Mittlerweile ist eine große Auswahl an 3D-Software am Markt erhältlich. In der Architekturvisualisierung spielen zurzeit besonders Autodesk 3ds Max und Maxon Cinema4D eine Rolle.

3ds Max bietet Spezialfunktionen für Architekten wie prozedurale Türen, Fenster, dwg-Import und eigene Architektur-Materialien an.

Cinema 4D verfügt über viele Schnittstellen für einen einfachen Import von Dateien aus CAAD-Software und gute Exportmöglichkeiten für die Postproduktion.

Weitere bekannte 3D-Programme sind Avid Softimage XSI, Autodesk Maya und NewTek LightWave 3D.

[vgl. Immler|2004|S.21 | Sonder|2006|S.7]

III|3 Austauschformate, Kompatibilitätsproblematik

Auch im Bereich der Architekturvisualisierung kann es notwendig sein, Daten entweder zwischen 2D- und 3D-Software oder auch zwischen unterschiedlicher 3D-Programme zu transferieren.

Für den Austausch zwischen 3D- und 2D-Anwendungen gibt es nur wenige standardisierte Dateiformate, eines davon das iges-Format. Darüber hinaus verfügen viele Programme lediglich über Austauschschnittstellen, die 10 Jahre und älter sind.

Deshalb stößt man beim Im- und Exportieren der Daten immer wieder auf Probleme.

Es gibt 3 verschiedene Möglichkeiten, Daten zwischen 2D und 3D bzw. auch zwischen verschiedenen 3D-Programmen auszutauschen, die in den folgenden Kapiteln beschrieben werden.

[vgl. KuHa|06|2005|S.56 | StreWei|1996|S.40]

III|3.1 Verwendung von alten Dateiformaten

Zum Datenaustausch können folgende Dateiformate verwendet werden. Diese sind jedoch, außer das iges-Format, nur Quasistandardformate, das heißt verschiedene Funktionen werden von den jeweiligen Softwareherstellern unterschiedlich interpretiert und es gibt auch so gut wie keine Informationen, welche Funktionen von welchem Format genau unterstützt werden.

Hier eine Liste der bekanntesten Austauschformate:

- 3ds, prj [3d Studio]
- obj [Alias]
- wrl [VRML95/97]
- hrc, dsc [Softimage]
- stl [Stereolitographie trianguliert]
- igs, iges [Initial Graphics Exchange Specification]
- dxf, dwg [AutoCAD]

[vgl. KuHa|06|2005|S.57 | StreWei|1996|S.40]

Probleme bei der Konvertierung

Das dxf-Format, das bei vielen Anwendern als universelles Austauschformat gilt, birgt seine Tücken. Es ist im Prinzip dasselbe wie das dwg-Format von AutoCAD, nur in Textform. Aus diesem Grund entwickelte sich das dxf-Format parallel zu AutoCAD weiter. Das bedeutet, dass dxf bis zur Version 12 keine Ellipsen enthalten konnte, da man mit AutoCAD zu dieser Zeit noch keine Ellipsen zeichnen konnte. Das dxf-Format ist also kein Standard und kann sich mit jeder AutoCAD Version ändern. Außerdem gibt es verschiedene „dxf-Dialekte“.

[vgl. KuHa|06|2005|S.57 | Seitz|2004 |S.145]

Elemente aus CAAD-Programmen bestehen aus einzelnen Flächen, die zu einem 3D-Objekt, welches innen hohl ist, zusammengefügt werden. Über ein Austauschformat, wie zum Beispiel obj, kann dieses in ein 3D-Programm wie Cinema4D importiert werden.

Schwierig wird der Import von Modellen aus CAD-Software von der Industrie. Diese verfügen über so genannte Volumenkerne, die Volumenkörper [Solids] verarbeiten und berechnen.

Solids sind mathematisch beschriebene Elemente, denen auch Material- und Masseneigenschaften zugewiesen werden können, mit welchen in späterer Folge z.B. Festigkeitsanalysen direkt aus dem Modell durchgeführt werden können.

Da 3D-Software nicht über Volumenkerne verfügt, muss, um Solids z.B. in Cinema4D zu importieren, ein Austauschformat gewählt werden, das Volumenkörper in Flächen zerlegt. Dazu gehören z.B. das 3ds und das stl-Format.

[vgl. Seitz|2004|S.145]

Ein weiteres Problem entsteht dadurch, dass die alten Formate oft nicht mehr mit der steigenden Komplexität der Modelle zurechtkommen. So enthalten z.B. das 3ds und das obj-Format Begrenzungen in Bezug auf die Polygon- und Objektmenge pro Ebene und Datei. Das hat zur Folge, dass Szenen ab einer gewissen Komplexität nicht mehr richtig über 3ds oder obj exportiert werden können.

[vgl. Seitz|2004 | KuHa|06|2005|S.57]

Beim Export aus einem CAD-Programm muss auch die richtige Gruppierung von CAD Objekten beachtet werden.

Üblicherweise werden bei der Arbeit mit CAD-Software z.B. alle Fenster auf eine Ebene gelegt, beim Export in eine 3D-Anwendung wie z.B. Cinema4D kommen alle Fenster als einzelnes Objekt in Cinema 4D an. Deshalb muss vor dem Datenexport geklärt werden, welche Elemente getrennt behandelt werden sollen. Wenn zum Beispiel ein Fenster animiert werden soll, muss dem Objekt in ArchiCAD [wie auch in Nemetschek Allplan] eine eigene Textur oder Stifffarbe zugewiesen werden. Beim obj-Export fragt ArchiCAD, ob Objekte nach Stifffarbe oder Textur getrennt werden sollen. Lichtquellen und Kameras werden jedoch beim Export aus z.B. ArchiCAD in Cinema 4D bei Verwendung des obj- Formats nicht mitexportiert. Wird das 3ds-Format verwendet, werden auch Lichtquellen und Kameras übernommen.

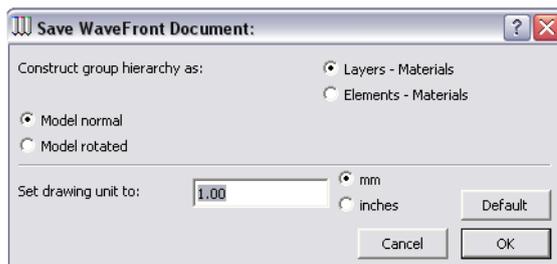


Abb. IV-3 | Export – Dialog in ArchiCAD 9

In AutoCAD funktioniert die Trennung der Objekte über die jeweils zugewiesenen *Ebenen*³. Es empfiehlt sich, alle Elemente, die später mit dem gleichen Material versehen werden sollen, auf eine Ebene zu legen.

[vgl. Seitz|2004|S.146 | Sonder|2006|S.36 | StreWei|1996|S.52]

Bei den Formaten igs, dwg und dxf sollte man Objekte, die aus demselben Material bestehen, immer vor dem Export gruppieren, da diese Dateiformate standardmäßig keine Materialinformationen enthalten und so diese Informationen für den Benutzer verloren gehen.

[vgl. KuHa|06|2005|S.58]

³ siehe Glossar

III|3.2 Verwendung eines speziellen Programms zum Datenaustausch

Mithilfe spezieller Programme können Dateien von einer Anwendung in eine andere überführt werden. So bietet z.B. Maxon, der Hersteller von Cinema 4D, Additional für Nemetschek Allplan, VectorWorks und ArchiCad an, mit welchen Dateien aus diesen Programmen problemlos in Cinema 4D importiert werden können.

Falls Daten zwischen 3D Programmen oder verschiedenen CAD Programmen und Animationssoftware ausgetauscht werden müssen, bieten sich Standalone-Konverter wie Okino Nugraf | Polytrans oder Right Hemisphere Deep Exploration an.

Da diese Programme natürlich einen zusätzlichen Kostenaufwand darstellen, greifen viele Anwender auf die Verwendung von alten Formaten zurück.

[vgl. KuHa|06|2005|S.58]

III|3.3 Eigens programmierte interne Software

Diese Variante ist offensichtlich die teuerste und wird deshalb nur in großen Produktionsfirmen genutzt. Es wird dadurch aber ein problemloser Datenaustausch zwischen verschiedenen Programmen gewährleistet.

IV. Architekturvisualisierung

Um architektonische Entwürfe auch nichtsachkundigen Personen zugänglich zu machen, wurde seit jeher nach geeigneten Visualisierungsformen gesucht.

Im einfachsten Fall kann eine schnell gezeichnete Handskizze ausreichend sein, um den Baukörper zu visualisieren. Doch für die dreidimensionale Darstellung war man früher auf analoge Maßstabsmodelle beschränkt.

Mit 3D-Programmen ist es nun möglich, fotorealistische Modelle des Bauobjekts zu erstellen und auch komplexe Lösungen und Darstellungen so zu vermitteln, dass sie für jeden verständlich sind.

Es sollte aber immer abgeschätzt werden, ob eine aufwendige 3D-Visualisierung zielführend ist oder ob auf traditionelle Darstellungsmethoden zurückgegriffen werden sollte.

[vgl. Mach|2000|S.16]

IV|1 Visualisierungsformen

Einteilung der Visualisierungsformen anhand der dimensionalen Ausdehnung (2D, 3D, 4D) und der Darstellung und der zugrunde liegenden Hilfsmittel (analog, digital).

2D	3D	4D	4Di
Skizze Plan Perspektive Axonometrie Bild Collage Plan Rendering Photomontage	Architekturmodell 1:1 Modell 3D-Plot	Animation (Film)	1:1 Modell QTVR Echtzeitgrafik Virtual Reality CAVE

[vgl. Kranzer|2005|S.25]

Im Folgenden werden die Visualisierungsformen näher beschrieben, bei welchen 3D-Modelle angewandt werden können oder müssen.

IV|1.1 2D [zweidimensional]

Rendering

Ein Rendering ist das aus einer 3D-Szene, unter Berücksichtigung von Oberflächenbeschaffenheit und Beleuchtung, berechnete Bild.

[vgl. Kranzer|2005|S.26]

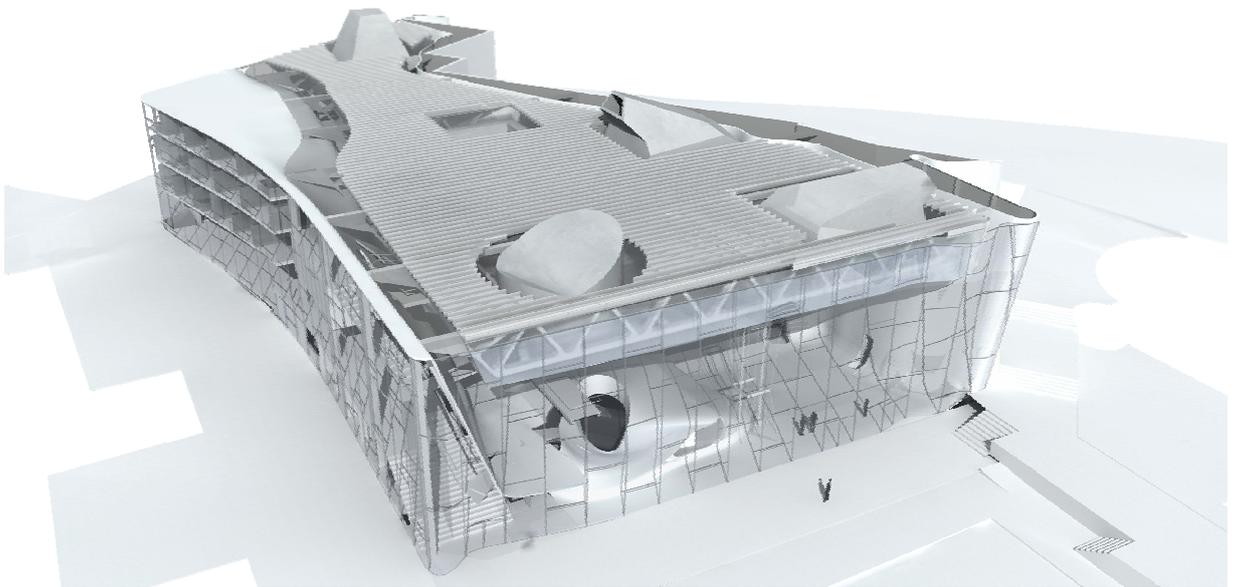


Abb. V-1 | Rendering, Museum Aachen, Architekt Wolfgang Tschapeller

Fotomontage

Eine Fotomontage hat den Vorteil, dass die Umgebung des Bauobjekts nicht im 3D-Programm selbst realisiert werden muss, sondern das Gebäude in eine Fotografie des Umfelds eingearbeitet wird. Dabei muss beim Erstellen des Baukörpers Gegend, Stimmung und Beleuchtung des Fotos beachtet werden.

[vgl. Schneider|2003|S.60]



Abb. V-2 | Originalfoto



Abb. V-3 | Fotomontage mit gerendertem Haus

IV|1.2 3D [dreidimensional]

[analoges] Architekturmodell

Die Herstellung von analogen Architekturmodellen hat sich ständig weiterentwickelt, war es doch auch vor Erfindung und Einsatz von 3D-Programmen die einzige Möglichkeit dreidimensionale Ansichten eines Bauobjekts darzustellen.

[vgl. *StreWei*|1996|S.28, S.33]

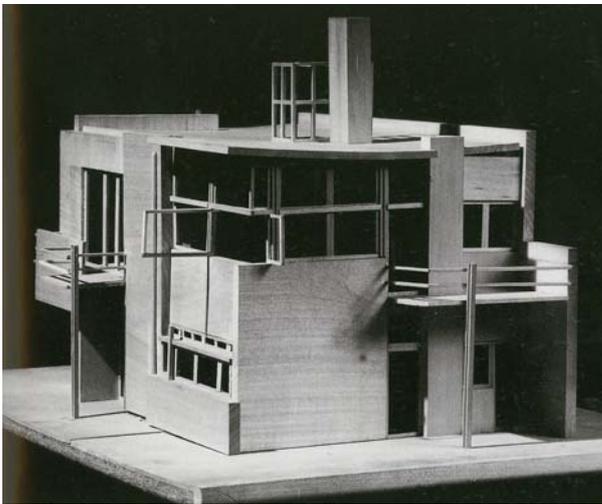


Abb. V-4 | analoges Architekturmodell von Hans Stark, Schröder Haus in Utrecht [1924] von Gerrit Rietveld

3D-Plot

Computergestützter Modellbau hat sich zunächst in der industriellen Fertigung entwickelt, da es in diesem Bereich schon sehr früh die Anforderung gab, den Modell- bzw. Prototypenbau zu beschleunigen. Daher stammt auch die Bezeichnung *Rapid Prototyping*. Da es sich aber beim Architekturmodell nicht um einen Prototypen im herkömmlichen Sinn handelt, sollte der Begriff in diesem Bereich besser nicht verwendet werden.

Für den Architekturmodellbau wurde es erst Ende der 80er interessant, da zuvor nur Bauteile mit relativ einfachen Geometrien mittels subtraktiver Verfahren erstellt werden konnten. Doch durch die Entwicklung von so genannten additiven Verfahren, die Freiformkörpermodellierung möglich machten.

[vgl. StreWej|1996|S.45-46]

IV|1.3 4D [vierdimensional]

Animation

Anstatt eines einzelnen Renderings kann aus einer 3D-Szene auch eine Animation [also eine Folge von Einzelrenderings] generiert werden. In der Architektur wird diese Technik angewandt um beispielsweise einen Rundgang durch ein Gebäude oder das Bauobjekt von außen zu zeigen.

Darüber hinaus können auch Fahrzeuge, Personen und ähnliches animiert werden um dem Betrachter die gewünschte Stimmung zu vermitteln. Auch der Einsatz von Musik ist möglich.

IV|1.4 4Di [vierdimensional, interaktiv]

QuickTime VR [Virtual Reality]

QuickTime VR von Apple ist ein Format zur Darstellung von Panoramabildern, wobei der User die Möglichkeit hat durch eine vorgerenderte, oder auch fotografierte, Szene zu navigieren.

Außer Panoramabilder können auch Objekte von allen Seiten interaktiv betrachtet werden.

Schöne Beispiele für eine QTVR-Anwendung in der Architekturvisualisierung finden sich unter www.swissinteractive.ch. Hier werden zwei Wohnungen mit Hilfe dieser Technik präsentiert.



Abb. V-5 | Screenshot, Architekturvisualisierung mit QuickTime VR

Echtzeitgrafik

Der Einsatz Echtzeitgrafik ist noch kein Standardverfahren zur Architekturvisualisierung. Bis jetzt wird sie hauptsächlich in Computerspielen angewandt. Die Grafikdarstellung am Computer entwickelte sich sehr schnell weiter, so ist es heute schon möglich, wie im Screenshot aus dem Computerspiel Unreal Tournament 2004 in Abb. V-6, realistische Bilder in Echtzeit zu zeigen.

Für die Visualisierung von Bauobjekten ist Echtzeitgrafik sehr gut einsetzbar, da dem Betrachter die Möglichkeit gegeben sich interaktiv durch ein Gebäude zu bewegen.

[vgl. Kranzer|2005|S.25]



Abb. V-6 | Screenshot aus Unreal Tournament 2004

IV|2 Arbeitsablauf

Im Verlauf der Planung eines Bauobjekts gibt es mehrere Phasen, in welchen digitale 3D-Modelle erstellt werden können. Wann und ob sie eingesetzt werden, hängt sehr stark vom Projekt und der Arbeitsweise des Architekten ab.

IV|2.1 Ideenfindung

Am Anfang des Arbeitsablaufs steht die Ideenfindung. Im Zuge dessen wird oft ein so genanntes Ideen- oder Massenmodell gestaltet. Dabei handelt es sich um ein ungenaues, maßstabsloses, meist analoges Miniaturmodell, das die Grundidee darstellen und den Architekten bei der weiteren Planung unterstützen soll.

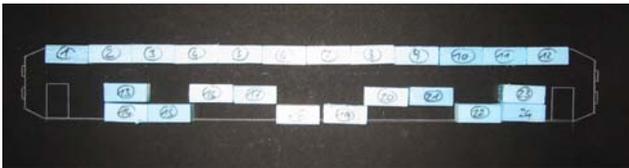


Abb. V-7 | analoges Ideenmodell für einen Zug mit beweglichen Modulen von oben



Abb. V-8 | fertig modellierter und gerenderter Zug, Innenansicht

Diese groben Miniaturen gehen in weiterer Folge fließend in ein Arbeitsmodell über. Diese sind detaillierter, schon maßstabsgerecht und bilden die Grundlage zur Abstimmung zwischen den am Entwurf beteiligten Personen.

Manchmal werden Ideen- und Arbeitsmodelle auch schon mit 3D-Programmen wie z.B. *SketchUp*⁴ modelliert. Viele Architekten bevorzugen aber nach wie vor das analoge Modell, das man angreifen und schnell verändern kann. Durch Umstellen, Umgruppieren und Variieren wird mit der Zeit die Struktur des Baukörpers entwickelt.

Auch Handskizzen spielen in dieser Entwurfsphase eine große Rolle, sie dienen zum Festhalten von Ideen und als Grundlage für weitere Diskussionen.

