

Karl - Steinbuch - Stipendium

Abschlussbericht

"Konzeption und Entwicklung eines speziellen, mobilen Augmented Reality Systems für die VFX Filmproduktion"

vorgelegt von:

Jörg Unterberg

Entstanden in Zusammenarbeit mit

Andreas Dahn

Filmakademie Baden-Württemberg

**Martina Brümmer, Claus Weymann
und Andreas Gutsche**

Fachbereich 4: Informatik, Universität Koblenz-Landau

Abgegeben am:

13. Juli 2007

Inhaltverzeichnis

1 Einführung	2
1.1 Zielsetzung	2
1.2 Aufbau	2
2 Vorbereitung	3
2.1 Konsultation und Kooperation	3
2.2 Bestehende Arbeiten	4
3 Konzeption, Entwurf und Implementierung	6
3.1 Anforderungen	6
3.2 Konzeption	7
3.3 Komponenten	8
3.3.1 AR Toolkit Plus	8
3.3.2 AR Pre Viz Tool (Derivative Touch Designer)	12
3.3.3 AR Post Produktion Interface (Autodesk Maya)	19
4 Ergebnisse und Anwendungsbeispiele	23
5 Fazit und Ausblick	27
Abbildungsverzeichnis	29
Literaturverzeichnis	30

1 Einführung

Beim Film werden bereits in der Vorproduktionsphase 3D-Elemente erstellt, um auf Grundlage des Storyboards eine Vorvisualisierung des Films zu erstellen, die "Prävisualisierung" oder "Animatic" genannt wird. Anschließend werden basierend auf dem "Animatic" die weiteren Film- und Dreharbeiten geplant. Für Filmemacher ohne 3D Erfahrung ist es schwer möglich die Animatics selber zu erstellen.

Das Ziel ist die Entwicklung eines Augmented Reality Prävisualisierung-Systems (im weiteren Verlauf kurz AR PreViz Tool), dass es Beteiligten einer Filmproduktion (Regisseur, Kameraleuten etc.) ermöglicht, die Erstellung von „Animatics“ eigenhändig vorzunehmen und Einstellungen an beliebigen Orten zu planen. Anschließend lassen sich die Animationen aufzeichnen, und können den verschiedenen Beteiligten in geeigneter Form übermittelt werden. Ebenfalls von großem Nutzen wäre eine Integration des AR PreViz Tools in die digitale Produktionspipeline, möglich z.B. als Plug In für eine 3D Animationssoftware.

Das Verfahren bietet eine Menge denkbarer Anwendungsmöglichkeiten, die entwickelt und erprobt werden sollen.

1.2 Aufbau

Im nächsten Kapitel wird die Vorbereitungs- und Planungsphase des Projekts beschrieben und ein kurzer Überblick über die für unsere Arbeit relevanten Entwicklungen im Bereich Augmented Reality gegeben. Im 3. Kapitel werden zunächst die Anforderungen zusammengefasst und das im Rahmen des Karl-Steinbuch-Stipendium realisierbare Augmented Reality Pre Visualisierungssystem konzipiert (kurz AR PreViz Tool). Anschließend wird Entwurf und Implementierung der Komponenten beschrieben. Zuletzt werden die Ergebnisse ausgewertet und ein Ausblick gegeben auf weitere Entwicklungen.

2. Vorbereitung

2.1 Konsultation und Kooperation

Zunächst bestand die Idee, die Augmented Reality Technologie für die Prävisualisierung von Filmszenen einzusetzen, d.h. für die Erstellung von 3D Animationen und Animatics. Andreas Dahn und ich haben uns über die Realisierbarkeit und die Möglichkeiten eines solchen Vorhabens beraten. Andreas Dahn ist Student der Szenischen Regie an der Filmakademie und verfügt über Erfahrungen als VFX-Artist /-Coordinator. Zusammen haben wir die Anforderungen an ein Prävisualisierungs Tool für die Filmproduktion erörtert. Parallel zu dem Prävisualisierungstool sollte eine Schwester-Anwendung entstehen, die VFX-Artists die Möglichkeit bietet, direkt in einer professionellen VFX-Software Parameter mittels Marker-Tracking in Echtzeit zu manipulieren.

Im Rahmen der Bewerbung für das Karl-Steinbuch-Stipendium haben wir Prof. Haegele vom Animationsinstitut der Filmakademie unser Vorhaben vorgestellt und Einsatzmöglichkeiten und den Nutzen für die Filmproduktion diskutiert. Nach Erhalt des Stipendiums folgte eine Konsultation bei Professor Stefan Müller von der Universität Koblenz Landau (Lehrstuhl Computervisualistik), wo in den letzten Jahren eine Vielzahl verschiedener, spezieller Augmented Reality Systeme entwickelt wurden. In Zusammenarbeit mit Studenten der Universität Koblenz wurde ein Projektplan ausgearbeitet, ein solches Projekt umzusetzen.

Bereits bei einem ersten Treffen wurde entschieden, ein markerbasiertes AR Tracking zu benutzen. Die Evaluierung von AR-Systemen fiel den Studenten der Universität Koblenz zu. Ihre Entscheidung fiel auf AR Toolkit Plus, dass von den Studenten für unsere Anwendung angepasst wurde.

Meine Hauptaufgabe bestand in der Entwicklung der Touch PreViz Anwendung und in der Anbindung des Tracking Moduls an die Software Touch Designer, die auf die Echtzeit Verarbeitung von 3D- und Videodaten optimiert ist.

Als Professionelle Anwendung in der VFX-Produktion wurde Autodesk

Maya ausgewählt, für das vor allem die Schwesteranwendung als Plug In entwickelt werden sollte. Das Plug In bietet den Vorteil, das AR Trackingsystem direkt mit der im Film-Bereich etablierten 3D-Animationssoftware zu nutzen. Oft liegen die 3D Szenen mit Charakteren und/oder Hintergründen bereits im Maya Format vor. Die Software Touch dagegen dient der Erstellung interaktiver 2D und 3D Animationen und bietet zudem die Möglichkeit, eine grafische Benutzeroberfläche zu entwickeln, die zur Steuerung einer bestimmten Anwendungen benötigt wird. Die Software ist auf Echtzeitdarstellung optimiert und führt einen wesentlichen Teil der Berechnungen direkt auf der GPU aus, im Gegensatz zu 3D Animationsoftware die zur Berechnung der Bilder einen Softwarerenderer verwenden.

Ich erhielt die Möglichkeit ein Training bei der Firma Derivative in Toronto zu machen, und die neueste Version der Software zu testen. So konnte ich bereits einige wesentliche Prinzipien / Arbeitsabläufe des geplanten AR Prävisualisierungs Tools in Touch umsetzen. Im Anschluß an das Training began die Entwicklung einer direkten Anbindung von Martinas AR Toolkit Plus an die Touch Software, in Rücksprache mit Derivative Inc., um die bestmögliche Anbindung an die Software zu gewährleisten.

2.2 Bestehende Arbeiten

Zur Vorbereitung und Planung des Augmented Reality Prävisualisierungstools wurde auch auf bereits veröffentlichte wissenschaftliche Arbeiten im Bereich Augmented Reality zurückgegriffen. [BR05] liefert eine Einführung die verschiedenen AR Displaysysteme, wobei für das AR Prävisualisierungstool eine Video-Based See-Through Lösung durchaus Sinn macht. [Azu06] gibt einen Überblick über bestehende AR Technologien und gibt dabei unter anderem auf mobile AR Anwendungen ein und gibt eine Einführung in das AR Toolkit. Das AR Toolkit Plus von [SW07] basiert auf dem AR Toolkit, und ist für den Einsatz auf mobilen Endgeräten entwickelt worden. Es bietet einige neue Features, die auch für die AR Prävisualisierung auf einem Laptop von entscheidendem Vorteil sind.

