Ein projektiver Ansatz computergestützten Zeichnens

UNIVERSITÄT MANNHEIM

Seminararbeit
von

cand. wirtsch.-inf.

Michael Möske

<mmoeske@rumms.uni-mannheim.de>

aus
Mannheim

vorgelegt am
Lehrstuhl für Praktische Informatik IV
Prof. Dr. Wolfgang Effelsberg
Universität Mannheim

April 2003

Betreuer:
Dipl.-Inf. Dirk Farin

Inhaltverzeichnis

1. Einleitung

Diese Arbeit ist im Rahmen des Seminars „3D Rekonstruktion“ am Lehrstuhl Praktische Informatik IV der Universität Mannheim unter der Leitung von Dirk Farin entstanden. Sie stellt einen „Projektiven Ansatz zum computerunterstützten Zeichnen“ vor, der von Osama S. Tolba, der einen Master in Landschaftsarchitektur besitzt, entwickelt wurde. Das in dieser Arbeit vorgestellte Computerprogramm entstand im Rahmen seiner Doktorarbeit im Juni 2001. Es unterstützt dreidimensionale Zeichnungen, wobei es auf einem projektiven zweidimensionalen Ansatz basiert.

Die Motivation für seine innovative Arbeit ist die folgende. Seitdem sich Graphikbearbeitung und -erstellung auf dem Computer etabliert haben, hat sich gerade im Bereich der Architektur einiges geändert. Konstruktionszeichnungen werden heute kaum noch von Hand gezeichnet und auch physikalische Modelle wurden inzwischen durch computergenerierte 3D Modellen abgelöst. Das perspektivische Zeichnen, das einst die wichtigste Technik war um Designideen darzustellen, ist im Zeitalter schneller und flexibler Computer-Aided Design and Drafting (CADD) Systeme geradezu überflüssig geworden.

Dabei haben perspektivische Zeichnungen einen besonderen Reiz. Sie schaffen es, dreidimensionale Informationen auf einer zweidimensionalen Oberfläche darzustellen. Die grundlegenden Prinzipien dieser Technik wurden bereits in der Renaissance entwickelt und sind seitdem in Kunst und Design weit verbreitet.

Es ist sehr aufwändig, perspektivische Zeichnungen zu konstruieren, da vor allem auf die richtigen Proportionen geachtet werden muss. Um diese Proportionen zu erhalten sind viele Hilfslinien nötig. Außerdem sind diese Bilder dann nur statisch und verringern damit ihren dreidimensionalen Eindruck. Noch zeit- und arbeitsaufwändiger ist eine realistische Ausarbeitung der Schatten und Schattierungen. Der letzte Nachteil ist, dass, wie bei allen Zeichnungen auf Papier, sie nur für den einmaligen Gebrauch angefertigt werden, denn sie sind nicht wiederzuverwenden.

Tolbas Arbeit stellt interaktive Techniken vor, die perspektivisches Zeichnen unterstützen. Bis jetzt wurde dieser Bereich in 2D Zeichenprogrammen weitgehend vernachlässigt. Fast alle handelsüblichen Programmen unterstützen nur ein kartesisches Koordinatensystem. Damit ein perspektivisches Bild zu zeichnen wäre ebenso zeitaufwändig wie auf dem Papier.

Der Mangel an Unterstützung für perspektivische computergestützte Zeichenprogramme ist vor allem durch den enormen Aufwand mit dem an dreidimensionalen Grafikprogrammen gearbeitet wird zu erklären. Doch auch diese haben Ihre Schwächen. 3D-Modelle halten sich an starre Geometrie und können ebenfalls sehr arbeitsaufwendig sein. Außerdem benötigen sie sehr genaue Daten, die meistens in der Designphase noch nicht vorhanden sind [1].

Im nächsten Kapitel wird ein kurzer Einblick in die Grundlagen des perspektivischen Zeichnens gegeben, indem traditionelle Techniken und Verfahren, wie zum Beispiel Fluchtpunktezeichnen, vorgestellt werden. Außerdem wird das dem Zeichenprogramm zugrundeliegende System der Projektionspunkte vorgestellt. Anschließend geht diese Arbeit auf die Repräsentation von Objekten im System, grundlegende Mechanismen zur Veränderung der Betrachtungsperspektive und zwei Grundfunktionalitäten, das Verschieben und das Drehen, näher ein und untersucht die dazugehörigen Algorithmen der Implementierungen. Im weiteren Verlauf der Arbeit werden die übrigen Features wie Schattierungen oder Extrusion angeführt, ohne jedoch detailliert auf Ihre Implementierung einzugehen, da dies den Rahmen des Seminars sprengen würde. Ein kurzer Einblick in die vom Programm unterstützte Integration anderer Medien und ein Fazit runden die Arbeit ab.

2. Grundlagen

Das Programm, über das diese Arbeit berichtet, basiert auf zwei Dingen, nämlich traditionellem perspektivischen Zeichnen zum einen und projektiver Geometrie zum anderen. In diesem Kapitel wird zunächst auf die Grundsätze des perspektivischen Zeichnens eingegangen um dem Leser ein gewisses Grundwissen über die Materie zu vermitteln. Danach wird eine kurze Einführung in die projektive Geometrie gegeben, wobei vor allem die Zentralprojektion im Dreidimensionalen und die daraus resultierende Projektion in den zweidimensionalen Raum im Mittelpunkt stehen.

2.1 Techniken des Perspektivischen Zeichnens

Ursprünglich wurde eine genaue perspektivische Zeichnung aus zwei oder mehr orthogonalen Ansichten gezeichnet. So konnte zum Beispiel das exakte perspektivische Abbild eines Hauses aus seinem Grundriss und einer Seitenansicht konstruiert werden. Dazu müssen Sichtstrahlen der beiden Ansichten geschnitten werden. Dieser sehr zeitaufwändige und arbeitsintensive Prozess, dessen Konstruktion eine Menge Hilfslinien erfordert, ist in Abbildung 2.1 dargestellt. Alternativ wurden im Laufe der Zeit eine Vielzahl von Techniken entwickelt, die es geübten Künstlern erlauben, den Prozess zu verkürzen.

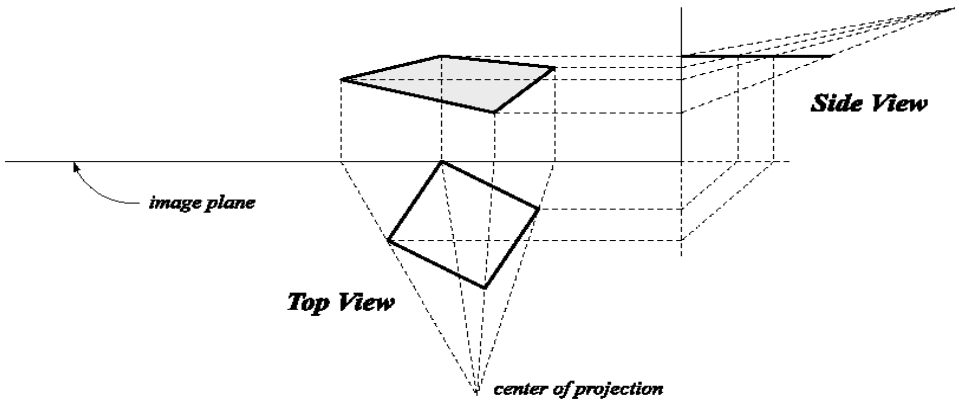

Abb. 2.1 Konstruktion eines Objekts aus Grund und Aufriss [1]

Die einfachste und wohl auch am weitesten verbreitete Vereinfachung ist die Technik des *Fluchtpunkts* oder der *Fluchtpunkte*. Wie alle Vereinfachungen zielt auch sie darauf ab, die

Anzahl der Konstruktionslinien, die durch die orthogonalen Ansichten erzeugt werden, zu verringern. Parallele Geraden im dreidimensionalen scheinen auf einen Punkt zuzulaufen, wenn die perspektivisch dargestellt werden. Ein solcher Punkt wird Konvergenz- oder, häufiger, *Fluchtpunkt* genannt. Zeichnungen, auf denen Gebäude zu sehen sind, enthalten oft Kanten, die in drei orthogonal zueinander stehenden Richtungen liegen. Diese Richtungen erzeugen jeweils einen Fluchtpunkt. Ist nun eine Richtung parallel zur Sichtebene, so laufen die Geraden dieser Richtung nicht in einem Punkt zusammen, es ergibt sich also eine *Zwei-Fluchtpunktperspektive* (zu sehen in Abbildung 2.2). Falls nun also zwei Ebenen parallel zur Sichtebene sind so spricht man von *Ein-Fluchtpunktperspektive*.