[vgl. *StreWei*|1996|S.28]

Oft müssen noch, bevor das Bauobjekt richtig geplant werden kann, finanzielle Mittel aufgetrieben werden, dabei kann ein gutes 3D-Modell sehr hilfreich sein. In diesem Fall muss eine grobe Idee oder eine Vision, noch ohne genaue Pläne, möglichst spektakulär umgesetzt werden.



Abb. V-9 | 3D-Modell, entstanden ohne genaue Pläne,
Architekturbüro Eichinger oder Knechtl

⁴ siehe Glossar

IV|2.2 Pläne | 3D Modell

Pläne werden nur noch selten mit Hand gezeichnet, fast immer kommen dafür CAD- oder CAAD-Programme zur Anwendung.

Oft wird in dieser Phase schon mit der Erstellung von digitalen 3D-Modellen begonnen, da die Pläne schon über ausreichende Genauigkeit verfügen, um darauf aufzubauen.

Bei der Erstellung von digitalen 3D-Modellen gibt es verschiedene Vorgehensweisen.

Wurde zum Anfertigen des Plans z.B. ein CAD-Programm wie AutoCAD verwendet und der Plan dort schon in der *3D-Ansicht des Programms*⁵ erstellt, besteht die Möglichkeit dieses direkt in ein 3D-Programm wie z.B. 3ds Max zu importieren und dort dann weiter zu bearbeiten. Der Datenaustausch zwischen AutoCAD und 3ds Max funktioniert sehr gut, da beide Anwendungen aus dem Hause Autodesk stammen.

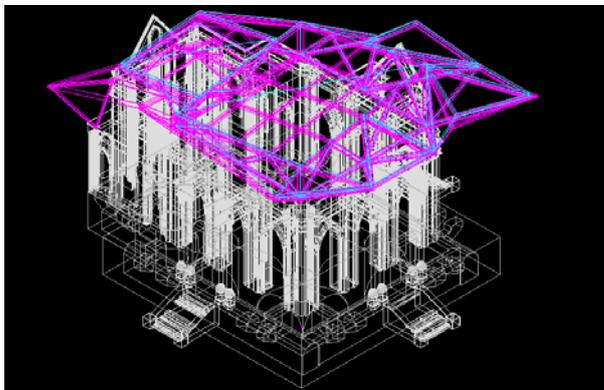


Abb. V-10 | AutoCAD Plan, 3D Ansicht

Wird ein CAAD-Programm verwendet, wird das Gebäude fast immer schon als dreidimensionales Modell gezeichnet. Aus diesem werden in weitere Folge zweidimensionale Schnitte und Ansichten berechnet.

⁵ in der neuen Version von AutoCAD [AutoCAD 2007], wurden spezielle Features für 3D-Modellierung implementiert, welche dem Benutzer die Arbeit an 3D-Entwürfen mit AutoCAD wesentlich erleichtern werden. [<http://www.autocad-magazin.de/index.php3?page=03-06/fokus.html>]

Oft ist aber die dreidimensionale Darstellung der CAAD-Programme in punkto Fotorealismus nicht ausreichend, da diese noch nicht über die Renderleistung verfügen, die notwendig ist, um Effekte wie Transparenz, Lichtbrechung, *diffuse Reflexion*⁶ etc. zu berechnen.

In diesem Fall kann das Modell wiederum in eine 3D-Anwendung zur weiteren Bearbeitung importiert werden.

Gelegentlich wird das 3D-Modell auch nur auf Basis von 2D-Ansichten, wie Grundriss oder Fassade des Bauobjekts, erstellt. Dazu können diese als Vorlage in den Hintergrund des jeweiligen Fensters im 3D-Programm gelegt und so das dreidimensionale Gebäudemodell erarbeitet werden. Schnitte können auch aus diesem Modell gerendert werden.

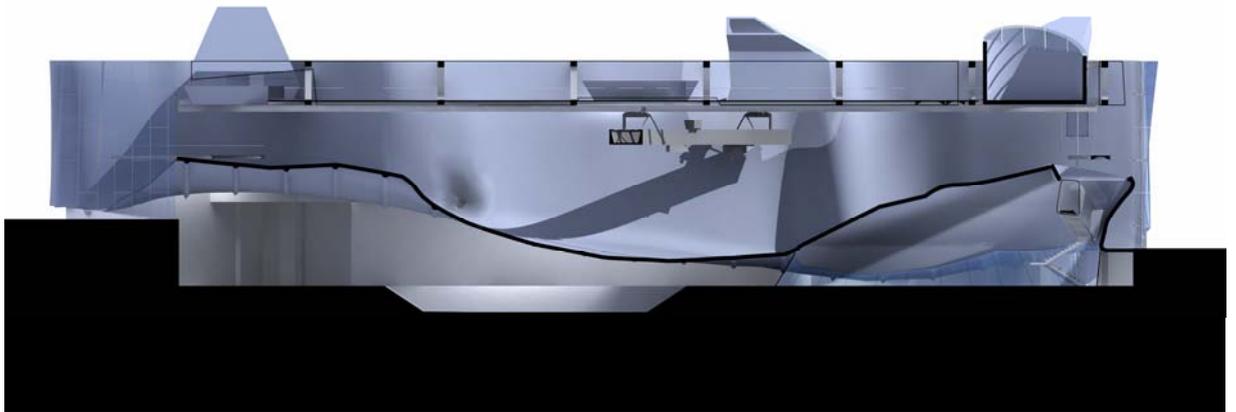


Abb. V-11 | gerenderter Schnitt, Architekt Wolfgang Tschapeller

Die Pläne werden parallel dazu immer weiter ausgearbeitet, oft sind sie aber für das digitale 3D-Modell aufgrund ihrer Genauigkeit nicht mehr relevant.

In vielen Fällen, wie z.B. für Studenten der TU Wien, ist ein analoges Miniaturmodell immer noch Teil der Projekt-Abgabe. Im universitären Bereich werden diese meistens per Hand angefertigt.

Daneben gibt aber auch die Möglichkeit, diese durch ein *3D-Plotter Verfahren*⁷ zu generieren. Dazu ist wieder ein spezielles, digitales 3D-Modell notwendig.

⁶ siehe Glossar

⁷ siehe IV|1 Visualisierungsformen

Nach der Fertigstellung des eigentlichen digitalen Modells muss dieses noch richtig ausgeleuchtet werden. Darüber hinaus besteht meistens die Anforderung auch die Umgebung des Bauobjekts zu visualisieren.

Handelt es sich bei der Visualisierungsform um eine Animation, müssen der Szene Kameras hinzugefügt und diese auch animiert werden.

IV|2.3 Nachbearbeitung

Nachdem das 3D-Modell fertig gestellt und gerendert ist, ist oft noch die Nachbearbeitung in einem Bildbearbeitungsprogramm oder, bei einer Animation, in einem Videobearbeitungsprogramm, notwendig.

Objekte wie z.B. Pflanzen, Personen oder ganze Hintergrundbilder können hinzugefügt und Fehler korrigiert werden.

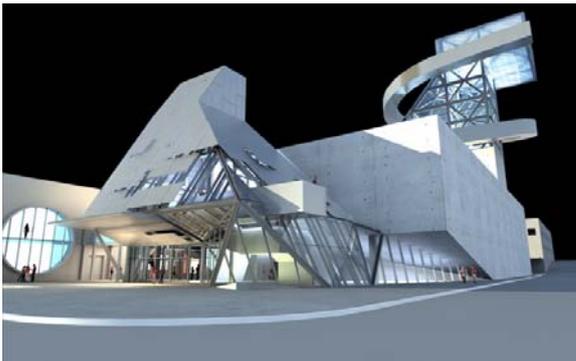


Abb. V-12 | Rendering ohne Nachbearbeitung, Akron Art Museum, Architekt Coop Himmelblau



Abb. V-13 | Rendering mit Nachbearbeitung, Akron Art Museum, Architekt Coop Himmelblau



Abb. V-14 | Rendering ohne Nachbearbeitung BMW Welt, Architekt Coop Himmelblau



Abb. V-15 | Rendering mit Nachbearbeitung, BMW Welt, Architekt Coop Himmelblau

Wurde das Modell, was häufig der Fall ist, in mehreren *Passes*⁸ aus der 3D-Software gerendert [Multipass-Rendern], müssen diese wieder zusammengefügt werden.

⁸ siehe Glossar

IV|3 Materialien | Texturen

Die Wahl der richtigen Materialien bzw. Texturen ist ein wichtiger Bestandteil der Architekturvisualisierung. Einerseits soll der Baukörper möglichst realistisch dargestellt werden, andererseits können auch kurze Rechenzeiten gefordert sein. Um diese beiden Faktoren gut abschätzen zu können ist es notwendig, über die verschiedenen Techniken Objekte mit Materialien zu versehen bescheid zu wissen.

In folgendem Beispielrendering [Abb. V-16] wurden beinahe alle in diesem Kapitel vorgestellten Techniken verwendet und im jeweiligen Unterkapitel genauer erklärt.

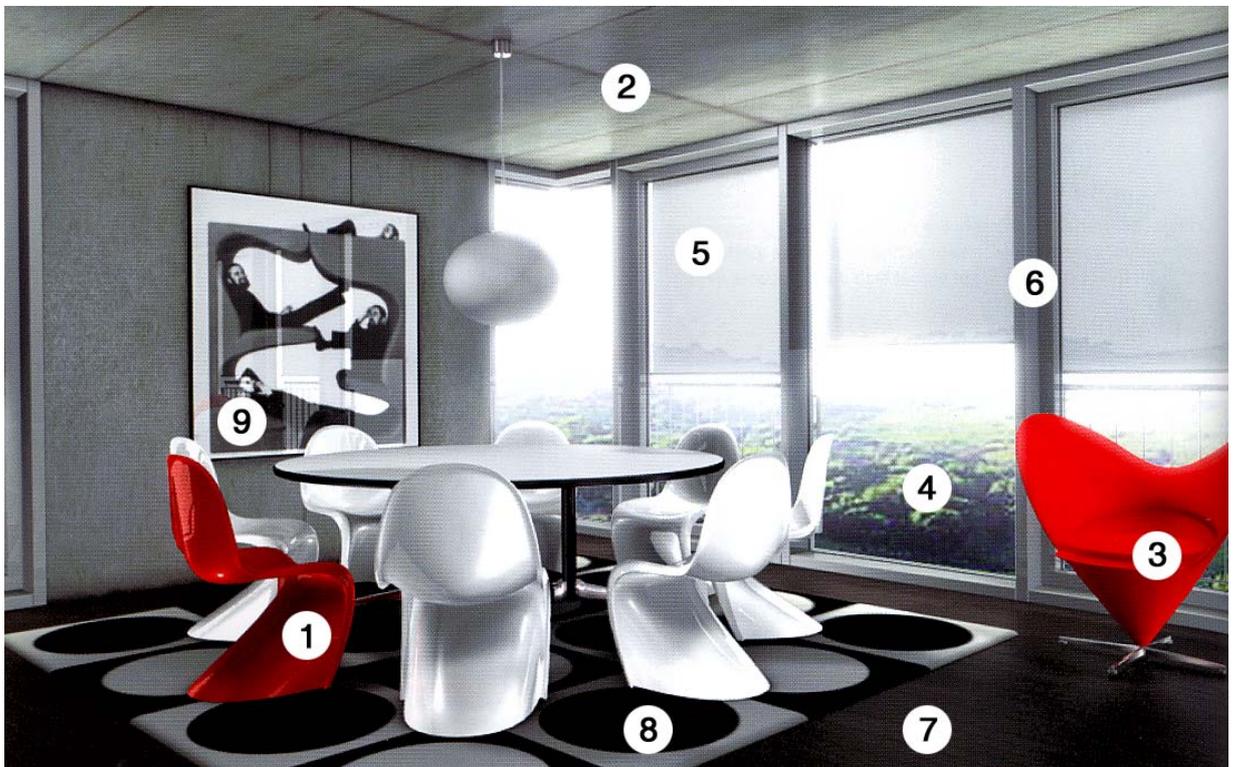


Abb. V-16 | Beispielrendering, Innenraumvisualisierung, Cinema 4D

- | | |
|--|---|
| 1. Specular Shader, Spiegelreflexion | 6. Diffuse Reflexion |
| 2., 7. Diffuse Reflexion, Image Map, Color Map, Bump Map, planare Projektion | 8. Image Map, planare Projektion |
| 3. Materialfarbe, Spiegelreflexion | 9. Image Map, Color Map, Spiegelreflexion, planare Projektion |
| 4., 5. Transparenz | |

IV|3.1 Surface Shading Techniques | Schattierungsmethoden

Das Erscheinungsbild einer gerenderten 3D-Szene wird sehr stark vom Shading-Prozess beeinflusst. Dieser generiert Oberflächen auf den modellierten Objekten. Shading ist der Zeitpunkt im Rendering-Prozess, an dem sichtbaren Oberflächen ein Shading-Wert zugewiesen wird. Dieser wird über die *Normalvektoren*⁹ [engl.: surface normals] und die Lichtquellen, die auf der Oberfläche auftreffen, berechnet.

Jede Schattierungsmethode hat ihre eigene Art Licht und Oberfläche zu kalkulieren. Die Berechnungsmethoden variieren sehr stark zwischen unterschiedlicher 3D-Software.

In diesem Kapitel werden die wichtigsten Shading-Basismethoden beschrieben, die in manchen Fällen auch kombiniert angewandt werden. Die gleichzeitige Verwendung von mehreren Shading-Methoden wird auch als *Hybrid-Shading* bezeichnet.

[vgl. Kerlow|2004|S.228-229]

Faceted Shading | Polygonal Shading

Basierend auf dem Winkel der Normalvektoren im Verhältnis zur Lichtquelle wird bei dieser Technik jedem Polygon eines Objekts ein Shading-Wert zugewiesen. Die meisten 3D-Programme messen dazu nur den Lichtanteil in der Mitte des entsprechenden Polygons, andere verwenden einen Mittelwert aus den Lichtanteilen, die an den Ecken des Polygons anfallen.

Mit Faceted Shading ist es nicht möglich, komplexere Oberflächeneigenschaften wie Texturen und Transparenz darzustellen. Ein Beispiel für Faceted Shading ist Lambert Shading.

[vgl. Kerlow|2004|S.229]

⁹ siehe Glossar

Smooth Shading

Smooth Shading berechnet den Durchschnitt des Shading-Werts von benachbarten Polygonen und schafft so eine Überblendung zwischen den einzelnen Polygonen eines Objekts. Mit dieser Technik ist es möglich, auch auf Körpern mit wenig Detail eine weiche Oberfläche zu generieren. Da Smooth-Shading Techniken keine Highlights berechnen können, sind sie nur zur Simulation von matten Oberflächen geeignet. Ein bekannter Smooth-Shader ist der Gourad Shader.

[vgl. Kerlow|2004|S.229-230]

Specular Shading

Mittels Specular Shading-Techniken lassen sich Oberflächen mit Highlights, also reflektierende Materialien berechnen. Wie beim Smooth-Shading werden auch hier weiche Übergänge zwischen angrenzenden Polygonen kalkuliert, allerdings ist die verwendete Technik komplexer. Der Shading Wert wird nicht wie bei Smooth-Shading für jedes Polygon, sondern nur für jeden einzelnen Punkt einer Oberfläche ermittelt.

Bei Specular Shading werden Streulicht [diffuses Licht], Umgebungs- und Glanzlicht miteinbezogen.

Variationen von Specular Shading sind Blinn und Cook Shading.

[vgl. Kerlow|2004|S.230]

Ein in Cinema 4D verfügbarer Specular Shader ist der *Danel Shader* von BhodiNut, verwendet in Abb. V-16 [1] als Material für die glänzenden Stühle. Das Material ist hochglänzend und wirkt beinahe wie Lack.

[vgl. Seitz|2006|S.147]

IV|3.2 Material | Surface Shaders

Ein Surface Shader ist im Prinzip eine Sammlung von Materialeigenschaften. Folgende Informationen können in ihm enthalten sein: Der Name des Shaders, die Shading-Technik und die veränderbaren Parameter, die Materialeigenschaften und die dazugehörigen Parameter und die Render-Methode mit ihren Parametern. Durch diese wird das Aussehen eines dreidimensionalen Objekts bestimmt.

Das Shader-Konzept wird in verschiedener Software unterschiedlich interpretiert. So kann es sein, dass in manchen Programmen alle Shader-Informationen auf verschiedene Menüpunkte verteilt sind.

Wenn Shader nur Informationen über das Oberflächenmaterial, nicht aber über das verwendete Render-Verfahren enthalten, werden diese auch als Materialbibliotheken bezeichnet.

Selbst ein Material zu kreieren kann viel Detailarbeit erfordern, aber wenn er gut gemacht ist, trägt er wesentlich zur Qualität des Endergebnisses bei.

Bei der Zuweisung des Shaders muss auch die Beleuchtung miteinbezogen werden, da diese das Erscheinungsbild des Materials wesentlich prägt. Ein und derselbe Shader kann unter verschiedenen Lichtverhältnissen sehr unterschiedlichen Ergebnissen führen.

[vgl. Kerlow|2004|S.230-231]

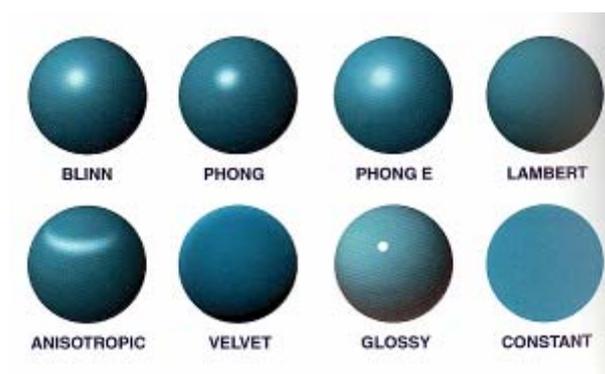


Abb. V-17 | Beispiele für Shader

IV|3.3 Reflexion

Als Reflexion wird die Fähigkeit eines Materials bezeichnet, einfallende Lichtstrahlen zurückzuwerfen. Verspiegelte Oberflächen stellen eine besondere Schwierigkeit bei der Erstellung von 3D-Szenen dar.

Die drei Grundreflexionen sind Umgebungsreflexion, diffuse Reflexion und Glanzreflexion. Diese werden im Folgenden noch genauer beschrieben; mit ihnen können aber nur Reflexionen, die sich auf das Reflektieren von Licht beziehen, berechnet werden.

Kombinationen aus diesen Grundreflexionen können verwendet werden, um verschiedene Materialien zu simulieren. Metall kann beispielsweise durch eine Mischung aus Umgebungs- und Glanzreflexion dargestellt werden. Für Plastik wird üblicherweise eine Kombination aus allen drei Grundformen verwendet.

Soll aber die Umgebung des Objekts gespiegelt werden, so muss die Raytracing Render-Methode oder eine *Reflection Map*¹⁰ bzw. *Environment Maps*¹¹ angewendet werden.

[vgl. Mach|2004|S.120 | Kerlow|2004|S.242-243 | Immler|2000|S.287]

Umgebungsreflexion | Ambient Reflection

Diese Art der Reflexionssimulation bezieht sich nur auf das Reflektieren des Umgebungslichts. Eine spezielle Eigenschaft ist, dass die Stärke der Reflexion unabhängig von der Entfernung der Lichtquelle und des Lichteinfallswinkels ist. Deshalb haben alle Polygone eines Objekts, die nur mit Ambient Reflection versehen wurden dieselbe Reflexionsstärke und die Körper wirken dadurch völlig flach.

