

Reliefperspektivische Modelle aus dem 3D-Drucker

Daniel Lordick

Institut für Geometrie
Technische Universität Dresden
E-Mail: lordick@math.tu-dresden.de

Die Herstellung reliefperspektivischer Modelle mit traditionellen Mitteln ist im Allgemeinen konstruktiv und handwerklich anspruchsvoll. Hier wird im Detail vorgeführt, wie man einen 3D Modeller (Cinema4D) mit einem Rapid-Prototyping-System kombinieren kann, um relativ unkompliziert zum Ziel zu gelangen. Einige Beispiele beleuchten die kulturelle Bedeutung der Reliefperspektive.

1 Einleitung

Seit der Renaissance werden in Reliefs Architekturelemente und andere Objekte perspektivisch verzerrt dargestellt. Das Relief wird dabei so auf einen Augpunkt hin konstruiert, dass die Objekte aus selbigem betrachtet unverzerrt erscheinen. Der Eindruck entsteht zum Teil dadurch, dass der Betrachter aufgrund seiner Seherfahrung die wahre Gestalt der Objekte in das Relief hineininterpretiert.

Geometrische Grundlage der Reliefkonstruktion ist die sogenannte Reliefperspektive, die seit Einführung der projektiven Geometrie im 19. Jahrhundert streng wissenschaftlich gefasst werden kann [6, 9 u.a.]. Es handelt sich um eine perspektive Kollineation (Zentralkollineation) im Raum mit speziellen Annahmen, die durch die Anwendung motiviert sind.

Ein frühes Beispiel ist ein bronzenes Altarrelief von Donatello in Padua (S. Antonio, ca. 1447). Die Darbietung der Hostie an einen Maulesel findet vor drei Tonnengewölben statt, die das Geschehen thematisch gliedern. Die Architektur ist im schmalen Bildraum so kunstvoll wiedergegeben, dass man meint, man blicke schräg von unten in einen großzügigen Altarraum und durch die gestaffelten Fenster weiter hinaus ins Freie (Abb. 1).



Abb. 1: Altarrelief von Donatello (S. Antonio, Padua, ca. 1447)

Im großen Maßstab nutzt Bramante das Phänomen 1483-86 in Mailand, um bei der Erweiterung der Kirche S. Maria presso S. Satiro einen Chorraum hinter dem Altar vorzutauschen, wo wegen einer vorbeiführenden Straße keiner sein kann [3, S. 53-56]. Die gesamte Architektur des Chores ist streng gesetzmäßig in ein Terrakottarelief



Abb. 2: Bramante, Scheinchor in Mailand (1483-86)



Abb. 3: Bramante, Scheinchor links im Bild

von kaum zwei Metern Tiefe zusammengestaucht. Steht man im Hauptschiff der Kirche, so ist der Raumeindruck vollkommen (Abb. 2). Nur aus dem Querschiff schauend kann man das Ausmaß der Täuschung erkennen (Abb. 3).

Bemerkenswert ist hier: Bramante möchte durch den Scheinchor die Proportionen des Kircheninnenraumes nach einer Idealvorstellung harmonisch ausbalancieren. Dabei nimmt er in Kauf, dass ebendiese Proportionen im Chor durch das Relief zerstört werden. Schließlich ist die zugrundeliegende perspektive Kollineation nicht teilverhältnistreu. Bramante stellt also die optische Wirkung über die messbare Wirklichkeit.

Auch in Bühnenbildern geht es mehr um Wirkung als um Wirklichkeit. Soll etwa ein großer Raum auf einer kleinen Bühne dargestellt werden, so bedient sich die sogenannte Theaterperspektive [2] der Reliefperspektive und passt sie den Erfordernissen des Schauspiels an. Das lässt sich bereits bei Palladio beobachten. In seinem Teatro Olimpico (Vicenza, ca. 1585) wird der Blick der Zuschauer in drei Straßenfluchten gelenkt, die reich detailliert und in Holz ausgeführt einen urbanen Raum vorstellen (Abb. 4).

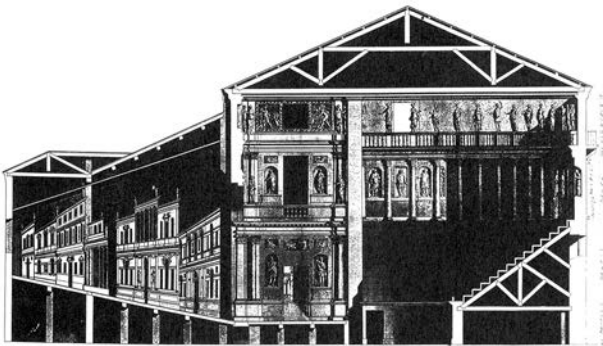


Abb. 4: Palladio, Teatro Olimpico (Vicenza, ca. 1585)

Ein brillantes Beispiel für eine fast makellose Reliefperspektive um ihrer selbst willen ist die Kolonnade des Borromini im Palazzo Spada (Rom, ca. 1635). Lediglich die Säulen sind rund statt elliptisch verzerrt. Michael Ende war von der begehbaren Kolonnade so beeindruckt, dass er sie mehrfach literarisch verarbeitet hat. Im Kern kreisen seine metaphysischen Beschreibungen um das vermeintlich gestörte Raum-Zeit-Kontinuum in der Kolonnade. Ein hölzernes Modell der Kolonnade im Maßstab 1:1 war auf der 42. Biennale in Venedig ausgestellt (Abb. 5) [5].

In einer aktuellen Inszenierung von »Alice im Wunderland« im Schauspielhaus Dresden (Regie: Bernarda Horres, Bühne: Andreas Jander, Abb.6) unterstützt ein trichterförmig gestaltetes Bühnenbild Wachsen und Schrumpfen der Alice. Weitere Beispiele für die Anwendung der Reliefperspektive sind die Fassade des Kinos Pathé de Munt in Amsterdam (Büro G. J. Van Delft, Abb.7) und die Reliefmodelle »Atlantis« (1989) und »The Security



Abb. 5: Holzmodell der Kolonnade des Borromini, Venedig 1986
690 x 890 x 500 cm

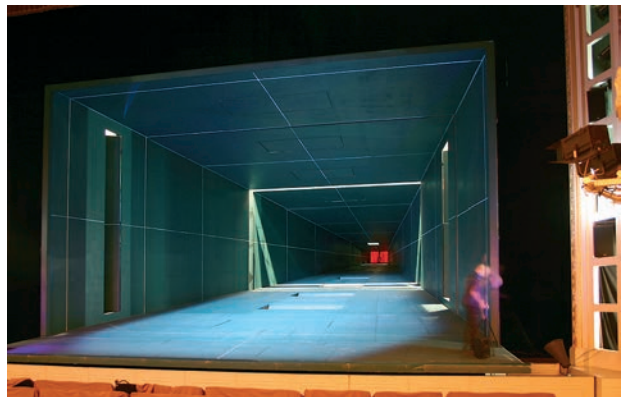


Abb. 6: Bühne zu »Alice im Wunderland« während des Aufbaus (Dresden 2004)

Council« (1989, Abb. 8) des englischen Künstlerduos Langlands & Bell [4].