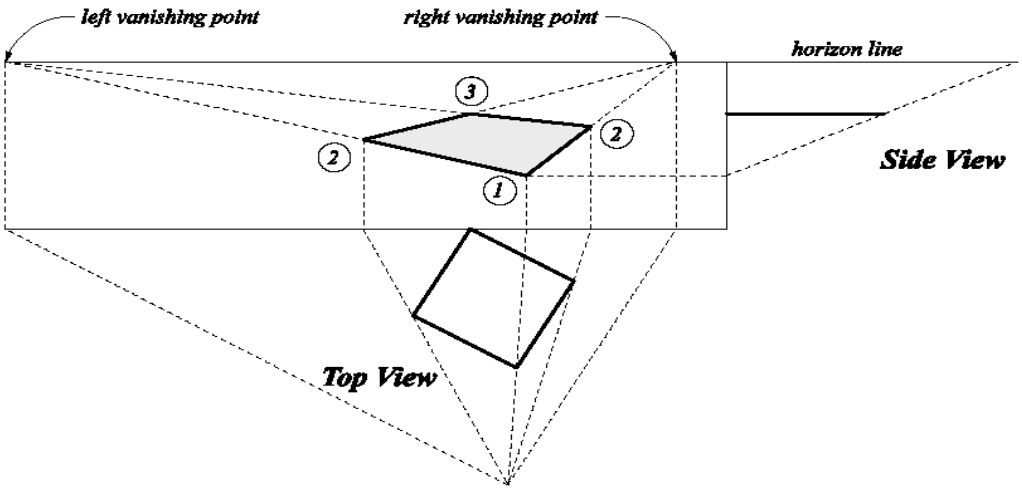

Abb. 2.2 Konstruktion eines Objekts mit Hilfe von Fluchtpunkten [1]

Zusätzlich zu den Fluchtpunkten, die mit den Richtungen helfen, benötigen die Künstler ein Hilfe bei der Längebestimmung verschiedener Kantenabschnitte und bei der relativen Lage perspektivisch gezeichneter Flächen. Eine solche Hilfe stellen *Gitternetzlinien* bereit. Die Diagonalen dieses *Gitters* zeigen auf Hilfsfluchtpunkte. Manchmal kann man solche *Gitter* auch noch in fertigen Bildern erkennen, z.B. in Form von gefliesten Fußböden. Sie können außerdem dazu benutzt werden, um komplexe Formen von der orthogonalen in die perspektivische Ansicht zu übertragen. Dies ähnelt dann der Anwendung von *Gittern* zur Vergrößerung von Bildausschnitten. Obwohl es eine Vielzahl von vorgefertigten *Gitternetzlinien* für perspektivische Zeichnungen gibt, empfinden manche Künstler diesen Ansatz als zu statisch.

Es gibt außerdem spezielle Fluchtpunkte, so genannte *Maßpunkte*, die es dem Zeichner ermöglichen, Längenkonstruktionen direkt in der perspektivischen Sicht auszuführen ohne dazu die orthogonalen Ansichten hinzuzuziehen.

Eine weitere Technik ist die sogenannte *Extrusion*. Das bedeutet, dass bei einer perspektivischen Zeichnung zuerst der Grundriss mit den oben genannten Techniken konstruiert wird. Erst danach werden die Kanten in vertikaler Richtung, also z.B. die Wände, eingezeichnet.

Des weiteren erhöhen *Schattierungen* den dreidimensionalen Effekt eines Bildes um ein Vielfaches, da sie Texturen und Lichteffekte hinzufügen. Wie genau *Schattierungen* aussehen hängt stark von der Art des Gemäldes zusammen, also ob es ein Ölgemälde ist oder eine Bleistiftskizze. *Schattierungen* werden benutzt, um zu zeigen, wie weit eine Fläche von den Augen des Betrachters, im Verhältnis zu einer anderen Fläche, entfernt ist. Außerdem werden *Schattierungen* benutzt, um Beleuchtungsverhältnisse darzustellen, also z.B. ob eine Fläche direkt von einer Lichtquelle angestrahlt wird oder nicht.

Zuletzt gibt es noch *Schatten*. Sie haben die Eigenschaft die relative Lage zweier Objekte im Bild zueinander hervorzuheben, nämlich die Lage des *schattenwerfenden* Objekts zum Objekt auf das der *Schatten* geworfen wird. Bilder unter freiem Himmel enthalten oft nur eine gerichtete Lichtquelle, die Sonne, während Bilder von Gebäudeinneren oftmals mehrere lokale Lichtquellen enthalten.

2.2 Projektive Geometrie

Eine Vielzahl von zweidimensionalen Räumen existiert, darunter der wohl am meisten benutzte *Euklidische Raum*, oftmals dargestellt als kartesisches Koordinatensystem, in dem die Lage eines Punkte durch Ihren Abstand von zwei orthogonalen Achsen definiert wird. Ergänzt man diesen Raum durch eine Gerade der Unendlichkeit, oder Horizont, so erhält man einen etwas allgemeineren Raum, den man auch Projektionsebene nennt [2]. Ein anderes Modell des Projektionsraums ist das *Kugelmodell* [3].

Projektionspunkte werden durch ihre *homogenen Koordinaten* definiert, wobei jeder Punkt durch ein Tripel reeller Zahlen dargestellt wird und skalare Vielfache als äquivalent angesehen werden. Ein Punkt wird demnach durch einen Spaltenvektor $m = [x, y, w]^T$ repräsentiert. Eine Gerade kann dementsprechend durch ihre *homogenen Koeffizienten* $l = (a, b, c)$ dargestellt werden, wobei für alle Punkte auf der Gerade $lm = ax + by + cw = 0$ gilt. Im Kugelmodell werden durch die Vorschrift $x^2 + y^2 + w^2 = 1$

alle Punkte normiert. Wenn man das Kugelmodell als eine *Einheitskugel*, die im dreidimensionalen Raum eingebettet ist, betrachtet, kann man dieses Modell benutzen, um perspektivisches Zeichnen zu interpretieren und zu manipulieren.

Die einfachste Form der *Zentralprojektion* kann durch die folgende Gleichung ausgedrückt werden:

$$\begin{pmatrix} x \\ y \\ w \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix}$$

Wobei $(X,Y,Z,W)^T$ die Koordinaten im dreidimensional Raum sind, und $(x,y,w)^T$ die im zweidimensionalen.

Im Kugelmodell entsprechen Geraden einem Kreis auf der Kugeloberfläche. Um das zu verdeutlichen muss man sich nur vorstellen, dass eine Ebene eine Kugel in einem Kreis schneidet. Dabei ist der Normalenvektor der durch den Kreis repräsentierten Ebene äquivalent zu der Gerade, die ebenfalls durch den Kreis repräsentiert wird. Der Schnittpunkt zweier Geraden ist demnach das Kreuzprodukt Ihrer homogenen Koordinaten.