[vgl. Kerlow|2004|S.243-244]

¹⁰ siehe Reflection Maps

¹¹ siehe Environment Maps

Diffuse Reflexion

Bei diffuser Reflexion werden Lichter, die sich näher am Objekt befinden, stärker reflektiert als die Lichtquellen, die weiter davon entfernt sind. Außerdem wird auch der Lichteinfallswinkel berücksichtigt, so dass die Stellen am 3D-Körper, auf die das Licht rechtwinklig einfällt, stärker reflektieren. Die Bereiche, die nicht von Lichtstrahlen erreicht werden, reflektieren nur schwach oder überhaupt nicht.

[vgl. Kerlow|2004|S.244]

In Abb. V-16 reflektieren der Beton [2], die Fensterrahmen [6] und der Parkettboden [7] das Licht diffus.

[vgl. Seitz|2006|S.147]

Spiegelreflexion | Specular Reflection

Objekte, die mit Specular Reflection versehen wurden, erscheinen sehr glänzend, da sie das Licht wie ein Spiegel reflektieren.

Die Intensität des zurückgeworfenen Lichts hängt hauptsächlich von der Relation des Winkels des reflektierten Lichts und dem Winkel der Kamera zum Objekt ab. Wenn die beiden Winkel zusammenfallen ist die Intensität des zurückgestrahlten Lichts maximal. Je weiter die Größe der Winkel differiert, desto schwächer wird das Licht reflektiert.

[vgl. Kerlow|2004|S.244]

Eine Spiegelreflexion wurde im Beispielrendering [Abb. V-16] bei dem Bild an der Wand [9] und den glänzenden Sesseln [1] eingesetzt. Auch der Fuß des gepolsterten Sessels [3] entstand durch den Einsatz von Spiegelreflexion.

[vgl. Seitz|2006|S.147]

IV|3.4 Selbstillumination

Die Darstellung von selbst leuchtenden Materialien wie z.B. Lava kann nicht mit Hilfe von Lichtquellen realisiert werden, da Licht in 3D-Software nur dann sichtbar ist, wenn es auf Flächen fällt und diese beleuchtet.

Ein Objekt, das mit einem selbst leuchtenden Material versehen wurde, ist deshalb aber noch keine Lichtquelle und beleuchtet Objekte in seiner Umgebung nicht. Ein Scheinwerfer beispielsweise lässt sich nur durch eine Kombination aus Lichtquelle und selbst leuchtendem Material [in diesem Fall das Glas des Scheinwerfers] simulieren. Zu beachten ist, dass selbst leuchtende Objekte in der Realität keine Schatten werfen, in 3D-Programmen muss der Schatten eventuell deaktiviert werden.

[vgl. Immler|2000|S.72-73 | Mach|2004|S.116 | Kerlow|2004|S.247-248]

IV|3.5 Materialfarbe

Die Farbe ist die offensichtlichste Eigenschaft eines Materials und sie beeinflusst neben der Beleuchtung wesentlich die Gesamtwirkung der Szene. Beim Zuweisen von Materialfarben muss aber beachtet werden, dass diese auch von anderen Faktoren, wie Einfallswinkel und Farbe des auf die Oberfläche auftreffenden Lichts verändert werden. Beim Verwenden von Render-Techniken wie *Raytracing*¹² oder *Radiosity*¹³ wirkt auch die Farbe von benachbarten Objekten auf die Materialfarbe ein.

Die eigentliche Farbe eines Objekts wird als Streufarbe bezeichnet. Die Schattenfarbe, auch Umgebungsfarbe, sollte bei einer Tageslichtszene die direkte Komplementärfarbe der Streufarbe sein. Bei Szenen in geschlossenen Räumen kann die Umgebungsfarbe etwas dunkler sein, da die allgemeine Beleuchtung mehr von den Lichtquellen im Innenraum abhängt.

Die Glanzfarbe wird so eingestellt, dass sie der Farbe der Lichtquelle entspricht, die das Objekt direkt beleuchtet.

¹² siehe Raytracing

¹³ siehe Radiosity

Zum Bestimmen der Farben im 3D-Programm werden *additive Farbmodelle*¹⁴ wie RGB oder HLS verwendet.

Eine weitere Möglichkeit, die Farbe eines Objekts zu bestimmen, ist der Einsatz von *Color Maps*¹⁵.

[vgl. Mach|2004|S.116 | Kerlow|2004|S.248]

Der gepolsterte Sessel in Abb. V-16 [3] wurde nur mit einer Materialfarbe versehen.

[vgl. Seitz|2006|S.147]

IV|3.6 Transparenz

Transparenz wird verwendet um durchsichtige Materialien wie Glas oder Wasser zu simulieren. Der Realität entspricht am meisten die Berechnung der Durchsichtigkeit mit der Raytracing Render-Methode.

Ein anderer Ansatz zur Darstellung von transparenten Objekten ist die Verwendung von *Transparency Maps*¹⁶.

[vgl. Kerlow|2004|S.255]

Im Beispielrendering [Abb. V-16] wurde das Material für das Fensterglas [5] mit einer Transparenz von 90% und die Rollos [4] mit einem Transparenzfaktor von 10% versehen.

[vgl. Seitz|2006|S.147]

¹⁴ siehe Glossar

¹⁵ siehe Color Maps

¹⁶ siehe Transparency Maps

IV|3.7 Image Mapping | Texturierung

Das Prinzip von Image Mapping ist die Projektion eines zweidimensionalen Bildes auf einen dreidimensionalen Körper. Damit lässt sich erstens jede erdenkliche Oberfläche realistisch darstellen, zweitens können mittels Texturierung auch Oberflächeneigenschaften wie Struktur, Farbeffekte, Transparenz und Reflexion simuliert werden.

Dasselbe Bild kann auf verschiedene Arten auf eine Oberfläche gemapped werden, um deren Eigenschaften zu berechnen. Bei Bedarf besteht auch die Möglichkeit jedes einzelne Attribut mit einem eigenen Bild zu versehen.

Image Maps beeinflussen Oberflächenbeschaffenheiten indem die Helligkeit oder Farbe eines Pixels in der Map mit einer bestimmten Eigenschaft des entsprechenden Pixels im Objekt verbunden wird.

So bestimmt beispielsweise die Helligkeit eines Pixels in der Image-Map die Transparenz des dazugehörigen Pixels des Objekts.

[vgl. Kerlow|2004|S.232-233]

Erstellen einer Image Map

Es gibt verschiedene Arten zu einer Image Map zu gelangen. Verwendet werden digitale bzw. eingescannte Fotos oder andere eingescannte Oberflächen. Außerdem können Image Maps auch selbst mit Hilfe eines Zeichenprogramms erstellt werden, Fotos müssen manchmal auch nachbearbeitet werden. In Bildbearbeitungsprogrammen können Sättigung, Farbbalance, Kontrast etc. einer Fotografie verändert werden. Soll die Image Map gekachelt werden, muss sie vor der Verwendung „kachelfähig“ gemacht werden, in Adobe Photoshop gibt es dafür eine spezielle Funktion.

Außerdem gibt es auch *Materialbibliotheken*¹⁷, die fertige Image Maps anbieten. Manche Materialien wurden allerdings schon sehr oft eingesetzt, so dass sie schon einschlägig bekannt ist aus welcher Materialbibliothek die verwendete Image Map oder Textur stammt.

¹⁷ Eine bekannte Website auf der Materialien verkauft werden ist <http://www.doschdesign.com/>.

Projektionsmethoden

Es gibt verschiedene Arten ein zweidimensionales Bild auf eine 3D-Struktur zu projizieren.

Bei der *planaren Projektion* [engl.: flat projection method] wird ein Bild von einer Ebene aus, ähnlich wie bei einem Film- oder Diaprojektor, auf ein Objekt projiziert. Diese Methode ist ideal um Image Maps auf flache Oberflächen zu mappen.

Planare Projektion wird unter anderem eingesetzt um Hintergründe in einer 3D-Szene zu simulieren.

Eine andere Projektionsart ist die *kubische Projektion* [engl.: cubical projection method]. Sie entspricht der planaren Projektion, nur wird die Textur sechs Mal, also auf jeder Seite des Würfels, wiederholt. Besonders gut funktioniert diese Projektions-Methode natürlich mit Würfeln, solange eine der Flächen parallel zur Projektionsebene ist. Um unerwartete Effekte zu erreichen, kann sie aber auch auf andere Objekte angewandt werden.

Die *zylindrische Projektion* [engl.: cylindrical projection method] funktioniert, indem eine Map quasi um das Objekt gewickelt wird bis die beiden Enden wieder zusammentreffen. Der User kann jedoch auch einen Spalt zwischen den beiden Enden mit dem Parameter „angle of mapping“ bestimmen.

Eine weitere Art Image Maps zu projizieren ist die *sphärische Projektion* [engl.: spherical projection method]. Dabei wird eine rechteckige Textur auf ein rundes Objekt wie z.B. einen Globus gemappt. Die Map wird um den Körper gewickelt bis die Enden zusammentreffen, die obere und untere Kante der Textur laufen in Polen zusammen.

Bei der Schrumpfwicklung [engl.: wrapping projection method] wird die Image Map über den Körper gespannt bis alle Ecken in einem Punkt zusammenlaufen. Angewandt wird diese Methode um Landschaften und komplexe Objekte mit einer Image Map zu versehen.

[vgl. Ferschin|2001|S.27-28 | Immler|2000|S.99 | Kerlow|2004|S.237-238]

In Abb. V-16 wurde für den Beton [2], den Parkettboden [7], den Teppich [8] und dem Bild [9] die planare Projektionsmethode eingesetzt.

[vgl. Seitz|2006|S.146]

Color Maps

Mit Color Maps wird die Farbe des dreidimensionalen Objekts bestimmt. Ein einfaches Beispiel für die Anwendung einer Color Map, auch Picture Map, ist das Generieren einer Milchpackung. Dabei werden auf einen Quader Fotos der Packung auf die entsprechenden Flächen des Quaders projiziert. Auch eine Ziegelwand wie in Abb. V-18 kann durch Verwendung einer Color Map relativ einfach simuliert werden. Dazu wird nur ein Ausschnitt der Ziegelmauer benötigt, der dann mit Hilfe von *Kachelung*¹⁸ einfach so lange wiederholt wird, bis die ganze Wand mit der Textur bedeckt ist.

Darüber hinaus ist es mit dieser Technik auch möglich, Fotos einer Person auf einen modellierten Körper aufzubringen. Die Color-Map ist ein Beispiel für eine *Visual Texture*¹⁹.

[vgl. Kerlow|2004|S.248]

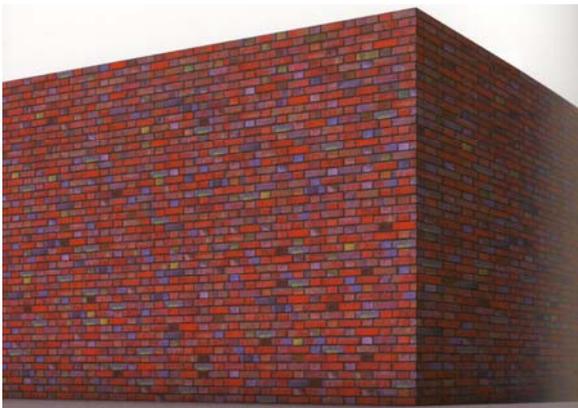


Abb. V-18 | Ziegelmauer, erstellt mit Hilfe von Kachelung

In Abb. V-16 wurden der Parkettboden, die Betonwände und die Betondecke [2], der Teppich [8] und das Bild an der Wand [9] mit Hilfe von Color Maps visualisiert.

[vgl. Seitz|2006]

¹⁸ siehe Glossar

¹⁹ siehe Glossar

Transparency Maps

Beim Transparency Mapping bestimmen die Helligkeitswerte der Image Map die Transparenzen auf dem Objekt, auf das sie projiziert wurde. Je nach Helligkeit des Pixels auf der Textur erscheint das entsprechende Pixel auf der Oberfläche transparent, opak oder transluzent. Bei einigen Programmen bedeutet ein schwarzes Pixel völlige Transparenz, bei anderen ein weißes Pixel. Grau ist für alle Programme ein Indikator für Transluszenz. Üblicherweise sind auch Transparency Maps Graustufenbilder, da beinahe alle Programme nur die Helligkeitswerte der Textur berücksichtigen.

[vgl. Kerlow|2004|S.256]

Die Lichtdurchlässigkeit der Glasbausteine in Abb V-19 wurde durch eine Transparency Map realisiert.

[vgl. [www|trans](http://www.trans)]



Abb. V-19 | Lichtdurchlässigkeit realisiert durch Transparency Map

Reflection Maps

Eine Reflection Map wird eingesetzt, um den gesamten Körper oder Teile davon reflektierend zu machen. Die Helligkeitswerte einer Reflection Map werden verwendet, um zu bestimmen welche Bereiche des Objekts reflektieren und welche nicht.

Als Reflection Maps werden wie bei Transparency Maps Bilder in Graustufen verwendet, da nur die Helligkeitsstufen die Stärke der Reflexion bestimmen und Farbinformation sogar hinderlich sein könnte.

[vgl. Kerlow|2004|S.245

Die Reflexion am Parkettboden in Abb. V-20 entstand durch eine Reflection Map. Um die Spiegelung realistischer erscheinen zu lassen wurde der Boden zusätzlich mit einer Bump Map versehen.

[vgl. [www|reflection](#)]



Abb. V-20 | Reflexion mit Reflection Map und Bump Map

Environment Maps

Environment Maps sind eine spezielle Form von Reflection Maps. Sie reflektieren nicht nur die Objekte, die den texturierten Körper umgeben, sondern auch die Umgebung reflektierender Objekte. Darüber hinaus werden Environment Maps von allen spiegelnden Objekten in der Szene und nicht nur von dem gemappten Körper reflektiert. Der gleiche Effekt könnte auch mit Hilfe von Raytracing berechnet werden, jedoch ist die Verwendung einer Environment Map die effektivere Lösung.

Im Prinzip wird bei dieser Technik ein zweidimensionales Bild der dreidimensionalen Umgebung des Objekts erzeugt und dieses dann auf alle reflektierenden Oberflächen projiziert.

Zur Erzeugung von Environment Maps stehen verschiedene Techniken zur Verfügung. *Spherical Environment Mapping*, eine Variation von kugelförmiger Projektion und *Cubical Environment Mapping*, eine Abwandlung der kubischen Projektion. Bei diesen beiden Verfahren wird zuerst ein Bild der Umgebung in eine Kugel bzw. einen Würfel projiziert, in welchen sich die spiegelnden Objekte befinden. In weiterer Folge wird dieses Umgebungsbild auf die reflektierenden Körper gemappt.

Darüber hinaus existieren noch Verfahren, die Environment Maps durch prozedurale Berechnung erstellen.

[vgl. Kerlow|2004|S.245-246]

In der Architekturvisualisierung werden Environment Maps häufig zur Beleuchtung von Modellen mittels *Image Based Lightning*²⁰, wie in Abb. V-21, angewandt.



Abb. V-21 | Environment Map

²⁰ siehe IV|4.4 Image Based Lightning [IBL]

Bump Maps | Relief-Maps

Bei Bump Maps werden die Helligkeitswerte der Image Map verwendet, um den Eindruck von Unebenheiten auf der Oberfläche zu erzeugen. Dabei wird die Orientierung der Normalvektoren während des Renderprozesses verändert. Dadurch wird das Licht wie von einer rauen Oberfläche reflektiert. Da bei diesem Verfahren aber die Oberflächenbeschaffenheit des Objekts nicht wirklich verändert wird – auch der Schatten des Körpers bleibt gleich, kann es nur dann eingesetzt werden, wenn sich der betreffende Körper nicht zu nahe an der Kamera befindet. Sonst ist eine *Displacement Map* besser geeignet.

[vgl. Demers|2001|S.204 | Kerlow|2004|S.250-252 | Mach|2004|S.113]

Bump Maps [Abb. V-22 | Abb. V-23] wurden im Beispielbild [Abb. V-16] bei der Simulation der Betonstruktur [2] und dem Parkettboden [7] eingesetzt.

[vgl. Seitz|2006]

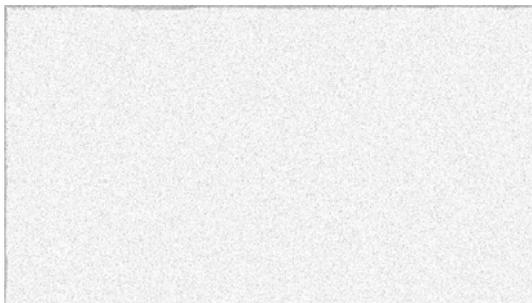


Abb. V-22 | Bump Map für Beton



Abb. V-23 | Bump Map für Parkettboden

Displacement Maps

Displacement Maps verändern, im Gegensatz zu Bump Maps, nicht nur die Normalvektoren, sondern auch die Geometrie der Oberfläche auf die sie projiziert werden. In machen Fällen eignet sich eine Kombination aus Bump und Displacement Map am besten um eine Körper darzustellen.

Darüber hinaus werden sie oft eingesetzt, um Landschaften oder, wie in Abb. V-24, Gras zu generieren.

[vgl. Demers|2001|S.204 | Kerlow|2004|S.252]



Abb. V-24 | Außenvisualisierung, Gras gerendert mit Displacement Map

Zweidimensionale Prozedur Maps

Prozedurale Texturen werden mit Hilfe mathematischer Algorithmen erstellt. Durch mathematische Funktionen die zufällige oder rythmische Muster generieren, entstehen abstrakte Image Maps. Diese zweidimensionalen Bilder können dann in weiterer Folge mit jeder beliebigen Projektionsmethode auf einen Körper gemappt werden.

[vgl. Kerlow|2004|S.253 | Mach|2004|S.113]

Dreidimensionale Prozedur Maps

Diese Art von Textur wird durch dreidimensionale Berechnungsmethoden generiert und befindet sich sowohl auf der Oberfläche des Objekts als auch im Körper selbst. Dadurch müssen dreidimensionale Prozedur Maps auch nicht projiziert werden. Wenn der Körper beispielsweise in der Mitte geteilt wird so setzt sich die Textur im Objekt fort.

Viele natürliche und künstliche Materialien können mit dreidimensionalen Prozedur Maps simuliert werden. Das Aussehen des Materials wird durch verschiedene veränderbare Parameter, wie Farbe, Frequenz, Skalierung, etc., beeinflusst.

Meist ist auch ein Zufallswert, oft als Noise bezeichnet, einstellbar. Diese Zufälligkeit ist sehr wichtig bei der Generierung von natürlichen Materialien wie z.B. Marmor oder Holz.

[vgl. Kerlow|2004|S.253-255]

IV|4 Licht

Die Beleuchtung eines Gebäudemodells ist das heikelste Thema in der Architekturvisualisierung. Die Wirkung, die ein computergeneriertes Bild auf den Betrachter hat, hängt wesentlich von der Wahl der Lichtquellen und der Lichtsetzung ab. Deshalb ist es wichtig über die Grundlagen von Licht in 3D-Software Bescheid zu wissen, wenn man professionelle und zum Bauwerk passende Ergebnisse erzielen will.