2 Reliefperspektivische Modelle aus Dresden

Anwender der Reliefperspektive stützen sich gelegentlich auf nur rudimentäre geometrische Kenntnisse. An dieser Stelle setzt Ludwig Burmester 1883 an und veröffentlicht am Königlichen Polytechnikum Dresden eine Schrift speziell für



Abb. 7: Kino Pathé de Munt in Amsterdam

Kunstakademien und Kunstgewerbeschulen [1]. Burmesters Verdienst ist eine didaktisch ambitionierte Präsentation aller Tricks und Kniffe, mit denen man Reliefmodelle so elegant und präzise wie möglich herstellen kann. Andererseits weist er aber auch darauf hin, dass ein künstlerisch wertvolles Relief gelegentlich die Modifikation – aber nicht die Missachtung – der geometrischen Regeln erfordert.

Herzstück seiner Veröffentlichung sind drei reliefperspektivische Modelle aus Gips (Abb. 9). Die ausgesprochen ästhetischen Modelle befinden sich in hölzernen Guckkästen. Bei zwei Modellen kann man die Rückwand des Kastens wegklappen, um die perspektivische Verzerrung quasi von der »Hinterbühne« aus zu betrachten. Für die Herstellung der Urmodelle bittet Burmester die Direktion um einen Modellbaukostenzuschuss von 100 Mark. (Zum Vergleich: ein Professor verdient zu jener Zeit ca. 500 Mark monatlich.) Burmester, selbst gelernter Feinmechaniker, resümiert über den Bau der Modelle: »Die mannigfaltigen Schwierigkeiten, welche bei der hier kurz beschriebenen praktischen Herstellung der beiden Reliefmodelle der Halle und der Basilika auftraten, kann nur der Eingeweihte ermessen; und die Besiegung dieser Schwierigkeiten erfordert Geschicklichkeit und

Geduld im allerhöchsten Masse.« [1, S. 27]

An der Technischen Universität Dresden existiert leider keines dieser in Kleinserie produzierten Meisterwerke mehr. Es muss davon ausgegangen werden, dass eventuell noch vorhandene Exemplare im Februar 1945 mit den übrigen Teilen der Modellsammlung durch Bomben vernichtet wurden. An der TU Wien sind Kopien der drei Modelle in mehr oder weniger gutem Zustand erhalten.

Heute beherbergt das Institut für Geometrie der TU Dresden wieder eine beachtliche Sammlung mathematischer Modelle. Ihr Kern geht auf Rudolf Boreis zurück, der in seiner Amtszeit (1957-66) größere Mittel investierte. Der Wunsch, die Modelle Burmesters in der neuen Sammlung an ihrem Ursprungsort zu präsentieren, führte zu Überlegungen, wie man die Modelle nachbauen könne.

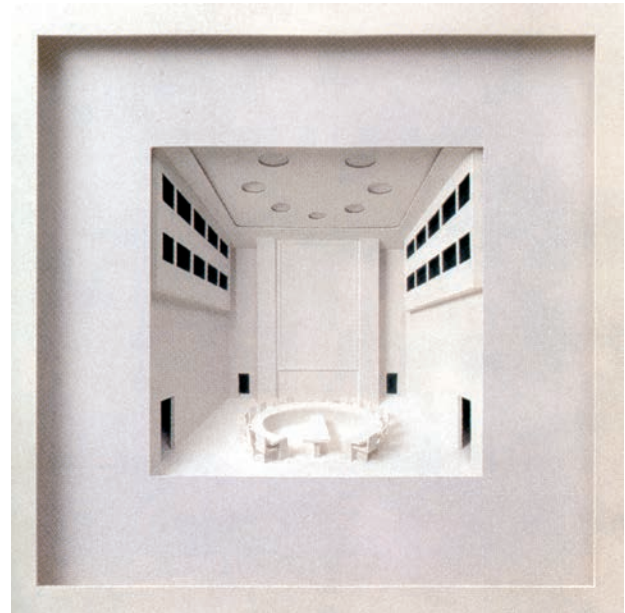
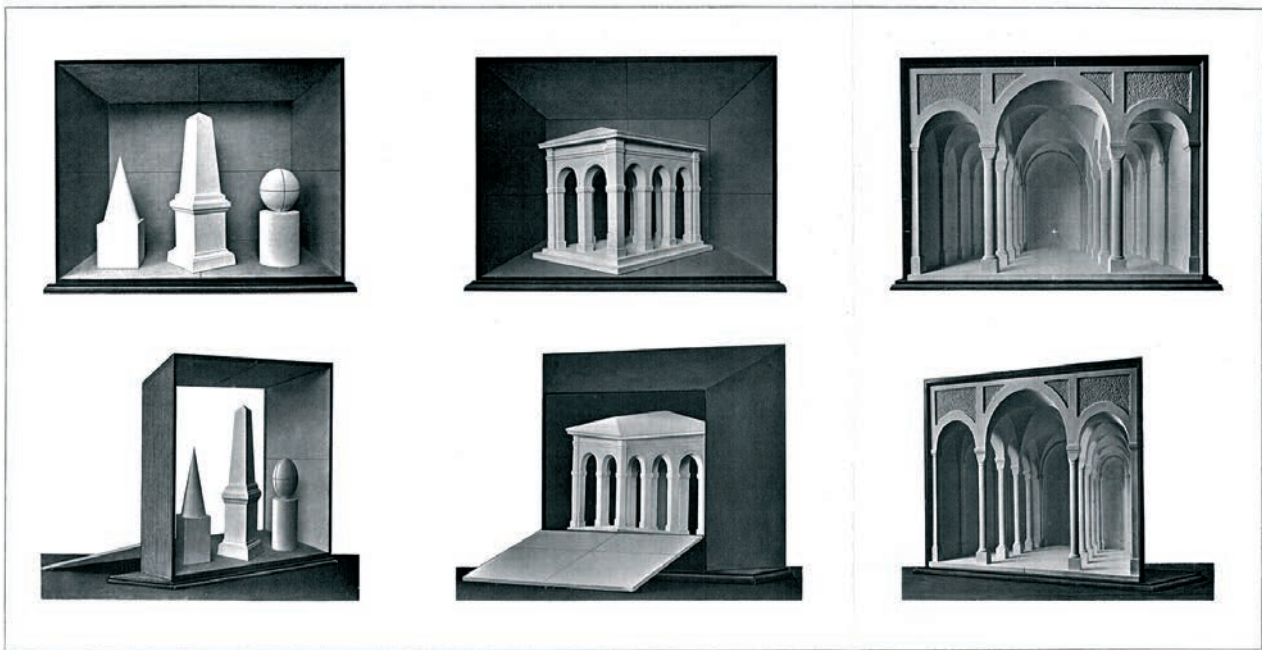


