

Optimierung des AR-Einrichtungsplaners KPS Click & design und Entwicklung einer Online-Version des Programms

Studienarbeit

im Studiengang Computervisualistik

vorgelegt von

Fabian Jäger

Betreuer: Herr Markus Schmitz, SHD KPS GmbH und Co. KG, Andernach
Herr Daniel Jäger, metaio GmbH, München

Koblenz, im März 2006



Aufgabenstellung für die Studienarbeit

Fabian Jager

(Matr.-Nr. 202120732)

Thema: Optimierung des AR-Einrichtungsplaners KPS Click&Design und Entwicklung einer Online-Version des Programms

Die SHD Holding GmbH hat sich auf EDV-gestützte Komplettlösungen im Möbelhandel spezialisiert und ist in diesem Bereich Marktführer. Bei dem Augmented Reality basierten Produkt KPS Click&Design besteht die Idee darin, dass per Mausklick dreidimensionale, virtuelle Möbelstücke in das Digitalfoto eines realen Raumes eingefügt und nach Belieben positioniert werden können. KPS Click&Design wurde von metaio Augmented Solutions in Garching bei München im Auftrag von SHD entwickelt.

Augmented Reality ist ein noch relativ junges Forschungsgebiet, das allmählich seinen Weg in kommerzielle Produkte findet. Außerdem werden immer mehr Applikationen heutzutage online angeboten. In dieser Arbeit soll versucht werden, Augmented Reality, Computergrafik und 3D-Online-Applikationen zu vereinigen.

Ziel der Arbeit ist es, die Darstellungsqualität in Click&Design in Zusammenarbeit mit metaio Augmented Solutions zu optimieren und aufgrund der gesammelten Erfahrungen eine Online-Version des Produktes zu entwickeln, die den Kunden ohne Installation einen Einblick in die Anwendung geben soll.

Schwerpunkte dieser Arbeit sind:

1. Einarbeitung in die verwendeten Technologien (ARToolkit, OpenGL, C++, C#)
2. Einarbeitung in die bestehenden Komponenten bei metaio Augmented Solutions, Verbesserung der Darstellungsqualität der virtuellen Möbelstücke
3. Ermittlung der Anforderungen und Erstellung eines Konzepts für die Online-Version
4. Implementierung der Online-Version
5. Dokumentation und Ausarbeitung

Die Studienarbeit wird bei der SHD Kreative Planungs-Systeme GmbH und Co. KG durchgeführt.

Betreuer: Herr Markus Schmitz, SHD KPS GmbH und Co. KG, Andernach

Herr Daniel Jäger, metaio Augmented Solutions GmbH, Garching bei München

Koblenz, den 4. Oktober 2005

- Prof. Dr. Stefan Müller-

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmitteln angefertigt habe. Diese Arbeit wurde in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Koblenz, 02.04.2006

Fabian Jager

Inhaltsverzeichnis

1	Einleitung	6
2	Grundlagen	7
2.1	Stand der Technik	7
2.2	Aufbau von Click & design aus Benutzersicht	8
2.3	Interne Architektur von Click & design	8
2.4	Verwendete Technologien	8
2.5	Zielgruppe	9
3	Anforderungsanalyse	10
3.1	Anforderungen von SHD	10
3.2	Anforderungen von metaio	10
3.3	Ziel der Arbeit	11
3.3.1	Anti-Aliasing	11
3.3.2	Schatten	12
3.3.3	Online-Version	12
4	Konzeption	13
4.1	Anti-Aliasing	13
4.1.1	Mehrmaliges Rendern mit Verschiebung und Überlagerung	13
4.1.2	Weichzeichnen der Kanten mit Shadern	13
4.1.3	Resampling von höherer Auflösung (Oversampling)	13
4.1.4	Bewertung und Auswahl	14
4.2	Schatten	14
4.2.1	Fake-Shadows	14
4.2.2	Shadow-Maps in VRML mithilfe von Shadern	15
4.2.3	Multitexturing	15
4.2.4	Planare Projektionsschatten	15
4.2.5	Bewertung und Auswahl	15

4.3	Online-Version	17
4.3.1	VRML/Javascript/HTML	17
4.3.2	Shockwave	17
4.3.3	ActiveX	17
4.3.4	Java-Applet	18
4.3.5	Bewertung und Auswahl	18
5	Implementierung	19
5.1	Anti-Aliasing	19
5.2	Schatten	20
5.2.1	Shadow Creator	20
5.2.2	Erweiterung von Click & design	22
5.3	Online-Version	23
6	Ergebnisse	28
6.1	Anti-Aliasing	28
6.2	Schatten	29
6.3	Online-Version	29
7	Zusammenfassung und Ausblick	31
A	Literaturverzeichnis	34
B	Abbildungsverzeichnis	35

1 Einleitung

Augmented Reality ist eine Technologie, die es erlaubt, virtuelle Objekte mit der realen Welt zu verbinden, indem sie in das Sichtfeld des Betrachters eingeblendet werden. Momentan sind nur wenige kommerzielle Anwendungen auf dem Markt, die auf dieser Methode basieren. So wird zum Beispiel bei TV-Sportübertragungen Augmented Reality zur Visualisierung von Abständen und Bestzeiten eingesetzt.

An der Universität Koblenz-Landau wird seit 2003 die Veranstaltung „Virtuelle Realität und Augmented Reality“ von der Arbeitsgruppe Computergrafik angeboten, in der unter anderem verschiedene Tracking-Verfahren und Ein-/Ausgabegeräte nach dem aktuellen Stand der Technik gelehrt werden.

Im Rahmen eines Kooperationstreffens zwischen der Arbeitsgruppe Computergrafik und Vertretern von SHD¹ kam es zur Vermittlung dieser Studienarbeit. Es ist die erste Studienarbeit an der Universität Koblenz-Landau, die von SHD betreut wird.

Ziel dieser Arbeit ist die Optimierung und Weiterentwicklung des SHD-Produktes Click & design. Es ist ein Möbelplanungsprogramm, welches Augmented Reality verwendet, um dem Benutzer eine virtuelle Neueinrichtung seiner eigenen Räume zu ermöglichen, ohne dass diese dreidimensional rekonstruiert werden müssen.

Immer mehr Anwendungen werden heutzutage dezentral eingesetzt. So ist es bereits möglich, auch von unterwegs seine E-Mails abzurufen oder Termine zu verwalten. Weil die Internetzugänge immer schneller werden und mittlerweile meist nicht mehr nach Zeiteinheiten abgerechnet wird, werden online basierte Applikationen ständig attraktiver. So ist meist keine Installation notwendig, die Anwendung kann mit persönlichen Daten auf verschiedenen Rechnern betrieben werden und Updates können vom Anbieter ohne Eingriff des Benutzers eingespielt werden. Das zweite Ziel der vorliegenden Arbeit wird durch die Frage bestimmt, in wie weit eine Portierbarkeit des Programms Click & design auf eine Online-Plattform möglich ist. Dies soll durch Implementierung eines Prototyps evaluiert werden.

¹<http://www.shd.de>

2 Grundlagen

2.1 Stand der Technik

Markerbasiertes, optisches Tracking hat sich in den letzten Jahren durch die fortschreitende Entwicklung hochauflösender Digitalkameras etabliert. So ist es möglich, auch bei kleiner Markergröße im Bild genaues Tracking durchzuführen. Ein großer Vorteil dieses Tracking-Verfahrens ist die hohe Verfügbarkeit der notwendigen Ausrüstung. So sind Digitalkameras heutzutage sehr verbreitet und Marker können bequem mit jedem handelsüblichen Drucker ausgedruckt werden. ARToolkit¹ wurde 1999 auf der SIGGRAPH Konferenz vorgestellt und wird mittlerweile als Sourceforge²-Projekt weiterentwickelt. Diese Software-Bibliothek stellt die gesamten benötigten Funktionen für die Entwicklung für Augmented Reality Anwendungen, die auf Markertracking basieren, zur Verfügung. Die Firma metaio³ aus München entwickelt seit 2003 ein eigenes Tracking-System mit dem Namen Unifeye SDK, das verschiedenste Tracking- und Rendering-Verfahren unterstützt.

Das Fraunhofer-Institut für Graphische Datenverarbeitung in Darmstadt⁴ veröffentlichte 2003 eine Presse-Mitteilung mit dem Titel „Virtuell Möbel rücken“ [igd, 2003], in der das Ziel gesetzt wurde, aus einem Foto die Geometrie des Raumes sowie die aktuelle Beleuchtung automatisch zu errechnen. Dieses Vorhaben wurde im Rahmen des europäischen Forschungsprojektes ARIS (Augmented Reality Image Synthesis) gemeinsam mit Partnern aus der Industrie gestartet.

Auch an der Universität Koblenz-Landau wird in dieser Richtung geforscht. Dipl. Inform. Thorsten Grosch befasst sich mit Augmentierter Bildsynthese. Bei diesem Verfahren sind fotorealistische Ergebnisse möglich, allerdings wird zurzeit als Input außer dem Foto noch ein grobes 3D-Modell des Raumes sowie eine Lightprobe benötigt. Dieses Vorgehen ist für den Einsatz beim Kunden nicht praktikabel, da die notwendige Ausrüstung dort in der Regel nicht vorhanden ist und der Aufwand für 3D-Rekonstruktion und Lightprobe zu groß ist.

¹<http://www.hitl.washington.edu/artoolkit/>

²<http://www.sourceforge.net>

³<http://www.metaio.com>

⁴<http://www.igd.fhg.de>

2.2 Aufbau von Click & design aus Benutzersicht

Nach der Installation hat der Benutzer die Möglichkeit, virtuelle Möbel in seiner Wohnung zu platzieren. Dabei geht er folgendermaßen vor:

1. Der Marker wird auf dem Boden des Zimmers positioniert. Dabei sollte auf die richtige Ausrichtung zur Wand geachtet werden.
2. Der Raum wird mit einer Digitalkamera fotografiert, wobei der Marker im Bild sein muss.
3. Nach dem Starten von Click & design kann das Digitalfoto geladen werden. Es stellt den Hintergrund für den zu planenden Raum dar und aus der Position und Verzerrung des Markers im Foto, wird die Position und Orientierung der Kamera errechnet.
4. Möbel werden mit dem entsprechenden Button in das Bild eingefügt und mit der Maus im Raum verschoben. Die Höhe und Drehung der Objekte kann über die Benutzer-Oberfläche eingestellt werden.
5. Der fertig geplante Raum kann als Screenshot und Bestückungsliste gespeichert werden.