Jede Gerade hat eine Richtung, die von Ihrer Lage unabhängig ist. Diese Richtung kann durch einen Punkt dargestellt werden, der auf der Kugel im Unendlichen des dreidimensionalen Raums liegt. Die Kugel im Unendlichen ist das Äquivalent des Horizont im zweidimensionalen Ansatz. Durch die Zentralprojektion wird dieser Punkte wiederum auf einen Punkt auf der Einheitskugel abgebildet. Da eine Projektion paralleler Geraden normalerweise zu diesem Punkt konvergiert, nennt man ihn auch *Fluchtpunkt*.

Außerdem spricht mal von der *Dualität von Punkten und Geraden*. Zu jedem Punkt gehört auch eine Gerade, d.h. die Worte Punkt und Gerade sind jederzeit beliebig austauschbar. Man kann sich den Punkt als den Pol der Einheitskugel vorstellen. Die dazugehörige Gerade wäre ein Kreis der dem Äquator entspricht, man nennt sie deshalb auch *polar Komplement* des Punktes.

Ebenso wie parallele Geraden konvergieren auch parallele Ebenen im Unendlichen. Diese bilden einen Umkreis auf der Kugel im Unendlichen. Da alle Punkte auf diesem Kreis lotrecht zum Normalenvektor der Ebene stehen, kann man sich leicht vorstellen, dass der Kreis das *polar Komplement* des Punktes ist, der den Normalenvektor der Ebene repräsentiert. Normalenvektoren von Ebenen können als Kreuzprodukt zweier beliebigen Fluchtpunkte der Ebene (vgl. Abbildung 2.3)

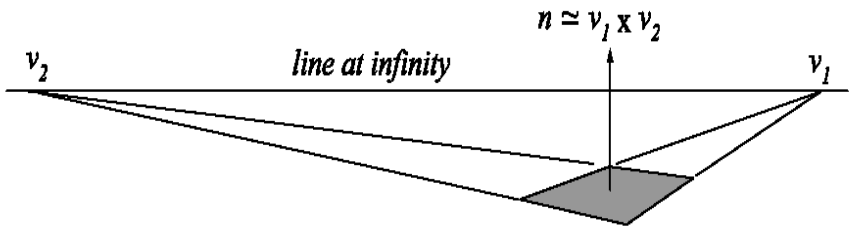

Abb. 2.3 Normalenvektor als Kreuzprodukt zweier beliebiger Fluchtpunkte [1]

Zuletzt geht dieses Kapitel noch auf *Kollineationen* ein. Eine Transformation im zweidimensionalen Projektionsraum heißt Kollineation, wenn sie eine Menge Punkte und Geraden in eine andere Menge überführt, wobei kollineare Punkte kollinear bleiben, Geradenbündel erhalten bleiben und die Inzidenz erhalten bleibt, d. h. alle Punkte die auf einer Geraden liegen, liegen danach wieder auf einer Geraden. Eine Kollineation wird durch die folgende Gleichung repräsentiert.

$$\begin{pmatrix} x' \\ y' \\ w' \end{pmatrix} = \lambda \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ w \end{pmatrix}$$

Hier ist H eine nicht singuläre Matrix und λ ein frei wählbarer Faktor. An dieser wird für den Rest der Arbeit das Zeichen $\cong$ eingeführt mit $a \cong b \Leftrightarrow a = \lambda b$

3. Grundfunktionen

Das in dieser Arbeit vorgestellte Zeichensystem unterstützt Veränderungen des Blickwinkels des Betrachters durch Drehen und Kippen sowie Manipulationen des Bildes, die wie 3D Rotation oder Verschiebung aussehen. Hierdurch wird die einfache Erstellung von Bildern mit symmetrischen oder sich wiederholenden Elementen ermöglicht. Für diese Techniken werden weder Entfernungs- noch Tiefeninformationen über das Objekt benötigt. Die Benutzerschnittstelle, die in diesem Kapitel vorgestellt wird, ermöglicht es dem Benutzer die Objekte durch einfaches Ziehen zu bewegen. Zuerst wird auf die Repräsentation von Objekten im System und die Veränderung der Perspektive des Betrachters eingegangen. Im zweiten Teil dieses Kapitels wird die Vorgehensweise des Programms beim Verschieben erläutert. Der dritte Teile legt den Algorithmus des Drehens dar.

3.1 Architektur des Systems

Tolbas Zeichenprogramm liegt der Kugelmodell-Ansatz zu Grunde. Jeder Strich und jede Fläche wird im System als eine Menge von den im letzten Kapitel vorgestellten Projektionspunkten dargestellt. Diese Punkte erhält man durch Projektion eines gezeichneten Punkts auf die Einheitskugel. Bildlich gesprochen blickt der Betrachter aus der Mitte einer Kugel auf die auf der Einheitskugel gespeicherten Objekte (vgl. Abbildung 3.1).

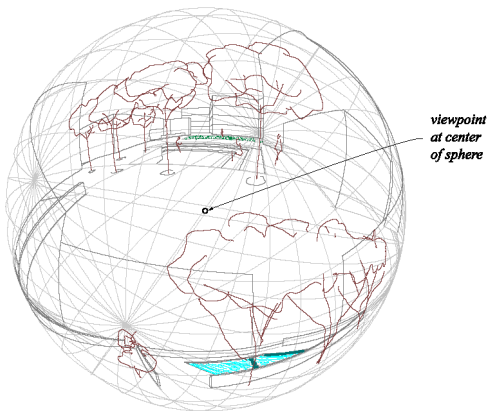

Abb. 3.1 Der Benutzer blickt vom Mittelpunkt der Kugel auf seine Umwelt.

Striche und Objekte erhalten zusätzliche Attribute wie Dicke und Farbe. Die gespeicherten Formen können leicht auf eine vom Benutzer definierte Sichtebene zurückprojiziert werden und erwecken dadurch den Anschein eines veränderten Blickwinkels. Diese Rückprojektionen können als Rotationen und Zooms ausgehend von einem fixen Punkt, dem Kugelmittelpunkte betrachtet werden. Es ist jedoch nicht möglich, von einem anderen Punkt aus etwas darzustellen. Abbildung 3.2 zeigt die Rückprojektion auf eine benutzerdefinierte Sichtebene.

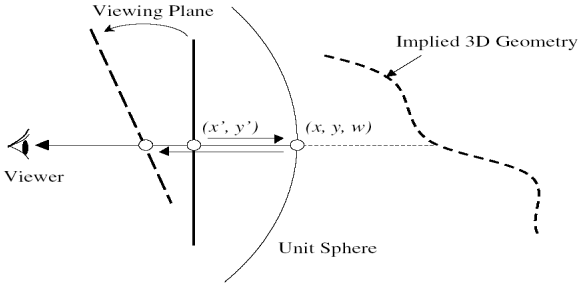

Abb. 3.2 Rückprojektion der Daten auf eine andere Sichtebene .

Die Veränderungen können durch die im letzten Kapitel vorgestellte Kollineationsgleichung erreicht werden. Im System gilt die folgende Gleichung:

$$H = \begin{pmatrix} f & 0 & x_0' \\ 0 & f & y_0' \\ 0 & 0 & 1 \end{pmatrix} R_{\theta\Phi}$$

x' und y' spezifizieren hierbei das Projektionszentrum und f der Abstand des Betrachters von der Sichtebene. R ist die Rotationsmatrix, die drei Winkel θ, Φ und Ψ enthält, die für eine Drehung des Bildes um die verschiedenen Achsen verantwortlich sind. Der Winkel Ψ kann in den meisten Fällen vernachlässigt werden, da er die Rotation um den Gerade von Betrachtungspunkt zu Objekt darstellt.Abbildung 3.3 veranschaulicht alle Parameter.