Abb. 8: Langlands & Bell, »The Security Council«, 1989. Vitrine, Holz, Glas, Farbe 75 x 75 x 20

Der traditionelle Modellbau an der mathematischen Fakultät ist aus finanziellen und personellen Gründen fast zum Erliegen gekommen. Dagegen erscheint es zeitgemäß, den Bau der Modelle im virtuellen Raum eines 3D Modellers durchzuführen. Anschließend kann das Modell dann mit einem Rapid Prototyping System ausgegeben werden.

Der direkte Bau eines perspektivischen Reliefs ist aufgrund der unregelmäßigen Winkel und Längen auch im Computer lästig. Wesentlich sinnvoller ist, die darzustellenden Objekte in ihrer wahren Gestalt zu konstruieren, weil man dann alle Vorzüge des jeweiligen CAD-Systems ausschöpfen kann. Erst im zweiten Schritt werden die Objekte so verzerrt, dass sie den Gesetzen der Reliefperspektive gehorchen. Das geschieht durch Anwendung der Abbildungsgleichung auf die Punktkoordinaten der Objekte. Man benötigt also eine Software, die Raumtransformationen über Formeln unterstützt.



Nr. 1. Typische Körper.

Nr. 2. Bogenhalle.

Nr. 3. Romanische Basilika.

Abb. 9: Reliefmodelle des Ludwig Burmester, Dresden 1883, Gips, Holz, 50 x 34 x 15 cm

Die Wahl fällt auf Cinema4D von Maxon. Als Rapid Prototyping System kommt der 3D-Drucker ZTM310 von Z-Corporation zum Einsatz (Abb. 10).



Abb. 10: 3D-Drucker ZTM310 von Z-Corporation, 74 x 81 x 109 cm

3 Abbildungsgleichungen der Reliefperspektive

Für die Reliefperspektive denken wir uns eine perspektive Kollineation im Raum mit *Kollineationszentrum* O (Augpunkt) und vertikaler *Fixpunktebene* π (*Spurebene*). Der nicht verschwindende Abstand von O und π ist die *Distanz* d. Die Reliefperspektive weist jedem Punkt im Urraum umkehrbar eindeutig einen Reliefpunkt zu und ist somit *bijektiv*.

Im Hinblick auf den idealen Betrachter des Reliefs, der mit einem Auge aus O schaut und π vor sich hat, bilden wir nur jenen Halbraum H^3 ab, der die

Randebene π_0 durch O und parallel zu π besitzt und π enthält. Üblicherweise wird das *charakteristische Doppelverhältnis* der Reliefperspektive durch die Wahl der *Fluchtebene* π_u^r festgelegt (das hochgestellte r bezeichnet das *Reliefbild* bzw. *Relief* eines Objektes). Die Fluchtebene ist das Relief der zu H^3 hinzugenommenen *Fernebene* π_u . Weil sich π_u und π in einer Ferngeraden schneiden, ist π_u^r parallel zu π . Damit dem Betrachter das Relief eines Gegenstandes im Vergleich zum Gegenstand nicht umgestülpt erscheint, muss π_u^r von O aus gesehen hinter π liegen. Der Abstand von π und π_u^r heißt *Relieftiefe* t. Das Relief von H^3 befindet sich vollständig zwischen π_0 und π_u^r . Wenn t verschwindet, liegt an Stelle eines Reliefs eine gewöhnliche Zentralperspektive vor. Die Normale zu π durch O heißt *Hauptsehstrahl*, ihr Schnittpunkt mit π *Hauptpunkt* H und mit π_u^r *Hauptfluchtpunkt* H_u^r .

Aus der Konstruktion des Reliefs P^r eines allgemeinen Punktes P lässt sich die Abbildungsgleichung leicht ableiten. Dazu führen wir ein möglichst geschickt gewähltes Koordinatensystem ein: Der Ursprung liegt in O, damit die Ortsvektoren von P und P^r linear abhängig sind, also

$$\vec{p}^r = \lambda \vec{p}.$$

Das illustriert die Faserung des Raumes mit Sehstrahlen (*Kollineationsstrahlen*).

In Cinema4D kommt ein kartesisches Linkskoordinatensystem mit vertikaler y-Achse zum Einsatz, weshalb wir die x-Achse horizontal in π_0 wählen und die y-Achse nach oben ausrichten. Die z-Achse fällt in den Hauptsehstrahl (Abb. 11).