2.3 Interne Architektur von Click & design

Click & design wurde von der Firma metaio aus München auf Basis der hauseigenen Plattform Unifeye SDK (Abbildung 2.1) entwickelt, in der Capturing (Bilderfassung), Tracking (Positions- und Orientierungsberechnung) und Visualisation (Darstellung der erweiterten Realität) vereinigt werden. Diese Bibliothek kann als OCX-Element in Anwendungen integriert werden. So lassen sich mit geringem Aufwand AR-Applikationen entwickeln.

2.4 Verwendete Technologien

Das Unifeye SDK verwendet den BS-Contact Renderer zur Visualisierung. Dieser von Bitmanagement Software GmbH⁵ aus Berg entwickelte Renderer basiert auf dem VRML-Konzept und erweitert dieses um einige Funktionen. Die Benutzeroberfläche von Click & design wurde in C# implementiert, was eine einfache Anpassung der Interaktionselemente erlaubt.

⁵<http://www.bitmanagement.de>

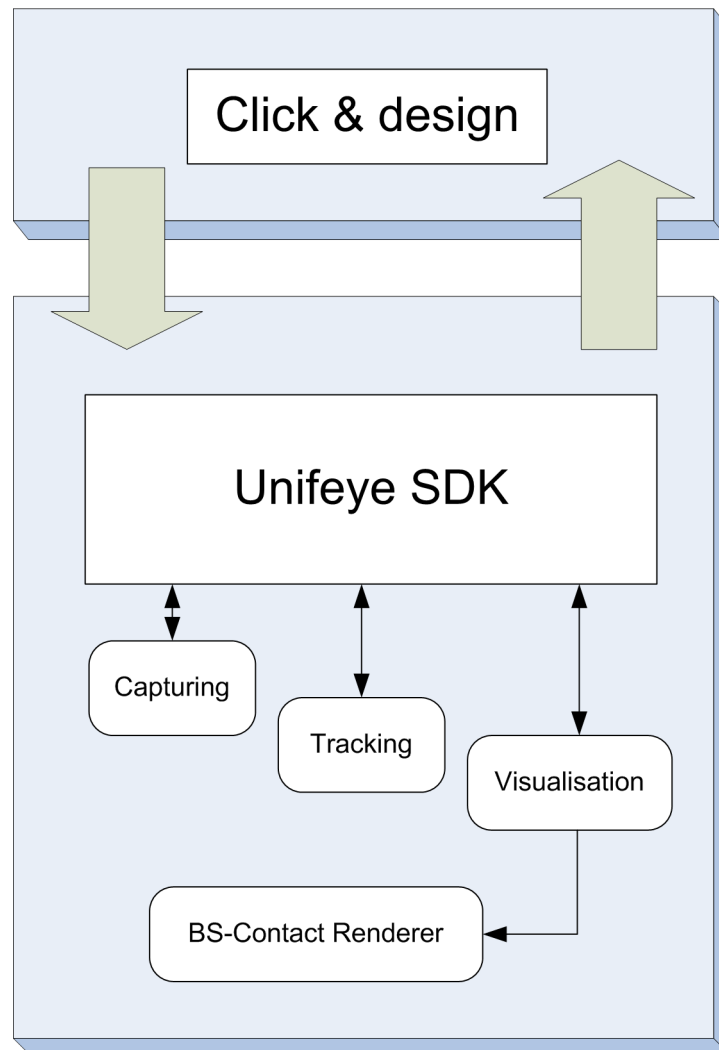


Abbildung 2.1: Interne Struktur von Unifeye SDK

2.5 Zielgruppe

Das Programm richtet sich an Endbenutzer, denen die Kaufentscheidung für Produkte eines Möbelhauses erleichtert werden soll, indem sie das Aussehen der Möbel vor der Bestellung in ihren eigenen Räumen virtuell ausprobieren können. Die Software, sowie die aktuelle Möbelkollektion des entsprechenden Möbelhauses werden auf CD zusammen mit dem normalen Katalog ausgeliefert. Es soll jedoch nicht außer Acht gelassen werden, dass das System aufgrund seiner Flexibilität auch in vollkommen anderen Bereichen, wie zum Beispiel zur Planung der Standorte von Plakatwänden in der Außenwerbung, zum Einsatz kommen kann.

3 Anforderungsanalyse

Das Programm Click & design entstand im Rahmen einer Kooperation von SHD Kreative Planungssysteme GmbH und der metaio GmbH. metaio ist dabei für die Entwicklung des Programmes verantwortlich und SHD übernimmt als Marktführer für Möbelindustrie-Software in Europa den Vertrieb.

Es wird ständig versucht, neue Einsatzfelder für das Programm zu finden. So werden derzeit Tests mit Hausfassaden, Autofelgen und Außenwerbung durchgeführt.

3.1 Anforderungen von SHD

SHD pflegt auf Fachmessen Kontakt zur Möbelindustrie und den Endkunden und hat so einen direkten Einblick in die Bedürfnisse und Erwartungen der Benutzer des Programms. Der wichtigste Aspekt ist und bleibt dabei die einfache Bedienbarkeit. Ziel ist außerdem, dass sich die virtuellen Möbelstücke möglichst realistisch in die reale Umgebung einbinden lassen. Ein großer Teil des Kapitals von SHD bildet die Erstellung und Verwaltung der 3D-Modelle von Möbeln. Deshalb werden in der Offline-Version die normalen VRML-Dateien in das eigens entwickelte verschlüsselte Dateiformat KPSCD umgewandelt, um zu verhindern, dass andere als von SHD erstellte Objekte in dem Programm verwendet werden. Diese Absicherungsmaßnahme soll insbesondere hinsichtlich der Online-Version fortgeführt werden.

3.2 Anforderungen von metaio

Weil die Integration von neuen Funktionen in Zusammenarbeit mit metaio geschieht, werden von dieser Seite auch einige Anforderungen gestellt:

- Einhaltung der firmeneigenen Qualitätsrichtlinien
- Möglichst einfache Integration der Erweiterungen in das bestehende System
- Enge Zusammenarbeit während der Integration

Zur Vorbereitung der Studienarbeit wurde ein zweiwöchiges Praktikum bei metaio absolviert, bei dem der grundlegende Aufbau und die Funktionsweise von Click & design sowie Unifeye SDK erarbeitet wurden.

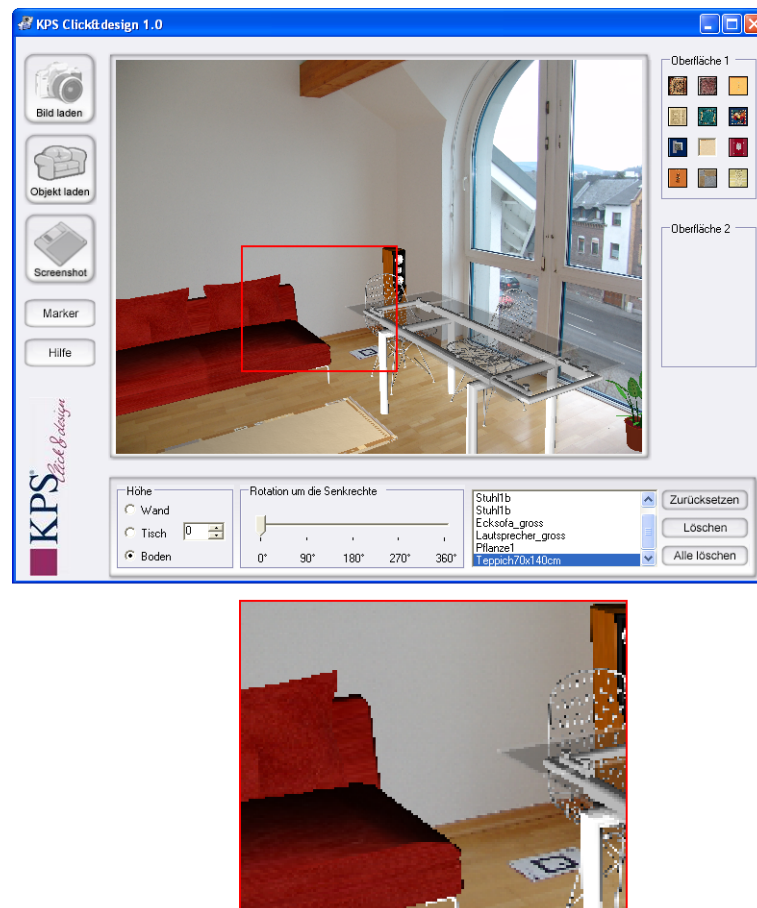


Abbildung 3.1: Aliasing in Click & design

3.3 Ziel der Arbeit

Nach Absprache mit SHD, metaio und Herrn Prof. Dr. Müller wurden folgende Teilaufgaben als Ziele der Arbeit festgelegt:

3.3.1 Anti-Aliasing

Je nach verwendeter Grafikhardware sind die Kanten der virtuellen Objekte zackig und treppenförmig (Abbildung 3.1). Der Grund liegt darin, dass bislang nur hardware-basiertes Anti-Aliasing zum Einsatz kommt und einige Systeme (speziell Bürocomputer) diese Funktion noch nicht unterstützen. Ziel ist es also, ein Anti-Aliasing zu entwickeln, das hardware-unabhängig die Kanten der virtuellen Objekte glättet und so auf allen Systemen eine zufriedenstellende Darstellungsqualität erzielt.