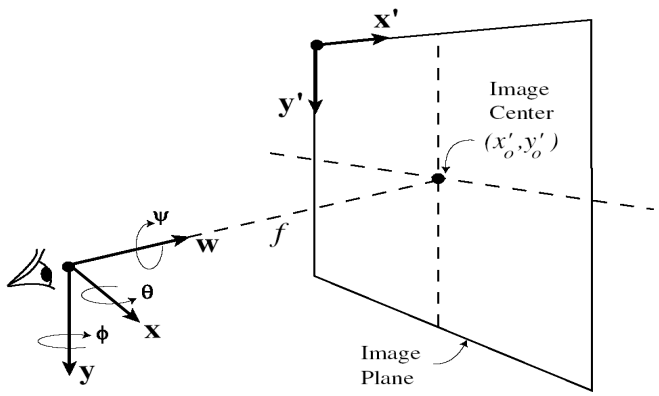

Abb. 3.3 Darstellung der verschiedenen Parameter bei der Rückprojektion.

3.2 Verschiebungen

Eine Möglichkeit ein Objekt zu verschieben ist durch Zuhilfenahme von vielen Konstruktionslinien. In Abbildung 3.4 sieht man an einem Beispiel das Verschieben eines Vierecks P_1, P_2, P_3, P_4.

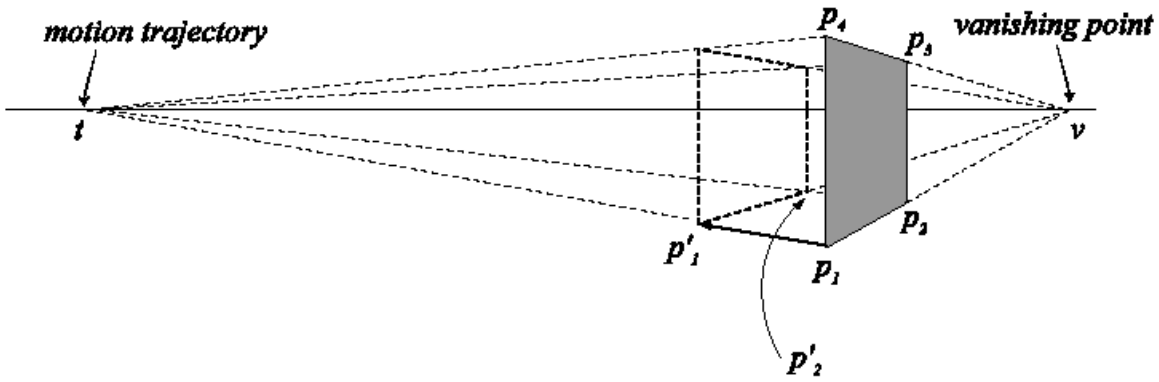

Abb. 3.4 Verschiebung unter Zuhilfenahme von Konstruktionslinien.

Zuerst benötigt man eine Bewegungsrichtung, hier dargestellt durch den Verschiebungsfluchtpunkt *t*. Dann können anhand eines verschobenen Punktes, z.B. P_1' die anderen Punkte des Vierecks unter Zuhilfenahme des Fluchtpunkts v konstruiert werden. Eine Implementierung der Verschiebung durch Fluchtpunkte wäre zwar denkbar, doch die entscheidende Schwachstelle dieser Methode offenbart sich, wenn der Verschiebungsfluchtpunkt mit dem Fluchtpunkt der Zeichnung zusammenfällt.
Daher wird in der Umsetzung von dieser Methode abgesehen, stattdessen wird eine Abbildung der Projektionsebene, in diesem Fall also der Kugel, durchgeführt. Eine solche Kollineation ist definiert durch Multiplikation mit der 3 x 3 Matrix H:

$$m' \cong Hm$$

Bekanntermaßen ist das Abbild einer 3D Ebene auf eine andere Ebene durch die folgende Homographie gekennzeichnet [6]:

$$H \cong I + \frac{\delta}{d} t n^T$$

wobei I die Einheitsmatrix darstellt, t den Verschiebungsvektor, δ die Verschiebungsentfernung und n der Normalenvektor der Bewegungsebene ist. Der Normalenvektor n ist im dreidimensional Raum durch die Gleichung $n * p = d$ definiert (vgl. Abbildung 3.5)

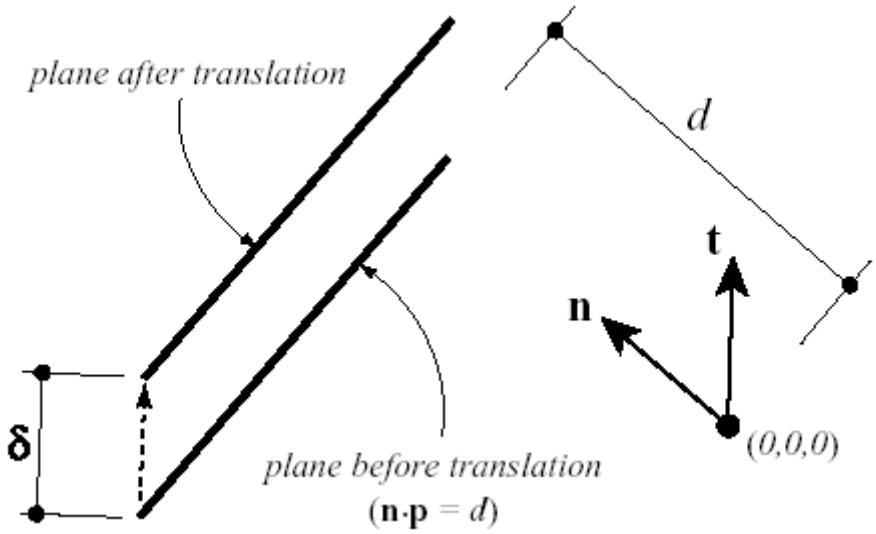

Abb. 3.5 Verschiebung einer dreidimensionalen Ebene.

Da in einem zweidimensionalen Projektionsansatz weder Informationen über den Abstand d des Sichtpunktes von der Oberfläche noch über die tatsächliche Verschiebung der Ebene δ vorliegen, wird eine neue Größe $\alpha = \delta / d$ eingeführt. Dies führt zu einer Familie von Homographien mit einem Parameter, die kompatibel zur Verschiebung einer dreidimensional Ebene sind:

$$T(\alpha) \cong I + \alpha t n^T$$

In dieser Gleichung ist auf der rechten Seite alles außer α bekannt und dieses kann leicht aus einem Punktepaar (m, m') berechnet werden, wobei m der Punkt vor und m' der Punkt nach der Verschiebung ist. Dieses Punktepaar kann mit dem Mauszeiger ausgewählt werden, muss aber auf der zuvor durch den Benutzer definierten Bewegungsrichtung liegen.

Sollte der Fall auftreten, dass eine Fläche genau durch den Ursprung geht oder nur Ihre Kante zu sehen ist, dann kann keine Homographie verwendet werden um eine 3D-Bewegung zu simulieren.

3.3 Rotationen

Herkömmliche Techniken zur Drehung von Objekten sind weitaus weniger bekannt als die Techniken zur Verschiebung. Die in Kapitel 2 vorgestellten *Maßpunkte* haben ihren Namen von der Eigenschaft, dass alle Streckenendpunkte bei der Drehung um einen fixen Punkt auf einer Gerade liegen, die den Horizont im Maßpunkt schneidet. Abbildung 3.6 veranschaulicht diese Eigenschaft.