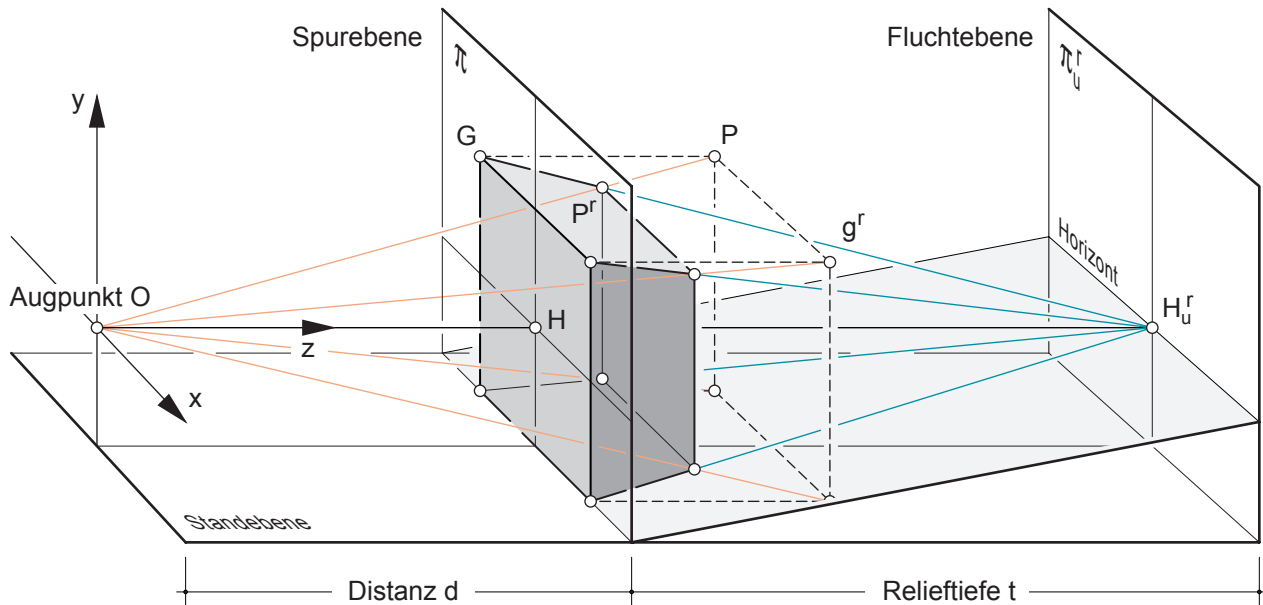


Abb. 11: Schrägriss der Reliefperspektive eines Würfels

Die Standardkonstruktion von P^r benutzt eine Hilfsgerade g normal zu π durch P . Das Relief g^r von g enthält den Fixpunkt $G = g \cap \pi$ und fluchtet zu H_u^r . Der Schnitt von g^r mit dem Sehstrahl durch P ist P^r . In der Ebene durch Hauptsehstrahl und P entsteht bei der Konstruktion eine Strahlensatzfigur mit dem Zentrum P^r , aus der ohne Rechnung folgt (Abb. 12):

$$\frac{|p - p^r|}{z_p - d} = \frac{|p^r|}{d + t}$$

wobei z_p der Abstand von P zu π_0 ist. Somit erhält man

$$\lambda = \frac{d + t}{z_p + t}$$

Aus den Koordinaten x_p, y_p, z_p von P berechnet man die Koordinaten von P^r :

$$x_p^r = \frac{d + t}{z_p + t} x_p,$$

$$y_p^r = \frac{d + t}{z_p + t} y_p,$$

$$z_p^r = \frac{d + t}{z_p + t} z_p$$

Die Ursprünge außerhalb des betrachteten Halbraums mit dem Abstand t von π_0 erfüllen die sogenannte *Verschwindungsebene* der Reliefperspektive. Ihre Bilder sind die Fernpunkte des Reliefs. Das macht sich im Verschwinden des Nenners für $z_p = -t$ bemerkbar. Die Formeln behalten für $d=0$ bzw. $t=0$ ihre Gültigkeit. Auch für $d = 0, t \neq 0$ (*Elation*) erhält man ein Relief.

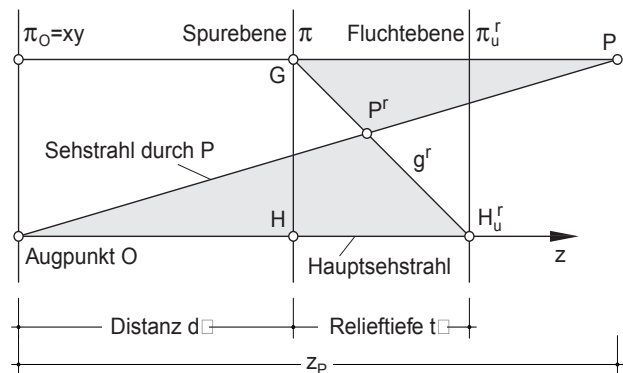


Abb. 12: Sehebene durch P und H

4 Implementierung in Cinema4D

Cinema4D ist ein Programmpaket zum Modellieren, Texturieren, Animieren und Rendern. Für die Verformung von Objekten bietet Cinema4D eine Reihe von Funktionen, die ihrerseits als Deformations-Objekte in eine Szenerie eingefügt werden können. Unsere Zwecke erfüllt das *Formel-Objekt*, weil es die Eingabe mathematischer Ausdrücke zulässt. Alle Objekte werden im *Objekt-Manager* verwaltet. Um ein *Deformations-Objekt* anzuwenden, positioniert man es im *Objekt-Manager* mit dem zu verformenden Objekt in einer *Objektgruppe* und auf einer gemeinsamen *Hierarchie-Ebene*.

Zur Demonstration der Reliefperspektive platzieren wir einen Würfel der Kantenlänge 200 (Einheiten) und verschieben ihn mithilfe des *Koordinaten-Manager* auf die Position $x, y, z = 0, 0, 400$. Anschließend erzeugen wir ein *Formel-Objekt* (Abb. 13). Seine Position entspricht dem Augpunkt der Reliefperspektive. Es verbleibt vorerst im Ursprung des *Welt-Koordinatensystems*. Im *Objekt-Manager* aktivieren wir Würfel und *Formel-Objekt* und wählen »Objekt / Objekte gruppieren«.

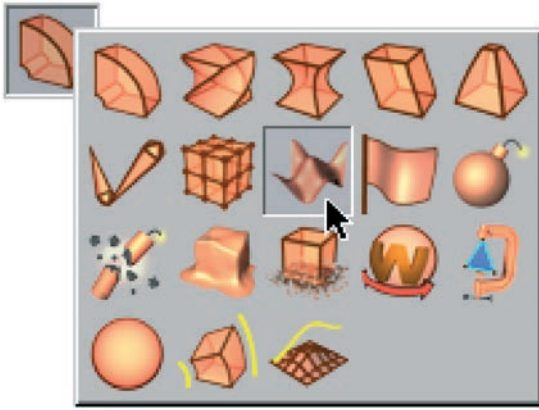


Abb. 13: Das Formel-Deformations-Objekt in Cinema4D

Die Eigenschaften der Objekte regelt *der Attribute-Manager*. Ist das Formel-Objekt ausgewählt, so schaltet man hier den *Effekt* auf »manuell« und hat anschließend Zugriff auf die einzelnen Koordinaten des Zielobjektes. Die Bezeichnung $X(x,y,z,t)$, etwa für die x-Koordinate, signalisiert, dass jeder Punkt koordinatenweise in Abhängigkeit von den Ausgangskordinaten transformiert werden kann. Zusätzlich steht der Parameter Zeit (t) zur Verfügung. Über diesen lässt sich die Formel automatisch animieren.

Wir wählen die Werte $d=300$ und $t=200$ und geben in der Zeile » $X(x,y,z,t)$ « die Formel für x_p^r in einer etwa aus *Maple* geläufigen Diktion ein: $500/(z+200)*x$ (» x « bedeutet x_p und » z « z_p). Analog verfahren wir in den beiden anderen Zeilen.