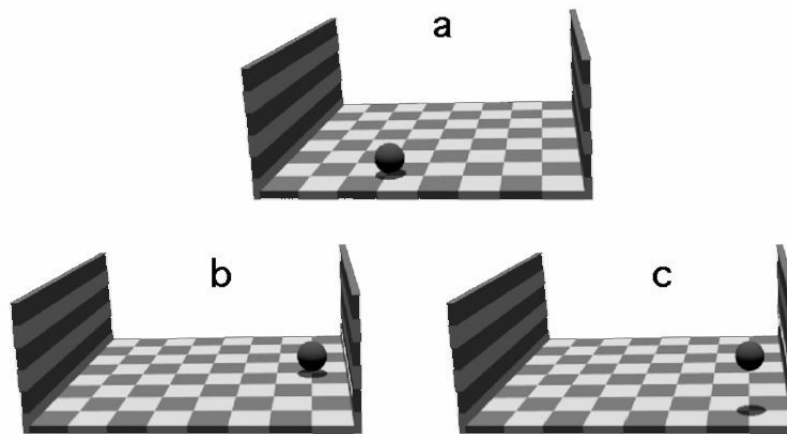


Abbildung 3.2: „ball-in-a-box“-Szenario von Kersten, Quelle: [kersten, 1997]

3.3.2 Schatten

Schatten sind sehr wichtig, um sich die Position virtueller Objekte in einer Umgebung vorstellen zu können. So kam Daniel Kersten 1997 [kersten, 1997] zu dem Schluss, dass Schattenwurf der Hauptfaktor ist, um die Beziehungen zwischen Objekten zu erkennen. Abbildung 3.2 zeigt das so genannte „ball-in-a-box“-Szenario. In Fall b befindet sich der Ball im Bild an der selben Stelle wie in Fall c und ist auch genauso groß. Allerdings werden durch die verschiedenen Schatten völlig unterschiedliche Positionen impliziert. Würden die Schatten fehlen, könnte der Betrachter nicht identifizieren, ob sich der Ball vorne und oben oder hinten und unten befindet. Im Rahmen der Studienarbeit soll deshalb ein Verfahren entwickelt werden, das die Objekte mit einfachen Schatten versieht, damit der Benutzer den Abstand vom Boden besser einschätzen kann. Schatten tragen außerdem dazu bei, dass die auf dem Boden stehenden virtuellen Objekte durch die fehlenden Abdunkelungen an den unteren Kanten nicht mehr „schwebend“ wirken.

3.3.3 Online-Version

Da die Software hauptsächlich bei Endbenutzern zum Einsatz kommt, ist die Anzahl der Installationen sehr groß. Deshalb sollen die Kosten pro Endnutzer und der Installationsaufwand so gering wie möglich gehalten werden. Hier liegt es nahe, sich Gedanken um eine Online-Version des Programms zu machen. Eine Verteilung der Software wäre dann wesentlich günstiger und schneller möglich und der Installationsaufwand für die wäre Kunden geringer. Die Online-Version könnte vom Möbelhäusern flexibel eingesetzt werden um per Internet neue Kunden zu gewinnen. In dieser Studienarbeit soll deshalb die Umsetzbarkeit einer Online-Version von Click & design untersucht werden.

4 Konzeption

Während der Themafindung war noch nicht bekannt, dass das Unifeye SDK zur Visualisierung den externen Renderer BS-Contact verwendet. Das Problem besteht hierbei darin, dass metaio zwar durch die Kooperation mit Bitmanagement indirekt an der Weiterentwicklung von BS-Contact mitwirken kann, jedoch keinen unmittelbaren Zugriff auf den Quellcode hat.

4.1 Anti-Aliasing

Als Methoden, ein Anti-Aliasing auch auf Systemen ohne hardwarebasiertes Anti-Aliasing zu ermöglichen wurden folgende Verfahren in Betracht gezogen:

4.1.1 Mehrmaliges Rendern mit Verschiebung und Überlagerung

Die klassische Methode, Kanten eines gerenderten Bildes in der Computergraphik weichzuzeichnen ist, die Szene mehrmals unter leicht verschobenen Kamerapositionen zu rendern. Dabei werden die gerenderten Bilder im Accumulationbuffer gesammelt und akkumuliert ausgegeben. Dieses Verfahren ist sehr einfach zu implementieren, wenn man Zugriff auf die OpenGL-Rendering-Ebene der betreffenden Applikation hat.

4.1.2 Weichzeichnen der Kanten mit Shadern

Auf Systemen, die programmierbare Shader zur Verfügung stellen ist es möglich, diese zu verwenden, um Kanten weichzeichnen. Dieses Verfahren ist sehr schnell, da die Berechnungen auf der Grafikhardware stattfinden.

4.1.3 Resampling von höherer Auflösung (Oversampling)

Eine weitere Methode zur Glättung der Kanten besteht darin, die Szene in einer höheren Auflösung zu rendern, als sie später dargestellt wird. Hier sind bereits ab einer Auflösungsvergrößerung von 50-100% gute Ergebnisse zu erreichen.

4.1.4 Bewertung und Auswahl

Durch die Verwendung des externen Renderers BS-Contact ist der direkte Eingriff in das Rendering nicht möglich, die zur Verfügung gestellten Funktionen sind sehr spezialisiert für bestimmte Anwendungsfälle vorgesehen. So stellt der Renderer zum Beispiel eine Funktion zur Verfügung, mit dem der Szenengraph verändert werden kann und zusätzliche Modelle in die Szene geladen werden können. Diese Objekte können ausgewählt und verschoben werden, eine Löschen-Funktion ist vom Renderer jedoch nicht vorgesehen und musste deshalb auf anderem Weg implementiert werden.

BS-Contact stellt eine Schnittstelle zur Verfügung, mit der eigene Shader verwendet werden können. Allerdings ist Grafikkhardware, die programmierbare Shader unterstützt auch in der Lage, hardwarebasiertes Anti-Aliasing zu betreiben. Durch die Implementierung eines Shaders würde jedoch keinerlei Mehrwert erreicht werden, da diese Variante bei Systemen ohne programmierbare Shadereinheiten nicht funktioniert.

Was als Möglichkeit bleibt ist das Rendern in höherer Auflösung und das anschließende „Runterskalieren“, wodurch eine Kantenglättung erreicht wird. Das Problem besteht hier darin, den Renderer dazu zu bringen, die Szene in höherer Auflösung zu rendern, als sie später angezeigt wird. Hier wurde als einzige Möglichkeit die Screenshot-Funktion von BS-Contact ins Auge gefasst, mit der Bilder in hoher Auflösung exportiert werden können. Dieser Funktion wird als Parameter ein Dateipfad mit Dateiname übergeben. Es ist nicht möglich, die berechneten Bild-Daten direkt abzugreifen, ohne den Umweg über das Dateisystem zu nehmen. Da dies jedoch die einzige Möglichkeit ist, die Szene in höherer Auflösung zu rendern ohne den Renderer zu modifizieren, fiel die Entscheidung zugunsten dieser Lösung.

4.2 Schatten

4.2.1 Fake-Shadows

Unter Fake-Shadows versteht man Schatten, die als einfache Flecken unter den Objekten platziert werden. Eine Optimierung besteht darin, statt einfacher Flecken senkrechte Projektionen der Objekte auf dem Boden einzublenden. Die Position dieser Silhouetten auf dem Boden kann durch den Lichteinfallswinkel variieren, wobei bei zu flachem Lichteinfall nicht mehr die korrekte Silhouette angezeigt werden würde, da diese für senkrechten Lichteinfall berechnet wurde. Es lassen sich jedoch bereits mit der vereinfachten Variante, dass die Silhouette unabhängig vom Lichteinfall unter dem Objekt eingeblendet wird sehr realistisch wirkende Ergebnisse erzielen.

4.2.2 Shadow-Maps in VRML mithilfe von Shadern

Beim Shadow-Maps-Verfahren wird die Szene einmal aus Sicht der Lichtquelle gerendert und so ein Tiefenbild der Szene erzeugt, das in einer depth map zwischengespeichert wird. Im zweiten Rendering-Schritt aus Kameraperspektive kann dann festgestellt werden, ob sich ein Punkt im Schatten befindet, indem mithilfe der depth map die Sichtbarkeit des Punktes für die Lichtquelle berechnet wird. Dazu wird der Tiefenwert aus der depth map mit dem Tiefenwert des Punktes aus Sicht der Lichtquelle verglichen. Ist der Tiefenwert aus der depth map kleiner, liegt der Punkt im Schatten.

4.2.3 Multitexturing

Der BS-Contact-Renderer unterstützt Multitexturing. So ist er in der Lage, eine virtuelle Welt in eine Textur zu rendern und diese Textur dann wieder zum Texturieren von Objekten zu verwenden. Mit dieser Funktionalität ist es möglich, Schatten der aktuellen Welt zu erzeugen, indem die Szene wie beim Shadow-Maps-Verfahren aus Sicht der Lichtquelle in eine Textur gerendert wird und diese Textur dann auf der Grundebene dargestellt wird. Es gibt bereits Tech-Demos dieser Funktionalität, die zeigen, dass diese Funktion echtzeitfähig ist.

4.2.4 Planare Projektionsschatten

Benjamin Blachnitzky hat in seiner Diplomarbeit [blachnitzky, 2004] das Verfahren von planaren Schatten in VRML umgesetzt. Dabei werden die Vertices des schattenwerfenden Objektes zwischengespeichert und daraus sowie der Position der Lichtquelle die projizierten Punkte berechnet. Diese bilden das Schattenpolygon und werden mit einem halbdurchsichtigen dunklen Material versehen, um einen Schatten zu simulieren.

4.2.5 Bewertung und Auswahl

Aus den Ergebnissen von [blachnitzky, 2004] kann entnommen werden, dass mit Fake-Shadows gute Resultate erzielt werden. Das Problem bestand darin, diese Fake-Shadows zu erzeugen. Dazu wurde in [blachnitzky, 2004] folgendes Verfahren vorgestellt:

- 3D-Modell in einem VRML-Viewer so positionieren, dass eine senkrechte Draufsicht gegeben ist
- Screenshot dieser Ansicht in ein Bildbearbeitungsprogramm laden
- Nachzeichnung der Form des Objekts für den Schatten
- Filter verwenden, um Schatten mit weichen Kanten zu versehen



Abbildung 4.1: Planare Schatten von Benjamin Blachnitzky, Quelle: [blachnitzky, 2004]

- Export als PNG-Datei mit Alpha-Kanal

Dieses Vorgehen erfordert einige manuelle Tätigkeiten, welche vom Umfang her nicht praktikabel sind.