Mit gut gesetzter Beleuchtung kommen Materialien, Formen und Farben erst richtig zur Geltung, eine schlechte Beleuchtungssituation kann die gesamte Szene ruinieren.

Besonders bei Bauobjekten berücksichtigen, dass sie bei Tag ein anderes Erscheinungsbild zeigen als bei Nacht. Untertags beeinflussen auch Wetter, Tages- und Jahreszeit die Lichtsituation. In der Nacht, wenn Kunstlichtbeleuchtung simuliert werden soll, gibt es für deren Darstellung viele verschiedene Möglichkeiten.

[vgl. Immler|2004|S.189 | Kerlow|2004|S.211]

IV|4.1 Lichtquellen

In 3D-Programmen stehen dem Benutzer verschiedene Arten von Lichtquellen zur Verfügung, die im folgenden Kapitel näher beschrieben werden.

Beim Setzen von Licht ist zu beachten, dass jede zusätzliche Quelle, besonders bei Spiegelungen und komplizierten Schatten, die Rechenzeit verlängert. Deshalb müssen diese von Vornherein richtig gewählt und sorgfältig gesetzt werden.

[vgl. Immler|2004|S.196]

Ambient Light | Umgebungslicht

Das Licht, das von einem Ambient Light ausgeht, erhellt die Szene gleichmäßig und beleuchtet den ganzen Entwurf. Obwohl Umgebungslicht eine bestimmte XYZ-Position zugewiesen werden kann, ist sie wohl am besten als Lichtquelle, die von allen Richtungen kommt zu beschreiben. Das Umgebungslicht kann dazu eingesetzt werden, die Grundhelligkeit einer Szene zu bestimmen oder auch scharfe Übergänge von Licht in Schatten aufzuweichen.

Dabei kann die Tiefenwirkung eines Bildes verloren gehen und dieses flach und spannungslos wirken. Eine Szene mit zuviel Ambient Light künstlich und unnatürlich. Deshalb ist es oft besser das Umgebungslicht zuerst abzudunkeln, die anderen Lichtquellen zu setzen und erst danach bei Bedarf ein Ambient Light zu setzen.

In manchen Programmen wird das automatisch für jede Szene generierte Punktlicht [engl.: point light] als Umgebungslicht bezeichnet.

[vgl. Ablan|2002|S.64 | Kerlow|2004|S.207 | Mach|2000|S.133-134]

Omni Light | Point Light | Punktlicht

Ein Punktlicht strahlt in alle Richtungen, weshalb es auch als omnidirektionales Licht bezeichnet wird. Sie die einfachste Lichtquelle und kann überall in einer Szene platziert werden. Punktlichter sind z.B. Glühbirnen, Sterne oder Kerzen.

Eine spezielle Form von Point Lights ist das Kameralicht, eine Punktlichtquelle, die sich an derselben Position wie die Kamera befindet.

[vgl. Sonder|2006|S.88 | Kerlow|2004|S.204-205 | Mach|2000|S.134]

Die Szene in Abb. V-25 wurde mit einem Ambient Light, die sich außerhalb des Raumes befindet, und Ambient Omni Lights, die innerhalb des Gebäudes an den Fenstern platziert sind, beleuchtet.

[vgl. [www|ambient\]](#)

In Abb. V-27 wurde unter anderem mit einem Punktlicht, das an direkt an der Kamera angebracht wurde ein Kameralicht simuliert.



Abb. V-25 | Beleuchtung mit Ambient und Omni Lights

Spot Light | Scheinwerfer

Ein Spot Light wirft das Licht kegelförmig in eine Richtung. Diese Lichtquelle hat einen veränderbaren Lichtkegel und *Fall-Off*²¹.

Der Schatten, den Spot Lights erzeugen, ist perspektivisch und hat einen weichen Randbereich zwischen Kernschatten und Halbschatten.

[vgl. Kerlow|2004|S.204-205 | Mach|2000|S.134]

Infinite Light | Directional Light

Infinite Lights sind so weit von den Objekten der Szene entfernt, dass die Lichtstrahlen parallel zueinander auftreffen. Der Übergang von *Schatten*²² [engl.: umbra] zu hellen Bereichen ist schärfer als bei Spot Lights, da Kernschatten und *Halbschatten*²³ [engl.: penumbra] miteinander verschmelzen.

Eine spezielle Form von Infinite Light ist das Sonneobjekt, das in manchen Programmen vorhanden ist. Bei dieser Lichtquelle können Längen- und Breitengrad bzw. der Ort sowie Zeit und Datum eingegeben werden, um ein relativ realistisches Sonnenlicht zu erhalten.

[vgl. Kerlow|2004|S.206 | Mach|2000|S.135]

In Abb. V-26 wurde ein Sonneobjekt für die Beleuchtung des Gebäudemodells verwendet. Die Umgebung ist ein Foto, die Schwierigkeit dabei ist, dass die Beleuchtungssituation des Modells mit dem auf der Fotografie übereinstimmen sollte.

[vgl. [www|sun](http://www.sun)]

²¹ siehe Lichtabnahme und Fall-Off

²² siehe Schatten

²³ siehe Schatten



Abb. V-26 | Gebäudevisualisierung mit Beleuchtung durch Sonnenobjekt

Area Light | Flächenlichter

Manche Programme simulieren Area Light durch mehrere einzelne Lichtquellen, andere durch eine große Lichtfläche. Flächenlichter können in jeder Größe generiert werden, am besten funktionieren aber relativ kleine, runde oder rechteckige Lichtquellen.

Sie werden verwendet, um kleine Bereiche gleichmäßig zu beleuchten oder auch um die Reflexion von Licht, das in ein Zimmer durch ein offenes Fenster fällt, zu simulieren.

Area Lights haben sehr weiche Übergänge zwischen hellen und dunklen Bereichen, die durch den Kernschatten und einen Halbschatten hervorgerufen werden.

[vgl. Kerlow|2004|S.206-207 | Mach|2000|S.135]

In Abb. V-27 wurden zur Simulation von diffusem Licht und Sonnenlicht 2 Flächenlichter links am Fenster gesetzt.

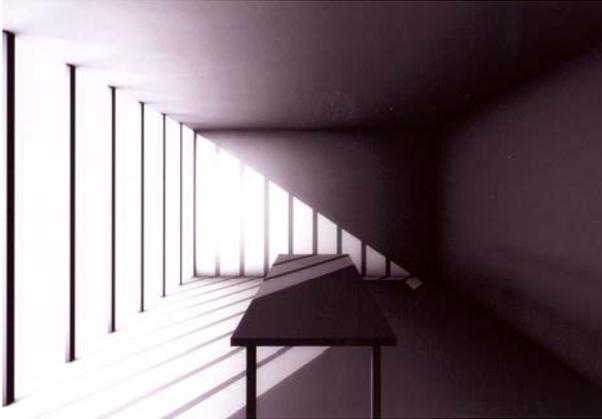


Abb. V-27 | Innenraum – Sonne und diffuses Licht, realisiert durch Sonnenobjekt, Kameralicht, Flächenlicht und diffusen Schatten

Linear Light

Leuchtstoffröhren können durch lineare Lichtquellen dargestellt werden. Sie haben nur eine Länge aber keine Breite und könne auf jede Größe skaliert werden.

Das Berechnen von Linear Lights ist allerdings zeitaufwendig. Deshalb empfiehlt es sich, wenn möglich, anstatt eines Linear Lights mehrere Punktlichtquellen zu verwenden.

[vgl. Kerlow|2004|S.207]

Volumenlicht | Volume Light

Volumenlichter werden eingesetzt um beispielsweise von Lichtquellen ausgehende Lichtkegel, die in staubiger oder feuchter Luft sichtbar werden, auch im 3D-Programm sichtbar zu machen. Anstatt eines Kegels können auch andere geometrische Körper verwendet werden, in welchen das Licht sichtbar wird.

Volume Lights können atmosphärische Effekte zugewiesen werden, um Rauch und Staub zu simulieren.

[vgl. Immler|2004|S.224]

In Abb. V-28 wurden die Straßenlaternen und die Lichtquellen am Gebäude selbst mit Volume Lights realisiert.



Abb. V-28 | Gebäudevisualisierung mit Volume Lights bei Nacht

IV|4.2 Lichteigenschaften

Lichtfarbe und Intensität

In einem 3D-Programm kann Licht in jeder Farbe simuliert werden.

Die Lichtfarbe hat einen wesentlichen Einfluss auf die Wirkung der Lichtquelle und somit auf den Gesamteindruck der Szene. Mit dem Einsatz von farbigem Licht können beeindruckende Effekte erzielt werden.

In der Realität hängt die Lichtfarbe von Tageszeit, Wetter, Längen- und Breitengrad des Ortes und der Umgebung ab.

Auch Lichtquellen, die auf uns im ersten Moment weiß wirken, haben fast immer einen leichten Farbton. Eine Leuchtstoffröhre beispielsweise hat wirklich weißes Licht, bei Glühbirnen wirkt die Lichtfarbe eher Beige bis hell Orange.

Licht mit einem Orange – oder Gelbton lassen die Szene wärmer wirken, durch Blau- und Grüntöne wirkt sie kühler. Auch durch die Kombination von warmen und kühlen Farbtönen können, wie in Abb. V-29, interessante Effekte erzeugt werden. Die Intensität kann in den meisten 3D-Programmen unabhängig von der Lichtfarbe eingestellt werden. Lichtfarbe und Intensität stehen miteinander in Wechselwirkung, das heißt, dass jede Farbänderung in der Regel auch die Lichtintensität beeinflusst.

[vgl. Abla|2002|S.64 | Kerlow|2004|S.202-203, S.208]



Abb. V-29 | Visualisierung mit farbigem Licht, Modell für den CGarchitecture AVC 2006 Wettbewerb

Lichtabnahme und Fall-Off

Der Lichtabnahme-Faktor bestimmt die Stärke der Lichtquelle und die Reichweite des von ihr ausgehenden Lichts. Nahe an der Lichtquelle befindliche Objekte werden heller beleuchtet als Objekte in weiterer Entfernung.

Zu beachten ist auch, dass der Effekt der Lichtabnahme bei Tageslicht nicht auftritt.

Schwache Lichtquellen verlieren, bezogen auf die Entfernung, schneller an Leuchtkraft während das Licht einer starken Lichtquelle auf größere Entfernungen sichtbar bleibt und langsamer abnimmt.

In der Realität ist die Lichtabnahme untrennbar mit der Lichtintensität verbunden, in 3D-Programmen kann diese aber meist unabhängig von der Intensität der Quelle geregelt werden.

Bei einem Punktlicht ist die Lichtabnahme nach allen Richtungen gleich. Bei einem Spot Light nimmt die Lichtintensität einerseits mit der Entfernung zur Quelle ab. Andererseits verliert sie an Stärke vom Zentrum des Lichtkegels zu dessen Rand. Diese Art von Lichtabnahme wird auch Fall-Off genannt. Mit dem Fall-Off-Faktor kann der Lichtkegel eines Scheinwerfers scharf oder weich eingestellt werden.

[vgl. Immler|2004|S.203| Kerlow|2004|S.208-209 | Mach|2000|S.130]

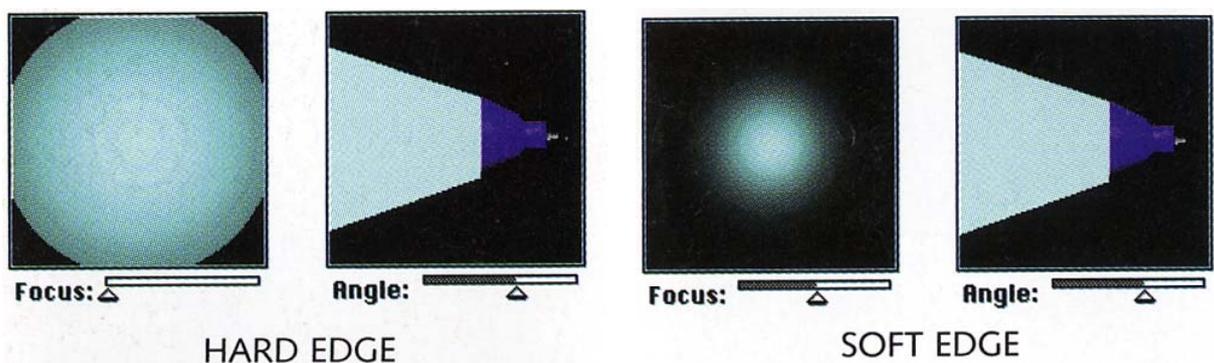


Abb. V-30 | rechts Scheinwerfer mit scharfem Lichtkegel und wenig Fall-Off, links weicher Lichtkegel und hoher Fall-Off Wert

Licht-Einfallswinkel

Wenn eine Fläche genau senkrecht zu einer Lichtquelle ausgerichtet ist, wird die Fläche mit voller Intensität angeleuchtet und der Einfallswinkel ist 0.

[vgl. Mach|2000|S.131]

Schein | Glow | Linseneffekte

Der Schein einer Lichtquelle ergibt sich aus der Brechung und Reflexion von Licht durch Teilchen wie Eis, Staub oder Rauch in der Umgebung.

In der 3D-Technik existieren zwei verschiedene Verfahren dieses Phänomen zu berechnen, und zwar Refraktion und Bleeding.

Refraktion basiert auf dreidimensionalen Berechnungen während Bleeding sich auf zweidimensionale Berechnungsverfahren stützt.

Ein Punktlicht hat einen ringförmigen Glow, ein Scheinwerferlicht einen Schein in Form eines Lichtkegels.

Als weitere Linseneffekte stehen Ringe, Strahlen [engl.: rays] und Sterne [engl.: stars] zur Verfügung.

*Lens Flare*²⁴ ist eine weitere Variation des Scheins, die mit 3D-Programmen simuliert werden kann. Dieser kann auch dazu verwendet werden eine Lichtquelle in einer 3D-Szene sichtbar zu machen.

In Abb. V-31 wird ein Lens Flare verwendet um das Gebäudemodell realistischer erscheinen zu lassen.

[vgl. Immler|2004|S.234 | Kerlow|2004|S.209-210]



Abb. V-31 | Architekturvisualisierung mit Lens Flare



Abb. V-32 | Lens Flare, eingesetzt um Lichtquellen sichtbar zu machen

Globale und Lokale Lichtquellen

Globale Lichtquellen beleuchten alle Objekte in der Szene, in 3D-Software sind alle Lichtquellen per default als globale Lichtquellen eingestellt.

Bei einer lokalen Lichtquelle scheint das Licht auf den Gegenstand, mit dem es zuvor im 3D-Programm verbunden wurde. Dabei wird zwischen exklusiven und inklusiven Verbindungen [engl.: Link] unterschieden.

Wird ein exklusiver Link verwendet, scheint das von der verbundenen Lichtquelle ausgehende Licht ausschließlich auf das gelinkte Element.

²⁴ siehe Glossar

Bei inklusiven Verbindungen werden auch Objekte beleuchtet, die sich direkt im Wirkungsbereich der Lichtquelle befinden.

[vgl. Kerlow|2004|S.210]

Schatten

Jede Lichtquelle wirft Schatten, doch gibt es in 3D-Programmen auch die Möglichkeit diese abzuschalten bzw. zu beeinflussen. Es können also auch Lichter eingesetzt werden, die nur zur Beleuchtung der Szene dienen, aber keine Schatten werfen – eine Möglichkeit, die es in der Fotografie nicht gibt.

Der innere Teil des Schattens, in dem das Licht vollständig blockiert wird, wird Kernschatten genannt. Hingegen wird der Rand, in dem der Schatten mit anderen Lichtern in der Umgebung überblendet, als Halbschatten bezeichnet. Körperschatten wird jener Schatten genannt, der auf dem Objekt selbst entsteht.

Das Aussehen des fertig gerenderten Schattens hängt nicht nur von den veränderbaren Eigenschaften des Schattens, wie Schattenfarbe, Deckkraft, Farbe des Halbschattens und der Schärfe der Schattenkante, sondern auch von den Eigenschaften des Schatten werfenden Objekts und der gewählten Rendermethode ab. Im ersten Moment erscheinen uns Schatten grau. Aber die Schattenfarbe hängt stark von der Beleuchtung ab, in der Regel entspricht die Farbe des Schattens der Komplementärfarbe der Hauptlichtquelle.

Objekte, die Lichtquellen darstellen, wie beispielsweise eine modellierte Lampe mit einer darin befindlichen Lichtquelle, werfen in der Realität keine Schatten. Bei diesen können in der 3D-Software die Schatten deaktiviert werden.

Schatten beeinflussen die Raumwirkung, heben Form und Struktur eines Objektes erst richtig hervor. Auch Dreidimensionalität und Perspektive werden wesentlich durch Schatten geprägt.

Durch das Zusammenspiel von Licht und Schatten wird die Stimmung des Gesamtbilds definiert.

[vgl. Immler|2004|S.213 | Kerlow|2004|S.210-211 | Mach|2000|S.136, S.139]

In Abb. V-33 ist das Gebäudemodell noch ohne Schatten zu sehen, in Abb. V-34 wurde der zuvor separat gerenderte Schatten hinzugefügt.



Abb. V-33 | Visualisierung ohne Schatten



Abb. V-34 | Visualisierung mit Schatten

IV|4.3 Lichtfunktionen

Hauptlicht | Führungslicht [Key Light]

Das Hauptlicht ist die wichtigste Lichtquelle der Szene. Durch das Setzen des Führungslichts wird die Richtung des Lichts und dadurch auch die Richtung der Schatten bestimmt. Eine Szene kann auch nur durch ein Hauptlicht beleuchtet werden.

Als Führungslicht werden meist Spotlichter oder Flächenlichter verwendet. Das Setzen des Key Lights ist der erste Schritt beim Beleuchten eines Objekts, mit den anderen Lichtquellen werden nur seine Eigenheiten verfeinert und abgerundet.

[vgl. Mach|2004|S.142 | Kerlow|2004|S.215]

Fülllicht

Das Fülllicht wird erstens eingesetzt, um die allgemeine Lichtfarbe der Szene zu bestimmen, zweitens, um die von den anderen Lichtquellen erzeugten Schatten aufzuhellen. Als Fülllicht werden Spotlichtquellen oder Infinite Lights ohne Schattenwurf eingesetzt.

[vgl. Mach|2004|S.143 | Kerlow|2004|S.215]

IV|4.4 Image Based Lighting [IBL]

Bei Image Based Lightning wird eine Szene mit Hilfe eines Bildes beleuchtet, dabei wird die Lichtinformation beim Rendern aus dem Bild extrahiert und auf das Objekt angewandt.

Dazu wird ein so genanntes *light probe image*, ein *High Dynamic Range Image*²⁵ [HDRI] einer realen Beleuchtungssituation, benötigt. Dieses wird dann mit Hilfe von environment Mapping in die Umgebung des Objekts projiziert.

Die Szene wird in weiterer Folge mit einem *Image Based Rendering*²⁶ Verfahren berechnet.

Mit dieser Technik können besonders realistische Beleuchtungssituationen geschaffen werden.

[vgl. Debevec|2002]

Das Gebäudemodell in Abb. V-35 wurde mit IBL beleuchtet.