J. Park et. Al. [Par05] haben ein Augmented Reality Interface vorgestellt, mit dem sich aus Einzelbildern der Augmentierten Szene ein 3D Storyboard erstellen lässt. Erstellung und Aufnahme von 3D Animationen und Animatics geht einen Schritt weiter. Im Rahmen des Forschungsprojects MR PreViz von [Ohs01] ,[Par05] wird ein Pre-Production Tool für die Filmproduktion entwickelt, das ebenfalls die Planung der Einstellungen und die Erstellung von Animatics ermöglichen soll. Im Gegensatz dazu ist unser System wesentlich preiswerter und man benötigt mit Hinblick auf den mobilen Einsatz nur einen Laptop, eine Firewire Camera und ein paar spezielle Trackingmarker. Es gibt zudem eine ganze Reihe an wissenschaftlichen Veröffentlichungen, die eine Anbindung von AR Systemen an andere Authoring und Content Creation Tools beschreiben. Bei [DART04] und [Ber04] steht die Einbeziehung von Designern in den Entwicklungsprozess von AR Anwendungen im Vordergrund. In [Ber06] wird eine Touch Anwendung realisiert, bei der man mit Hilfe von markerbasiertem Tracking mit einem Character interagieren kann und z.B. Motion Capture Daten von einem Character filtern kann. In unserem Fall wird neben der AR Tracking Anbindung besonderes Augenmerk auf die Entwicklung einer robusten Anwendung gelegt, um Filmemachern und 3D Animatoren ohne technischen Background die Content Creation mit AR zu vereinfachen.

3 Konzeption, Entwurf und Implementierung

3.1 Anforderungen

Das Augmented Reality Prävisualisierungs-Tool sollte es ermöglichen 3D Szenen samt virtueller Charaktere, Lichter und Kameras auf das Lifebild eine Videokamera zu augmentieren. Dazu sollte ein möglichst robustes und stabiles Tracking vorhanden sein. Die Bedienung der Kamera, die Benutzung des Trackings und der Einsatz der Marker soll intuitiv ohne Fachkenntnisse in 3D-Animation von Filmemachern benutzbar sein. Das Tracking soll sich einfach von der Anwendung aus starten lassen bzw. einfach zusammen mit der Anwendung gestartet werden können. Die Trackingmarker müssen handlich sein und einfach im Raum bewegt werden können. Sie müssen eindeutig zuweisbar sein mittels einer ID oder einem angelernten Muster. Es wird unterschieden zwischen einem Weltmarker und vielen Objektmarkern mit festen IDs. Der Weltmarker markiert den Ursprung der virtuellen Szene und läßt sich in der Trackingsoftware abspeichern, um bei einer Verdeckung Darstellungsfehler zu vermeiden. Die Verwaltung der Objektmarker sollte in Touch bzw. vom Maya Plug In aus geschehen. Die Bewegungsdaten der Objektmarker und der assoziierten 3D Objekte müssen sich als Keyframes oder in anderer Form in der Anwendung speichern und wiedergeben lassen.

Die 3D Szenen und Elemente müssen einfach und im richtigen Maßstab zu erstellen sein bzw. sich importieren und editieren lassen. Die 3D Bewegungsdaten müssen sich auch exportieren lassen und von anderen 3D Programmen aus der VFX Produktion und Setdesign gelesen werden können. Mit Hilfe von Trackingmarkern lassen sich auch die Positionen von realen Objekten, wie z.B. Lichtquellen und Kameras bestimmen. Für das Shading der 3D Objekte sollten verschiedene Shader bereit gestellt werden. Um aus dem VR / AR Szenario ein Animatic zu erstellen, müssen sich die augmentierten Kamerabilder als Movie Clip speichern lassen. Das System soll mit mind. 25Hz auf einem serienmäßigen PC mit leistungsfähiger GPU laufen und mit den Mitteln aus dem Karl-Steinbuch-Stipendium finanziert werden können.

3.2 Konzeption

Das AR PreViz Tool stellt eine Ergänzung zu bestehende Prävisualisierungs-verfahren in der Filmproduktion dar. mit dem sich 3D Animationen und Animatics erstellen lassen, wie sie zur Planung von Dreharbeiten und VFX Produktionen benutzt werden. Das AR PreViz soll auch für On-Set Kamera PreViz und als Postproduktions Interface zum Aufnehmen von Bewegungsdaten eingesetzt werden können. Der modulare Aufbau ermöglicht es, Um das Tool auf unterschiedliche und möglicherweise neue Anwendungsszenarien anzupassen ist es modular aufgebaut. Ein wesentlicher Teil besteht also in der Entwicklung von geeigneten Schnittstellen zwischen den bestehenden Programmen. Das AR PreViz Tool besteht aus mehreren Komponenten:

- Leistungsfähiger Laptop mit GPU
- Firewire Kamera, wobei wir neben der Allied Vision Marlin auch eine Webcam und eine Mini DV Kamera von Panasonic eingesetzt haben.
- Markerbasierten Trackingsystem (AR Toolkit Plus) mit dazugehörigen Markern
- AR Pre Viz Tool basierend auf Touch und die dazugehörige Shared Memory Schnittstelle zur Übertragung der Bild und Trackingdaten
- Maya Plug In, das sowohl die geometrisch korrekte 3D Darstellung sowie die Darstellung des Videobildes im Hintergrund ermöglicht. Es fungiert als Schwesteranwendung und Post Production Interface, mit dem sich auch komplexe Maya Szenen previzen lassen und beliebige Attribute durch Markerbewegung steuern lassen
- Stativ

Um mit den Trackingmarkern arbeiten zu können, wird basierend auf Marker IDs ein Workflow erarbeitet, der es ermöglicht, die Marker und die Objekte zu assoziieren und in der 3D Szene zu verwalten. Zur Organisation der Szene werden für die im Einsatz befindlichen Marker automatisch 3D Knoten erstellt, wenn diese dem System sichtbar gemacht werden, und in einer Liste gespeichert. An die 3D Knoten lassen sich beliebige 3D Objekte binden und wieder lösen. Die Bewegungsdaten

lassen sich Keyframes oder in anderer Form in der Anwendung speichern und wiedergeben. Die Verbindung bleibt bestehen, auch wenn der Marker nicht mehr im Bild zusehen ist. Sollte ein Marker nicht mehr gebraucht werden, kann die Verbindung aufgelöst werden und der erstellte 3D Knoten aus der Liste wieder gelöscht werden.

3.3 Komponenten

3.3.1 AR Toolkit Plus

Tracking

Als markerbasiertes Tracking für das AR PreViz Tool wird die AR Toolkit Plus Implementierung von Martina Brümmer [Bru07] von der Universität Koblenz-Landau benutzt. Das AR Toolkit Plus basiert auf der Funktionsweise des AR Toolkit, wobei aber auch verschiedene Algorithmen verändert, erweitert und ergänzt wurden. Es handelt sich bei dem ARToolkitPlus jedoch nicht um eine Komplettlösung, sondern um eine reine Bibliothek für das Tracking [ART+].

Beim AR Toolkit wird mittels der Algorithmen des ARToolkits [ART] nach schwarzen quadratischen Markern im Videobild gesucht. Wenn ein solcher Rahmen gefunden wurde, wird durch mathematische Berechnungsroutinen die Position des Markers zur Kamera bestimmt [ART]. Im nächsten Schritt wird das Symbol im Inneren des quadratischen Rahmens identifiziert. Dabei wird das Muster (Pattern) mit bereits zuvor in das Programm Geladenen verglichen, und das mit der höchsten Übereinstimmung ausgewählt. Ab diesem Zeitpunkt ist also die Information über ID und Position des Markers bekannt. Beim AR Toolkit Plus verläuft das Erkennen der Marker zusätzlich über codierte ID-Marker, und nicht mehr nur über das Abgleichen von Mustern (ID-Detektionsalgorithmus vom ARTag inspiriert [Mar04]) Details über die Kodierungsart findet sich auf der Webseite des ARToolkitPlus [ART+].