Jetzt wird folgender Widerspruch sichtbar: Der Würfel sollte nach unseren numerischen Annahmen mit der dem Betrachter zugewandten Facette in der Fixpunktebene liegen. Sein Relief wird aber deutlich jenseits seiner ursprünglichen Lage abgebildet und eine Verzerrung ist ebenfalls kaum feststellbar. Verantwortlich hierfür ist der Parameter *Größe* im Attribute-Manager, der die Skalierung des Formel-Objektes regelt. Um zu gewährleisten, dass die Angaben von d und t absolut übernommen werden, darf das Formel-Objekt in

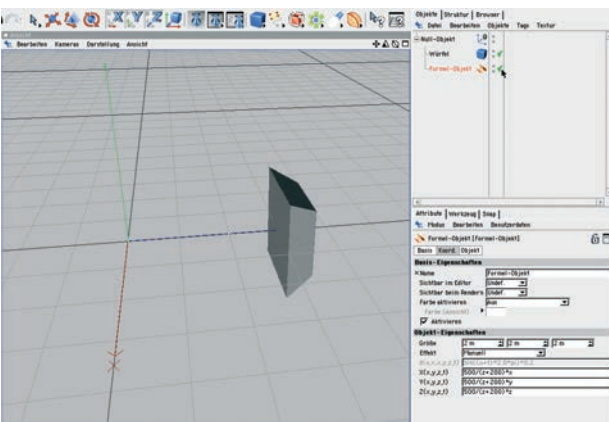


Abb. 14: Reliefperspektive eines Würfels in Cinema4D

keiner Richtung skalieren. Das erreicht man mit dem Wert 2 in allen drei Feldern, weil so das Formel-Objekt in jede Achsen-parallele Richtung mit dem Faktor 1 skaliert (Abb. 14).

Endlich ist das Relief des Würfels klar erkennbar und die dem Betrachter zugewandte Facette bleibt fix. Davon kann man sich durch An- und Abschalten des Formel-Objektes mithilfe des grünen Häkchens im Objekt-Manager überzeugen. Außerdem ist es ein Leichtes, etwa durch Drehen des Würfels über den Koordinaten-Manager, Varianten des Reliefs zu erstellen (Abb. 15). Wünschenswert erscheint aber außerdem die bequeme Regelung der Parameter d und t , um freier experimentieren zu können.

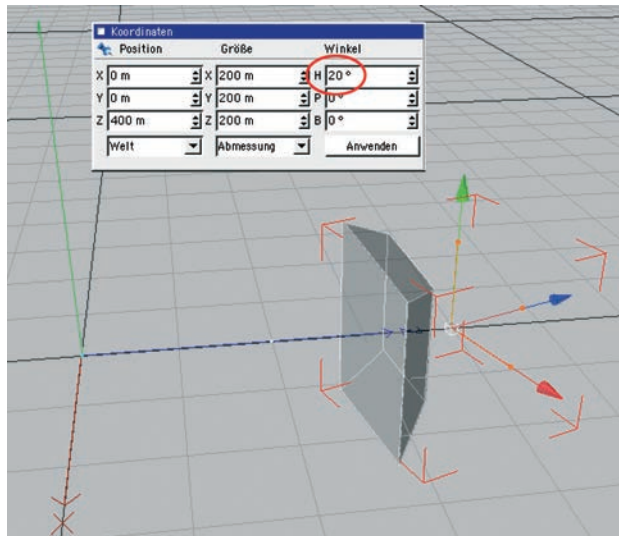


Abb. 15: Drehung des Würfels unter der Reliefperspektive

5 Regelung des Reliefs mit geeigneten Parametern

Cinema4D bietet mit dem *XPresso-Editor* eine Art Baukasten für einfache Programmieraufgaben. Auf grafischem Wege lassen sich Funktionen und Parameter koppeln und damit beliebige Objekte steuern. In unserem Fall werden wir eine Schaltung generieren, die unsere Reliefperspektive über Distanz und Relieftiefe regelt.

Zuerst aktivieren wir das Formel-Objekt im Objekt-Manager und erzeugen zwei Eingabefelder für d und t . Ein derartiges Eingabefeld erhält man im Attribute-Manager über »Benutzerdaten / Benutzerdaten hinzufügen ...«. Als Werte im zugehörigen Dialogfenster setzen wir: *Name* = »Distanz d «, *Daten-Typ* = »Fließkomma«, *Interface* = »Fließkomma«, *Einheit* = »Real«. Damit keine negativen Werte für d auftreten, setzen wir ein Häkchen bei *Min* und tragen eine »0« ein. Bei *Max* erhöhen wir auf »1000000« und *Schritt* bleibt auf »1« (Abb. 16). Nach »Ok« erscheint das Eingabefeld unter *Benutzerdaten* im Attribute-Manager.

Mit den selben Werten erzeugen wir ein zweites



Abb. 16: Benutzerdaten hinzufügen

Eingabefeld, nur lautet der Name diesmal »Relieftiefe t«. Eventuell nötige Korrekturen führt man durch, indem man bei gedrückter Steuerungstaste auf den Namen des Eingabefeldes klickt.

Als Nächstes benötigt man einen *XPRESSO-Expression-Tag*. Diese Funktion findet man im Objekt-Manager im Menü »Datei / Neue Expression / XPRESSO Expression«. Die eigentliche Schaltung wird dann im zugehörigen XPRESSO-Editor erzeugt, der sich automatisch öffnet. Der Editor ist eine gerasterte Fläche, auf der die *Nodes* (engl. Knoten) – die Bausteine der Schaltung – arrangiert und verknüpft werden.

Der Datenfluss erfolgt diagrammartig von links nach rechts. Als Startpunkt der Schaltung benötigen wir die Eingabewerte aus den Benutzerdatenfeldern »Distanz d« und »Relieftiefe t«. Man greift also das Formelobjekt aus dem Objekt-Manager und zieht es in den XPRESSO-Editor auf die linke Seite. Dort erscheint nun ein Node mit dem Namen Formel-Objekt. Seine linke Seite (blau) ist sein Eingang und rechts befindet sich der Ausgang

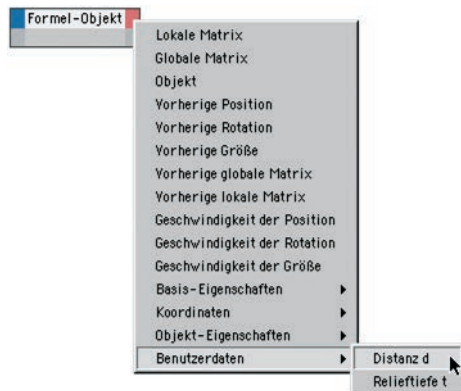


Abb. 17: Pull-down-Menü eines Nodes im XPRESSO-Editor

(rot). Wir klicken das rote Rechteck an und wählen aus dem Pull-down-Menü »Benutzerdaten / Distanz d« (Abb. 17). Anschließend wählen wir aus derselben Ecke »Benutzerdaten / Relieftiefe t«. Eventuell muss man durch Ziehen am unteren Rand des Nodes seine Fläche etwas vergrößern, um beide Ausgänge gut sehen zu können.