Abb. V-35 | Rendering mit IBL, Musem Lyon, Architekt Coop Himmelblau

²⁵ siehe Glossar

²⁶ siehe IV|7.7 Image Based Rendering

IV|5 Kamera

In der Computergrafik werden beinahe dieselben Begriffe und Grundlagen verwendet wie in der realen Welt, da sich diese auch an echten Kameras orientieren. Ohne eine virtuelle Kamera kann die Szene dem Betrachter nicht vermittelt werden. Durch die Kamera wird bestimmt, was der Betrachter wann und in welcher Form sieht.

[vgl. Mach|2004|S.156 | Kerlow|2004|S.183-184]

IV|5.1 Allgemein

Kameralinsen, Brennweite

Die Linse bildet die Szene ab, indem sie die dreidimensionale Szene auf mit Hilfe von perspektivischer Projektion auf eine zweidimensionale Fläche – die Image Plane – projiziert.

Die Brennweite einer Kamera bestimmt das von der Kamera erfasste Blickfeld. Die Brennweite eines Objektivs wird in mm angegeben.

In 3D-Software steht dem Benutzer eine unendliche Anzahl an Brennweiten zur Verfügung. Diese können im 3D-Programm in den Kameraeigenschaften definiert werden. Bei einer realen Kamera benutzt der Fotograf verschiedene Objektive mit unterschiedlichen Brennweiten.

Die Kameralinse ist die wichtigste Komponente einer Kamera, da sie bestimmt wie die 3D-Szene in ein 2D-Bild umgerechnet wird.

In der Fotografie und im Film werden hauptsächlich folgende Brennweiten, Objektive verwendet.

Die *Standardbrennweite* [50-55mm] entspricht dem Sehwinkel des Auges. Mit einer langen Brennweite [135mm] – einem *Teleobjektiv* – wird nur ein kleiner Ausschnitt der Szene erfasst, dafür werden Details genauer abgebildet. Wenn die Brennweite zu lang ist, geht allerdings der perspektivische Eindruck verloren. Ein Objektiv mit einer kurzen Brennweite [28mm] wird als *Weitwinkelobjektiv* bezeichnet. Dadurch könne mehr Details der Szene erfasst werden, allerdings wird die Perspektive verzerrt.

In der folgenden Tabelle werden die Objektive mit der entsprechenden Brennweite wie sie in 3D-Programmen angenommen werden können angeführt.

[vgl. Mach|2004|S.157-159 | Kerlow|2004|S.192-194]

Objektiv	Bezeichnung	Brennweite
Superweitwinke	Kurze Brennweite	10-25 mm
Weitwinkel	Kurze Brennweite	25-40 mm
Standard	Mittlere Brennweite	40-60 mm
Leichtes Tele	Lange Brennweite	60-100 mm
Mittleres Tele	Lange Brennweite	100-200 mm
Tele	Lange Brennweite	200-300 mm
Supertele	Lange Brennweite	300-... mm

[Mach|2004|S.158]

Bildformat

Beim Rendern eines Standbilds stehen 3 Bildformate zur Auswahl, Quadratisch, Querformat und Hochformat. Das quadratische Format wirkt ausgeglichen und neutral, kann aber auch spannungslos wirken. Das Querformat entspricht der Art, wie der Mensch sieht, es wirkt ausladender als die anderen beiden Formate. Ein Extrem wäre das Panoramabild. Durch das Hochformat wird die Horizontale betont und ist zur Darstellung hoher Objekte und es hat eine erhabene und spannungsgeladene Wirkung auf den Betrachter.

[vgl. Mach|2004|S.164]

Kamerastandpunkt | POV [Point of View], POI [Point of Interest]

Der Kamerastandpunkt ist der Ort an dem die Kamera platziert ist. Als POI wird der Bereich bezeichnet, auf welchen die Kamera fokussiert.

[vgl. Kerlow|2004|S. 186]

Unschärfe | Tiefenschärfe | DOF [Depth of field]

Bei einer realen Kamera erscheinen die Bereiche unscharf, die nicht im Brennpunkt liegen. Das ist bedingt durch die Linsenunschärfe. In 3D-Software gibt es diese Unschärfe standardmäßig nicht, weshalb computergenerierte Bilder auf uns oft unnatürlich und zu sauber wirken. Doch kann auch mit 3D-Programme Tiefenschärfe simuliert werden. Ein hoher DOF-Wert lässt beinahe alles in der Szene scharf erscheinen. Wird ein niedriger Wert eingestellt, bleibt nur ein kleiner Bereich fokussiert.

[vgl. Ablan|2002|S.14 | Mach|2004|S.174]

Abb. V-36 zeigt eine Visualisierung ohne Unschärfe, in Abb. V-37 dieselbe Szene mit Tiefenunschärfe.



Abb. V-36 | Visualisierung ohne Unschärfe



Abb. V-37 | Visualisierung mit Unschärfe

IV|5.2 Kameraanimation

Typische in der Architekturvisualisierung angewandte Animationen sind ein Walkthrough, ein Flug über das Bauobjekt, oder eine Fahrt um das Bauwerk. Als Walkthrough wird das Durchwandern einer Szene genannt, die Kamera befindet sich dabei auf Augenhöhe.

Änderung der Kameraposition

Die Position einer Kamera kann in drei Richtungen verändert werden. Die Bewegung einer Kamera entlang der horizontalen Achse wird Dolly genannt. Ein Truck ist die Bewegung der Kamera nach hinten oder nach vorne. Das Bewegen nach oben oder nach unten ist ein Boom.

[vgl. Kerlow|2004|S. 314]

Änderung der Kameraorientierung

Die verschiedenen Änderungen der Kameraorientierung sind Tilt, Roll und Pan. Als Tilt wird das Rotieren der Kamera entlang der horizontalen Achse genannt. Die Rotation entlang der Z-Achse ist Roll und als Pan wird die Rotation entlang der vertikalen Achse bezeichnet.

[vgl. Kerlow|2004|S. 314]

Pfadanimation

Eine relativ einfache Technik komplexe Kamerafahrten zu generieren, ist die Animation der Kamera entlang eines vordefinierten Pfads. Dabei wird zuerst ein Pfad gezeichnet mit welchem die Kamera in weiterer Folge verknüpft wird. Auch die Geschwindigkeit der Kamera kann während der Kamerafahrt verändert werden. Darüber hinaus kann die Pfadanimation durch die oben beschriebenen Standardkamerabewegungen erweitert werden.

In vielen 3D-Programmen kann die Pfadanimation in die einzelnen Animationsteile aufgegliedert und diese so einzeln weiterbearbeitet werden.

[vgl. Kerlow|2004|S. 315-316]

IV|6 „Zubehör“

Um architektonische Entwürfe realistischer und interessanter aussehen zu lassen, wurden schon in analogen Architekturmodellen Personen, Pflanzen und eventuell auch Möbel platziert.

Durch die Weiterentwicklung von Computerhardware und Software gibt es nun schon viele Möglichkeiten, digitale Architekturmodelle mit „Zubehör“ auszustatten. Geschieht die Einbindung von Pflanzen, Personen oder Möbeln nicht schon beim Rendern, kann sie anschließend im Compositing nachgeholt werden.

Abb.V-38 bis Abb. V-40 zeigen die Fotomontag einer Person in ein fertig gerendertes Architekturmodell.



Abb. V-38 | Originalfoto



Abb. V-39 | nachbearbeitetes Rendering mit fotomontierter Person



Abb. V-40 | Rendering

IV|6.1 Billboards

Eine weitere einfache Form Pflanzen, Personen oder Möbel zu simulieren sind so genannte Billboards.

Als Billboards werden Flächen bezeichnet, auf die Bilder oder Animationen des darzustellenden Objekts projiziert werden. Je nach Bedarf können auch mehrere Projektionsflächen ineinander verschachtelt werden, um auch aus anderen Blickwinkeln ein Objekt realistisch zu simulieren.

Unterschieden werden statische und view-aligned Billboards. View-aligned Billboards ändern ihre Position so dass sie immer frontal zum Betrachter ausgerichtet sind. Anwendung finden sie hauptsächlich in Computerspielen. Statische Billboards bleiben wie der Name schon sagt immer in der gleichen Position.

Ein Problem, das sich durch das reine Projizieren ergibt, ist der Schatten des Billboards, der sich auf die Fläche und nicht auf das darauf gemappte Bild bezieht. Dieser Schönheitsfehler kann aber durch *Alpha-Kanäle*²⁷ in der verwendeten Grafik korrigiert werden.

Der große Vorteil dieser Art der Darstellung ist der relativ geringe Rechenaufwand und der Effekt ist für viele Anwendungen im Bereich der Architekturvisualisierung völlig ausreichend.

[vgl. Kerlow|2004|S.260-261 | Umlauf|2004|S.11]

IV|6.2 RCP [Real Photorealistic Content]

RCP ist eine von der Firma ArchVision entwickelte Technologie zur Darstellung von Personen und Objekten, die auf IMBR [Image Based Rendering] -Methoden basiert. Es werden also 2D-Bilder verwendet um die Illusion eines 3D-Objekts zu erzeugen. Durch dieses Verfahren wird erstens die Renderzeit verkürzt und zweitens eine sehr realistische Darstellung der Objekte generiert.

Ein Nachteil von RCP-Objekten ist, dass sie nur in einem Winkel von höchstens 40° betrachtet werden können, da es keine Daten zu Ansicht von oben gibt.

²⁷ siehe Glossar

Es gibt RPC-Bibliotheken von Bäumen, Personen, Fahrzeugen und anderen Objekten. Um RCP-Bibliotheken zu nutzen, muss zuerst ein Plug-In installiert werden. Diese gibt es gratis und zu fast jeder bekannten 3D-Software.

Neben statischen RCP-Elementen existieren es seit 2001 auch animierte RCP-Objekte, wie gehende oder sich unterhaltende Personen und fahrende Autos.

[vgl. www|rnp]

Die Personen in Abb. V-41 stammen aus einer RCP-Bibliothek.

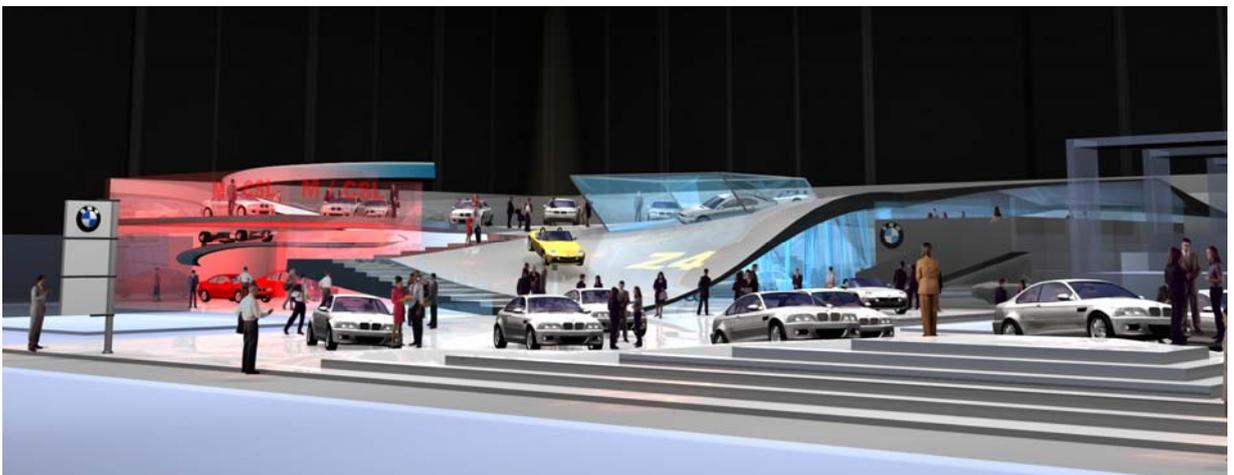


Abb. V-41 | Visualisierung Autosalon Genf BMW mit RCP-Personen

IV|6.3 Crowd Simulation

In der Städteplanung können auch Systeme zur *Crowd Simulation*²⁸ zum Einsatz kommen. Oder beispielsweise zur Visualisierung von Personen von oben bei einem animierten Flug über ein Bauobjekt

²⁸ siehe Glossar

IV|6.4 Möbel, Fahrzeuge

Möbel und Fahrzeuge können selbst modelliert werden, was aber in den meisten Fällen zuviel Aufwand wäre.

Wie schon erwähnt können Billboards, RCP-Objekte eingesetzt werden. Auch ein nachträgliches Hinzufügen im Compositing ist möglich.

Eine weitere Möglichkeit ist es fertig modellierte und texturierte Möbel oder Fahrzeuge zu kaufen.

Eine bekannte Website, die diese zum Kauf anbietet ist www.turbosquid.com.



Abb. V-42 | 3D-Modell Barcelona Chair von Mies van der Rohe, www.turbosquid.com

IV|6.5 Pflanzengeneratoren

Auch Pflanzen können in der Architekturvisualisierung in Abhängigkeit von der Art des Projekts eine wichtige Rolle spielen.

Es gibt verschiedene Möglichkeiten Gewächse zu visualisieren, oft reicht ein Billboard um beispielsweise Bäume und Zimmerpflanzen glaubhaft darzustellen, manchmal werden aber auch Pflanzengeneratoren wie z.B. Greenworks Xfrog oder IDV SpeedTree herangezogen. In bestimmten Fällen, wie in Abb. V-43 und Abb. V-44, kann es auch notwendig sein Gewächse selbst zu modellieren und zu texturieren.



Abb. V-43 | Entwurf Büro Rockstar Games mit händisch modellierten Pflanzen, Detailansicht

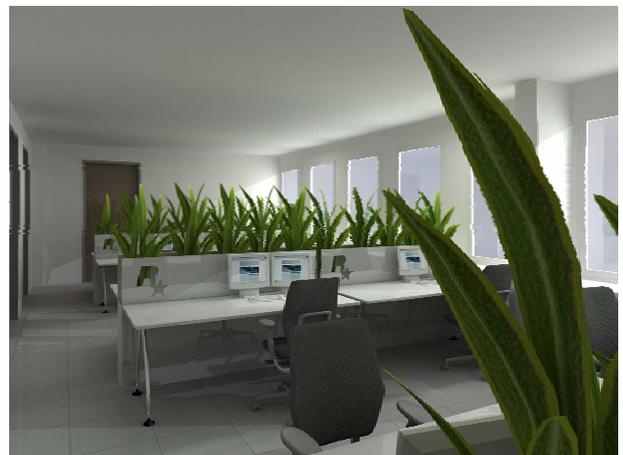


Abb. V-44 | Entwurf Büro Rockstar Games mit händisch modellierten Pflanzen

Durch die immer leichtere Handhabung von Pflanzengeneratoren und ihre wachsende Verbreitung werden auch diese interessant für den Bereich der Architekturvisualisierung, da mit diesen relativ schnell und einfach verblüffende Ergebnisse erzielbar sind. Der Nachteil ist nach wie vor die hohe Rechenkapazität, die benötigt wird, um mit solcher Software zu arbeiten.

Pflanzengeneratoren arbeiten mit verschiedenen Ansätzen, um möglichst naturgetreue Darstellungen zu erzeugen. Manche ziehen exakte wissenschaftliche Beschreibungen heran, die in enger Zusammenarbeit mit Botanikern entstehen wie Greenworks's XFrog, andere basieren auf einfacher Beobachtung botanischer

Prinzipien. Wachstumsprozesse werden oft mittels so genannter *L-Systems*²⁹ berechnet. Die Pflanzen in Abb. V-45 wurden mit XFrog generiert.



Abb. V-45 | Pflanzen generiert mit XFrog

Einen guten Pflanzengenerator macht neben der eingesetzten Technologie auch die Pflanzenvielfalt aus. Bei einigen Programmen ist diese Vielfalt schon im Lieferumfang enthalten, bei anderen besteht die Möglichkeit noch zusätzlich welche zu kaufen.

Das Problem bei diesen Anwendungen ist aber eindeutig die sehr hohe benötigte Rechenleistung. Doch da die Hersteller vermehrt auf Plug-Ins setzen, die in 3D Programme integriert werden können, kann eine effizientere Nutzung der Rechnerressourcen erfolgen.

Einen anderen interessanten Ansatz, um Rechenleistung zu sparen bietet das Programm natFX, ein vollständig auf Algorithmen basierender Pflanzengenerator, an. Bei der Plug-In Version für 3ds Max gibt es die Möglichkeit, Pflanzen in drei verschieden detaillierten Darstellungsformen einzusetzen. Der Benutzer hat die Wahl, die Pflanze als Billboard, als *Billboard Cloud*³⁰ oder als Polygonmodell berechnen zu lassen. Diese 3 Detailvarianten können je nach Bedarf zugleich eingesetzt werden. Zusätzlich können Alter der Pflanze und Jahreszeit simuliert werden.

²⁹ siehe Glossar

³⁰ siehe Glossar

Das Plug-In für Cinema4D bietet mit Hilfe des Plant Tuners differenziertere Möglichkeiten zur Beeinflussung des LODs³¹ [Level of Detail] der Pflanze. So kann man zwischen vier unterschiedlichen Geometrietypen [Full 3D, Volumic hybrid, Hybrid, Billboard] wählen, an diesen Einstellungen vornehmen, das Wood Detail Level und Organ Detail Level einstellen. Gleichzeitig zeigt natFX die Anzahl der Eckpunkte und *Triangles*³² der Pflanze an.

Ein weiteres interessantes Feature ist die Möglichkeit, einzelne Äste zu entfernen und so die Pflanze ohne großen Aufwand zu verändern.

[vgl. KuHa|06|2005|2|S.74]

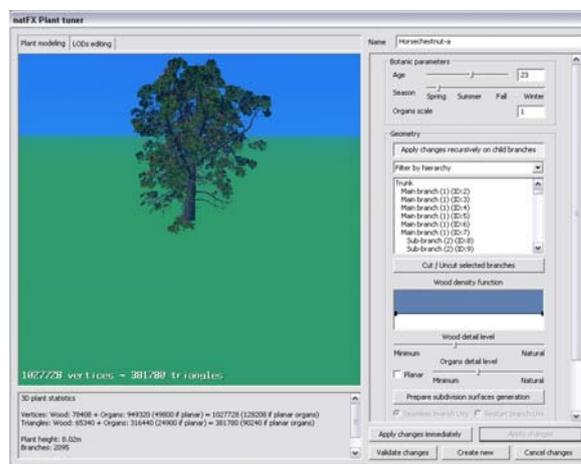


Abb. V-46 | Plant Tuner von natFX für Maya

³¹ siehe Glossar

³² siehe Glossar

IV|7 Rendertechniken

Das zweidimensionale Erscheinungsbild einer 3D-Szene wird neben der Lichtsetzung und der Wahl der Materialien auch wesentlich von der Rendermethode beeinflusst.

Rendersoftware kann vom User selbst nicht verändert werden, außer durch die Einstellung verschiedener Parameter wie Licht, Oberflächenbeschaffenheit und Shading. Eine Ausnahme bildet dabei Photorealistic RenderMan [PRMan] von Pixar, die vom Benutzer selbst mit Hilfe der RenderMan Shading Language um- und weiterprogrammiert werden kann.

Oft werden dieselben Rendertechniken von den verschiedenen Softwareherstellern anders implementiert. Aus diesem Grund können Ergebnisse mit denselben Einstellungen in verschiedenen Softwarepaketen anders aussehen. Zusätzlich können auch verschiedene Rendermethoden kombiniert werden, dieses Kombinieren wird als Hybrid Rendering bezeichnet.