Ausserdem stellt das AR Toolkit Plus einen robusteren Tracking

Algorithmus zur Verfügung als AR Tag und erlaubt es zusätzlich die Beachtung der Helligkeitsbedingungen mit ein zu bringen. Durch diesen Algorithmus ist es möglich, auch bei variierenden Lichtverhältnissen während des laufenden Programms, die Marker zu erkennen[SW07]. Die Berücksichtigung der Eckenverschattung bei Kameras beim AR Toolkit Plus trägt ausserdem zu einer besseren Erkennung der Marker bei. Ein weiterer Vorteil liegt darin, dass das ARToolkitPlus klassenbasiert implementiert wurde, so dass es kein Problem darstellte dieses auch objektorientiert zu erweitern.

Videoanbindung

In der Implementierung von Martina Brümmer wird die Videoanbindung vom AR Toolkit verwendet. Beim AR Toolkit ist die libARvideo Bibliothek für die Videoanbindung zuständig. Sie wiederum benutzt die DsVideolib von [DSVL], die sich um die Kommunikation mit dem Kameratreiber unter dem Betriebssystem Windows kümmert (DirectX 9.0 erforderlich).

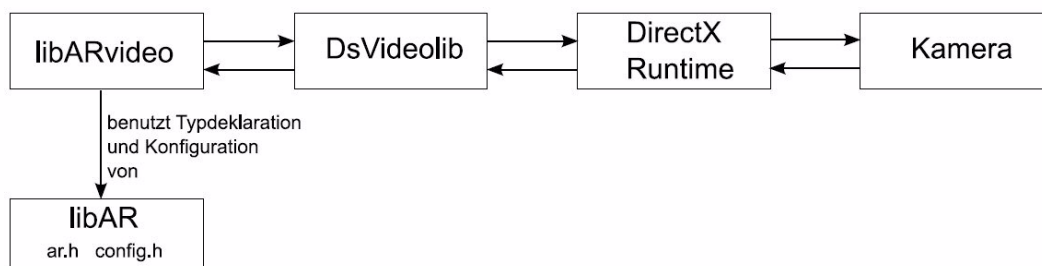


Abb.1: Videoanbindung [Bru07]

Um das Modul der Videoanbindung aus dem Toolkit zu isolieren, wurde als erstes eine neue Projektmappe erstellt, in der sich nur noch die libARvideo befand (Abhängigkeiten von der libAR wurden vorläufig mit in die libARvideo eingebunden). Danach wurde ein neues Projekt erstellt, mit den neuen Klassen "CamVideo" und "Image".

Die Klasse Image ist für die Verwaltung des Videobildes zuständig, das

von der Klasse CamVideo von der Kamera abgefragt wird. Dabei ist sie so konzipiert das Höhe, Breite und Pixelformat des Bildes mit abgespeichert werden, und so alle wichtigen Informationen des Bildes in dem Objekt gespeichert sind. Die Klasse CamVideo stellt Funktionen für das Öffnen und Schließen der Videoquelle zur Verfügung. Ebenfalls erstellt diese Klasse ein Objekt vom Typ Image, in dem das derzeitige Videobild mit Höhe, Breite und Pixelformat abgespeichert wird.

Neue Klassen

Es wurden weitere neue Klassen entwickelt, um Funktionen bereit zu stellen, die für das Previz-Tool benötigt würden, die es aber so in der AR Toolkit Plus Bibliothek nicht gibt. Eine wichtige Anforderung war es, beliebig viele Marker in Abhängigkeit zu einem Weltmarker zu detektieren, und sich ausgeben zu lassen. Hierfür wurde eine neue Klasse erstellt.

Sie enthält den Hauptalgorithmus, der für das Detektieren von mehreren Markern zueinander und zu einem Weltmarker verwendet wird. Hier wird der Marker mit der ID = 0, aus dem Repertoire des ARToolkitPlus von "SimpleId", als Weltmarker festgelegt. Dabei wird die maximale Anzahl an detektierten Markern durchlaufen und nachgesehen, ob sich der Weltmarker unter den gefundenen Markern befindet. Sollte dieser Durchlauf positiv sein, so wird die Position und Orientierung des Markers berechnet und als Matrix in "wTrans" abgespeichert. Die Inverse von "wTrans" wird in "camTrans" gespeichert. Mit der Matrix "camTrans" hat man nun die Position und Orientierung der Kamera im Weltmarkerkoordinatensystem. Sobald ein Weltmarker gefunden wurde, werden die restlichen gefundenen Marker durchlaufen.

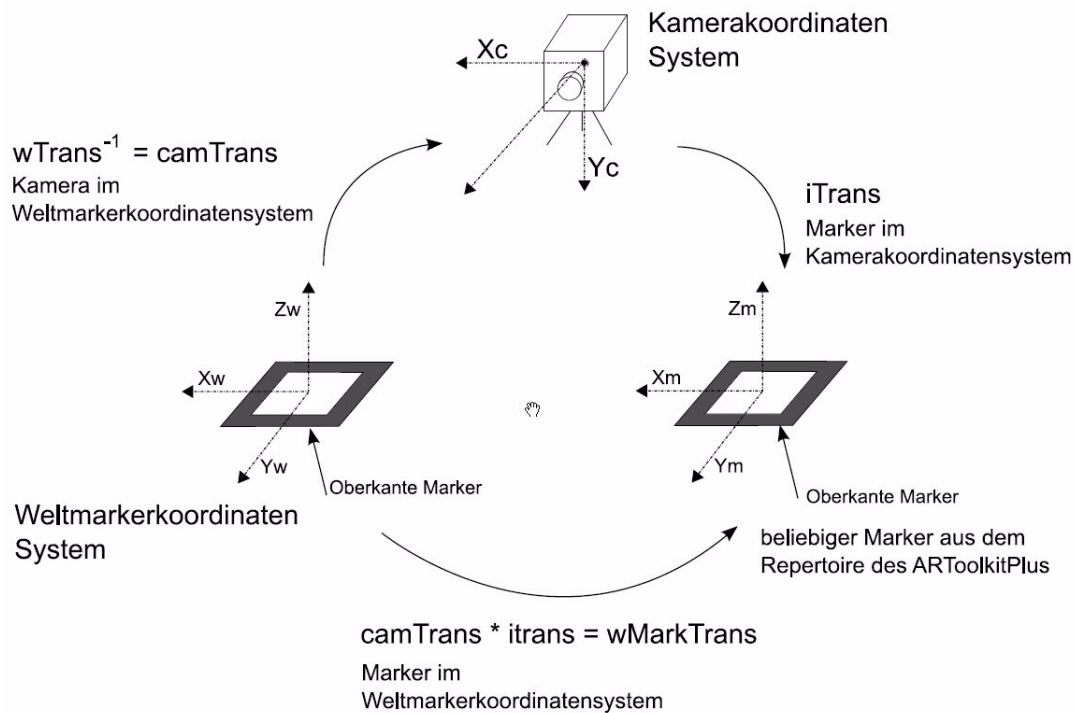


Abb. 2: Beziehung zwischen Markern und Kamera [Bru07]

Eine weitere Anforderung bestand darin, dass, wenn der Weltmarker nicht im Bild zu sehen wäre, die alte Position und Orientierung des Weltmarkers benutzt werden sollte. Zu Anfang wird eine neue Variable "isWorldMarkerCached" im Funktionsrumpf eingeführt, die dem aufrufenden Programm angibt, ob ein Weltmarker erkannt wurde oder nicht. Des Weiteren wird eine neue Variable "mOldWorldMarker" angelegt, welche die Daten des vorherigen Weltmarkers enthält, und dann wieder aufgerufen werden kann, wenn kein Weltmarker gefunden wurde. Danach werden die weiteren erkannten Marker durchlaufen, und in Relation zu dem alten Weltmarkerkoordinatensystem berechnet. Weiterhin wurde eine neue Klasse "TrackCoord" angelegt. Für jeden Marker, der im Kamerabild gefunden wurde, wird ein Objekt vom Typ dieser Klasse erstellt, und so zur Informationsabfrage bereit gestellt. Weitere Details finden sich in der Studienarbeit von Martina Brümmer [Bru07]