Die anderen Möglichkeiten sind zwar dynamischer, jedoch sind die Anforderungen an die Hardware größer und die Interaktion wird langsamer. So wurde in [blachnitzky, 2004] festgestellt, dass bei der Erzeugung von planaren Schatten allein das Parsen der Objektkoordinaten für ein Modell auf einem Testrechner 7 Sekunden dauerte und dieses Verfahren damit nicht echtzeitfähig ist.

Dadurch, dass die Punktdichte an einigen Stellen des Objektes höher ist, sind dort bei den projizierten Punkten mehrere Polygone überlagert (Abbildung 4.1). Aufgrund der Halbtransparenz zeichnet sich so die Struktur des Objektes innerhalb des Schattens ab. Ein weiteres Problem besteht darin, dass zu jedem Objektpunkt ein projizierter Punkt existiert, sich die zu rendernde Vertex-Anzahl also verdoppelt. In der Testapplikation zeigt sich das durch merklich langsamere Interaktionsmöglichkeiten mit dem Objekt. Eine Verwendung dieses Verfahrens in der Anwendung wurde deshalb ausgeschlossen.

Da das einzig störende an den Fake-Shadows die aufwendige Erstellung war, bestand die Herausforderung darin, eine einfachere, schnellere Möglichkeit zu finden, um dieses Verfahren in der Praxis einsetzen zu können. So kam die Idee auf, ein Programm zu entwickeln, das die Erzeugung von Fake-Shadows aus den vorhandenen 3D-Objekten im Vorhinein vereinfacht und beschleunigt. Die erstellten Fake-Shadows sollten dann zusammen mit dem 3D-Modell des jeweiligen Möbelstückes abgespeichert werden. Nach dem Laden des Objektes in Click & design sollte sich die Schattenebene dynamisch verhalten, also sich mit dem Objekt drehen und bewegen lassen, aber auf der Grundebene bleiben, wenn das Objekt angehoben wird. Hier wird wie oben beschrieben mit der vereinfachten Methode gearbeitet, bei der das Licht stets von oben kommt. Deshalb wandert der Schatten beim Anheben des Objektes nicht auf der Grundebene.

4.3 Online-Version

Die Umsetzbarkeit des Programmes Click & design als Online-Version wurde mit der Implementierung eines Prototypen überprüft. Das Vorgehen ist dabei an das Prototypisierungsmodell angelegt, nach dem bei einer schwer überschaubaren Problemstellung als erstes ein Prototyp entwickelt wird, um die Realisierbarkeit zu prüfen und auftretende Probleme besser abschätzen zu können. Im Anschluss folgt dann im Falle einer erfolgreichen Entwicklung die Neuimplementierung der Applikation.

Fest steht, dass es bei einer Online-Version keinen Sinn macht, das Tracking lokal auf dem Client-Computer durchzuführen, da pro Foto nur einmal getrackt werden muss (die Kamera bewegt sich nicht mehr), was sich sehr gut auf den Server verlagern lässt und so einen großen Teil des Programmes bei der Client-Applikation einspart.

4.3.1 VRML/Javascript/HTML

Die erste Möglichkeit, die in Betracht gezogen wurde war die Verwendung von VRML zur 3D-Visualisierung, eingebettet in eine HTML-Oberfläche, die dynamische Elemente mithilfe von Javascript zur Verfügung stellt. Dadurch, dass die Objekte bereits als VRML vorliegen, müssten diese nicht konvertiert werden, um in der Online-Version verwendet werden zu können. Für die Verwendung der Applikation wäre es dann notwendig, ein VRML-Plugin für den Webbrowser installiert zu haben.

4.3.2 Shockwave

In der Verbindung von 3D und Internet ist der Shockwave-Standard, der von Macromedia entwickelt wurde, bereits stark verbreitet. Diese Technologie hat den Vorteil, dass Plugins für nahezu jede Plattform und jeden Browser zur Verfügung stehen und so eine große Kompatibilität gegeben ist. Durch die Monopolstellung von Macromedia existiert nur ein Plugin für das Shockwave-Format auf dem Markt. So kann sichergestellt werden, dass die Darstellung auf allen Systemen gleich ist.

4.3.3 ActiveX

Da das Unifeye SDK als OCX zur Verfügung steht, kann es nicht nur flexibel in Applikationen eingebunden werden, sondern auch als ActiveX-Element in eine HTML-Seite integriert werden. So ist es möglich, sämtliche Funktionen, die das Unifeye SDK zur Verfügung stellt auch in einem Webbrowser zu verwenden.

4.3.4 Java-Applet

Java stellt zusammen mit der Erweiterung Java3D auch eine Plattform für Rendering zur Verfügung. Java-Applets lassen sich genauso wie ActiveX-Elemente in eine Webseite einbinden. Hier ist zur Verwendung eine Installation der JRE¹ erforderlich.

4.3.5 Bewertung und Auswahl

Bei der Einbettung von VRML in HTML wäre ein VRML-Renderer als Browser-Plugin wie zum Beispiel BS-Contact erforderlich. Ein Problem besteht darin, dass die VRML-Dateien unkodiert übertragen werden müssten um von VRML-Plugins gelesen werden zu können, das KPSCD-Format könnte nicht eingesetzt werden. Bei HTML und Javascript sind außerdem alle Quelltexte ohne Hindernisse einsehbar. Dadurch würde sich die Applikation leicht kopieren und modifizieren lassen. Das ist jedoch ein Ausschlusskriterium, weshalb diese Methode nicht in Frage kommt.

Shockwave ist ein kleines Plugin, das eine sehr hohe Verbreitung hat und auf verschiedensten Plattformen und Browsern gut funktioniert. Es stellt zudem vielfältige Schnittstellen zur Client-Server-Kommunikation zur Verfügung und mit dem Programm Director bietet Macromedia eine grafische Entwicklungsumgebung an. Damit ist Shockwave flexibler als ActiveX-basierte Programme, die meist nur im Internet Explorer mit niedrigen Sicherheitseinstellungen korrekt funktionieren. Bei der für Java-Applets notwendigen Java Runtime Environment verhält es sich genau entgegengesetzt. Hier hat Microsoft vor einigen Jahren die Auslieferung mit dem Internet Explorer gestoppt und so die Verwendung von Java-Applets in diesem Browser erschwert. Im Übrigen ist die Java Runtime Environment mit 16 MB in der aktuellen Version sehr groß.

Wegen der genannten Vorteile und weil SHD eine Lizenz von Macromedia Director besitzt, fiel die Entscheidung für die Online-Version auf die Implementierung als Shockwave-Applikation.

¹Java Runtime Environment

5 Implementierung

5.1 Anti-Aliasing

Wie bereits in der Konzeption beschrieben sind die Lösungsmöglichkeiten für das Anti-Aliasing-Problem stark eingeschränkt. Um eine gerenderte Ansicht in einer höheren Auflösung zu generieren stellt BS-Contact eine Screenshot-Funktion zur Verfügung, mit der entweder eine BMP- oder eine JPEG-Datei exportiert werden kann. Um dieses Bild wieder darstellen zu können wird die exportierte Datei geladen und über der 3D-Szene angezeigt. Es soll folgendermaßen vorgegangen werden:

1. Abspeichern der gerenderten Szene als Bild-Datei in höherer Auflösung
2. Laden der Bild-Datei
3. Anzeigen des geladenen Bildes vor der Szene in angepasster Größe
4. Ausblenden des statischen Bildes, sobald eine Interaktion stattfindet

Da das Rendern der Szene in eine Datei und das anschließende Laden und Anzeigen je nach System einige Sekunden dauern kann, ist dieses Verfahren nicht echtzeitfähig. Deshalb wurde entschieden, das optimierte Rendering erst durch den Klick auf einen Button zu initiieren.

Die Ladegeschwindigkeit bei Verwendung eines BMP-Bildes ist wegen der nicht vorhandenen Komprimierung schneller, deshalb wurde dieses Dateiformat zum Zwischenspeichern ausgewählt. Damit sich die Dateien nicht anhäufen, wird die Datei im Template-Verzeichnis des Systems abgelegt und sofort nach dem Laden wieder gelöscht.

Eine Resample-Funktion zur Skalierung des Bildes in eine kleinere Auflösung wird von C# zur Verfügung gestellt. Der Algorithmus zum Rendern in hoher Auflösung und Anzeige über dem Renderfenster ist in Abbildung 5.1 zu finden.

Jedes Interaktionselement, mit dem Änderungen an der 3D-Darstellung vorgenommen werden können, musste im Anschluss um einen Funktionsaufruf zum Deaktivieren der statischen Darstellung erweitert werden, weil die statische Ansicht nach dem Ändern der 3D-Szene nicht mehr aktuell ist.

Ein Problem entstand dadurch, dass zur Verschiebung der Objekte auch innerhalb der 3D-Welt eine Interaktion möglich ist. Damit die Objekte mit der Maus „festgehalten“ werden können,

```

//save scene as screenshot
axAS_AR_Controller1.saveScreenshot(tempFile);

//load image from temp-image
System.Drawing.Image tempImage = System.Drawing.Image.FromFile(tempFile);

//create new bitmap to put in the resized image
System.Drawing.Image myNewImage =
new Bitmap(aaPicture.Width, aaPicture.Height, tempImage.PixelFormat);
Graphics myGraphic = Graphics.FromImage(myNewImage);

//adjust resize-quality
myGraphic.CompositingQuality = System.Drawing.Drawing2D.CompositingQuality.HighQuality;
myGraphic.SmoothingMode = System.Drawing.Drawing2D.SmoothingMode.HighQuality;
myGraphic.InterpolationMode = System.Drawing.Drawing2D.InterpolationMode.HighQualityBicubic;

//create rectangle for resized image
Rectangle myRectangle = new Rectangle(0, 0, aaPicture.Width, aaPicture.Height);
myGraphic.DrawImage(tempImage, myRectangle);

//put resized image into picture container
aaPicture.Image = myNewImage;

```

Abbildung 5.1: Rendering in Datei und resample-Darstellung (C#)

ist es erforderlich, dass das MouseDown bei einem Klick noch einmal simuliert wird, da das erste MouseDown von der davor liegenden statischen Darstellung „verschluckt“ wird. Wenn ein Objekt verschoben werden soll, bewirkt ein Klicken mit der linken Maustaste auf die 3D-Szene, dass die statische Ansicht ausgeblendet wird und das Objekt bewegt werden kann.