IV|7.1 Scanline Rendering

Das Scanline Rendering Verfahren, auch Tastlinienrendern, von Watkins ist eine *image space rendering*³³-Methode, die zeitgleich mit John E. Warnocks *area subdivision*³⁴ und Newells *depth sort*³⁵ in den 70er Jahren erfunden wurde.

Mit dieser Methode können grafische Primitive auf Rastergeräten in 2D dargestellt werden. Im 3D-Bereich kann dieses Verfahren erst nach der *projektiven Abbildung*³⁶ möglich.

Das Bild wird dann von oben nach unten, Zeile für Zeile, abgetastet. Eine solche Zeile wird Tastlinie oder auch Scanline genannt.

³³ siehe Glossar

³⁴ siehe Glossar

³⁵ siehe Glossar

³⁶ siehe Glossar

IV|7.2 Z-Buffer

Die Z-Buffer Methode, erfunden von Dr. Edwin Catmull, bestimmt die Sichtbarkeit einzelner Pixel an jeweils einem Objekt. Jedes Pixel wird nach seiner z-Position, auch Tiefenwert [engl.: depth] genannt, sortiert und in einem *Puffer*³⁷ als zweidimensionales Array [x-y] gespeichert. Wird ein zweites Pixel mit denselben x und y Werten gefunden, vergleicht der Algorithmus die beiden Pixel und *shaded*³⁸ das Pixel, das dem Beobachter am nächsten ist. Wird in späterer Folge ein anders Pixel gefunden, das noch näher an der Kamera liegt, wird der Wert wieder überschrieben. So wird ermittelt, welches Pixel sichtbar ist und welches von einem anderen Pixel des Objekts verdeckt wird.

Ist die Sichtbarkeit der Pixel in einem Objekt geklärt, geht der Algorithmus zum nächsten Objekt über.

Die Z-Buffer Rendertechnik ist neben Raytracing eines der häufigsten Verfahren zur Verdeckungsberechnung [engl.: hidden surface determination]. Es handelt sich bei dieser Methode um eine *image space rendering*-Technik, die aber schon Ansätze von *object space rendering*³⁹ zeigt.

[vgl. Kerlow|2004|S.162-163 | www.wiki.zbuffer]

IV|7.3 Raytracing

Bei Raytracing oder Strahlenverfolgung, handelt es sich um einen Algorithmus mit dem es möglich ist, fotorealistische Bilder aus dreidimensionalen Szenen zu generieren.

³⁷ siehe Glossar

³⁸ siehe Glossar

³⁹ siehe Glossar

Der Algorithmus wurde in den 60er Jahren von Appel, Goldstein und Nagel erstmals veröffentlicht. In seiner Grundform konnte mit Raytracing nur direkte Beleuchtung berechnet werden, doch wurde die Technik im Laufe der Jahre soweit verbessert, dass auch indirektes Licht berücksichtigt wird.

Im Prinzip wird bei Raytracing für jedes im Kameraausschnitt sichtbare Pixel ein Strahl [engl.: ray], auch als Samplingstrahl bezeichnet, bis zur Lichtquelle verfolgt. Ein Strahl ist eine gedachte gerade Linie, die auf dem Weg zur Lichtquelle Renderinformation sammelt. So wird berechnet, wie sich der Lichtstrahl verhält, wenn er auf verschiedenen Oberflächen auftrifft oder sie durchdringt.

Es wird also nicht der Weg der Lichtstrahlen von der Lichtquelle ausgehend berechnet, sondern der Weg der sichtbaren Lichtstrahlen zur Lichtquelle, da sonst auch jene Strahlen berücksichtigt werden müssten, die im eigentlichen Kameraausschnitt gar nicht zu sehen sind.

Aus diesem Grund spricht man auch von backward ray tracing.

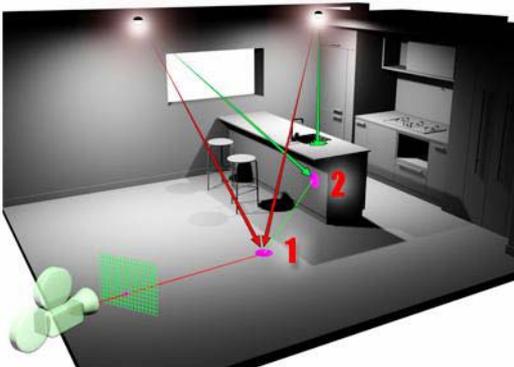


Abb. V-47 | Raytracing Funktionsweise

Der Hauptgrund, warum Raytracing so gute Ergebnisse liefert, ist, dass das Bild, welches aus einer 3D-Szene entsteht, auch in einer dreidimensionalen Umgebung berechnet wird. Deshalb kann mittels dieser Technik das Verhalten von Licht im dreidimensionalen Raum genauer simuliert werden als bei image space rendering-Methoden.

Die Hauptparameter bei Raytracing sind die *ray tracing depth*⁴⁰, die Pixelanzahl des zu berechnenden Bildes und die Anzahl der Lichtquellen.

⁴⁰ siehe Glossar

Die Pixelanzahl der Szene bestimmt die Anzahl der Strahlen, die zum Rendern des Bildes benötigt wird. Pro Pixel wird ein Strahl zur Lichtquelle verfolgt. Hat das zu berechnende Bild beispielsweise eine Auflösung von 1024x768 so sind, ohne Berücksichtigung von Schatten- und Brechungsstrahlen, 786.432 Strahlen zur Kalkulation notwendig.

Die meisten Raytracer verfügen drei verschiedene Arten von Strahlen, Reflektionsstrahlen [engl.: reflections rays], Schattenstrahlen [engl.: shadow rays] und Brechungsstrahlen [engl.: refraction rays]. Die ray tracing depth kann für jede Strahlenart unabhängig bestimmt werden.

Reflektionsstrahlen bewegen sich geradlinig in der Szene bis sie auf eine reflektierende Oberfläche treffen. Dort werden sie im selben Winkel wie ihr Einfallswinkel reflektiert.

Nachdem ein bestimmter Punkt im dreidimensionalen Raum von einem Strahl getroffen wurde, wird ein Schattenstrahl von diesem Punkt aus in die Lichtquelle gesendet. Der Punkt wird nur dann dargestellt, wenn der Schattenstrahl auf kein anderes Objekt auf seinem Weg in die Lichtquelle trifft.

Trifft ein Strahl auf seinem Weg durch die 3D-Szene auf eine transparente Oberfläche, werden Brechungsstrahlen generiert, mit welchen die Lichtbrechung im Objekt berechnet wird.

Raytracing eignet sich sehr gut dazu, scharfe Kanten, Spiegelungen, Reflexionen und Refraktionen darzustellen.

Das Raytracing Renderverfahren wird von allen hochwertigen Renderern, eventuell in Kombination mit anderen Rendertechniken angewandt.

[vgl. Kerlow|2004|S.163-165 | www.wiki.raytracing | Mach|2000|S.181]

IV|7.4 globale Beleuchtungsverfahren

Die Bezeichnung globales Beleuchtungsverfahren [engl.: Global Illumination] bedeutet in der 3D-Computergrafik die Simulation aller Möglichkeiten der Ausbreitung von Lichtstrahlen in einer 3D-Szene. Dadurch werden die Gesetze der geometrischen Optik sowie der Energieerhaltung vollständig erfüllt und ein relativ realistisches Bild erzeugt. Allerdings sind diese Verfahren viel langsamer und rechenintensiver als direkte Beleuchtungsverfahren. Im folgenden Abschnitt werden die globalen Beleuchtungsverfahren Radiosity, Photon Mapping und Image-Based Rendering genauer beschrieben.

[vgl. Kerlow|2004|S.165 | www.wiki.global/]

IV|7.5 Radiosity

Bei Radiosity bzw. Radiosität handelt es sich um ein *globales Beleuchtungsverfahren*⁴¹, das auf dem *Energieerhaltungssatz*⁴² beruht. Es wurde erstmals 1984 von Forschern der Cornell University [C. Goral, K. E. Torrance, D. P. Greenberg and B. Battaile] in ihrer Publikation „Modeling the interaction of light between diffuse surfaces“ vorgestellt.

⁴¹ siehe globale Beleuchtungsverfahren

⁴² siehe Glossar

Zuerst werden die vorhandenen Objekte in Lichtquellen, Licht blockierende und Licht absorbierende Oberflächen eingeteilt. Bei dieser Einteilung entsteht ein numerisches Rechengitter, das Ähnlichkeit mit einem Subdivision Grid hat. Dann wird mit Hilfe diese Gitters solange der Anteil an Licht berechnet, der jeweils von einer Oberfläche auf die andere übertragen wird, bis die Lichtenergie vollständig von den Oberflächen absorbiert oder im Raum zerstreut ist.



Abb. V-48 | Radiosity Rechengitter



Abb. V-49 | mittels Radiosity gerenderte Szene

Einer der beeindruckendsten Lichteffekte, der mit Radiosity simuliert werden kann, ist Color Bleeding [Abb. V-50], welches auftritt, wenn farbige Objekte einen Teil ihrer Farbe durch diffuses Licht an benachbarte Objekte weitergeben.

Einer der Vorteile von Radiosity ist, dass der Algorithmus im Vergleich mit Raytracing nicht vom betrachteten Bildausschnitt abhängig ist. So muss für jede Szene die Lichtverteilung nur einmal gerendert werden. Der Rest der Szene kann dann in Echtzeit – oft über Scanline Rendering – berechnet werden, was besonders für den Bereich der Architekturvisualisierung interessant ist.

Mit dem Radiosity Verfahren ist es aber nicht möglich spiegelnde und transparente Objekte darzustellen, was mit Raytracing kein Problem war. Da Raytracing mit der Zeit laufend verbessert wurde und es jetzt möglich ist, alle von Radiosity unterstützen Effekte sogar noch naturgetreuer zu simulieren, ist Radiosity im Bereich des fotorealistischen Renderns weitgehend aus der Mode gekommen.

In der Architekturvisualisierung wird Radiosity besonders in den Bereichen der Innenarchitektur, der Lichtsimulation und Lichtgestaltung eingesetzt, da es sich sehr gut zur Darstellung von Innenbeleuchtungen eignet. Ein weiterer

Einsatzbereich ist, wie oben erwähnt, das Echtzeitrendering von Architekturmodellen.

[vgl. Kerlow|2004|S.165-166 | Mach|2000|S. 182| [www|wiki|radiosity/](http://www.wiki|radiosity/) | [www|wiki|radiosityII/](http://www.wiki|radiosityII/)]

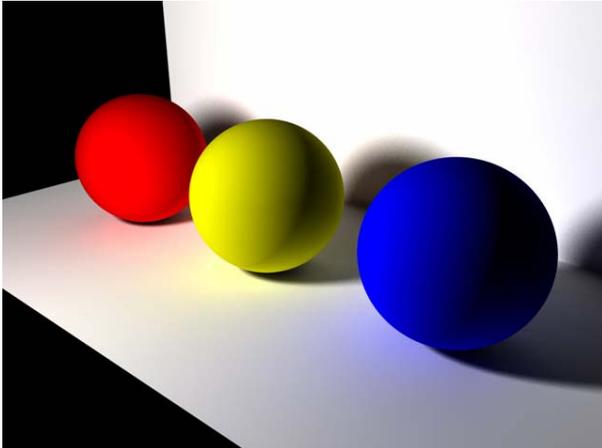


Abb. V-50 | Color Bleeding

IV|7.6 Photon Mapping

Photon Mapping oder auch Photonenkartierung ist kein eigenständiges Renderverfahren, sondern dient als Erweiterung von Raytracing-basierenden Verfahren. Es wurde 1995 von Henrik Wann Jensen veröffentlicht.

Bei dieser Technik werden vor dem eigentlichen Rendervorgang Photonen, ausgehend von den Lichtquellen, durch die Szene geschossen. Je nach Beschaffenheit der Oberfläche, auf die sie treffen, werden sie reflektiert, gestreut und gebrochen. Wenn sie auf eine diffuse Oberfläche aufkommen, werden sie auf einer so genannten [globalen] Photonenkarte [engl.: photon map] gespeichert. Dieser Vorgang wird Photon Tracing genannt. Die Photonenkarte ist völlig unabhängig von der Geometrie der Szene, somit kann indirekte Beleuchtung auch in sehr komplexen Szenen berechnet werden.

Dabei bezieht sich die Bezeichnung *Photon* nur auf die Tatsache, dass Lichtenergie transportiert wird. Es handelt sich bei Photon Mapping aber nicht um eine quantenphysikalisch korrekte Simulation der Ausbreitung von Licht.

Mittels Photon Mapping können besonders Kaustiken, eine Erscheinung, die auftritt, wenn Licht durch Kristall oder Wasser scheint, berechnet werden. Zu

diesem Zweck gibt es auch eine eigene Kaustik-Photonenkarte. Kaustiken waren lange Zeit die am schwersten zu simulierenden Phänomene; sie lassen sich nur von Verfahren gut darstellen, die von Lichtquellen aus arbeiten.

[vgl. Kerlow|2004|S.166-167 | www.wiki.kaustik | www.wiki.photon | www.StruTrau]

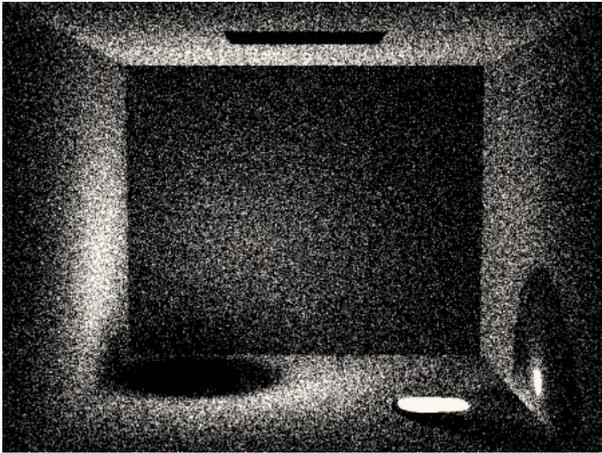


Abb. V-51 | Kaustik Photonen

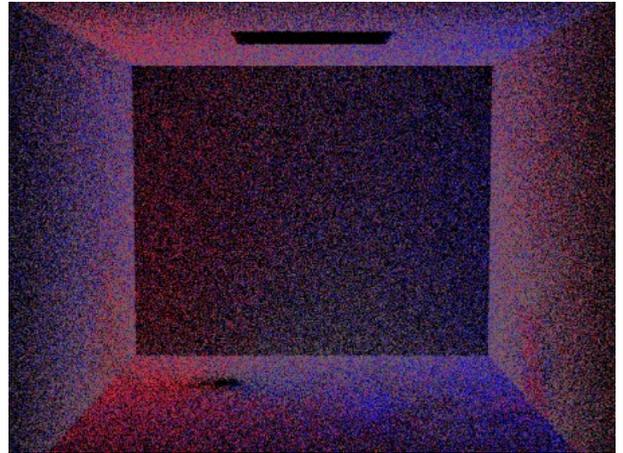


Abb. V-52 | Globale Photonen

IV|7.7 Image Based Rendering

Bei Image Based Rendering-Methoden werden dreidimensionale Umgebungen mit Hilfe von zweidimensionalen Fotos berechnet, dadurch hängt das Resultat sehr stark vom gesammelten Ausgangsmaterial ab. Um gute Ergebnisse zu erzielen, müssen die verwendeten Fotos so viel Qualität und Auflösung wie möglich haben. Zusätzlich sollten auch Informationen über Belichtungszeit, Blende, Dynamikumfang, Lichtintensität und Leuchtkraft gesammelt werden. Weiters sind Kameraposition und der Abstand zu wichtigen Punkte in der zu berechnenden Szene notwendig.

Auch sollte die Art von Licht, die auf die Objekte fällt, wie z.B. Sonnenlicht oder künstliche Lichtquellen bekannt sein.

Für Image Based Rendering eignen sich HDR Fotos sehr gut, da dabei keine Informationen verloren gehen und die oben genannten, für IBR-Verfahren benötigten Informationen im HDR-Bild enthalten sind. Mit Hilfe dieser Fotos kann eine realistische 3D-Szene mit sehr schönen Schatten und Reflexionen berechnet werden. Dies bedeutet für den Benutzer, dass er sich nicht mit dem Setzen von

Lichtquellen und Experimentieren mit der Farbgebung und Stärke der Lichter auseinandersetzen muss.

HDRIs enthalten auch Informationen darüber, ob es sich um eine direkte Beleuchtung wie z.B. Sonne, Halogenstrahler mit harten Schatten, oder um diffuse Beleuchtung mit weichen Schatten handelt.

[vgl. Kerlow|2004|S.167-170]

IV|7.8 Render-Software | Renderer

Ein Renderer ist die Software, die ein zweidimensionales Bild aus der 3D-Szene berechnet. In beinahe jedes 3D-Programm ist zumindest eine Render-Software integriert, doch gibt es auch Plug-Ins und Standalone-Software, die oft noch bessere Ergebnisse, als die integrierten Renderer erzielen.

Eine in der Architekturvisualisierung häufig eingesetzte Render-Software ist V-Ray [siehe Abb. V-53] von Chaos Group. Dieses Programm gibt es aber nur als Plug-In für 3ds Max.

Weitere Beispiele für Renderer sind SplutterFish Brazil, Mental Images Mental Ray [siehe Abb. V-54] und Pixar PRMan [Photorealistic RenderMan].



Abb. V-53 | Visualisierung gerendert mit V-Ray



Abb. V-54 | Visualisierung Berlin-Schönefeld, gerendert mit Mental Ray, Architekten Gerkan Marg und Partner

V. Zusammenfassung

V|1 Beantwortung der forschungsleitenden Fragestellung

I. Was ist im Bereich der Architekturvisualisierung besonders zu beachten?

In der Architekturvisualisierung wird ein besonderes Augenmerk auf Fotorealismus gelegt. Ein besonders wichtiger Punkt stellt dabei die richtige Beleuchtung dar. Die zum Thema vorhandene Literatur wie beispielsweise das 2006 erschienene Buch „Licht Schatten Raum“ von Horst Sondermann oder auch „3ds Max – Visualisierung und Animation“ von Christian Immler, der seit über 10 Jahren zum Schwerpunkt CAD und Visualisierung publiziert und zudem als Dozent für 3D-Visualisierung im Fachbereich Architektur tätig ist, beschäftigen sich intensiv mit Lichtsetzung.

II. In welchen Phasen der Planung wird Architekturvisualisierung mittels 3D-Programmen angewandt?

Die Hauptanwendung von computergenerierten 3D-Modellen ist die Endpräsentation des Projekts. Doch auch während der Planung des Bauobjekts kommt sie zum Einsatz, wie etwa zur Absprache mit dem Kunden.

III. Wie schaut der Arbeitsablauf im Allgemeinen aus?

Der Arbeitsablauf in der Visualisierung von Gebäuden ist nicht standardisiert. So gibt es verschiedene Herangehensweisen um zu einem 3D-Modell des Bauwerks zu gelangen.

IV. Welche Render-Techniken kommen der Architekturvisualisierung zugute?

Raytracing, Radiosity und Image Based Rendering – dieses Verfahren findet besonders bei der Beleuchtung von Architekturmodellen mittels IBL Verwendung – eignen sich durch die hohe Annäherung an die Realität besonders gut für die fotorealistische Darstellung von Architekturmodellen. Auch Scanline Rendering kommt in Kombination von Radiosity für das Echtzeitrendering von Architekturmodellen zum Einsatz.