3.3.2 AR PreViz Tool

Touch Designer

Touch Designer unterscheidet sich von 3D Animationssoftware in mehreren Punkten. Es ist in erster Linie designed zur Erstellung von interaktiven Echtzeit 3D Anwendungen und verfügt nicht über einen Softwarerenderer wie in der Filmproduktion üblich. Touch Designer ist eine professionelle OpenGL Anwendung, unterstützt OpenGL 2.0 mit Shadinglanguage und führt selber einen Großteil der 3D und 2D Bearbeitung auf der GPU aus. Die Content Erstellung basiert auf einem prozeduralen Workflow, bei der einfache Generatoren durch eine Reihe von Filtern modifiziert werden können. Die einzelnen Komponenten werden über wires, z.Dt. Drähte, miteinander verbunden. Diese prozedurale Kette wird kontinuierlich durchlaufen und Änderungen am Beginn der Kette haben Auswirkungen auf das letztlich sichtbare Objekt. Touch Designer ermöglicht den Austausch von Daten zwischen verschiedenen Komponenten, das heißt Geometrie- und Bilddaten können genauso modifiziert werden wie Channel Daten. Außerdem kann man Skripte zur Steuerung des Programmablaufs schreiben und Expressions zur Berechnung einzelner Parameter benutzen.

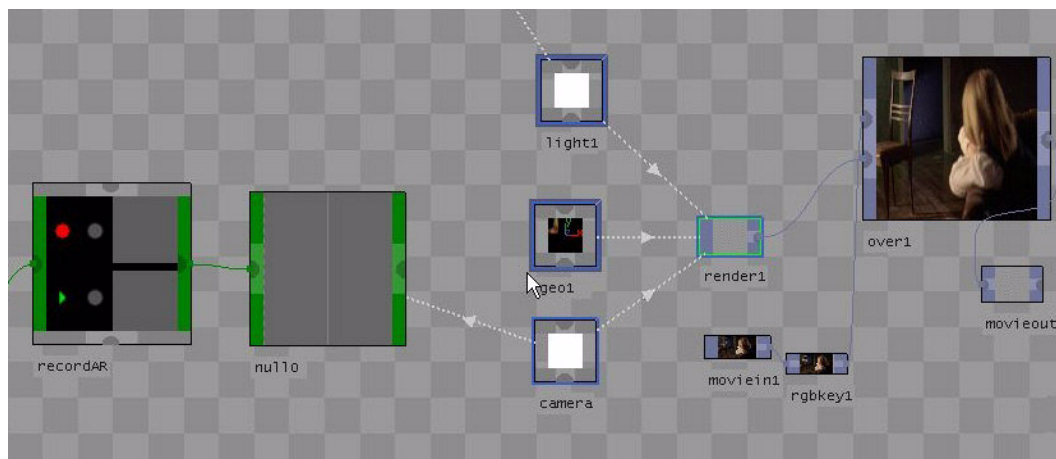


Abb. 3: Die Abbildung zeigt Channel Operatoren, 3D und 2D Compositing

Ich hatte die Gelegenheit, die Software bei Derivative Inc. zu testen und konnte bereits einige wichtige Prinzipien / Arbeitsabläufe des geplanten AR Previz Tools in Touch umsetzen, dazu zählten u.a. die Entwicklung von Interface Komponenten, die Umsetzung eines Workflow zur Erstellung und zur Verwaltung von Marker Nodes mit festen IDs, die Implementierung von Echtzeitshading und die Realisierung einer Datenanbindung zwischen einem C++ Programm und Touch über Netzwerk Schnittstelle.

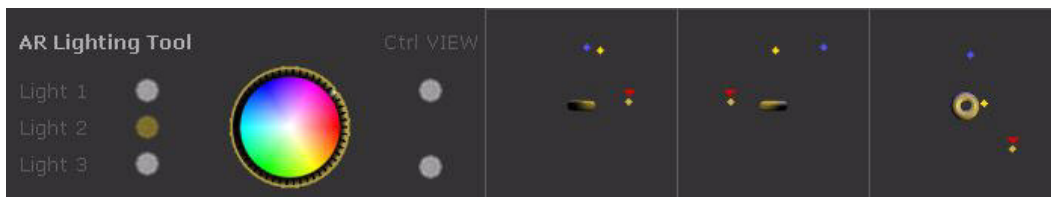


Abb. 4: Beispiel einer Benutzungsoberfläche mit Schaltflächen und Reglern zur Positionsbestimmung von Lichtquellen

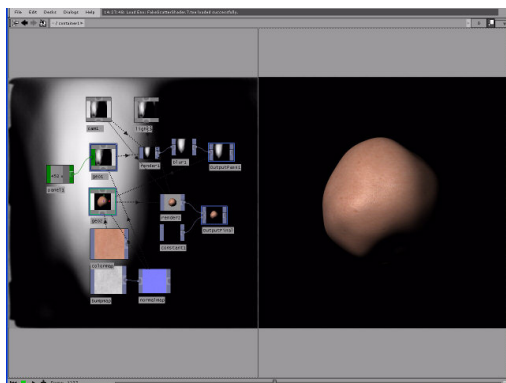


Abb. 5: Echtzeitshading [Joe04]

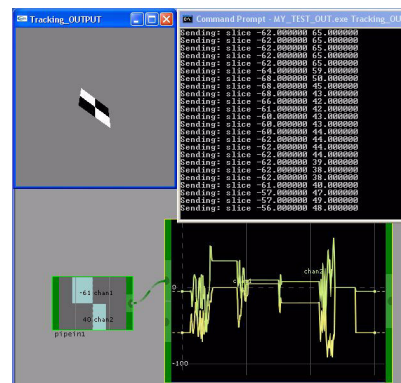


Abb. 6: Network socket Interface

Shared Memory Interface

Die Verbindung über einen Netzwerkport erwies sich als nicht ausreichend für Augmented Reality mit Video Anbindung und so wurde eine neue Video-/ Datenschnittstelle geschaffen. Ich habe eine Klasse implementiert, die die wichtigsten Methoden bereitstellt, um die Videodaten des AR Toolkit Plus (oder einer beliebigen anderen Anwendung) in einen reservierten Bereich im Speicher zu kopieren, so

dass von Touch aus direkt darauf zugegriffen und das Videobild dargestellt und / oder weiterverarbeitet werden kann.

Shared Memory wird benutzt, um von zwei oder mehr Prozessen auf den selben Speicherbereich zuzugreifen. Um sich den Speicher zu teilen, müssen die Prozesse ein File Mapping Objekt verwenden, welches einen eindeutigen Namen und Handle hat. Dann muss darauf geachtet werden, dass der Shared Memory Buffer die korrekte Größe hat. Am Anfang wird ein Header eingefügt, anschließend folgen die eigentlichen Videobild-Daten. Über die Datenstruktur des Shared Memory Headers können die zu übertragenden Daten definiert werden, einige sind selbsterklärend wie z.B. Bildgröße und Aspect Ratio, andere sind OpenGL Konstanten, wie z.B. dataFormat, dataType und pixelFormat [Open GL Spezifikation], weil hier OpenGL für die Videoanbindung / zum Rendering benutzt wird. Die korrekte Größe ist `sizeof(ShareMemoryHeader) + theSizeOfTheImageData`. Der Namen des CreateFileMapping Objekt wird durch Derivative Touch und die Schnittstelle des AR Toolkit Plus festgelegt. Um mögliche Konflikte mit anderen Programmen auf demselben PC zu vermeiden, die ebenfalls File Mapping verwenden wurde der Name intern mit „TouchSHM“ geprefixed. Um zu vermeiden das Touch und die Schnittstelle oder andere Programme gleichzeitig auf den Shared Memory Speicherbereich zugreifen, wird ein Mutex Objekt verwendet, mit dem sich Zugriffe verwalten lassen. Nach Beendigung der Programme wird der Speicherbereich wieder freigegeben.