5.2 Schatten

Die Umsetzung der Fake-Shadow-Funktionalitäten gliederte sich in zwei Schritte:

1. Entwicklung eines Programmes zur Schattenerstellung: Arbeitsname Shadow Creator
2. Erweiterung von Click & design, damit Schatten geladen und die Intensität verändert werden kann. Die Schatten sollen am Boden bleiben, wenn das Objekt angehoben wird und sich mitbewegen, wenn das Objekt gedreht oder verschoben wird.

5.2.1 Shadow Creator

Das Programm Shadow Creator soll aus vorhandenen 3D-Modellen der Möbel Fake-Shadows erstellen, die als Grafik gespeichert werden und später als Textur unter dem Objekt den Schatten simulieren.

Die Wahl der Entwicklungsumgebung fiel auf Microsoft Visual Studio .net unter Verwendung der Programmiersprache C#, weil auch Click & design selbst damit implementiert wurde. Die

Hauptidee besteht darin, dass nicht einfach eine Ansicht von oben zur Weiterverarbeitung aufgenommen wird, sondern stattdessen der Z-Buffer ausgelesen wird. So kann auf einfache Weise die Silhouette des Objektes erzeugt werden. Die Kamera wird dabei unter dem Objekt positioniert, um die Silhouette der Unterseite des Objektes zu extrahieren. Auch in diesem Programm wird für das Laden und Darstellen der VRML-Dateien der BS-Contact-Renderer verwendet.

Die Erstellung eines Schattens besteht aus mehreren Schritten:

1. Laden eines Modells und Darstellung im OCX-Fenster von BS-Contact
2. Positionierung der Kamera unter dem Objekt
3. Auslesen des OpenGL-Tiefenpuffers [opengl, 2004] mithilfe von CsGL¹ (Grafikbibliothek für OpenGL in C#) und Erstellen der Silhouette. Wenn an einer Stelle ein Wert im Tiefenpuffer steht wird ein schwarzes Pixel gesetzt, ansonsten ein Transparentes.
4. Wenn sich an einer Randlinie der Silhouette schwarze Punkte befinden, wird die Kamera weiter vom Objekt entfernt und zu Punkt 3 zurückgesprungen (das Objekt war nicht komplett im Bild). Die Breite der Randlinie ist abhängig vom Radius des Filters, damit genügend Platz für die spätere Vergrößerung der Silhouette durch den Weichzeichner vorhanden ist.
5. Die Silhouette wird mithilfe eines Mittelwertfilters in einen weichen Schatten verwandelt.
6. Abspeichern des Schattens als PNG-Datei. Diese ist komplett schwarz, die Schattenkontur wird nur mithilfe des Alpha-Wertes erreicht, der 256 Stufen erlaubt.
7. Abspeichern des veränderten VRML-Modells, so dass die PNG-Datei als Textur einer Fläche unter dem Objekt geladen wird. Hier wird die Größe der Fläche entsprechend des Abstandes der Kamera bei Akquirierung der Silhouette skaliert.

Die Aufteilung der Benutzeroberfläche gliedert sich in Visualisierungselemente (Darstellung des Modells, der Silhouette und der Schattentextur), sowie Interaktionselemente, die oben rechts mithilfe von Reiterelementen gruppiert wurden (Abbildung 5.2).

Neben der Möglichkeit, die einzelnen Schritte der Schattenerstellung manuell durchzuführen wird auch eine Batch-Funktion zur Verfügung gestellt, mit der automatisiert für komplette Verzeichnisse Modelle mit Schatten generiert werden können (Abbildung 5.3). Der verwendete Filter kann unter dem Reiter „Einstellungen“ individuell konfiguriert werden, um die Weichheit und Qualität sowie die Berechnungszeit des Schattens zu kontrollieren.

Der Kernschatten ist stets komplett schwarz, damit der gesamte Bereich der 256 Transparenzabstufungen ausgenutzt werden kann. Eine Abschwächung des Schattens und die daraus

¹<http://csgl.sourceforge.net>

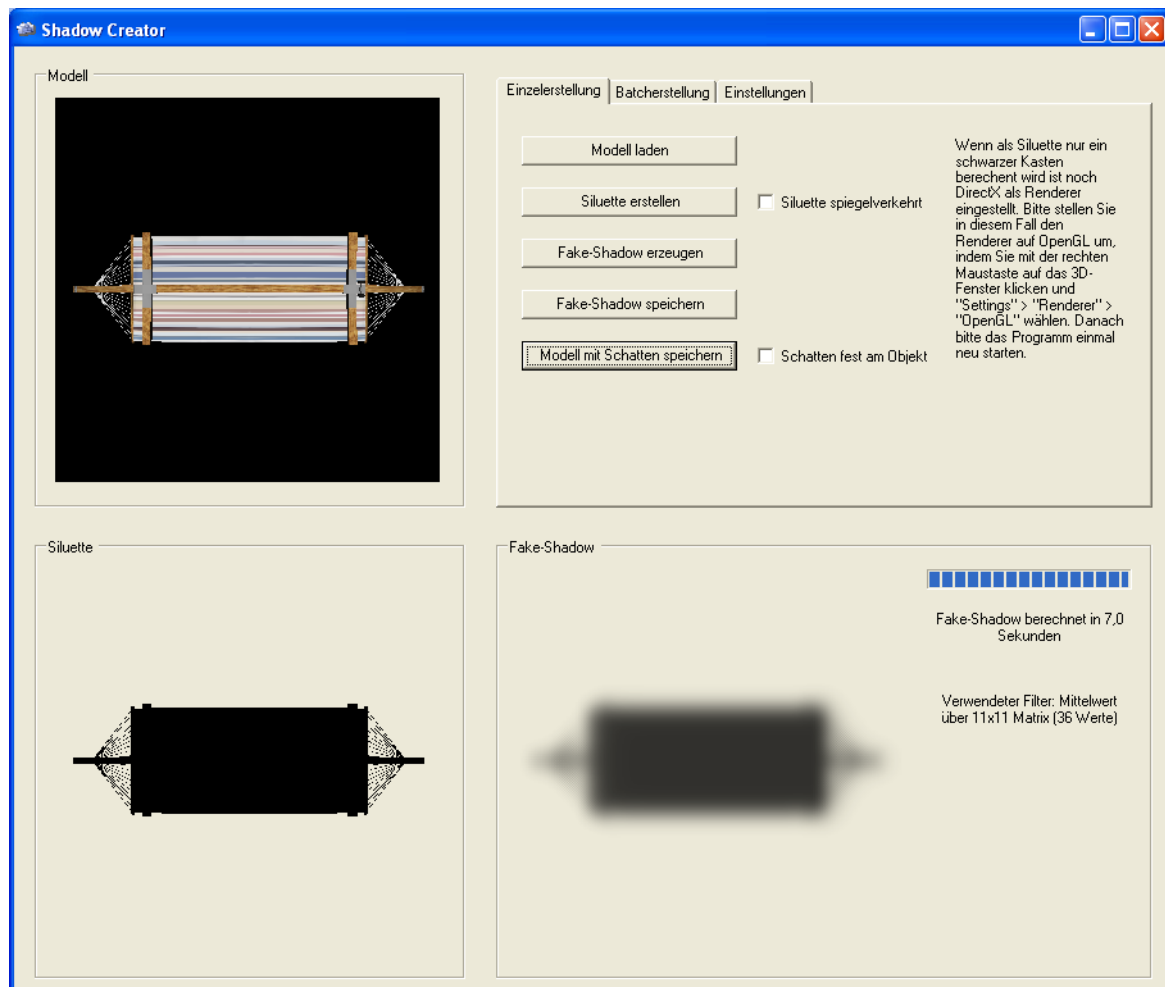


Abbildung 5.2: Benutzeroberfläche des Shadow Creators

resultierende Halbtransparenz des Kernschattens kann später in Click & design eingestellt werden.

5.2.2 Erweiterung von Click & design

Im zweiten Schritt muss Click & design an das neue Format der Objekte angepasst werden. Es ist zwar bereits möglich, ein Objekt mit Schatten zu laden, allerdings ist der Kernschatten schwarz und der Schatten ist stets fest mit dem Objekt verbunden, weil er im Szenengraph ein Teil des Objektes darstellt. Im Rahmen der Studienarbeit wurde Click & design um einen Slider erweitert, mit dem für das aktuell ausgewählte Objekt die Schattenintensität eingestellt werden kann.