V. Von welchen Funktionen der 3D-Programme profitiert dieser Bereich besonders?

Der Bereich der Architekturvisualisierung profitiert von den vielfältigen Möglichkeiten der Lichtsetzung, die 3D-Software bietet. Auch die Fähigkeiten der Renderer kommen der Architekturvisualisierung zu gute.

Darüber hinaus hilft Zusatzsoftware wie RCP-Plugins und Pflanzengeneratoren dem Anwender bei der fotorealistischen Darstellung von Bauobjekten.

V|2 Überprüfung der Hypothesen

Der Datenaustausch zwischen den einzelnen verwendeten Programmen ist kompliziert.

Der Austausch von Daten zwischen verschiedener Software [CAAD-Software, CAD-Software, 3D-Software] birgt seinen Tücken, besonders wenn alte Dateiformate zum Austausch verwendet werden.

Doch gibt es auch spezielle Software, die eine relativ problemlose Konvertierung der Files ermöglicht. Der Nachteil ist, dass diese extra erworben werden muss, während die Verwendung von Quasistandard-Formaten zum Datenaustausch keine zusätzlichen Kosten verursacht.

Der Datenaustausch kann also relativ kompliziert sein, besonders bei großen komplexen Modellen, durch die Verwendung von Spezialsoftware kann er aber wesentlich vereinfacht werden.

Beleuchtung, Materialien und Kameraperspektiven sind für die fotorealistische Darstellung von Architekturmodellen besonders wichtig.

Die über Architekturvisualisierung verfügbare Literatur beschäftigt sich intensiv mit Beleuchtung und Materialien.

Es gibt keinen standardisierten Arbeitsablauf im Bereich der digitalen 3D-Architekturvisualisierung.

Es gibt sehr viele unterschiedliche Wege zu einem 3D-Architekturmodell. Wie es entsteht hängt beispielsweise von der verwendeten Software ab, aber auch davon ob das Modell vom Planer selber oder von einem darauf spezialisierten Experten erstellt wird. Auch in welcher Phase der Planung das Architekturmodell entsteht verändert den Arbeitsablauf.

Architekturvisualisierung mit 3D-Programmen wird hauptsächlich zur Präsentation des fertig geplanten Bauobjekts verwendet.

Fotorealistische Modelle werden meist zur Präsentation angewandt, doch auch einfachere Modelle wie Ideen- oder Massenmodelle werden vermehrt auch mit Hilfe von 3D-Programmen erstellt.

Oft wird auch nicht das fertig geplante Bauobjekt, sondern eine grobe Idee, noch bevor die Planung fertig abgeschlossen ist bzw. in manchen Fällen auch bevor die Planung richtig begonnen hat, aufwendig visualisiert.

Digitale 3D-Modelle können also in jedem Stadium der Planung Anwendung finden.

Digitale 3D-Modelle werden analoge Architekturmodelle vollständig ersetzen.

Es wird eine Koexistenz von digitalen und analogen 3D-Modellen geben. Viele Architekten bevorzugen analoge Modelle als Ideen- und Massenmodelle. Außerdem sind oft auch zur Schlusspräsentation von Projekten digitale und analoge Modelle gefordert.

Doch können auch analoge Miniaturmodelle aus am computergenerierten Modellen entstehen [Computergestützter Modellbau].

V|3 Reflexion und Ausblick

In Zukunft wird wohl auch der Bereich der Echtzeitgrafik in der Visualisierung von Bauobjekten eine größere Rolle spielen. Durch diese Technologie ist es dem Betrachter möglich schon vor dem Bau das Gebäude zu betreten. Dadurch werden auch Aspekte, die in den anderen Visualisierungsformen nicht visualisiert werden können, wie beispielsweise das Raumgefühl, dargestellt.

Ob es einem Architekturbüro aber möglich ist, diese aufwendigen Visualisierungen noch selbst zu generieren, ist fraglich. Wahrscheinlich wird das eher eigenen Firmen überlassen, wie es auch jetzt schon bei aufwendigen 3D-Modellen üblich ist.

Wie exakt ein Modell der Wirklichkeit entsprechen muss, hängt vom Einsatzgebiet des Modells bzw. auch davon ab wie weit das Projekt schon in der Planung fortgeschritten ist. Auch der Stil des Architekten spielt dabei eine Rolle, so verzichtet beispielsweise von Wolfgang Tschapeller bei seinen Modellen bewusst auf die fotorealistische Darstellung von Personen, sondern deutet diese nur schattenhaft an [siehe Abb. V-1].

Im Zuge meiner Suche nach Beispielen für Architekturvisualisierungen sind mir auch folgende Renderings aufgefallen.

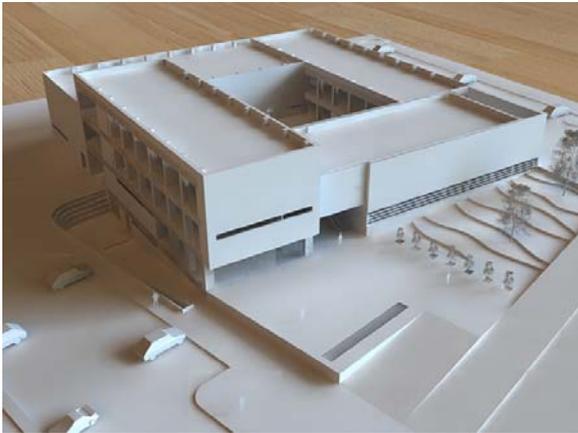


Abb. VI – 1 | gerendertes Modell

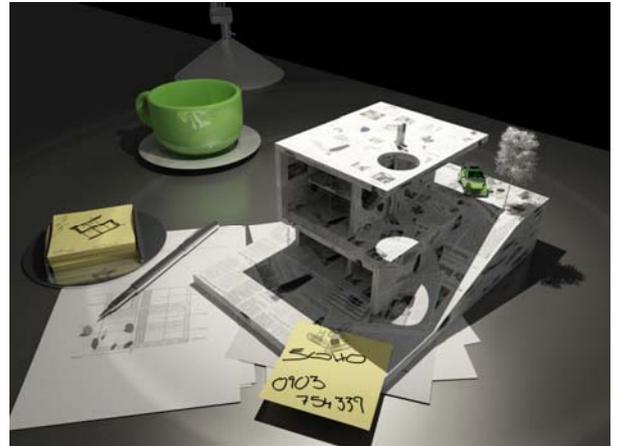


Abb. VI – 2 | Newspaper House

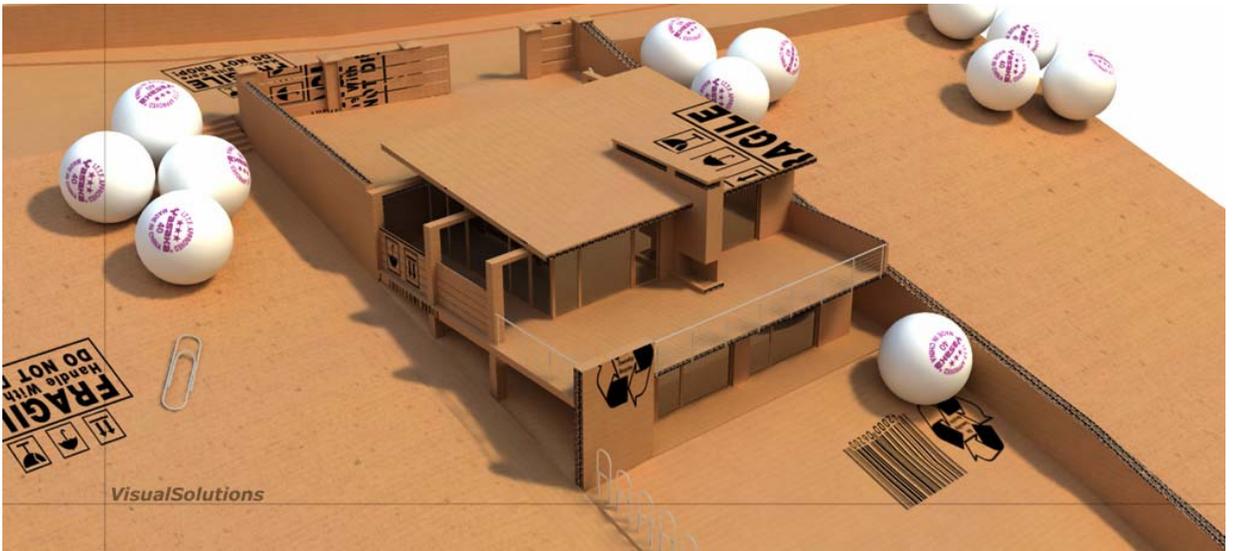


Abb. VI – 3 | Cardboard House

Im Prinzip sind das fotorealistische Renderings von analogen Architekturmodellen, die teilweise auch wirklich zur Präsentation der Projekte eingesetzt wurden. Das legt die Annahme nahe, dass es vielleicht schon eine Sättigung an realistischen Visualisierungen gibt.

Ein Bereich, der in dieser Diplomarbeit nicht behandelt wurde sind spezielle Render-Plug-Ins für CAAD Programme, wie Artlantis R, mit welchen ein fotorealistisches Rendern von CAAD Files möglich sein soll. Abb. VI-4 zeigt eine mit ArchiCAD und Artlantis R erstellte Innenraumvisualisierung.

So könnte CAAD-Software wirklich zu einem Komplettpaket für Architekten werden, da bei richtiger Anwendung das Zeichnen von Plänen, die Berechnung von Mengendaten etc. und auch die fotorealistische Visualisierung mit einer Anwendung erledigt werden kann.

Jedoch ist davon auszugehen, dass spezielle 3D-Software bessere Ergebnisse liefert als ein Plug-In.



Abb. VI-4 | Innenraumvisualisierung, modelliert mit Graphisoft ArchiCAD, gerendert mit Artlantis R

Ein weiterer interessanter Bereich, der in dieser Diplomarbeit nicht abgehandelt wurde ist der Einsatz von 3D-Software im Bereich der Visualisierung in der Landschaftsplanung und Landschaftsarchitektur. Besonders die ständige Weiterentwicklung von Pflanzengeneratoren könnte für dieses Gebiet interessant sein.

Die Visualisierung von Bauobjekten mit 3D-Programmen ist aus der Architektur nicht mehr wegzudenken und da sich die Software immer weiterentwickelt, werden sich auch die Möglichkeiten der fotorealistischen Darstellung weiter verbessern.

VI. Anhang

VI|1 Glossar

additive Farbmodelle

Ein Farbmodell, das auf der Addition von Lichtfarben basiert. Wenn die drei Grundlichtfarben Rot, Grün, Blau [RGB] im gleichen Anteil aufeinander treffen entsteht weiß, die Kombination von Rot und Grün ergibt Gelb. Ein Computermonitor arbeitet mit dem additiven Farbmodell.

[vgl. Demers|2001|S.94]

Alpha-Kanal

Ein Alpha Kanal ist ein Datenkanal [neben den 3 Farbkanälen, die die Farbintensität des Bildes bestimmen], der die Transparenz und Opazitäten der verschiedenen Pixel angibt.

area subdivision

Bei dieser 1969 von John E. Warnock veröffentlichten Methode zur Verdeckungsberechnung werden Bildausschnitte, beginnend mit einer Region, die die gesamte Szene beinhaltet, solange weiter unterteilt bis sie entweder einfach genug sind um beschrieben werden zu können, oder nur mehr 1 Pixel groß sind.

[vgl. http://wwwcs.uni-paderborn.de/cs/ag-domik/computergrafik/cg_skript/html/node96.htm, Abruf am 15.05. 2006]

Billboard Cloud

Eine Anordnung von mehreren texturierten Flächen durch welche ein Polygonmodell simuliert werden soll.

Crowd Simulation

= Gruppensimulation. Ist die Simulation des Verhaltens einer großen Gruppe von Personen. Dabei gibt es 2 Ansätze: Partikesysteme und Künstliche Intelligenz.

[vgl. http://de.wikipedia.org/wiki/Crowd_simulation]

depth sort

Ist ein von M. E. Newell, R. Newell and T. Sancha 1972 veröffentlichter Algorithmus zur Verdeckungsrechnung.

Zuerst werden alle Polygone nach ihrer z-Koordinate [beginnend mit dem am weitesten entfernten Polygon] sortiert, danach werden alle Mehrdeutigkeiten, die aus der Überlappung der Polygone hervorgehen aufgelöst.

Am Schluss werden die so berechneten Polygone vom entferntesten zum nächsten Polygon gezeichnet.

Ebenen

[engl.: layer] werden in vielen Programmen [z.B. Autodesk AutoCAD, Adobe Photoshop] verwendet, um die Übersichtlichkeit zu erhöhen. So können Elemente auf einer Ebene zusammengefasst und dann zusammen z.B. ein- und ausgeblendet werden.

Energieerhaltungssatz

Der Energieerhaltungssatz besagt, dass es unmöglich ist, innerhalb eines abgeschlossenen Systems Energie zu erzeugen oder zu vernichten. Im Bezug auf Lichtenergie bedeutet er, dass alles Licht, das eine Fläche empfängt und nicht absorbiert, sie wieder reflektieren muss. Weiters kann eine Fläche auch selbstleuchtend sein.

[vgl. <http://de.wikipedia.org/wiki/Energieerhaltungssatz>, Abruf am 13.05.2006]

High Dynamic Range Image

Eine digitale Bilddatei, deren Helligkeitswerte einen Dynamikumfang von mindestens vier Größenordnungen (1:10.000), eine relative Abstufung feiner als 1% und Abdeckung des ganzen sichtbaren Farbspektrums unterstützt.

[vgl. http://de.wikipedia.org/wiki/High_Dynamic_Range_Image, Abruf am 22.05.2006]

image space rendering

Ein Konzept zur Verdeckungsrechnung, bei dieser Technik werden die dreidimensionalen Objekte auf eine zweidimensionale Fläche projiziert, so geht viel dreidimensionale Informationen, die ansonsten für die Berechnung Schatten, Texturen und Anti-Aliasing verwendet werden könnte verloren, im Gegenzug sind diese Verfahren wenig rechenintensiv und sehr effizient.

[vgl. Kerlow|2004|S.162]

Kachelung

[engl.: tiling] Durch Kachelung kann eine Textur in alle Richtungen beliebig oft wiederholt werden. Diese Technik wird von fast jeder 3D-Software unterstützt. Der Vorteil an dieser Methode ist die geringe benötigte Rechenleistung.

[vgl. Kerlow|2004|S.241]

LOD

Bezeichnung für die unterschiedlichen Detailstufen, abhängig von der Anzahl der Polygone eines Computermodells.

L-Systems

Kurzform für Lindenmayer Systeme, ein in den 60er Jahren von Aristide Lindenmayer entwickelter mathematischer Algorithmus zur Berechnung des Wachstums von Pflanzen.

Lens Flare

Ein Lens Flare ist ein ringförmiger Schein, der durch die Brechung von Licht innerhalb der Kameralinse auftritt.

object space rendering

Methoden, die object space rendering zur Verdeckungsrechnung verwenden, berechnen die Daten in 3 Dimensionen, auf diese Weise werden auch Informationen gewonnen, die in weiterer Folge verwendet werden können um das Rendern von Schatten, Texturen und Anti-Aliasing zu verbessern.

[vgl. Kerlow|2004|S.162]

projektive Abbildung

= Projektion. In der darstellenden Geometrie eine Abbildung, die Punkte des dreidimensionalen Raumes auf Punkte einer gegebenen Ebene darstellt.

Passes

Im Prinzip Ebenen, die beim Rendern getrennt berechnet werden können. Beispielsweise eine Ebene [Pass] für Schatten [Shadow-Pass] und ein Pass für den Rest der Szene.

Puffer

In der Informatik wird ein Zwischenspeicher als Puffer [engl.: Buffer] bezeichnet.

ray tracing depth

Die Anzahl der Objekte, mit welchen ein Strahl in einer Szene maximal in Berührung kommen kann.

Rendering

= Bildsynthese, Erzeugung eines Rasterbildes aus einer dreidimensionalen Szene.

Shading

[von engl. to shade = schattieren] In der 3D-Computergrafik wird damit die Simulation der Oberfläche eines Objekts bezeichnet.

SketchUp

Eine relativ leicht erlernbare 3D-Software, die speziell für die Konzeptions- und Entwurfsphase von Designs entwickelt wurde.

[vgl. <http://de.sketchup.com/>, Abruf am 13.05.2005]

Spatial Textures

Diese Art von Textur verändert die Geometrie des Objekts.

[vgl. Kerlow|2004|S.250]

Normalvektoren

Normalvektoren sind Vektoren mit einer festgelegten Richtung, die sich an den Eckpunkten jedes Polygons eines Objekts befinden.

[vgl. Kerlow|2004|S.227]

Triangels

In der 3D-Computergrafik werden mathematisch beschriebenen Oberflächen, wie z.B. eine Kugel, durch Polygone oder Dreiecke [Triangels] angenähert.

[vgl. <http://de.wikipedia.org/wiki/3D-Computergrafik>, Abruf am 22.05.2006]

Visual Textures

Visual Textures beeinflussen im Gegensatz zu *Spatial Textures* die Geometrie des Körpers, auf den sie gemapped wurden, nicht. Zu den Visual Textures gehören Color Maps, Prozedurale Maps so wie Environment, Bump und Transparency Maps.