Die Shared Memory Schnittstelle des AR Toolkit Plus (kann auch für andere Programme verwendet werden), stellt die nötigen Methoden zu Verfügung, um sowohl die Videodaten als auch die Trackingdaten zu übertragen. Dies geschieht in der Display Funktion des AR Toolkit Plus nachdem das Videobild auf Trackingmarker hin durchsucht wurde und bevor es im OpenGL Testfenster angezeichnet wird.

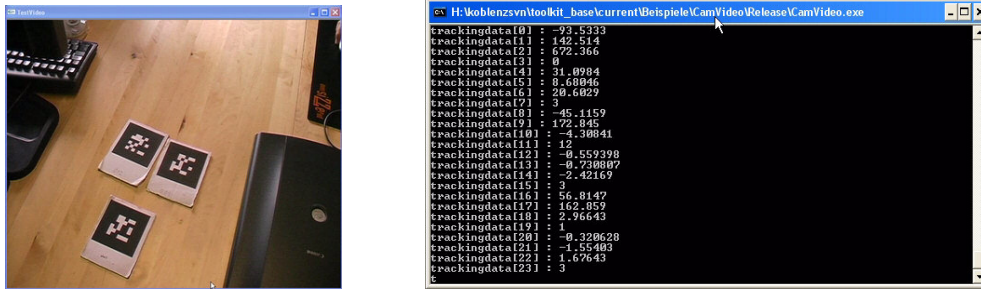


Abb. 7: OpenGL Videodarstellung und Ausgabe der Trackingparameter

Mit Hilfe einer speziellen Shared Memory In Node können die Video- und Trackingdaten von AR Toolkit eingelesen werden.

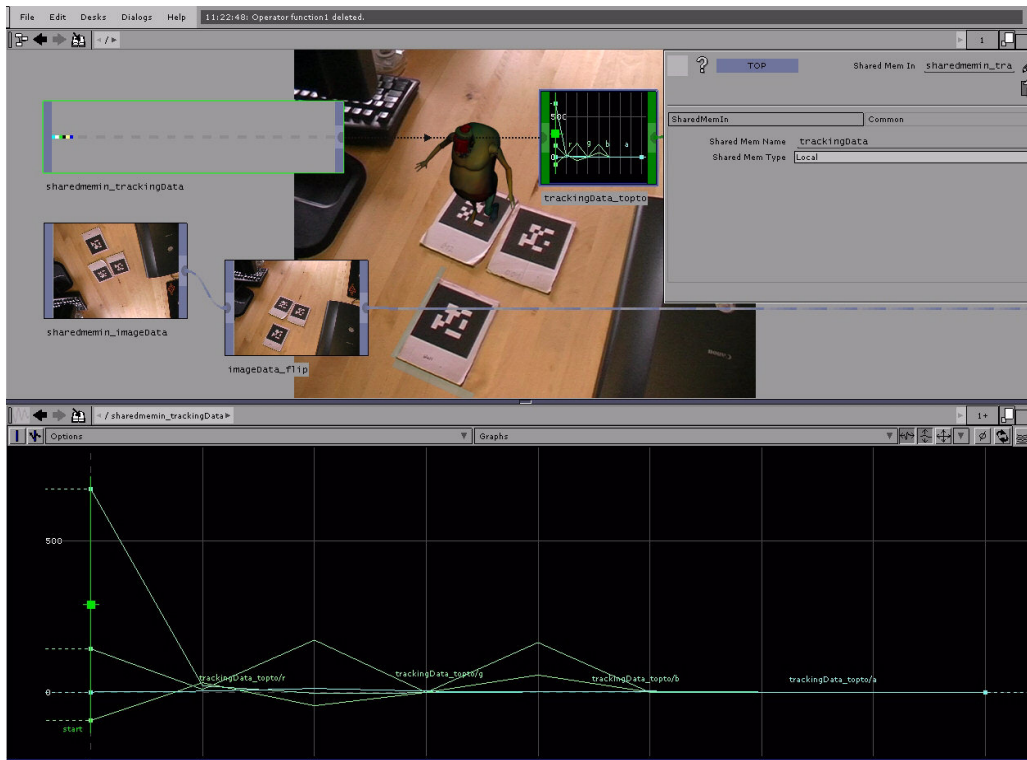


Abb. 8: Shared Memory In Top, Tracking Daten und augmented Videobild

Die Transformationsmatrix wird letztlich benutzt, um 3D Elemente perspektivisch korrekt zu rendern und dem Videobild zu überlagern.

Die Shared Memory Schnittstellenklasse, die für den Datenaustausch zwischen Touch und anderen Programmen benutzt werden kann, befindet sich im Anhang.

Die Performance wurde anhand von zwei unterschiedlichen Kameras mit unterschiedlicher Bildgröße getestet. Bei einer Bildgröße von 640*480 läuft AR Toolkit Plus (CPU) mit 37 Hz (Bildern pro Sekunde) und Touch (GPU und CPU) läuft mit 31 Hz auf einer Dell Precision 690 Workstation mit Intel Dual Xeon 5160 Prozessoren und Quadro FX 4500 Grafikkarte. Die Performance mit DV Video Camera, Vollbildmodus 25 Hz, Bildgröße 720*576 liegt etwas darunter.

Programmablauf

Um einen einfachen und intuitiven Arbeitsablauf mit dem AR Prävisualisierungs Tool zu ermöglichen, wurden bereits einige Verbesserungen am AR Toolkit Plus vorgenommen, dazu zählt die Speicherung des Weltmarkers bei Verdeckung oder entfernen des Weltmarkers aus dem Sichtfeld der Kamera [Bru07].

Der Touch Anwendung fällt die Verwaltung der übrigen Marker und der Aufnahme und Wiedergabe der Bewegungen zu. Alle Marker werden automatisch in eine Liste eingetragen und verwaltet. Sobald ein Marker im Bild erkannt wird, wird mit Hilfe eines Touch Skripts überprüft, ob der neu erkannte Marker bereits in Verwendung ist. Wenn nicht, wird der Marker der Liste hinzugefügt. Die Liste mit den Markern und IDs kann man sich anzeigen lassen, kann aber auch versteckt bleiben. Für die Marker in der Liste werden 3D Transformationsknoten, Marker Nodes genannt, erstellt. Sie enthalten die Marker ID und die Trackingdaten des entsprechenden im Einsatz befindlichen Markers, und haben eine kleine grafische Repräsentation in der virtuellen Szene.

Die 3D Camera wird automatisch mit dem Weltmarker (ID=0) verbunden. Parallel dazu können 3D Elemente, sofern sie bereits als Touch Komponente (.tox) oder im .fbx Format von Autodesk vorliegen, direkt in ein Fenster der Touch Anwendung gezogen werden, wodurch wiederum ein spezielles „drag and drop“ Skript ausgeführt werden soll, mit dem die 3D Elemente der Bibliothek hinzugefügt werden.

Wenn ein 3D Objekt aus der Library mit einem Marker Node verbunden wird, wird es direkt richtig im 3D Raum transformiert und im Viewport der Camera angezeigt. Die Verbindung Marker Node – 3D Objekt wird durch ein spezielles „drag and drop“ Skript erzeugt, das ausgeführt wird, wenn das 3D Objekt einfach mit der Maus über den Marker im Kamerabild gezogen wird und losgelassen wird.