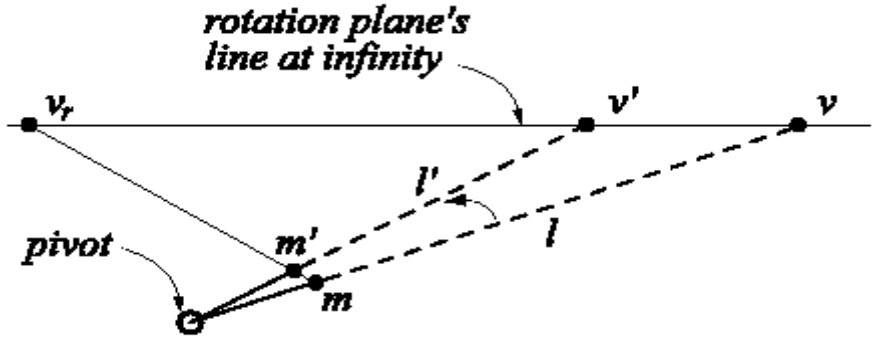

Abb. 3.6 Traditioneller Ansatz um eine Linie perspektivisch zu Drehen

Doch auch hier verwendet Tolba einen homographischen Ansatz um die Rotation eines zweidimensionalen Objekts um einen fixen Punkt zu simulieren. Hierzu wird durch den Benutzer ein Drehpunkt angegeben. Nachdem Rotationsachse, Drehpunkt und Winkel festgestellt wurden, führt das Programm die Drehung des Objekts in 2 Schritten durch.
Zuerst wird das ganze Objekt um den Sichtpunkt gedreht. Hierfür wird die gewünschte Drehachse und Rotationswinkel benutzt. Alle Punkte des Objekts, der Drehpunkt eingeschlossen, werden zu einer vorübergehenden Position bewegt:

$$m'' = R(a, \theta)m$$

Anschließend wird die im letzten Abschnitt vorgestellte dreidimensionale Verschiebung eingesetzt, wobei $t \cong p - p''$, um das Objekt wieder zurück zum ursprünglichen Drehpunkt zu bewegen:

$$m' \cong T(\alpha : p'' \rightarrow p)m''$$

Die gewünschte Rotation ist also eigentlich die Verknüpfung einer dreidimensionalen Rotationsmatrix mit einer pseudo-dreidimensionalen Verschiebung.

$$m' \cong T(\alpha)R(a,\theta)m$$

Eine sehr intuitiv bedienbare Benutzerschnittstelle erlaubt dem Anwender die Rotationsparameter zu spezifizieren. Als erstes wählt der Benutzer die Rotationsachse aus einer Anzahl aktiver Richtungen aus und benutzt den Mauszeiger um einen Drehpunkt zu definieren. Danach kann der Winkel durch das Ziehen des Mauszeigers um den Drehpunkt eingestellt werden.

3.4 Beispiele

Zum Abschluss dieses Kapitels soll anhand zweier Beispiele die Funktionsweise der im Kapitel vorgestellten Algorithmen aufgezeigt werden. Abbildung 3.7 zeigt ein Regal das aus verschobenen Kopien von drei verschiedenen Basisobjekten hervorgegangen ist. Objekt 1 diente jeweils als Vorlage für die Flächen an der Regalvorderseite, Objekt 2 wurde für die Seitenteile benutzt und aus Objekt 3 wurden die Regalbretter erschaffen.

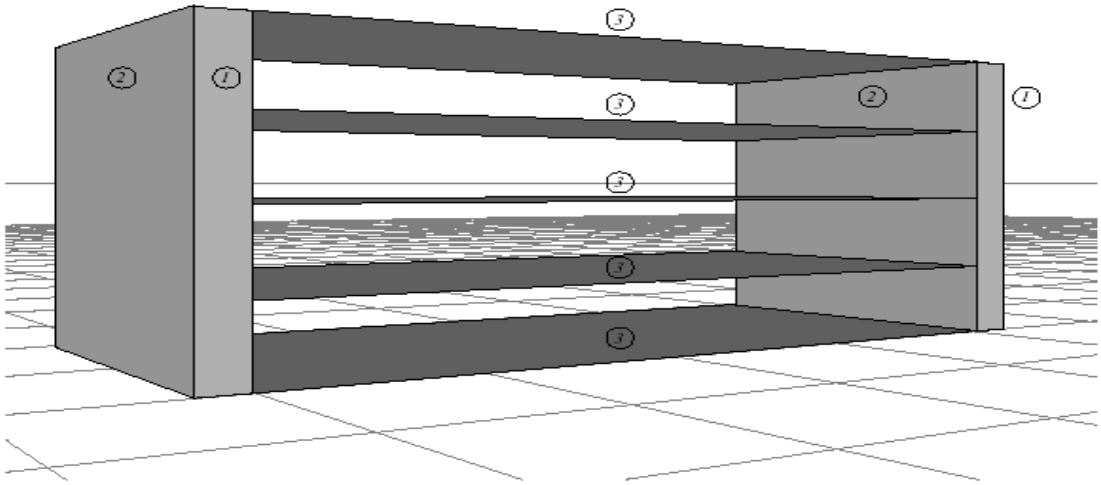

Abb. 3.7 Ein Regal wurde aus drei Basiselementen durch Kopieren und Verschieben konstruiert [1]

Das zweite Beispiel, dargestellt in Abbildung 3.8 , zeigt eine Fläche, die um Ihre Mittelachse rotiert. Sowohl Verschiebungen als auch Drehungen erwecken den Anschein, dass die Objekte wirklich dreidimensionalen Veränderungen unterzogen wurden. Diese pseudo-dreidimensionalen Operationen stellen dem Benutzer ein flexibles und zusammen mit der Kopieroperation ein mächtiges Werkzeug zur Verfügung, das die schnelle Erzeugung dreidimensional wirkender Objekte ermöglicht.

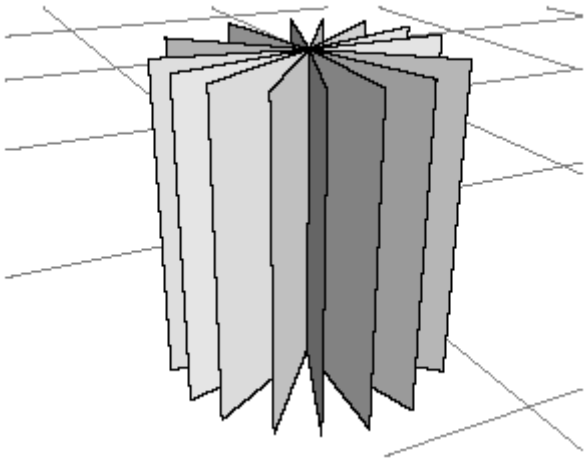

Abb. 3.8 Rotation eines Rechtecks um eine seiner Mittelparallelen.

4. Weitere Funktionen

Dieses Kapitel ist den vielen weiteren Funktionen des Zeichenprogramms gewidmet, die aufgrund des begrenzten Rahmens der Seminararbeit nur angerissen werden können ohne ihre Implementierung ausreichend detailliert beschreiben zu können. Der erste Abschnitt beschäftigt sich mit *aggregierten Formen* während *Schatten* und *Schattierungen* den Kern des zweiten Abschnitts bilden.

4.1 Aggregierte Formen

Die in Kapitel 3 vorgestellte Funktionalität bildet die Basis für Konstruktionszeichnungen mit Objekten, die dreidimensionale Modelle imitieren. Im nächsten Schritt bietet das Programm den Künstlern die Möglichkeit auch komplexere Formen mit der gleichen Leichtigkeit wie Freihandzeichnungen darzustellen. Selbstverständlich wird für diese Formen auch die Basisfunktionalität des Drehens und Rotierens unterstützt. Dazu werden solche Formen als Anhäufung von planaren Grundelementen repräsentiert.

Die *Extrusion* soll hier als Bespiel für solche aggregierten Formen dienen. Hierbei sollen Formen, die dreidimensionale extrudierte Oberflächen imitieren, erzeugt werden, indem eine Instanz einer dreidimensionalen Kurve mit einer anderen verbunden wird, die aus der ursprünglichen nur durch eine Umformung, typischerweise eine Verschiebung, hervorgegangen ist. Abbildung 4.1 zeigt eine solche Extrusion.