Distanz und Relieftiefe dienen als Eingabewerte für die Konstante λ . Den entsprechenden Ausdruck kann man mit einem Node namens *Mathe: Addieren* zusammenbauen. »Addieren« leistet nämlich nicht nur die Summenbildung, sondern auch das schlichte Aneinanderreihen verschiedener Parameter. Den Node findet man im *XPRESSO-Pool* unter »System-Operatoren / Allgemein / Berechne« oder auch im Kontext-Menü des XPRESSO-Editors. Damit der Node wie gewünscht wirkt, stellen wir im Attribute-Manager seinen *Daten-Typ*

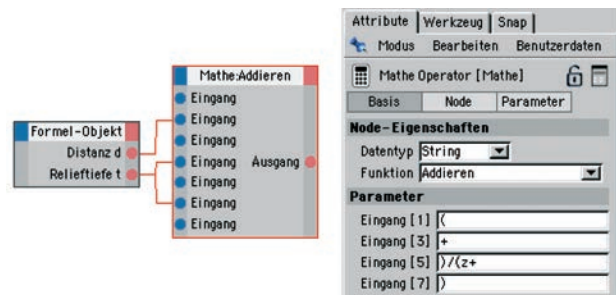


Abb. 18: Node für die Darstellung von λ

auf »String«.

Aus dem blauen Rechteck ziehen wir weitere Eingänge – wir benötigen insgesamt sieben. Durch Klick-Ziehen schicken wir jetzt den Ausgang »Distanz d« vom Formel-Objekt auf den zweiten Eingang von »Mathe: Addieren«, und »Relieftiefe t« auf Eingang vier und sechs. Die freien Eingänge werden im Attribute-Manager mit Zeichen belegt: (, +,)/(z+,) (Abb. 18). Durch einen weiteren Node namens »Ergebnis« (XPRESSO-Pool / System-Operatoren / Allgemein / Allgemein), den man auf »String« stellt, kann man den Ausgang von »Mathe: Addieren« prüfen.

Für die einzelnen Koordinatenfelder des Formel-Objektes muss λ jeweils mit x, y und z kombiniert werden. Das geschieht über drei weitere Nodes der Sorte »Mathe: Addieren« (Abb. 19). Der Daten-Typ ist wieder »string«. Auf dem ersten Eingang liegt jeweils λ und im zweiten steht »*« vor der jeweiligen Koordinate (z.B. »*x«).

All diese Einstellungen münden zuletzt in einen zweiten Node des Formel-Objektes. Man zieht das Objekt erneut aus dem Objekt-Manager in den XPRESSO-Editor und ruft am neuen Node folgende Eingänge auf: Aus *Objekt-Eigenschaften* »X(x,y,z,t)«, »Y(x,y,z,t)«, »Z(x,y,z,t)« und »Effekt«, sowie aus *Objekt-Eigenschaften / Größe* »Größe .X«, »Größe .Y« und »Größe .Z« (nicht zu verwechseln mit *Koordinaten / Größe*). Dann belegt man die drei ersten Eingänge mit den Ausgängen der

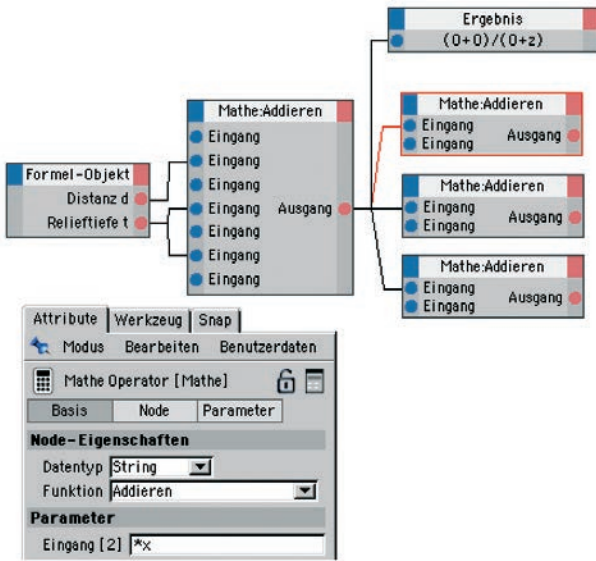


Abb. 19: Nodes für die Übergabe der Formel an das Formel-Objekt

drei zuletzt konfigurierten Mathe:Addieren-Nodes. Auf den Eingang »Effekt« schickt man aus einem Node namens »Konstante« (XPresso-Pool / System-Operatoren / Allgemein / Allgemein) mit dem Daten-Typ »String« den Wert »Manuell« und aus einem zweiten Node dieser Art den Wert »2 m« auf die letzten drei Eingänge. Damit wird sicher gestellt, dass diese Einstellungen nicht mehr am Formel-Objekt verändert werden können (Abb. 20).

Jetzt kann man das Fenster des XPresso-Editors

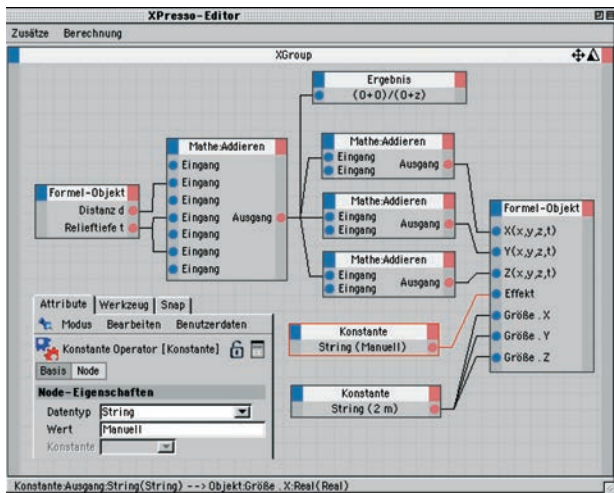


Abb. 20: Vollständige XPresso-Schaltung, Eingabe von Effekt=Manuell

schließen und die Schaltung durch Eingabe verschiedener Werte für d und t testen. Will man den Augpunkt verschieben, so bewegt man einfach das Formel-Objekt an die gewünschte Stelle.