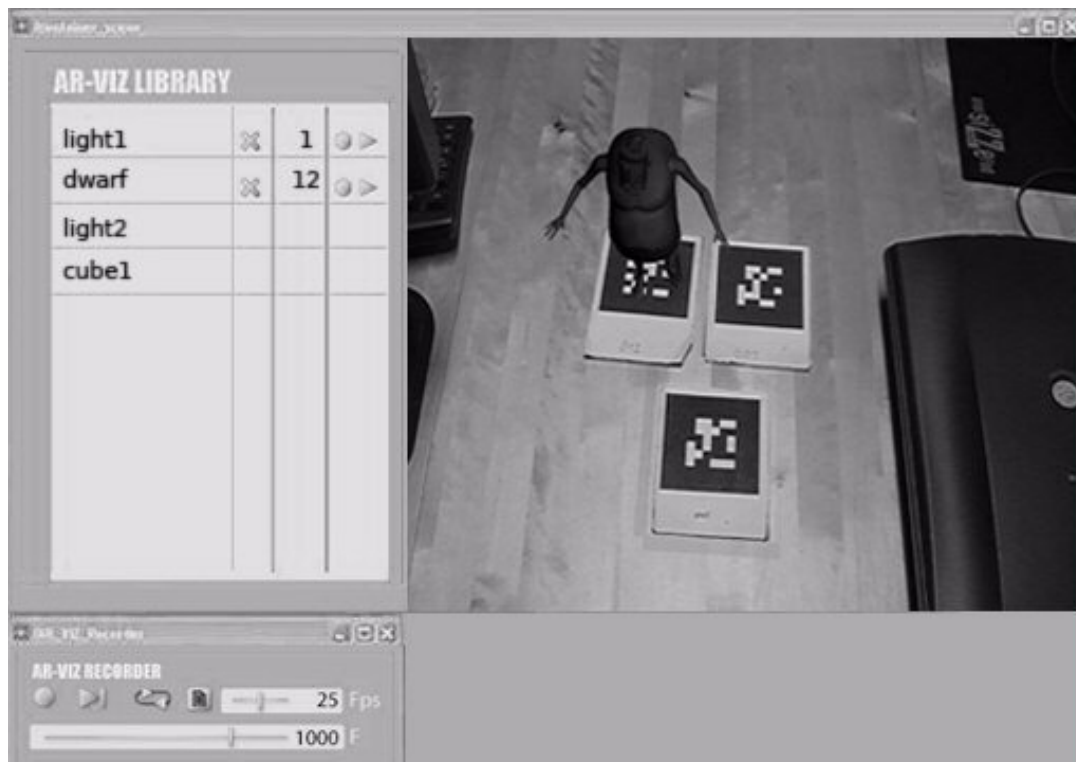


Abb. 9: AR PreViz Tool Graphic User Interface mit 3D Library

Die ID des Marker Nodes wird in der 3D Library direkt neben dem 3D Objekt angezeigt, mit dem es verbunden ist. (Anmerkung: Das GUI befindet sich immer noch in der Umsetzung, diese Funktionalität wurde bislang nur beispielhaft implementiert) Die Verbindung Marker Node – 3D Objekt kann jederzeit durch einen Button (Anm.: bislang nur beispielhaft implementiert) wieder aufgehoben und dem Marker Node ein anderes 3D Element zugewiesen werden. Die Verbindung Marker Node - 3D Objekt wird auch automatisch gelöscht, wenn ein 3D Objekt aus der Bibliothek entfernt wird (Anm.: bislang nur beispielhaft implementiert).

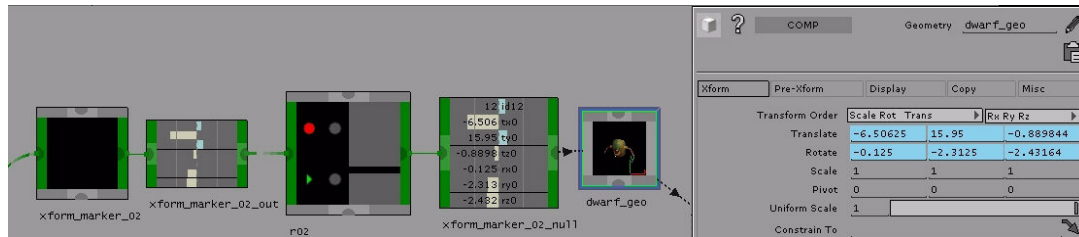


Abb.10: Interne Verbindung Marker Node – 3D Objekt

Für jeden Marker Node wird ein Record und ein Play Button generiert, mit denen sich die Markerbewegungen aufnehmen und abspielen lassen. Nach Aufzeichnen der Bewegungsdaten, kann man in den Play Modus wechseln und die Bewegung des 3D Objekts beobachten. Der Trackingmarker kann aus dem Sichtfeld der Kamera entfernt werden. Dieser Vorgang kann der Reihe nach für eine Vielzahl von Marker Nodes wiederholt werden.

Wenn alle Bewegungen aufgezeichnet wurden, kann man mit dem AR Recorder die 3D Daten exportieren, oder ein Video der geplanten Einstellung aufzeichnen.

3.3.3 Maya AR Post Production Interface

In unserem Workflow ist das AR Previz Maya Plug In eine Schwesteranwendung zur Touchanwendung, um beim Erstellen von Previsualizations auf alle Features von Maya zurückgreifen zu können. Dies lohnt sich vor allem bei VFX-Produktionen, die bereits mit Maya und entsprechendem Fachpersonal arbeiten.

Autodesk Maya

Maya von Autodesk wird zum Modellieren, Animieren und Rendern von virtuellen Objekten in 3D benutzt. Maya lässt sich durch Plug Ins erweitern. Dies geschieht über eine relativ gut dokumentierte C++ API und bietet die Möglichkeit, in nahezu alle Programmabläufe einzugreifen. Es gibt zwei wichtige Arten von Plug Ins für Maya, Command Plug Ins und Node Plug Ins.

Programmablauf

Im Gegensatz zur Touch Anbindung handelt es sich bei dem Maya AR Interface um ein Plug In, das Tracking kann also direkt von Maya aus gestartet werden. Das AR Previz Plug In ist von Claus Weymann und Andreas Gutsche [GW07] als eine Kombination von Command Plug In und Node Plug In umgesetzt worden. Die Anmeldung des Plug Ins funktioniert ähnlich wie bei Command Plug Ins über `initializePlugin()`, nur die `register`-Methode unterscheidet sich.

Ausserdem wurde eine Reihe neuer Node Plug Ins erstellt, die z.B. für die Darstellung der Videodaten auf der Image Plane bzw für die Erstellung von Marker Nodes verantwortlich sind. Das Kernstück eines Node Plug Ins ist die `creator` Funktion, mit der Knoten dieses Typs erzeugt werden. Die eigentlichen Daten des Knotens werden als Attribute abgespeichert, und sind über sogenannte Plugs also Stecker für andere Knoten benutzbar.

Das Basis Plug-In steuert das Anmelden sowie das Abmelden und bildet so das Grundgerüst für die zu Implementierende Funktionalität. Nachdem der User also im Plug In Manager das Plug In geladen hat, bietet initializePlugin() die Möglichkeit das Plug-In auf die Benutzung vorzubereiten. Hierzu gehört das Anlegen der Datenstruktur zur Knotenverwaltung.

Nach dem Laden erscheint die GUI. Durch den Play Button im Userinterface wird das Tracking gestartet (MEL-Befehl viskit -cmd start wird ausgelöst).

Die Marker werden automatisch im Kamerabild erkannt und die Bewegungen der Marker auf virtuelle Objekte (Marker Nodes) übertragen. Derzeit muss das Tracking für die Verknüpfung gestoppt werden.