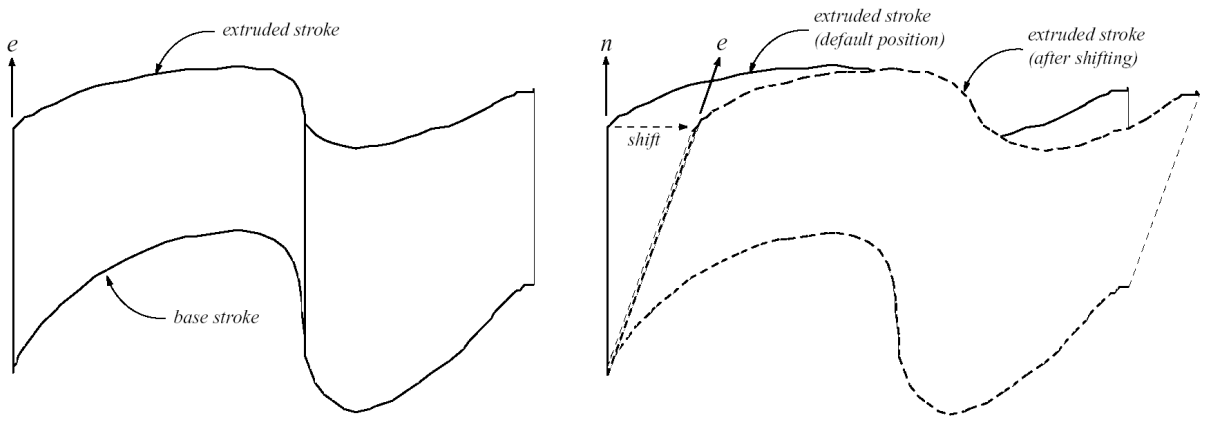

Abb. 4.1 Extrusion eine Basisstrichs. Links im Lot zur Basis, rechts verschoben

Dabei ist die Extrusionsrichtung standardmäßig senkrecht zum Ursprungsstrich, kann aber durch den Benutzer in jede Richtung verschoben werden. Die Technik der Extrusion ermöglicht es dem Benutzer ganze Gebäude aus dem Grundriss „herauszuziehen“. Zuerst kann der Grundriss durch einfache Zwei-Fluchtpunkt Technik konstruiert werden und danach wird durch einfaches Ziehen mit der Maus, nach Definieren der Extrusionsrichtung durch einen aktiven Fluchtpunkt, die Höhe der Extrusion bestimmt. Das System fertigt dann eine Kopie des Basisstrichs an und verschiebt diese nach einer Methode, die der im letzten Kapitel vorgestellten stark ähnelt.

Anstatt alle Teile einer Extrusion separat auszuführen unterstützt das System automatische Silhouettenbildung. Sobald ein Teil eines zusammenhängenden Objekts ein anderes verdeckt, blendet das System den verdeckten Teil aus. Ein einfache zweidimensionale Methode zur Silhouettenberechnung existiert [5].

Da im Programm keinerlei Tiefeninformationen gespeichert werden, stellt es dem Benutzer die Möglichkeit des sogenannten Stapelns zur Verfügung. Hierbei kann verschiedenen überlappenden Objekten eine *Priorität* zugewiesen werden, wobei diese *Priorität* definiert, ob ein Objekt vor einem anderen liegt. Einziger Nachteil ist, das ein Objekt ein anderes nicht umschließen kann, denn da einem Objekt nur eine *Priorität* zugewiesen werden kann, ist es für das Objekt unmöglich, gleichzeitig vor und hinter einem anderen Objekt zu sein. In solchen Fällen muss das betreffende Objekt getrennt werden. Um die Benutzerfreundlichkeit nicht allzu sehr einzuschränken bietet ein *Gruppierungstool* die Möglichkeit, das getrennte Objekt als Gruppe zusammenzufassen. Innerhalb dieser *Gruppe* sind verschiedene *Prioritäten* erlaubt. Abbildung 4.2 zeigt den Irrgarten im Hampton Court Palace der durch Extrusion aus seinem Grundriss hervorgegangen ist. Des weiteren hat man sich die automatische Silhouettenbildung zu Nutze gemacht sowie Objekte zerteilt, um ihnen verschiedene Prioritäten zu geben.

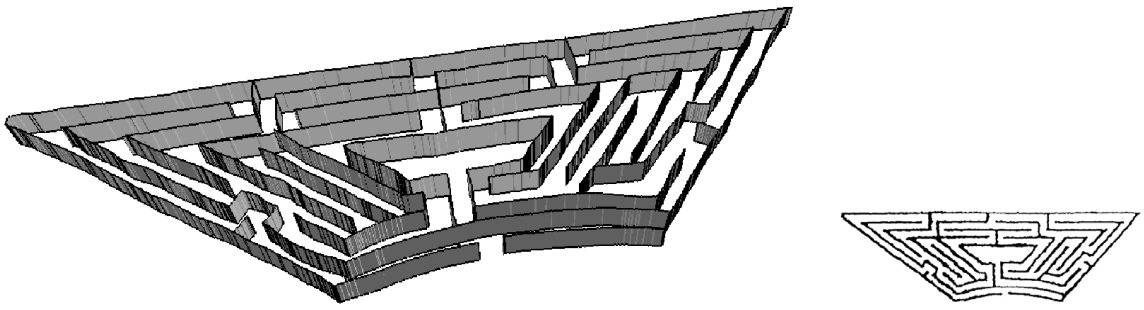

Abb. 4.2 Darstellung des Irrgartens im Hampton Court Palace durch Extrusion .

4.2 Schatten und Schattierungen

Perspektivische Zeichnungen, die *Schatten* und *Schattierungen* enthalten, wirken oftmals plastischer und damit realistischer als die besten, die nur Silhouetten enthalten. *Schattierungen* enthalten für den Betrachter Informationen über die Lage bzw. Ausrichtung der Fläche, während Schatten eine wichtige Rolle durch Ihre Fähigkeit relative Positionsinformationen zu enthalten spielen. Präzise Lichtsimulationen sind nur in dreidimensionalen Systemen möglich. Osama S. Tolba hat dennoch eine Möglichkeit gefunden, die beiden Techniken *Schattierungen* und *Schatten* in seinem System zu unterstützen.

Beleuchtungen und *Schattierungen* machen sich in Tolbas System das Vorhandensein der Normalen von Flächen zu Nutze. Der Benutzer kann Lichtquellen in Form von Lichtvektoren eingeben. Das System berechnet dann die Schattierungen wobei die beiden Schattierungstechniken *stippling* (tüpfeln) und *hatching* (schraffieren) unterstützt werden. Abbildung 4.3 zeigt an einem Beispiel wie die getüpfelte Schattierungen berechnet werden.

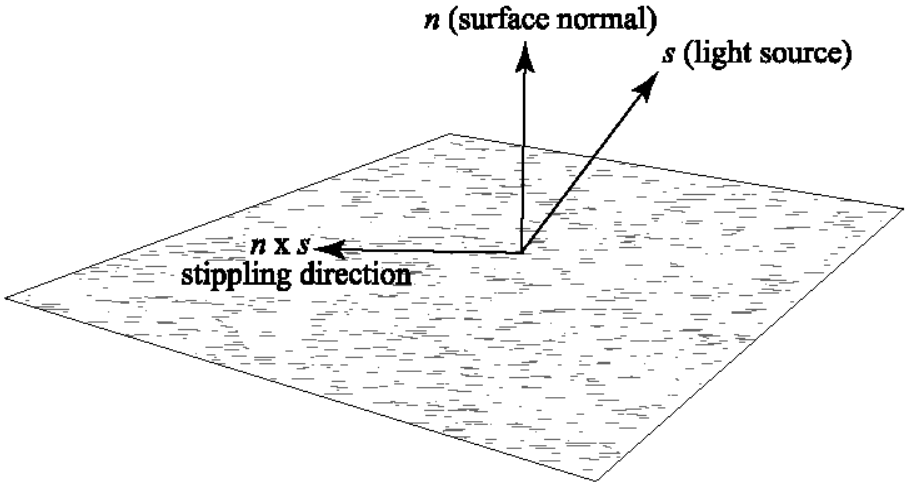

Abb. 4.3 Die Richtung der Schattierungen wird für *stippling* berechnet

Die Richtung der Striche gibt dem Benutzer beim *stippling* Informationen über die Ausrichtung der Fläche. Die Dichte der Striche wird nach der *Lambertian shading computation* durchgeführt während die Länge und Position der Striche zufällig bestimmt wird. Beim *hatching* hingegen gibt es nur vier verschiedene Schattierungsstufen. Welche Schattierung eine Fläche erhält hängt allein vom Winkel zwischen dem Normalenvektor der Fläche und dem Lichtvektor ab. Um den erhöhten Rechenaufwand beim *stippling* auszugleichen speichert das System die berechneten Schattierungen auf der Kugel. Wird das