6 Herstellung der Modelle

Wenn man sich für ein Relief entschieden hat, so geht man auf »Datei / Exportieren / STL« und speichert das Modell im *Stereolithographie*-Format. Dabei entsteht ein trianguliertes Polyedermodell. Die *ZPrint Software* des 3D-Druckers *Z™310* kann das STL-Format lesen und das Relief in Zehntelmillimeter-Schichten zerlegt an den 3D-Drucker schicken. Der *Z™310* druckt dann Schicht für Schicht, indem er Binder in ein Gipspulver spritzt. Der Gips wird aus einem Vorratsbehälter schichtweise in den Bauraum eingebracht. Der

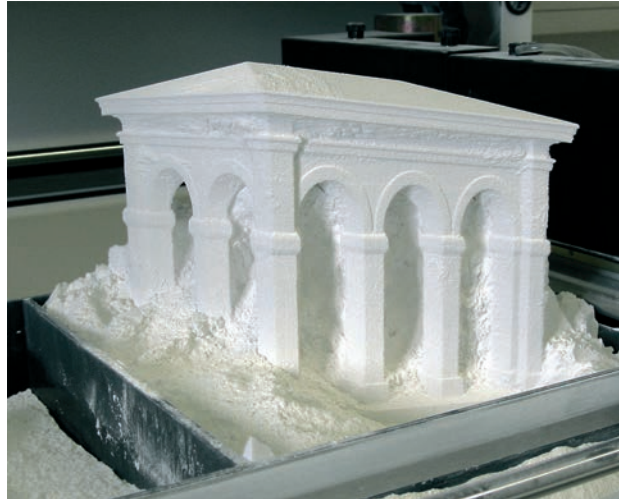


Abb. 21: Bergen eines Modells aus dem Bauraum des 3D-Druckers

Bauraum misst 200 x 250 x 200 cmm. Am Ende wird das Modell mit Pinsel und Staubsauger aus dem Gipsbett geborgen (Abb. 21) und mit *Epoxidharz* infiltriert.

Beim Modellieren der Objekte in Cinema4D sollte man einige Regeln berücksichtigen, damit *ZPrint* die Daten sauber interpretieren kann. Zum Beispiel sollte man möglichst auf den Einsatz *Boolescher Operationen* verzichten, da sie in Cinema4D zu Artefakten an den Schnittlinien führen können. Typisch sind Löcher, doppelte Polygone, verkannte Splitterpolygone, verdrehte Facettennormalen und ähnliches Datenchaos. Das ist dann nur mit zeitraubenden Nacharbeiten im *Polygon-* und *Punkt-Modus* zu beheben. Oder man investiert in zusätzliche Software. *ZPrint* benötigt nämlich ein »wasserdichtes« Polygonnetz, damit es Innen und Außen der triangulierten Polyeder unterscheiden kann. Eine große Erleichterung beim Modellieren ist, dass *ZPrint* zusammengesetzte Objekte verarbeiten kann. Dabei sind auch Überlappungen der Teilvolumina kein Problem, solange jedes Bauteil in sich konsistent ist.

Das STL-Format speichert nur Punkte und Polygone. Für ebenflächig begrenzte Volumen ist das kein Problem. Krumme Flächen müssen hingegen ausreichend unterteilt werden, um gekrümmt zu wirken. Damit man das Druckergebnis bereits in Cinema4D abschätzen kann, sollte man völlig auf das Glätten der Objekte mit dem *Phong-Tag* ver-

zichten. Standardmäßig wird aber jedes Objekt im Objekt-Manager mit Phong-Tag erzeugt. Man muss dieses anwählen und löschen.

7 Schlussbemerkungen

Bei der Herstellung der Reliefmodelle Burmesters



Abb. 22: Reliefperspektive typischer Körper aus dem Z™310



Abb. 23: Reliefperspektive einer Bogenhalle aus dem Z™310



Abb. 24: Das Rendering vom Relief der Basilika ist genau aus dem Augpunkt aufgenommen. Dass es sich tatsächlich um ein Relief handelt (und nicht den unverzerrten Raum), kann man nur erkennen, wenn man den Schlagschatten im Bild betrachtet.

in der hier skizzierten Prozesskette bleiben die von ihm beschriebenen handwerklichen Herausforderungen weitgehend außen vor. Lediglich das Infiltrieren der Z-Prints und die Ergänzung der Sockel und Guckkästen erfordern noch traditionelle Techniken. Die Modelle »Typische Körper« und »Bogenhalle« wurden bereits mit dem Z™310 realisiert, wobei das letztere nur nach Skalierung auf 80% der Originalgröße im Bauraum Platz hat (Abb. 21, 22, 23). Das dritte Relief, die »Romanische Basilika«, ist zwar schon modelliert aber noch nicht gedruckt (Abb. 24). Es kann mit Rücksicht auf die geringe Dicke der elliptisch verzerrten Säulen nur ab einer gewissen Größe gebaut werden, was zu erheblichen Materialkosten führt.

Das Formel-Deformations-Objekt in Cinema4D erfasst lediglich die Eckpunkte eines Objektes. Für die Reliefperspektive ist das völlig ausreichend, weil sie als *lineare Abbildung geraden- und ebenentreu* ist. Die Polygonkanten zwischen den Ecken werden also korrekt ins Relief übertragen. Bei anderen Transformationen, die mit dem Formel-Objekt realisiert werden können, ist das keineswegs der Fall. So muss etwa für ein *Zylinder- oder Kugelrelief* [7, 8] jedes Objekt ausreichend unterteilt werden, damit sich die Kanten im Relief jenen Kegelschnitten annähern können, die den Trägergeraden im Allgemeinen entsprechen. Abb. 25 zeigt das Ergebnis einer *Inversion im Raum*. Die Polygone des Rekonstruktionsmodells der *Maxentius-Basilika* sind im Hinblick auf die auftretenden Verformungen gezielt unterteilt, damit die Ebenen

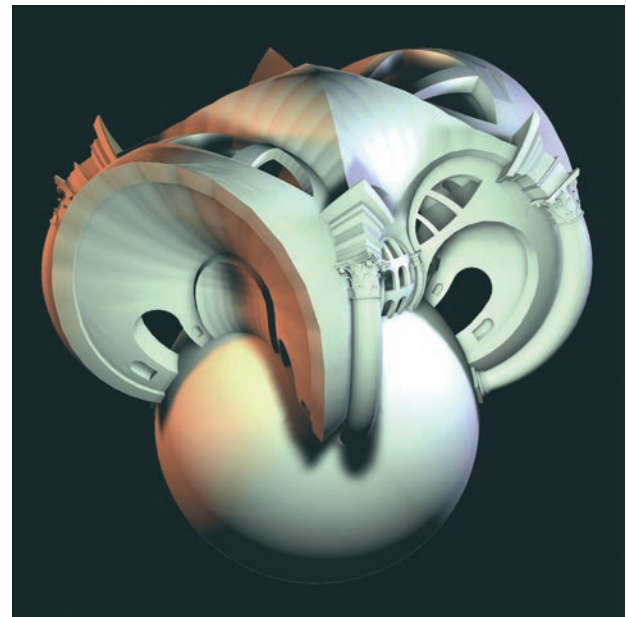


Abb. 25: Rendering von der Inversion eines Rekonstruktionsmodells der Maxentius-Basilika, Rom

als Kugeln dargestellt werden können und andererseits die Datenmenge nicht unkontrollierbar anwächst.