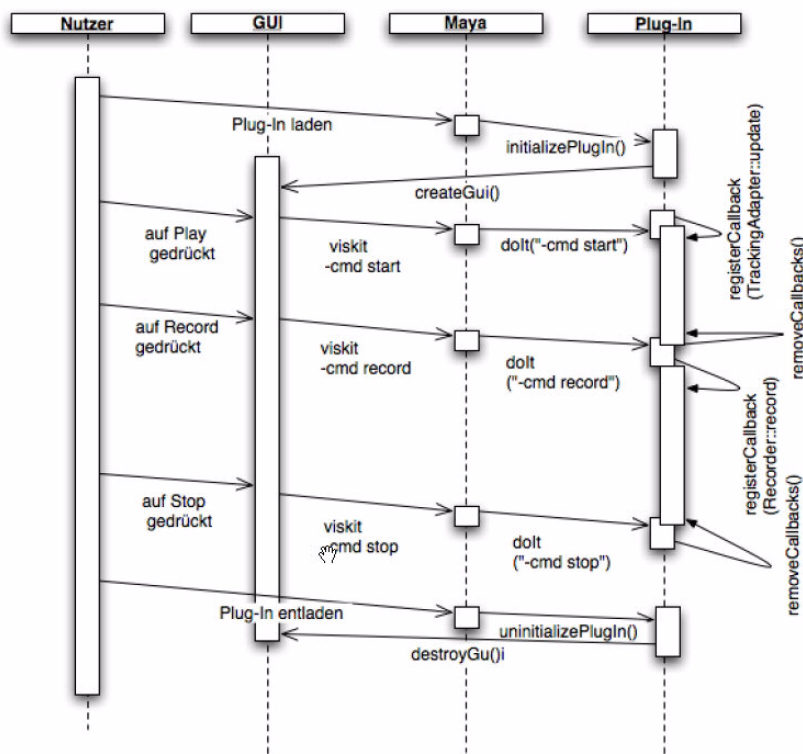


Abb. 11: Programm Ablauf Diagramm [GW07]

Im Record Modus sollen diese Bewegungen als Keyframes zu den entsprechenden Objekten in Maya gespeichert werden. Dieser wird durch klicken auf den Record Button gestartet, welcher seinerseits viskit -cmd

record auslöst. Beim Record Modus wird zum Aufzeichnen der Keyframes eine Kapselung geschaffen, um die dem Marker untergeordneten Knoten zusammenzufassen und für diese Gruppe Animationskurven zu erstellen. Die Knotenstruktur verwaltet die Daten, die für die beiden Modi benötigt werden. Eine besondere Herausforderung besteht darin Menge der Markerknoten die Maya bekannt sind und die im Plug In verwaltet werden, zu synchronisieren. Dies ist besonders problematisch beim Laden von Szenen .

Echtzeitdarstellung

Um das Kamerabild im Maya Hintergrund möglichst flüssig anzuzeigen werden zwei Möglichkeiten angeboten, zwischen denen vor dem Kompilieren des Plug Ins gewählt werden muss in dem man die passende Einstellung in der config.h vornimmt. Dies soll bei der Weiterentwicklung über einen Installer gelöst werden.

VideoPlane

Die VideoPlane (Camera Image Plane) ist eine Ebene orthogonal zur Kamera, die als Hintergrundbild benutzt wird. Mit Hilfe der Klasse MPxImagePlane wurde ein Knotentyp erstellt, der jeweils den Pointer zum aktuellen Bild der Kamera speichern und anzeigen kann. Diese direkte Vorgehensweise ist allerdings nicht besonders schnell, da vermutlich das Kopieren der Bilddaten von einem char array in das Maya interne Format von MImage etwas Zeit in Anspruch nimmt. Großer Vorteil dieser Herangehensweise, ist das ein schon bekanntes Konzept aus Maya nun für Videoinhalte genutzt werden kann, ohne das man sich um Feinheiten kümmern muss.

Renderer

Ein neueres Feature seit Maya 8.0 ist die Möglichkeit seinen eigenen

Viewport zu zeichnen, entweder mit OpenGL oder mit DirectX. Technisch heißt das, dass man das eigene Plug In einem externen Viewport Renderer registrieren muss und sich von Maya danach einen Viewport Render Context übergeben lassen kann, in dem man alles machen kann, was mit der jeweiligen API (OpenGL/DirectX) möglich ist. Die main-loop bleibt jedoch unter Mayas Kontrolle, da ansonsten die Applikation auf ein Refresh des Viewports angewiesen wäre um weiterlaufen zu können. Dieses Feature bedeutet zwar völlige Flexibilität und Unabhängigkeit vom Maya Viewport und könnte z.B. für Spielentwickler interessant sein, die ihre Engine so implementieren können, jedoch bedeutet es ebenfalls, dass man das zeichnen der Maya Objekte und Entitäten komplett selbst übernehmen muss. Es wurde zwar ein leichter Performance Gewinn beobachtet, der jedoch erwies sich auf schnellen Maschinen als marginal. Trotzdem würden moderne Effekte wie Pixelshader (für Lichteffekte, Compositing usw.) oder eine Erweiterung des Viewport Konzepts mit optimiertem Code wahrscheinlich nur in dieser Konstellation mit einem eigenen Renderer möglich sein.

4 Ergebnisse

Wir haben einige Anwendungsbeispiele aus der Pre Production mit den entwickelten Tools durchgespielt, und festgestellt, dass unser Workflow funktioniert und in der Praxis getestet werden kann.

Bei Tests hat sich herausgestellt, dass der User die Freiheit braucht nach einander Positionen und Animationen für Figuren, Kamera und die Ausleuchtung zu definieren. Das System bietet die Möglichkeit, Markerbewegungen aufzuzeichnen. Dies geschieht z. Teil automatisch wenn ein Marker verdeckt wird bzw. auf Wunsch des Nutzers in der Software. Die Aufnahmefunktion ermöglicht es, die Bewegungen von verschiedenen 3D Elementen nach einander aufzuzeichnen, so die Szene Schritt für Schritt aufzubauen und die Marker danach aus der Szene zu nehmen oder bei Bedarf wieder reinzunehmen. Dieser Workflow entspricht den Gewohnheiten des Filmemachens und minimiert gleichzeitig die Gefahr, dass Marker durch andere Marker verdeckt werden.

Es hat sich gezeigt, das die Touchanwendung und das Maya Plug In unterschiedliche Vor- und Nachteile besitzen:

Touch Pro

- Optimale Echtzeit Render Architektur / Performance
- Überlegene Echtzeitdarstellung durch Open GL 2.0 Unterstützung (Realistische Darstellung von Schatten und Oberflächen)
- Compositing in Echtzeit möglich
- Lifeausspielung von Videos möglich
- Weiterverarbeitung der Daten mittels Channel Operatoren (CHOPS)
- GUI Design kann sehr gut den Anforderungen angepasst werden

Touch Contra

- Import / Export von 3D Objekten und Szenen nötig und nicht immer möglich
- Deutlich begrenzte Möglichkeiten bei der Manipulation von 3D Objekten

Maya Pro

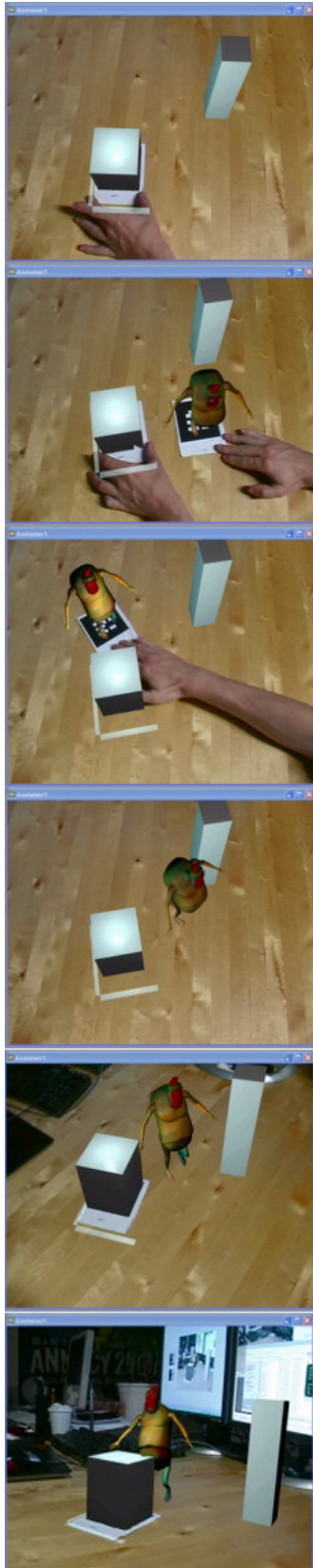
- Überlegene Möglichkeiten zum Bearbeiten von 3D Objekten und Animationen in der Software bei entsprechender Fachkenntnis
- Marker Bewegungen können beliebigen Maya Parametern zugeordnet werden