Bild nun gedreht, wird auf die gespeicherten Daten auf der Kugel zurückgegriffen um die verschobenen Objekte zu stricheln. Obwohl die Schattierungsinformationen dadurch etwas ungenau werden, ermöglichen sie dem Benutzer eine adäquate Vorschau ohne das Flackern, das eine Neuberechnung der Schattierungen nach sich ziehen würde.

Die Berechnung der *Schatten* im System hält sich eng an die klassischen Techniken mit Konstruktionslinien. Der *Schatten* eines Objekts durch eine zentrale Lichtquelle, z.B. die Sonne, wird auf ein anders Objekt projiziert indem der Lichtvektor das schattenwerfende Objekt Punkt für Punkt abarbeitet. Hierfür ist lediglich der Normalenvektor des schattenwerfenden und das schattenempfangenden Objekts zusammen mit dem Lichtvektor nötig. Da im Programm keinerlei Tiefeninformationen enthalten sind, wird der *Schatten* zu Beginn immer an das schattenwerfende Objekt angehängt. Der Benutzer kann den *Schatten* dann an die richtige Stelle ziehen. Wird danach die Lichtquelle noch einmal verändert, so merkt sich das Programm die Position der Schatten und berechnet sie für die neue Lichtquelle gleich richtig. Abbildung 4.4 zeigt die erste Projektion des Schattens und die Verschiebung durch den Benutzer.

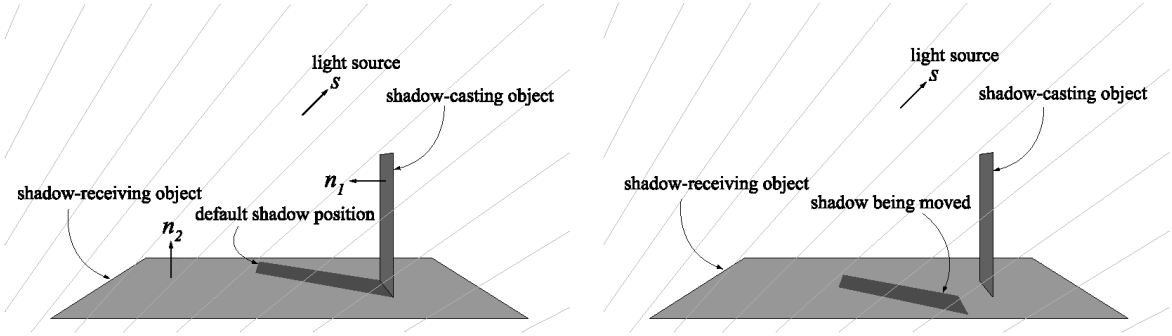

Abb. 4.4 Schattenberechung durch das System und Modifikation durch den Benutzer.

Zum Abschluss soll nochmal an Hand eines etwas aufwändigeren Beispiels die Mächtigkeit des Zeichenprogramms verdeutlicht werden. Die folgende Abbildung 4.5 zeigt einen Hof des Alhambra Palastes, auf dem die Bewegung der Sonne durch die sich verändernden *Schatten* veranschaulicht wird. Diese Sequenz von 3 Bildern wurde vom Programm automatisch berechnet.

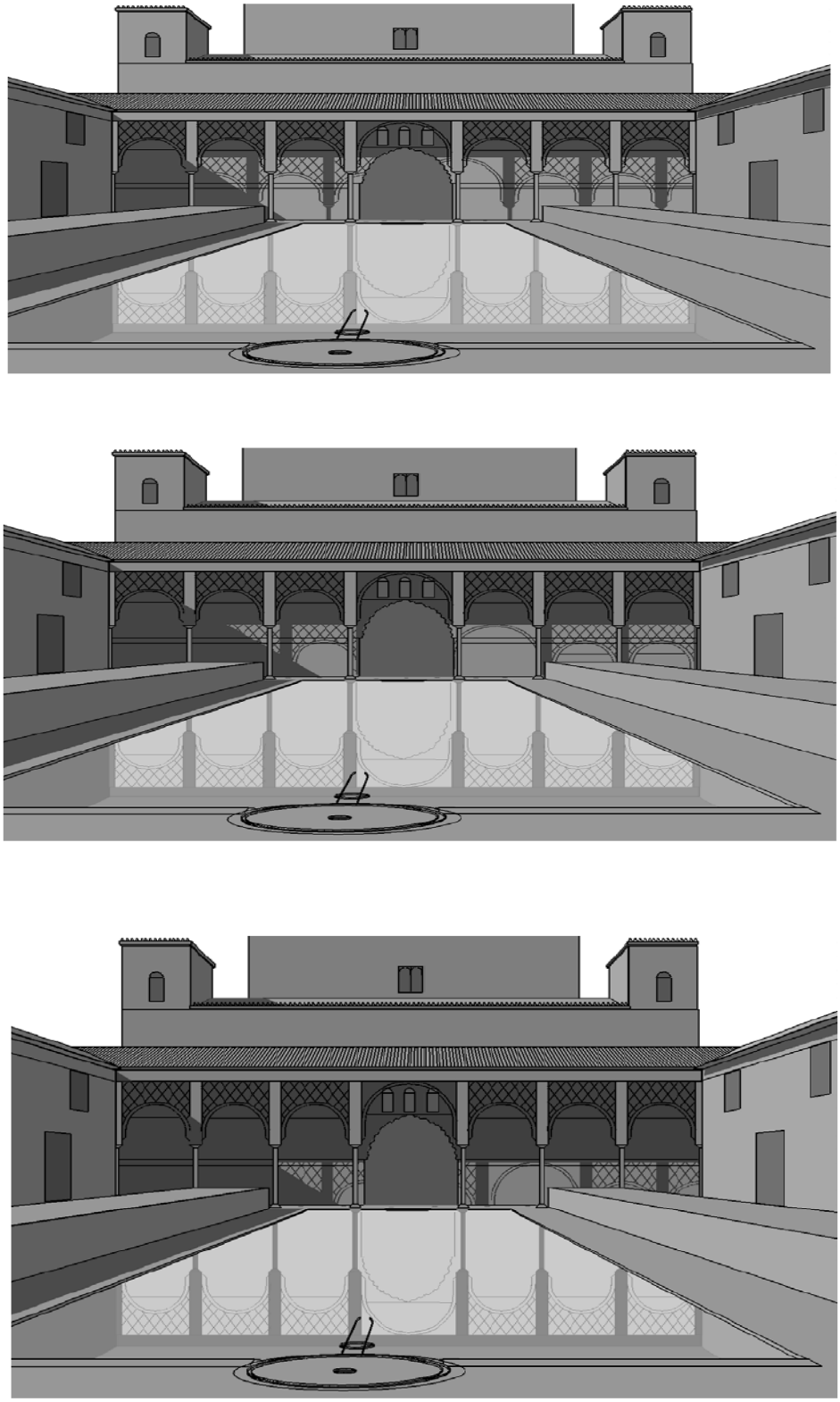

Abb. 4.5 Die Sonne wandert über einen Hof des Alhambra Palastes.