[vgl. Kerlow|2004|S.250]

VI|2 Literaturverzeichnis

VI|2.1 Buch | Zeitschrift

- [Ablan|2002] Dan Ablan, 2002, „[digital] Cinematography & Directing“, New Riders Publishing
- [AmSchö|2003] Wolfgang Amsoneit, Christoph Schöffel, 2003, „CAAD.PRAXIS, Computer Aided Architectural Design auf der Basis der HOAI“, Akademik Verlag
- [Debevec|2002] Paul Debevec, 2002, Computer Graphics and Applications, IEEE Volume 22, Issue 2, March, April, S.26-34,
- [Demers|2001] Owen Demers, 2001, „[digital] Texturing & Painting“, New Riders Publishing
- [Evers|1994] herausgegeben von Bernd Evers, 1994, „Architekturmodelle der Renaissance“, Prestel München New York
- [Ferschin|2001] Peter Ferschin, 2001, „Grundkurs Architektur und Darstellung – CAAD 272.043; Skriptum zur Vorlesung“
- [IEMAR|2003] IEMAR TU Wien, 2003, „Grundkurs Konstruktion und CAAD“
- [Immler|2004] Christian Immler, 2004, „3ds max 6 Visualisierung und Animation“, Markt und Technik
- [Kerlow|2004] Isaac V. Kerlow, 2004 „The Art of 3D Computer Animations and Effects“, John Wiley & Sons

- [Kranzer|2005] Jochen Kranzer, 2005 "Architekturvisualisierung mit Methoden der virtuellen Realität"
- [KuHa|2004] Antje Kunze, Jan Halatsch, 2004, "Amorphe Gebäude entwerfen: Der gläserne Reinraum", Digital Production (2004), Jänner, Februar, S. 40-45, ACT GmbH
- [KuHa|03|2005] Antje Kunze, Jan Halatsch, 2005, "Architektur Experten: CAAD - Programme", Digital Production (2005), Mai, Juni, S. 44-49, ACT GmbH
- [KuHa|06|2005] Antje Kunze, Jan Halatsch, 2005, "Von 2D zu 3D: Methoden und Lösungen", Digital Production (2005), November, Dezember, S. 53-58, ACT GmbH
- [KuHa|06|2005|2] Antje Kunze, Jan Halatsch, 2005, "Gärtnern leicht gemacht", Digital Production (2005), November, Dezember, S. 53-58, ACT GmbH
- [Mach|2000] Rüdiger Mach, 2000, „3D Visualisierung, Optimale Ergebnispräsentation mit AutoCAD und 3D Studio MAX“
- [MartPet|2002] Bob Martens, Herbert Peter, 2002, „ArchiCAD, Das Anwenderbuch für die Praxis“, SpringerWienNewYork
- [Schneider|2003] Thomas Schneider, 2003, "Architekturvisualisierung – Bauten optisch vorhersagen", Digital Production (2003), November, Dezember, S. 60-61, ACT GmbH
- [Seitz|2004] Stefan Seitz, 2004, "CAD Visualisierung mit Cinema 4D", Digital Production (2004), Mai, Juni, S. 144-149, ACT GmbH

- [Seitz|2006]* Stefan Seitz, 2006, "Architekturkonstruktion mit Cinema 4D", Digital Production (2006), März, April , S. 144-150, ACT GmbH
- [StreWei|1996]* Bernd Streich, Wolfgang Weisgerber,
„Computergestützter Architekturmodellbau“
- [Sonder|2006]* Horst Sondermann, 2006, "Licht Schatten Raum, Architekturvisualisierung mit Cinema 4D, SpringerWienNewYork
- [Umlauf|2004]* Eike Jens Umlauf, 2004, Diplomarbeit „Image-Based Rendering of Forests“

VI|2.2 Internet

- [*www|ambient*] <http://www.evermotion.org/vbulletin/showthread.php?t=41488&highlight=ambient+light> [Abruf am 01.08.2006]
- [*www|2Dcad*] <http://www.blien.de/ralf/cad/db/2dcadsys.htm> [Abruf am 07.06. 2006]
- [*www|RCP*] <http://www.archvision.com/> [Abruf am 27.06.2006]
- [*www|reflection*] http://www.cgarchitect.com/upclose/article19_TB.asp
- [*www|StruTrau*] Michael Strub, Stefan Traud, Diplomarbeit "Photon Mapping", 2004 | <http://www.fhbb.ch/informatik/bvmm/Projekte/0304%20Photon%20Mapping/Grundprinzip.html> [Abruf am 13.05.2006]
- [*www|sun*] <http://www.evermotion.org/vbulletin/showthread.php?t=54068&page=2> [Abruf am 04.08.2006]
- [*www|trans*] <http://www.evermotion.org/vbulletin/showthread.php?t=54014> [Abruf am 30.07.2006]
- [*www|wiki|catmull*] http://en.wikipedia.org/wiki/Edwin_Catmull [Abruf am 11.05.2006]
- [*www|wiki|global*] http://de.wikipedia.org/wiki/Globale_Beleuchtung [Abruf am 13.05.2006]
- [*www|wiki|kaustik*] http://de.wikipedia.org/wiki/Kaustik_%28Optik%29 [Abruf am 13.05.2006]

- [www|wiki|photon]* http://de.wikipedia.org/wiki/Photon_Mapping [Abruf am 13.05.2006]
- [www|wiki|qtv]* http://en.wikipedia.org/wiki/QuickTime_VR [Abruf am 24.06.2006]
- [www|wiki|radiosity]* <http://en.wikipedia.org/wiki/Radiosity> [Abruf am 11.05.2006]
- [www|wiki|radiosityII]* http://de.wikipedia.org/wiki/Radiosity_%28Computergrafik%29 [Abruf am 13.05.2006]
- [www|wiki|raytracing]* <http://de.wikipedia.org/wiki/Raytracing> [Abruf am 08.05.2006]
- [www|wiki|sketch]* <http://en.wikipedia.org/wiki/Sketchpad> [Abruf am 26.04.2006]
- [www|wiki|zbuffer]* <http://en.wikipedia.org/wiki/Z-buffering> [Abruf am 11.05.2006]

VI|3 Abkürzungsverzeichnis

CAD	Computer Aided Design Computer Aided Drafting
CAAD	Computer Aided Architectural Design
GDL	Geometric Description Language
IBL	Image Based Lighting
IMBR	Image Based Rendering
LOD	Level of Detail
QTVR	Quick Time Virtual Reality
RPC	Real Photorealistic Content

VI|4 Softwareverzeichnis

3D Studio Max	Autodesk	www.autodesk.com
Allplan	Nemetschek	www.nemetschek.de
ArchiCAD	Graphisoft	www.graphisoft.de
Artlantis R	Artlantis	www.artlantis.com
AutoCAD	Autodesk	www.autodesk.com
Brazil	SplutterFish	www.splutterfish.com
Cinema4D	Maxon	www.maxon.de
Deep Exploration	Right Hemisphere	www.righthemisphere.com
LightWave 3D	NewTek	www.newtek.com
Softimage XSI	Avid	www.softimage.com
Maya	Autodesk	www.autodesk.com
Mental Ray	Mental Images	www.mentalimages.com
Mental Ray	Mental Images	www.mentalimages.com
VectorWorks	Computerworks	www.computerworks.de
Nugraf Polytrans	Okino	www.okino.com
Photorealistic RenderMan	Pixar	renderman.pixar.com
Photoshop	Adobe	www.adobe.com
QuickTime VR	Apple	www.apple.com
SketchUp	SketchUp	de.sketchup.com
SpeedTree	IDV	www.speedtree.com
VRay	Chaos Group	www.chaosgroup.com
Xfrog	Greenworks	www.xfrogdownloads.com

VI|5 Abbildungsverzeichnis

Abb. III – 1 | Modell eines Hauses mit Garten, Theben, 2009-1998 v. Chr. [Quelle: „Architekturmodelle der Renaissance“, Prestel München New York]

Abb. III -2 | Holzmodell des Doms von Pavia, Cristoforo Rocchi und Giovan Pietro Gugazza, ca. 1490 [Quelle: „Architekturmodelle der Renaissance“, Prestel München New York, S.226]

Abb. IV-1 | GDL-Frau, GDL-Mann [Quelle: Bob Martens, Herbert Peter, 2002, ArchiCAD, Das Anwenderbuch für die Praxis“, SpringerWienNewYork, S51]

Abb. IV-2 | GDL- Objekt [Quelle: Eigenproduktion, 2006]

Abb. IV-3 | Export – Dialog in ArchiCAD 9 [Quelle: Eigenproduktion, 2006]

Abb. V-1 | Rendering, Museum Achen, Architekt Wolfgang Tschapeller [Quelle: freundlicherweise zur Verfügung gestellt von Isochrom|Arnold Faller⁴³, Armin Hess⁴⁴

Abb. V-2 | Originalfoto [Quelle: Thomas Schneider, 2003, „Architekturvisualisierung – Bauten optisch vorhersagen“, Digital Production (2003), November, Dezember, S. 61, ACT GmbH]

Abb. V-3 | Fotomontage mit gerendertem Haus [Quelle: Thomas Schneider, 2003, „Architekturvisualisierung – Bauten optisch vorhersagen“, Digital Production (2003), November, Dezember, S. 61, ACT GmbH]

⁴³ Dipl.Ing. Arnold Faller ist als Selbstständiger im Bereich der Architekturvisualisierung und Amination tätig und Univ.Ass. am Institut für Architekturwissenschaften - Digitale Architektur und Raumplanung der TU Wien.

⁴⁴ Dipl. Ing. SBA Armin Hess ist Mitbegründer von urban-filter [<http://www.urban-filter.com/>], einer Arbeitsgemeinschaft für Architektur, Design und urbane Strategien, und Univ. Lektor an der Univ. für Angewandte Kunst, Wien, Studio Prix.

Abb. V-4 | analoges Architekturmodell von Hans Stark, Schröder Haus in Utrecht [1924] von Gerrit Rietveld [Quelle: "Scale Models Houses of the 20th Century", Verlag Anton Pustet, S. 265]

Abb. V-5 | Screenshot, Architekturvisualisierung mit QuickTime VR [Quelle: http://www.swissinteractive.ch/html/visu_wohn_00index.htm, Abruf am 23.06.2006]

Abb. V-6 | Screenshot aus Unreal Tournament 2004 [Quelle: http://cerberus.gamershell.com/screenshots/178/6258_full.jpg, Abruf am 24.06.2006]

Abb. V-7 | analoges Ideenmodell für einen Zug mit beweglichen Modulen von oben [Quelle: freundlicherweise zur Verfügung gestellt von Christoph Lindner⁴⁵]

Abb. V-8 | fertig modellierter und gerenderter Zug, Innenansicht [Quelle: freundlicherweise zur Verfügung gestellt von Christoph Lindner]

Abb. V-9 | 3D-Modell, entstanden ohne genaue Pläne, Architekturbüro Eichinger oder Knechtl [Quelle: freundlicherweise zur Verfügung gestellt von Isochrom|Arnold Faller, Armin Hess]

Abb. V-10 | AutoCAD Plan, 3D Ansicht [Quelle: Christoph Lindner, 2006]

Abb. V-11 | gerenderter Schnitt, Architekt Wolfgang Tschapeller [Quelle: freundlicherweise zur Verfügung gestellt von Isochrom|Arnold Faller, Armin Hess]

Abb. V-12 | Rendering ohne Nachbearbeitung, Akron Art Museum, Architekt Coop Himmelblau [Quelle: freundlicherweise zur Verfügung gestellt von Isochrom|Arnold Faller, Armin Hess]

⁴⁵ Christop Lindner ist Architekturstudent im xxx. Semester.

Abb. V-13 | Rendering mit Nachbearbeitung, Akron Art Museum, Architekt Coop Himmelblau [Quelle: freundlicherweise zur Verfügung gestellt von Isochrom|Arnold Faller, Armin Hess]

Abb. V-14 | Rendering ohne Nachbearbeitung, BMW Welt, Architekt Coop Himmelblau [Quelle: freundlicherweise zur Verfügung gestellt von Isochrom|Arnold Faller, Armin Hess]

Abb. V-15 | Rendering mit Nachbearbeitung, BMW Welt, Architekt Coop Himmelblau [Quelle: freundlicherweise zur Verfügung gestellt von Isochrom|Arnold Faller, Armin Hess]

Abb. V-16 | Beispielrendering, Innenraumvisualisierung, Cinema 4D [Quelle: Stefan Seitz, 2004, "CAD Visualisierung mit Cinema 4D", Digital Production (2004), Mai, Juni, ACT GmbH, S. 144]

Abb. V-17 | Beispiele für Shader [Quelle: Owen Demers, 2001, „[digital] Texturing & Painting“, New Riders Publishing, S.200]

Abb. V-18 | Ziegelmauer, erstellt mit Hilfe von Kachelung [Quelle: Horst Sondermann, 2006, "Licht Schatten Raum, Architekturvisualisierung mit Cinema 4D, SpringerWienNewYork, S. 196]

Abb. V-19 | Lichtdurchlässigkeit realisiert durch Transparency Map [Quelle: <http://www.evermotion.org/vbulletin/attachment.php?attachmentid=3651&d=1153798906>]

Abb. V-20 | Reflexion mit Reflection Map und Bump Map [Quelle: <http://www.cgarchitect.com/upclose/TB/dec7.jpg>]

Abb. V-21 | Environment Map [Quelle: <http://www.artemis.pl/ppb/forty05render.jpg>]

Abb. V-22 | Bump Map für Beton [Quelle: Material für Tutorial, Stefan Seitz, 2004, „CAD Visualisierung mit Cinema 4D“, Digital Production (2004), Mai, Juni, ACT GmbH]

Abb. V-23 | Bump Map für Parkettboden [Quelle: Material für Tutorial, Stefan Seitz, 2004, „CAD Visualisierung mit Cinema 4D“, Digital Production (2004), Mai, Juni, ACT GmbH]

Abb. V-24 | Außenvisualisierung, Gras gerendert mit Displacement Map [Quelle: <http://www.elifserencioglu.com/cc/egew02.jpg>, Abruf am 01.08.2006]

Abb. V-25| Beleuchtung mit Ambient und Omni Lights [Quelle: http://www.pixelperfectgraphics.biz/cg_files/dirt-omnis.jpg, Abruf am 01.08.2006]

Abb. V-26 | Gebäudevisualisierung mit Beleuchtung durch Sonnenobjekt [Quelle: freundlicherweise zur Verfügung gestellt von Victor Hernandez⁴⁶]

Abb. V-27 | Innenraum – Sonne und diffuses Licht, realisiert durch Sonnenobjekt, Kameralicht, Flächenlicht und diffusen Schatten[Horst Sondermann, 2006, „Licht Schatten Raum, Architekturvisualisierung mit Cinema 4D, SpringerWienNewYork, S.82]

Abb. V-28 | Gebäudevisualisierung mit Volume Lights bei Nacht [Quelle: <http://img113.exs.cx/img113/1482/withoutanyposting1ky.jpg> ,Abruf am 01.08.2006]

Abb. V-29 | Visualisierung mit farbigem Licht, Modell für den CGarchitecture AVC 2006 Wettbewerb [Quelle: <http://static.highend3d.com/galleryimages/706/screen.jpg>, Abruf am 26.06.2006]

⁴⁶ Victor Hernandez hat in Mexiko Architektur studiert und am UPC [Universitat Politècnica de Catalunya] in Barcelona den Masters in computer science applied to architecture erlangt, er arbeitet zur Zeit in der Architekturabteilung von TEC4, einem Planungsbüro in Barcelona.

Abb. V-30 | oben Scheinwerfer mit scharfem Lichtkegel und wenig Fall-Off, unten weicher Lichtkegel und hoher Fall-Off Wert [Quelle: Isaac V. Kerlow, 2004 „The Art of 3D Computer Animations and Effects“, John Wiley & Sons ,S.210]

Abb. V-31 | Architekturvisualisierung mit Lens Flare [Quelle:
<http://www.evermotion.org/vbulletin/attachment.php?attachmentid=3225>]

Abb. V-32 | Lens Flare, eingesetzt um Lichtquellen sichtbar zu machen [Quelle:
<http://www.soulpix.de/content/gallery/character/bot.jpg>, Abruf am 07.06.2006]

Abb. V-33 | Visualisierung ohne Schatten [Quelle: freundlicherweise zur Verfügung gestellt von Victor Hernandez]

Abb. V-34 | Visualisierung mit Schatten [Quelle: freundlicherweise zur Verfügung gestellt von Victor Hernandez]

Abb. V-35 | Rendering mit IBL, Musem Lyon, Architekt Coop Himmelblau [Quelle: freundlicherweise zur Verfügung gestellt von Isochrom|Arnold Faller, Armin Hess]

Abb. V-36 | Visualisierung ohne Unschärfe

Abb. V-37 | Visualisierung mit Unschärfe

Abb. V-38 | Originalfoto [Quelle: freundlicherweise zur Verfügung gestellt von Christoph Lindner]

Abb. V-39 | Rendering [Quelle: freundlicherweise zur Verfügung gestellt von Christoph Lindner, 2006]

Abb. V-40 | nachbearbeitetes Rendering mit fotomontierter Person [Quelle: freundlicherweise zur Verfügung gestellt von Christoph Lindner, 2006]

Abb. V-41 | Visualisierung Autosalon Genf BMW mit RCP-Personen [Quelle: freundlicherweise zur Verfügung gestellt von Arnold Faller]

Abb. V-42 | 3D-Modell Barcelona Chair von Mies van der Rohe,

www.turbosquid.com [Quelle:

http://files.turbosquid.com/Preview/Content_on_4_1_2002_16_18_25/Barcelona_Chair.jpgA022AA70-648A-44C8-917F467CC3842B5C.jpgLarge.jpg, Abruf am 29.06.2006]

Abb. V-43 | Entwurf Büro Rockstar Games mit händisch modellierten Pflanzen,

Detailansicht, [Quelle: freundlicherweise zur Verfügung gestellt von Arnold Faller]

Abb. V-44 | Entwurf Büro Rockstar Games mit händisch modellierten Pflanzen,

[Quelle: freundlicherweise zur Verfügung gestellt von Arnold Faller]

Abb. V-45 | Pflanzen generiert mit XFrog

[Quelle:

<http://www.xfrogdownloads.com/greenwebNew/gallery/images/Architecture/achitecture12.jpg>, Abruf am 08.05.2006]

Abb. V-46 | Plant Tuner von natFX für Maya. [Quelle: Eigenproduktion 2006]

Abb. V-47 | Raytracing Funktionsweise. [Quelle: http://www.3dmax-tutorials.com/graphics/il_radiosity_raytrace.jpg, Abruf am 15.05.2006]

http://www.3dmax-tutorials.com/graphics/il_radiosity_raytrace.jpg, Abruf am 15.05.2006]

Abb. V-48 | Radiosity Rechengitter Szene [Quelle:

<http://accad.osu.edu/~waynec/history/images/radiosity.jpg>, Abruf am 15.05.2006]

Abb. V-49 | mittels Radiosity gerenderte Szene [Quelle:

<http://accad.osu.edu/~waynec/history/images/radiosity.jpg>, Abruf am 15.05.2006]

Abb. V-50 | Color Bleeding [Quelle: [http://de.wikipedia.org/wiki/Bild:Radiosity-](http://de.wikipedia.org/wiki/Bild:Radiosity-yes.jpg)

[yes.jpg](http://de.wikipedia.org/wiki/Bild:Radiosity-yes.jpg), Abruf am 08.05.2006]

Abb. V-51 | Kaustik Photonen [Quelle:

http://www.fhbb.ch/informatik/bvmm/Projekte/0304%20Photon%20Mapping/CausticMap_452.jpg, Abruf am 13.05.2006]

Abb. V-52 | Globale Photonen [Quelle:

http://www.fhbb.ch/informatik/bvmm/Projekte/0304%20Photon%20Mapping/GlobalMap_452.jpg, Abruf am 13.05.2006]

Abb. V-53 | Visualisierung gerendert mit V-Ray [Quelle:

<http://www.chaosgroup.com/galleryimages/mrb2.jpg>, Abruf am 24.06.2006]

Abb. V-54 | Visualisierung Berlin-Schönefeld, gerendert mit Mental Ray, Architekten Gerkan Marg und Partner [Quelle:

http://www.mentalimages.com/4_4_architecture/images/big/flughafen_berlin/_001.jpg, Abruf am 24.06.2006]

Abb. VI-1 | gerendertes Modell [Quelle:

<http://www.evermotion.org/vbulletin/attachment.php?attachmentid=3686&d=1153895112>, Abruf am 06.08.2006]

Abb. VI-2 | Newspaper House [Quelle:

<http://www.evermotion.org/vbulletin/attachment.php?attachmentid=3522&d=1153206508>, Abruf am 01.08.2006]

Abb. VI-3 | Cardboard House [Quelle:

<http://www.evermotion.org/vbulletin/attachment.php?attachmentid=3717&d=1153957150>, Abruf am 26.07.2006]

Abb. VI-4 | Innenraumvisualisierung, modelliert mit Graphisoft ArchiCAD, gerendert mit Artlantis R

[Quelle:

<http://www.artlantis.com/community/galerie/index.php?pages=3&render=1>, Abruf am 29.06.2006]