Im Shadow Creator kann für jedes erstellte Objekt festgelegt werden, ob der Schatten fest am Objekt oder auf dem Boden bleiben soll. Diese Information wird mit in der VRML-Datei

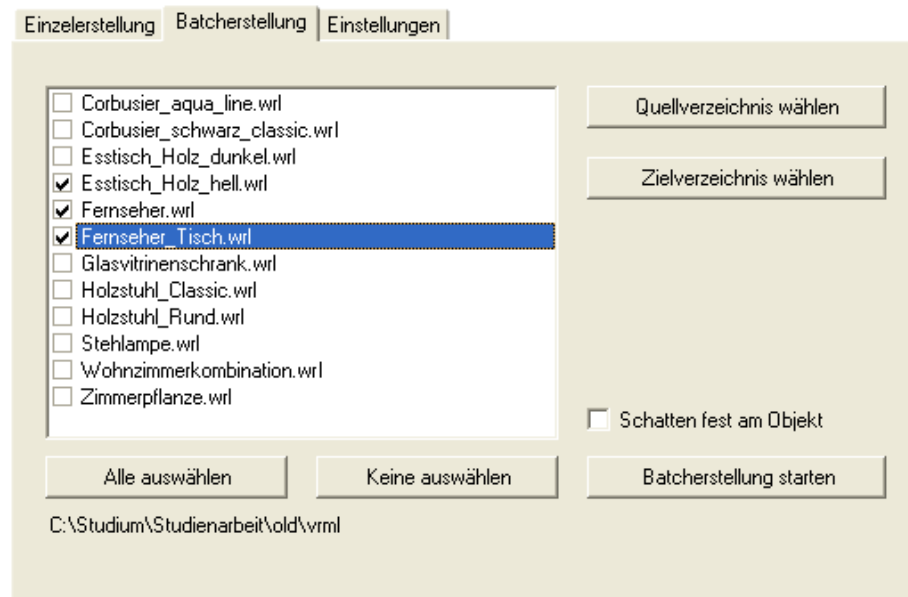


Abbildung 5.3: Batch-Erstellung von Objekten mit Schatten

gespeichert, indem die Schatten-Textur anders benannt wird. Für kleinere Objekte, wie zum Beispiel Vasen ist es sinnvoll, den Schatten fest am Objekt zu positionieren, weil diese in der Regel auf einem anderen Objekt stehen und sich der Schatten so direkt unter der Vase befindet und nicht unter dem Tisch. Wenn der Schatten auf dem Boden bleiben soll, wird bei jeder Änderung der Höhe des Objektes eine Gegentranslation auf die Schattenebene angewendet, um den Schatten auf Bodenhöhe zu belassen.

5.3 Online-Version

In der Konzeptionsphase wurde festgelegt, dass zur Realisierung des Prototyps einer Online-Version Macromedia Shockwave zum Einsatz kommt. Dynamische Elemente lassen sich hier mit so genannten Aktionen gestalten, die in der Skriptsprache Lingo implementiert werden (Abbildung 5.5). Auch auf Serverseite werden dynamische Komponenten benötigt. Diese werden in der verbreiteten Skriptsprache PHP und dem Datenbanksystem MySQL implementiert. Das Ziel soll sein, später auch in der Online-Version die komplette Funktionsvielfalt der normalen Version von Click & design abdecken zu können (Abbildung 5.4). Hier kommt jedoch erschwerend hinzu, dass die Daten (Objekte und Bilder) nicht lokal auf dem Client gespeichert sind, sondern vom Server zur Verfügung gestellt werden, weil Shockwave-Programme immer erst komplett zum Client übertragen werden müssen, bevor sie ausgeführt werden können. Im Übrigen kann durch eine Trennung von Applikation und Daten eine höhere Flexibilität erreicht werden und so später eine Oberfläche zur Administration zur Verfügung gestellt werden.

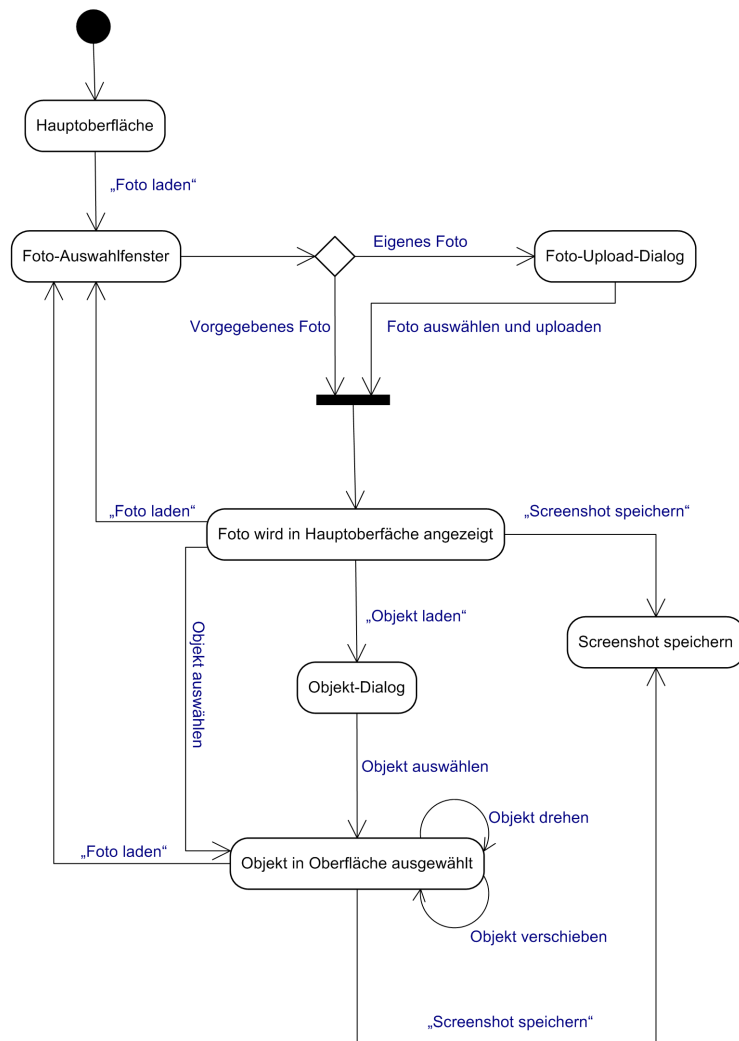


Abbildung 5.4: Zustandsdiagramm der Hauptfunktionen von Click & design

```

stringToFind = ";"
i = 0

--find all thumbnail urls, separated by ";"
repeat while input contains stringToFind
    i = i + 1

    currOffset = offset(stringToFind, input)

    --find thumbnail url
    thumbUrl = input.char [1..currOffset]
    delete the last char of thumbUrl
    delete input.char [1.. (currOffset)]

    --loading the image...
end repeat
  
```

Abbildung 5.5: Parsen einer Serverrückgabe (Lingo)

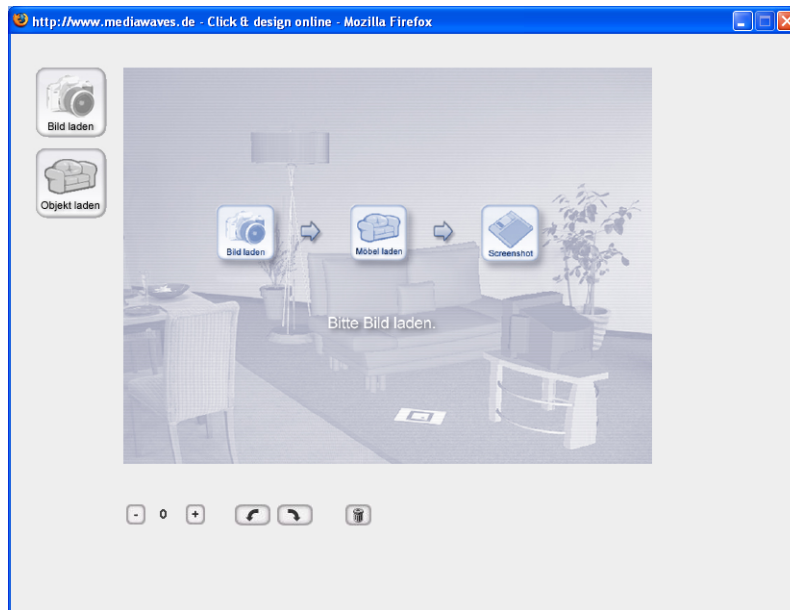


Abbildung 5.6: Oberfläche des Prototyps von Click & design online

Aus diesen Gründen muss es eine Kommunikationsschnittstelle zwischen Shockwave-Client und Server geben, über die vorhandene Daten als Listen vom Server geladen sowie benötigte Bilder und Objekte übertragen werden können. Hierzu wurden folgende Kommunikationsfälle implementiert:

- Abrufen eines Start-Skriptes, das dem Client eine eindeutige Session-ID zuweist, damit er wiedererkannt werden kann
- Abrufen der Foto-Liste mit Metainformationen pro Foto (Position und Orientierung der Kamera)
- Abrufen der Objekt-Liste
- Laden von Foto-Thumbnails
- Laden von Objekt-Thumbnails
- Laden von Fotos in Bildschirmauflösung
- Laden von Objekten als W3D-Datei
- Upload eigener Fotos

Die Gestaltung der Oberfläche fand in Anlehnung an die normale Version statt und es wurden weitgehend die vorhandenen Interaktionselemente und Grafiken verwendet (Abbildung 5.6).

In der Prototyp-Version werden die vorhandenen Beispielbilder vorab mit einem eigens entwickelten Testprogramm, das auch auf Unifeye SDK basiert, getrackt und Markerposition/-orientierung, sowie Öffnungswinkel der Kamera zu jedem Bild in einer MySQL-Datenbank gespeichert.

Das Unifeye SDK liefert die Position und Orientierung des Markers im Kamerakoordinatensystem. In der Shockwave-Applikation werden die Kameradaten für die Darstellung der Szene jedoch im Weltkoordinatensystem benötigt. Dieses soll sich mit dem Markerkoordinatensystem decken, damit sich Objekte, die im Nullpunkt des Weltkoordinatensystems geladen werden auf dem Marker befinden. Im Übrigen hat dieses Vorgehen den Vorteil, dass eine einfache Verschiebung auf der Nullebene des Markers (also dem Boden des Raumes) möglich ist.

Die Koordinatensystemtransformation vom Kamerakoordinatensystem ins Markerkoordinatensystem wurde in PHP realisiert, weil die Transformation später direkt vom Server durchgeführt werden soll, wenn der Benutzer eigene Fotos hochlädt (Abbildung 5.7). Dazu mussten in PHP Methoden zur Matrixmultiplikation implementiert werden.

Weitere Aufgaben der Serverkomponenten sind die Verwaltung von Session-IDs und das Auslesen und Übertragen von Serververzeichnis, in denen sich Objekte und Fotos befinden.