In einer etwas aufwändigeren XPresso-Schaltung lassen sich die hier angesprochenen Raumtransformationen zusammenfassen. Über einen Schie-

beregler gelangt man dann schrittweise von der Reliefperspektive über das Zylinder- und Kugelrelief zur Inversion.

Literatur

[1] Ludwig Burmester: *Grundzüge der Reliefperspektive nebst Anwendungen zur Herstellung reliefperspektivischer Modelle*; Leipzig 1883

[2] Ludwig Burmester: *Grundlehren der Theaterperspektive*. *Allgemeine Bauzeitung* 49 (1884), 39-40, 44-49, 53-57, Tafeln 25-27

[3] Joost Elffers: *Anamorphosen. Ein Spiel mit der Wahrnehmung, dem Schein und der Wirklichkeit*; Köln 1981

[4] Langlands & Bell: »Atlantis« (1989) und »The Security Council« (1989); in: *DaimlerCrysler Collection. Private/Corporate III*; Berlin 2005, 68/69

[5] Giulio Macchi: *Raum (aus dem Katalog der XLII ESPOSIZIONE INTERNAZIONALE D'ARTE LA BIENNALE DI VENEZIA, ARTE E SCIENZA 1986)*; in: *Kunstforum International 85*; *Von der Renaissance Perspektive zum Kybernetischen Raum (Oktober 1986)*, 154-161

[6] Emil Müller, Erwin Kurppa: *Lehrbuch der darstellenden Geometrie, fünfte, ergänzte Auflage*, Wien 1948, 373-376

[7] Fred Rößler: *Geometrische Betrachtungen über eine Verallgemeinerung der Reliefperspektive*; in: *Zeitschrift für angewandte Mathematik und Mechanik. Ingenieurwissenschaftliche Forschungsarbeiten, Band 28, Heft 10 (Oktober 1948)*, 311 - 316

[8] Fred Rößler: *Über verallgemeinerte Reliefperspektiven*; in: *Monatshefte für Mathematik, Band 53, Heft 3 (Wien 1949)*, 211-220

[9] Hermann Schaal: *Reliefperspektive*; in: *Der Mathematikunterricht* 27, Heft 3 (Stuttgart 1981), 69-90

[10] Georg Scheffers: *Lehrbuch der Darstellenden Geometrie*; Band 2; Berlin 1927, 424 ff

[11] Manfred Wundram, Thomas Pape: *Andrea Palladio (1508-1580). Architekt zwischen Renaissance und Barock*; Köln 1988, 226-233

Internet

Zu Cinema4D: <http://www.maxon.de/>

Zum Z™310: <http://www.zcorp.com/de/> und <http://www.4dconcepts.de/>

Zur Modellsammlung der TU Dresden: <http://www.math.tu-dresden.de/modellsammlung/>

Zum 3D Labor B25 am Institut für Geometrie der TU Dresden: <http://www.math.tu-dresden.de/3D-Labor>

Mathematischer Hintergrund von Rendering-Verfahren

Alexander Wilkie

Institut für Computergraphik und Algorithmen, TU Wien

1 Einführung

Zusätzlich und ergänzend zu den altbekannten Darstellungsmethoden des geometrischen Zeichnens haben sich im Bereich der Geometrie in den letzten Jahren zunehmend neuartige Verfahren wie etwa Raytracing zur Visualisierung von geometrischen Problemen und Objekten etabliert.

Die optische Qualität der damit erzielbaren Resultate ist sehr unterschiedlich, und die dafür verwendeten Programme weisen manchmal speziell für einen nicht aus dem Bereich der Computergraphik stammenden Anwender eine verwirrende Terminologie und Bedienung auf. Dies trifft oft gerade auf moderne Programme zu die moderne, leistungsfähige – aber eben auch komplexe und in der Bedienung manchmal unhandliche – Algorithmen verwenden. Die einschlägige Fachliteratur ist derzeit oftmals nur für Spezialisten gedacht und für reine Anwender nicht sehr hilfreich; Ziel dieses Artikels ist es hier durch Erklärung einiger der Grundlagen den Nutzern solcher Werkzeuge eine Hilfe anzubieten.

Dieser Artikel zerfällt dabei in zwei Teile: im ersten wird das Problem der Bildsynthese (die oft auch als **Rendering** bezeichnet wird) rein theoretisch betrachtet, und im zweiten werden dann einige der gängigen Verfahren in dieses theoretische Gerüst eingeordnet.

2 Die Rendering-Gleichung

Auch wenn die Forschung im Bereich der photorealistischen Bildsyntheseverfahren keineswegs abgeschlossen ist sind die theoretischen Grundlagen des Problems mittlerweile gut verstanden, und es ist auch für reine Anwender solcher Verfahren durchaus interessant sich diesen mathematischen Hintergrund vor Augen zu führen da er das Verständnis von manchen in der Anwendung auftretenden Eigenheiten sehr erleichtern kann.

Kern dieses ersten Teils ist daß sich alle bekannten Methoden zur Bildsynthese in einem gemeinsamen mathematischen Formalismus darstellen lassen [4]. Dieser leitet sich daher ab daß das Problem der Bildsynthese durch eine einzige Gleichung darstellbar ist, der sogenannten **Rendering-Gleichung**:

$$L(x, x') = g(x, x') [L_e(x, x') + L_R(x, x')]$$

mit

$$L_R(x, x') = \int_{\Omega} f_r(x, x', x'') L(x', x'') dx''$$

Alle Verfahren die auf mathematischem Wege synthetische Bilder erzeugen – gleich mit welchem konkreten Algorithmus dies geschieht – können als Lösungsstrategien dieser Gleichung (einer Fred-