5. Einbindung anderer Medien

Das in dieser Arbeit vorgestellte Zeichenprogramm unterstützt die Integration von Zeichnungen auf Papier und Computergraphiken zu vielen Zeitpunkten während des Design-Prozesses. Einerseits haben handgezeichnete Entwürfe viele Vorteile durch ihre Spontanität, Flüssigkeit und ihre Portabilität, andererseits erhält der Computer Pluspunkte in den Bereichen der Wiederverwertbarkeit, des Editierens und des Verfeinerns. Daher verfolgt das Programm einen Ansatz, der die Vorteile beider Seiten vereint.
Außerdem unterstützt das System das traditionelle Medium der Fotografie und der gescannten Zeichnungen. Diese können einem Entwurf ein großes Maß an Realitätsnähe geben. Zuerst wird in diesem Kapitel die Integration von handgezeichneten Entwürfen hervorgehoben. Anschließend wird auf die Integration von Fotos und Scans eingegangen.

5.1 Papierzeichnungen

Oftmals benutzen Designer gerne die herkömmliche Art Skizzen anzufertigen, nämlich Papier und Stift. Der Grund darin liegt im einfachen Freihandzeichnen, das in seiner ursprünglichen Form nur von sehr wenigen Programmen unterstützt wird. Sogar Flachbettscanner schaffen es nicht, Freihandzeichnungen akzeptabel in den Computer zu transferieren, da sie die Zeichnungen oft rastern und dadurch eckig wirken lassen. Um diesem Problem Abhilfe zu schaffen, hat das System die Verwendung des *CrossPad portable digital notepad* [4] integriert. Dieses Pad zeichnet Striche auf, während der Benutzer mit Tinte auf einem Blatt Papier zeichnet. Diese Aufzeichnung kann dann in das System übertragen und dort weiterverarbeitet werden. Nachdem das Bild im Computer editiert wurde, kann es gedruckt werden und wiederum per Hand durch das CrossPad modifiziert werden. Abbildung 5.1 zeigt diesen Zyklus der abwechselnden Bearbeitung des Entwurfs auf dem Papier und dem Computer.

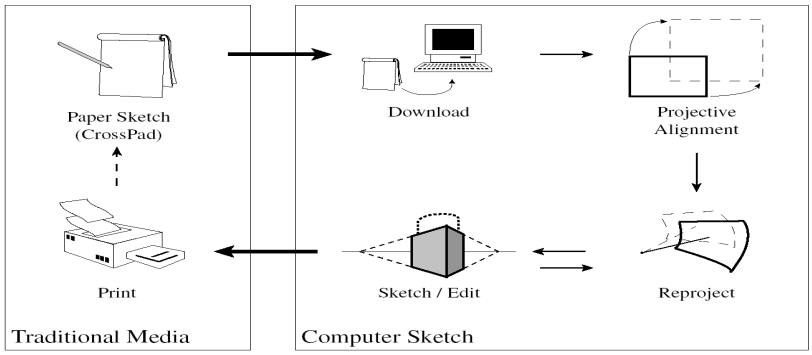

Abb. 5.1 Integration traditioneller Medien in einer Schleife.

5.2 Fotografien und Scans

Konventionelle Fotografien und eingescannte Bilder können im System als Texturen auf perspektivische Flächen abgebildet werden. Die Bilder und Fotos können zuvor mit herkömmlicher Bildbearbeitungssoftware modifiziert werden, um gewisse Teile des Bildes auszuschneiden oder nicht relevante Teile durchsichtig erscheinen zu lassen.

Natürlich können die mit Texturen versehenen Flächen anschließend mit dem Zeichenprogramm weiterhin modifiziert werden. Abbildung 5.2 zeigt den Entwurf für die Restaurierung des Peirene Brunnens. Auf der linken Seite sieht man den Entwurf, der auf der rechten Seite durch eine Fotografie des aktuellen Zustands ergänzt wurde.

Abb. 5.2 Entwurf für die Restaurierung des Pireine Brunnens [1]

6. Fazit

Der wohl größte Vorteil eines zweidimensionalen Ansatzes zur Repräsentation von dreidimensionalen Bildern ist die Zeitersparnis. Die Erzeugung einer einzelnen Ansicht mit einem handelsüblichen 3D Programm kann sehr zeitaufwändig sein. Außerdem kann der Designer keine freigezeichneten Elemente zu dem Bild hinzufügen. Normalerweise ist auch in zweidimensionalen Ansätzen das Erstellen eines dreidimensionalen Bildes sehr aufwändig, da die abzubildende Szene zumeist aus mehr als einer Perspektive dargestellt werden muss. Diesen offensichtlichen Nachteil überwindet Osama S. Tolba durch seinen Einsatz des Kugelmodells. Ein weiterer Vorteil ist, dass die Ausrüstung, die benötigt wird, um eine Zeichnung anzufertigen, sehr portabel ist. Zeichnungen können auf Papier oder einem CrossPad angefertigt werden und danach in das System integriert werden. Das unterstützt das Anfertigen von Skizzen vor Ort, was mit 3D Programmen nicht möglich ist.
Entscheidender Nachteil des zweidimensional Ansatzes ist das Fehlen von Tiefeninformationen. Dieser Nachteil kann zwar zum Teil durch das Zuordnen einer Priorität, welche die Sichtbarkeit regelt, ausgeglichen werden, doch beim Verschieben von ganzen Objektgruppen wird der Nachteil augenscheinlich. Der Verschiebungsvektor wird auf jedes Objekt der Gruppe separat angewandt. Dies kann zu inkonsistenten Ergebnissen führen, die der Benutzter in aufwändiger Handarbeit wieder modifizieren muss. Abbildung 6.1 zeigt eine solche Inkonsistenz die beim Verschieben eines Winkels, bestehend aus 2 Rechtecken, entstanden ist.

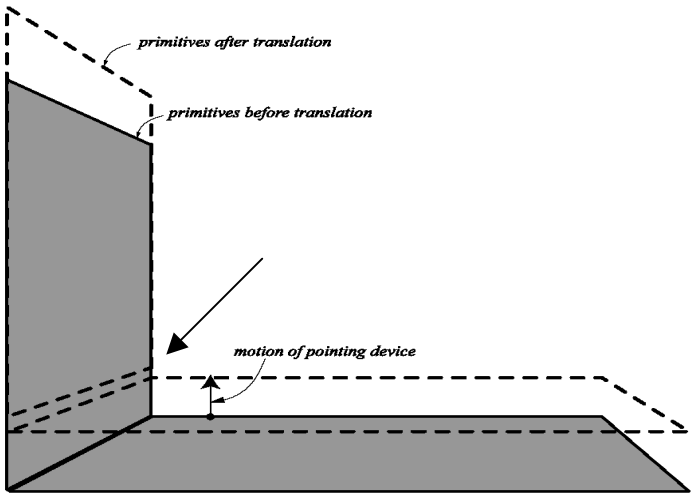

Abb. 6.1 Inkonsistenz die durch das Verschieben von Objektgruppen entsteht.

Literatur

[1] Osama S. Tolba. *A Projective Approach to Computer-Aided Drawing.* Master Thesis at the Massachusetts Institute of Technology. Juni 2001

[2] H.S.M. Coxeter. *The Real Projektive Plane*. McGrw-Hill, New York, 1949

[3] Jorge Stolfi. *Oriented Projective Geometry: a Framework fir Geometric Computations.* Academic Press, Boston, 1991.

[4] Cross Pen Computing Group. Portable Digital Notepad (no longer in production)

[5] S. C. Hsu, I. H. H. Lee, and N.E. Wiseman. Skeletal Strokes. In *UIST 93 Proceedings of the ACM SIGGRAPH & SIGCHI Symposium on User Interface Software & Technology*, November 1993

[6] Kenichi Kanatani. Computational projective geometry. *CVGIP: Image Understanding*, 54(3):333-348, 1991