Um den Upload von eigenen Fotos zu ermöglichen stellt der Server ein Upload-Skript zur Verfügung, mit dem Fotos in ein Session-abhängiges Verzeichnis hochgeladen werden. So werden hochgeladene Fotos nur dem Uploader selbst und nicht anderen Benutzern des Programmes angezeigt. Ein hochgeladenes Foto wird vom Server automatisch in die beiden notwendigen Größen für die Darstellung in der Online-Version (Thumbnail und Bildschirmgröße) heruntergerechnet. Die Applikation ruft nach dem Klick auf „Eigenes Foto verwenden“ so lange regelmäßig die Fotoliste vom Server ab, bis ein neues Bild in der Liste erscheint. So ist sichergestellt, dass eigene hochgeladene Bilder nach dem Upload in der Fotoliste erscheinen.

```

/*
    $cameraData includes translation and rotation of the marker
    in camera coordinate system. The data has been calculated
    by the unifeye tracking programm.
*/

//translation of marker in camera coordinate system
$tVec = array($cameraData[0], $cameraData[1], $cameraData[2]);

//rotation of marker in camera coordinate system
//convert from degrees to radiant
$rx = deg2rad($cameraData[3]);
$ry = deg2rad($cameraData[4]);
$rz = deg2rad($cameraData[5]);

//construct axis-rotation matrixes

$mat_rx = array(
    1, 0, 0,
    0, cos($rx), -sin($rx),
    0, sin($rx), cos($rx)
);

$mat_ry = array(
    cos($ry), 0, sin($ry),
    0, 1, 0,
    -sin($ry), 0, cos($ry)
);

$mat_rz = array(
    cos($rz), -sin($rz), 0,
    sin($rz), cos($rz), 0,
    0, 0, 1
);

//construct the complete rotation matrix
$mat_rot = mult3x3Matrix($mat_rz, mult3x3Matrix($mat_ry, $mat_rx));

//construct the basis vectors needed for coordinate system transformation

$basis_x = array(1, 0, 0);
$basis_x = multVectorMatrix($basis_x, $mat_rot);

$basis_y = array(0, 1, 0);
$basis_y = multVectorMatrix($basis_y, $mat_rot);

$basis_z = array(0, 0, 1);
$basis_z = multVectorMatrix($basis_z, $mat_rot);

//construct the coordinate system rotation as 4x4 matrix (basis vectors as row)
$mat_rot4x4 = array(
    $basis_x[0], $basis_x[1], $basis_x[2], 0,
    $basis_y[0], $basis_y[1], $basis_y[2], 0,
    $basis_z[0], $basis_z[1], $basis_z[2], 0,
    0, 0, 0, 1
);

//construct the coordinate system translation as 4x4 matrix
$mat_trans4x4 = array(
    1, 0, 0, -$tx,
    0, 1, 0, -$ty,
    0, 0, 1, -$tz,
    0, 0, 0, 1
);

//calculate the complete system transformation
$mat_absolute = mult4x4Matrix($mat_rot4x4, $mat_trans4x4);

//calculate the new translation vector (camera in marker coordinate system)
$tVec = multVectorMatrix4($tVec, $mat_absolute);

//calculate the new rotation vector (camera rotation)
$rVec = array(-$cameraData[3], -$cameraData[4], -$cameraData[5]);

```

Abbildung 5.7: Transformation vom Kamera- ins Markerkoordinatensystem (PHP)

6 Ergebnisse

Aufgrund der guten Zusammenarbeit mit metaio und SHD war es möglich, die Ergebnisse der Studienarbeit in das Produkt einfließen zu lassen.

6.1 Anti-Aliasing

Die Lösung für Anti-Aliasing wurde als Button mit der Aufschrift „Vorschau“ in die Applikation integriert. Damit ist es dem Benutzer jetzt möglich, nach dem virtuellen Einrichten die Szene in einer optimierten Darstellung zu betrachten. Dieses Vorgehen ist sinnvoll, da für den Benutzer das Ergebnis im Vordergrund steht. Die Objekte gliedern sich durch die Optimierung besser in das Foto ein und wirken realistischer.

Bei der Erstellung eines Screenshots wird die Szene in einem Vorschaufenster angezeigt. Hier wurde die Darstellungsqualität auch noch einmal unter der Verwendung von besseren Resampling-Funktionen optimiert. Zur Integration der zusätzlichen Funktionen in Click & design wurden



Abbildung 6.1: Augmentierte Szene vorher (ohne Anti-Aliasing) und nach Betätigen des Vorschau-Buttons (mit Anti-Aliasing)



Abbildung 6.2: Augmentierte Szene ohne und mit Fake-Shadows

von metaio die Quelltexte der Applikation zur Verfügung gestellt. Die Änderungen wurden nach der Entwicklung übernommen, indem von den modifizierten Dateien ein Diff erstellt wurde. So war eine komfortable Integration in den in der Zwischenzeit auch bei metaio weiterentwickelten Code möglich.

6.2 Schatten

Mit dem Shadow Creator können die Möbel, die als VRML-Objekte vorliegen, mit Schatten versehen werden. Dieses Programm wird bei SHD aktiv eingesetzt. Auch die Verarbeitung des Schattens in Click & design wurde von metaio aus der Studienarbeit übernommen.

6.3 Online-Version

Es wurde ein Prototyp entwickelt, der zeigt, dass eine Entwicklung der gesamten Funktionalität von Click & design als Online-Version grundsätzlich möglich ist.

metaio möchte das Programm gerne für ein AR-Portal einsetzen, um dem Kunden das Thema Augmented Reality näher zu bringen und besser verständlich zu machen.

Eigene Fotos können hochgeladen werden und erscheinen danach in der Liste. Allerdings gibt es noch keine Anbindung an ein Online-Trackingsystem, so dass eigene Fotos noch nicht zur virtuellen Einrichtung verwendet werden können, wenn sie nicht vorher mit dem Tracking-Testprogramm verarbeitet und die Kameradaten in der MySQL-Datenbank abgelegt wurden. In der Szenenansicht können Objekte angehoben, gedreht und verschoben werden. Die Verschiebung funktioniert momentan so, dass die X-/Y-Positionen der Maus die Translation in X-/Z-Richtung des Markerkoordinatensystems beschreiben. Es findet also eine Verschiebung auf

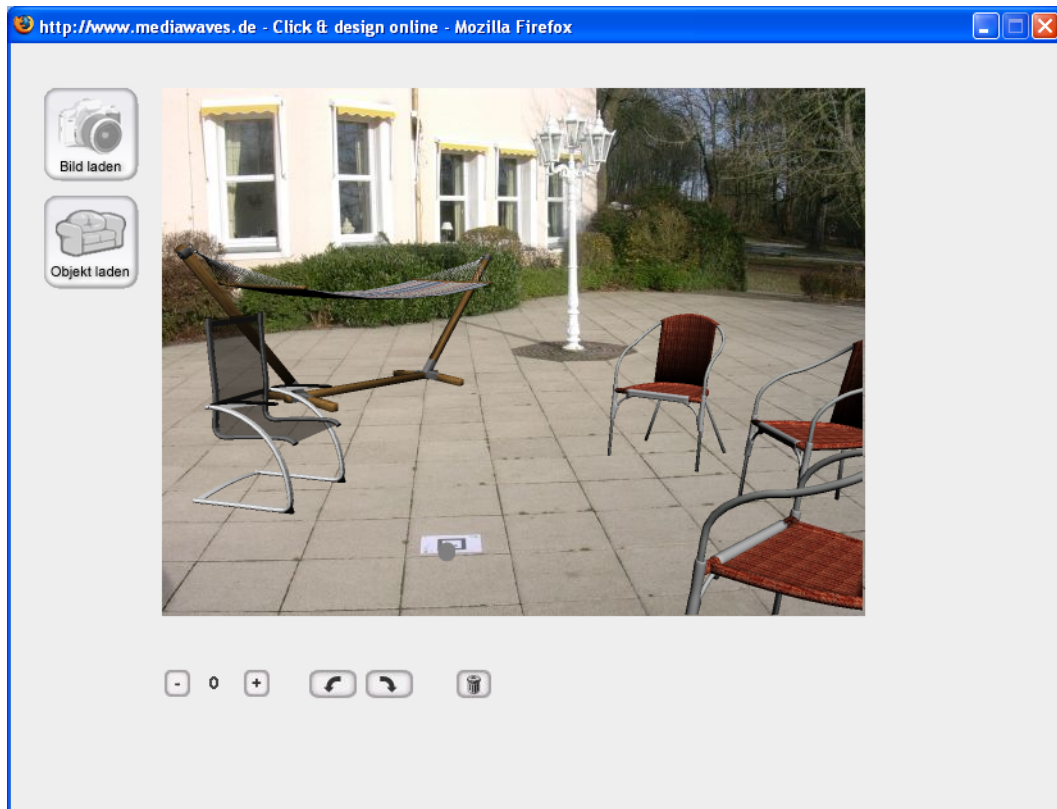


Abbildung 6.3: Virtuelle Stühle und Hängematte auf einer Terrasse (Online-Version)

der Grundebene statt. Auch das Löschen von einzelnen Objekten ist nach der Auswahl durch Anklicken eines Objektes möglich. Zum Auswählen und Verschieben von Objekten musste ein Picking-Algorithmus in Lingo entwickelt werden. Shockwave stellt bereits softwarebasiertes Anti-Aliasing zur Verfügung, so dass auf allen Systemen in Echtzeit eine gute Darstellungsqualität erreicht werden kann.

7 Zusammenfassung und Ausblick

Alle gesetzten Ziele wurden erreicht, so dass die Aufgabenstellung erfolgreich abgeschlossen wurde. In der aktuellen Version von Click & design sind Anti-Aliasing und die Schattenfunktionalität, die im Rahmen dieser Studienarbeit entwickelt wurden, integriert. Zum Erweitern der Objekte um Schatten kommt bei SHD der Shadow Creator aus dieser Arbeit zum Einsatz. Dieser könnte noch optimiert werden, indem die Filteralgorithmen hinsichtlich Effizienz verbessert werden und der Renderer auf orthografische Projektion umgestellt wird, damit die Silhouette wirklich eine senkrechte Projektion des Modells auf die Grundebene ist. So würden auch Verzerrungen bei zu großen Objekten vermieden.

Die Online-Version wurde als Prototyp umgesetzt, eine Demoversion findet sich unter http://www.mediawaves.de/cd_demo. Hier soll nach dem Prototypisierungsmodell eine Neuentwicklung stattfinden, die auf den erbrachten Erkenntnissen und Techniken des Prototyps basiert und um zusätzliche Funktionen erweitert wird:

- **Intelligentere Verschiebung der Objekte:** Eine korrekte Verschiebung der Objekte ist zurzeit nur möglich, wenn der Marker wie vorgesehen positioniert wird. Diese Ausrichtung kann nicht immer sichergestellt werden, zum Beispiel wenn der Marker aus verschiedenen Perspektiven fotografiert wird, um einen Rundumblick im Raum zu erzielen. Damit ein intuitives Verschieben der Objekte mit der Maus aus jeder Blickrichtung auf den Marker möglich ist, muss die Ausrichtung des Markers zur Kamera in die Berechnung der Verschiebungen mit einbezogen werden.
- **Anbindung eines serverbasierten Trackingprogrammes:** Im Prototyp geschieht das Berechnen von Position und Orientierung des Markers noch mithilfe eines auf dem Unifeye SDK basierenden Testprogramms, mit dem vorab für die Beispielfotos die Kameradaten berechnet werden. Dies könnte „on-the-fly“ geschehen, so dass Benutzer die Möglichkeit haben, eigene Marker-Fotos auf den Server zu laden, die dann von einem serverbasierten Programm getrackt werden. Dieses Serverprogramm sollte kommandozeilenorientiert entwickelt werden, damit eine einfache Anbindung an die PHP-Komponenten möglich ist.
- **Laden und Anpassung von Fake-Shadows:** Damit Schatten auch in der Online-Version geladen werden können, müsste es eine Funktion geben, die die PNG-Dateien

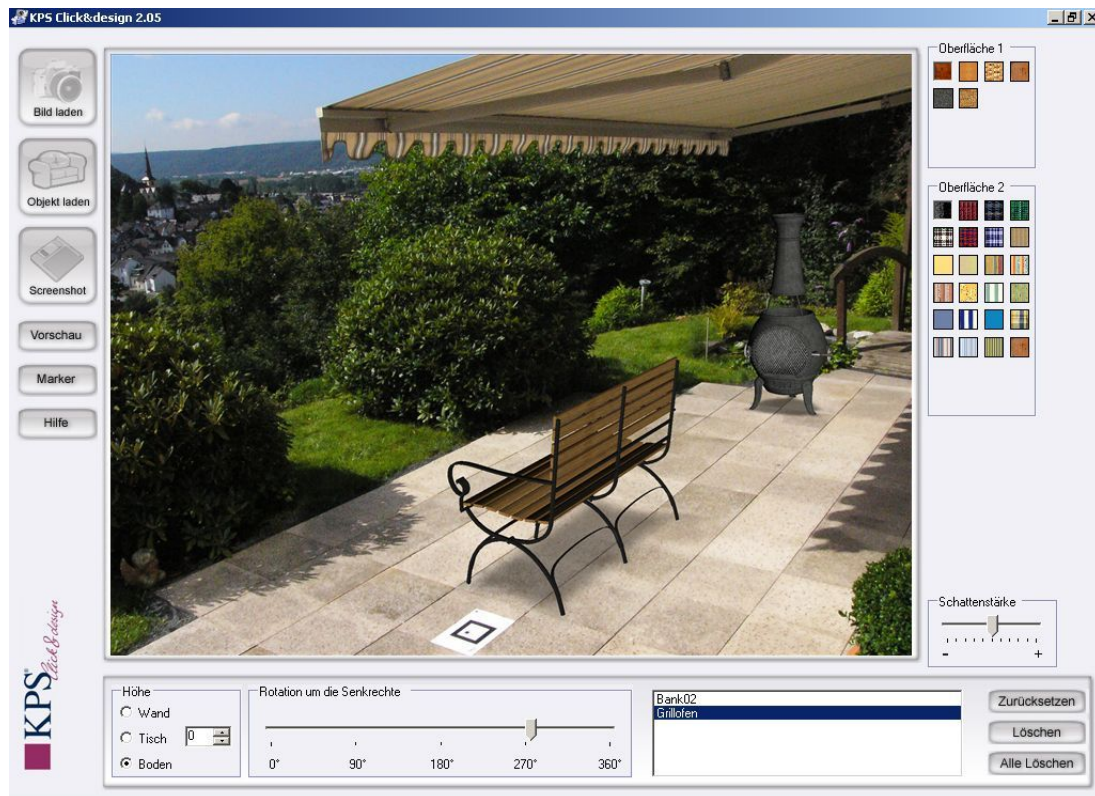


Abbildung 7.1: Neue Oberfläche von Click & design nach Einbindung von Anti-Aliasing und Schattenfunktionalität

als Textur unter die W3D-Objekte läd, da Alpha-Texturen beim W3D-Export¹ verloren gehen. Die Schatten sollten in der Applikation dynamisch angepasst werden, das heißt sie sollten am Boden bleiben, und sich mitbewegen, wenn das Objekt angehoben, gedreht oder verschoben wird.

- **Texturierung der Objekte ermöglichen:** Die Standardtexturierung der Objekte ist direkt in der W3D-Datei festgelegt und die Texturen werden bereits im Prototyp korrekt angezeigt. Dem Benutzer soll zusätzlich die Möglichkeit gegeben werden, einzelnen Objektteilen andere Texturen zuzuweisen. Dies funktioniert mit einem Texturwähler, wie er bereits in der normalen Version von Click & design zum Einsatz kommt. Dazu müssen die verfügbaren Texturen für ein bestimmtes Objekt vom Server übertragen werden und dem Benutzer als Interaktionselement angeboten werden. Nach der Auswahl einer Textur muss diese in voller Auflösung vom Server geladen und dem entsprechenden Teil des Objektes zugewiesen werden.
- **Screenshot-Funktion:** In der Online-Version existiert noch keine Screenshot-Funktion,

¹SHD verwendet Cinema4D

mit der die Szene vom Benutzer als Bild abgespeichert werden kann. Hier müsste zur Implementierung in der Online-Version noch die Machbarkeit überprüft werden, da Shockwave diese Funktion nicht standardmäßig zur Verfügung stellt.

- **Online-Administration:** Damit eine einfache Verwaltung der verfügbaren Objekte und Fotos in der Online-Version möglich ist, sollte diese über eine Online-Oberfläche für SHD-Mitarbeiter zugänglich gemacht werden. Hier wäre auch sinnvoll, eine Verwaltung verschiedener Pakete anzubieten, damit verschiedene Konfigurationen der Online-Version (zum Beispiel für verschiedene Möbelhäuser) realisierbar sind.

Allgemein gesehen sind noch weitere, komplexere Erweiterungen möglich. So könnte die reale Umgebung mithilfe von Fotos aus leicht verschobenen Blickwinkeln und Epipolargeometrie rekonstruiert werden, so dass Verdeckung von virtuellen Objekten durch reale Objekte möglich wäre. Hier würde auch die Verwendung eines Markers überflüssig werden. Für den professionellen Einsatz (zum Beispiel für Innenarchitekten) würde sich der Aufwand der Verwendung komplexer Geräte (HDR-Kamera, Lightprobe) eventuell lohnen, da mit dieser Ausstattung eine vollkommen korrekte Beleuchtung und optimale Integration der virtuellen Objekte möglich ist.

A Literaturverzeichnis

- [balzert, 2001] Helmut Balzert: *Lehrbuch der Software-Technik*. Spektrum, (2001).
- [blachnitzky, 2004] Benjamin Blachnitzky: *Usergerechte Licht- und Schattensimulation in einer Augmented Reality Anwendung*. Fachhochschule Furtwangen, (2004).
- [igd, 2003] Bernad Lukacin: *Virtuell Möbel rücken*. Fraunhofer-Institut für Graphische Datenverarbeitung IGD, (2003).
- [kersten, 1997] Daniel Kersten, Pascal Mamassian, David C. Knill: *Moving Cast Shadows Induce Apparent Motion in Depth*. University of Minesota, (1997).
- [huebner, 2004] Klemens Hübner, Hans Seifert: *Macromedia Director - Multimediaprogrammierung mit Lingo*. Springer Verlag, (2004).
- [opengl, 2004] Dave Shreiner: *OpenGL Reference Manual*. Addison Wesley, (2004).

B Abbildungsverzeichnis

2.1	Interne Struktur von Unifeye SDK	9
3.1	Aliasing in Click & design	11
3.2	„ball-in-a-box“-Szenario von Kersten, Quelle: [kersten, 1997]	12
4.1	Planare Schatten von Benjamin Blachnitzky, Quelle: [blachnitzky, 2004]	16
5.1	Rendering in Datei und resample-Darstellung (C#)	20
5.2	Benutzeroberfläche des Shadow Creators	22
5.3	Batch-Erstellung von Objekten mit Schatten	23
5.4	Zustandsdiagramm der Hauptfunktionen von Click & design	24
5.5	Parsen einer Serverrückgabe (Lingo)	24
5.6	Oberfläche des Prototyps von Click & design online	25
5.7	Transformation vom Kamera- ins Markerkoordinatensystem (PHP)	27
6.1	Augmentierte Szene vorher (ohne Anti-Aliasing) und nach Betätigen des Vorschau-Buttons (mit Anti-Aliasing)	28
6.2	Augmentierte Szene ohne und mit Fake-Shadows	29
6.3	Virtuelle Stühle und Hängematte auf einer Terrasse (Online-Version)	30
7.1	Neue Oberfläche von Click & design nach Einbindung von Anti-Aliasing und Schattenfunktionalität	32

Erklärung

Ja Nein

Mit der Einstellung der Arbeit in die Bibliothek bin ich einverstanden. ☐ ☐

Der Veröffentlichung dieser Arbeit im Internet stimme ich zu. ☐ ☐

.....
(Ort, Datum) (Unterschrift)