Maya Contra

- nicht so gute Echtzeitdarstellung über internes CGFX Plug In
- Verminderte System Performance
- Tracking muss gestoppt werden um die Verbindung zwischen Marker Node – 3D Objekt herzustellen

Je nach Aufgabenspektrum und Pipeline einer Filmproduktion werden die einen oder anderen Vorteile überragen. Der bestehende Entwicklungsstand ermöglicht die Erprobung und Weiterentwicklung in einem sehr breiten Anwendungsfeld.

Anwendungsbeispiele



Zunächst wird die virtuelle Szene im Viewport der realen Kamera ausgerichtet. Da die Video-/ Trackingkamera fest auf einem Stativ steht, und sich nicht bewegt, kann man den Weltmarker anschließend aus dem Sichtfeld der Kamera nehmen, und er wird gespeichert

Anschließend werden 3D Objekte mit den Markern verbunden

Der Regisseur bewegt über Marker die virtuellen Characters Abbildung
Nach einigen Tests wird die Bewegung aufgezeichnet.

Nun kann der Marker aus der Szene genommen und die Bewegung im Play-Modus überprüft werden (Abb. 4)

Um verschiedene Kameraeinstellungen mit der AR-Kamera auszuprobieren wird der Weltmarker wieder im Sichtfeld der Kamera positioniert (Abb5 und Abb6)

Abb.12: AR Camera Anwendung

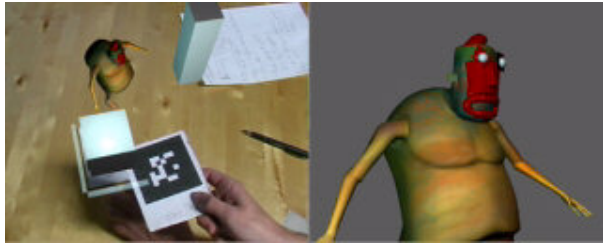


Abb.13: AR Camera und virtuelle Kamera

Zusätzliche virtuelle Kameras können Markern zugewiesen werden, was noch mehr Freiheit für Kamerawinkel und Brennweiten ermöglicht.

Mit dem Maya Plug In wurden ebenfalls beide Anwendungen, mit AR Kamera und mit virtueller Kamera, durchgespielt. In Abb ist im oberen Viewport das augmentierte Kamerabild zu erkennen. Es bietet einen sehr guten Überblick über das Szenario. Der Kameramann und der Regisseur planen eine Einstellung, wobei beide gleichzeitig jeweils einen Marker bewegen. Die Videokamera ist fest auf einem Stativ montiert. Der Kameramann führt mit einem Marker die virtuelle Kamera, wobei im unteren das Bild der virtuellen Kamera in Echtzeit angezeigt wird. Die Szene war bereits vorhanden, und konnte während der Arbeiten mit dem Maya Plug In editiert werden.

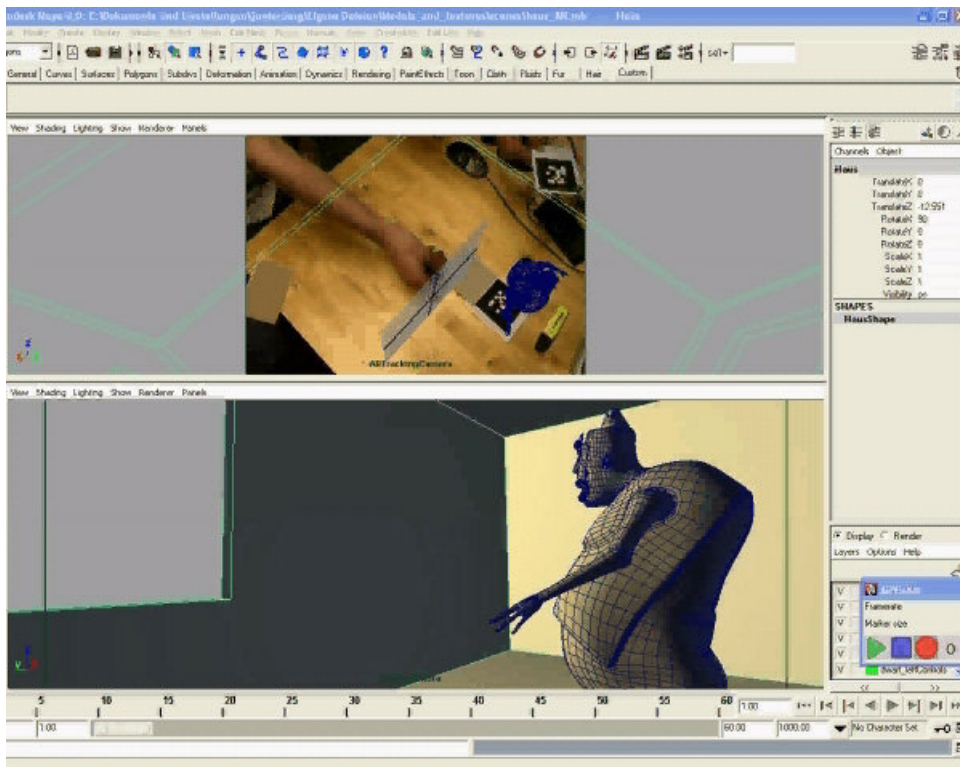


Abb.14: AR Camera und virtuelle Kamera in Maya

5 Fazit und Ausblick

Das System nutzt Augmented Reality Technologie als intuitive und kostengünstige Eingabemöglichkeit, um Filmaufnahmen in Echtzeit zu planen. Es bietet eine bisher nicht vorhandene Schnittstelle für die Zusammenarbeit von Szenenbild, Regie, Kamera, Schnitt und Visual Effects. Es bietet allen beteiligten die Möglichkeit in Echtzeit virtuelle Figuren, Sets, Kameras und Lichtquellen zu manipulieren und virtuelle Objekte im Zusammenspiel mit realen Hintergründen zu betrachten. Die virtuellen Objekte können durch die Marker, die sie repräsentieren, förmlich in die Hand genommen werden. Auf diese Weise können Filmszenen egal ob sie virtuell, real oder als Kombination umgesetzt werden sollen ohne spezielle Computerkenntnisse geplant, diskutiert und gestaltet werden. Die Ergebnisse können in Form von Animationsdaten und Videoclips gespeichert und exportiert werden und so von den einzelnen Abteilungen in den üblichen Produktionsprozessen weiter bearbeitet werden. Durch das Maya Plug In wurde ein Post Production Interface geschaffen, mit dem sich Kameraeinstellungen auch direkt in einer bereits vorhandenen Maya Szene proben lassen. Über die Touchanwendung lassen sich Szenen bei hoher Echtzeitperformance mit realistischeren Schatten und Oberflächen darstellen und mit Realaufnahmen auch über Blue- oder Greenscreens kombinieren.

Durch das markerbasierte Tracking ist die Bewegungsfreiheit auf das Sichtfeld der Kamera beschränkt. Dafür sind jedoch Bewegungen in allen Achsen möglich. Das gesamte System ist modular aufgebaut und individuell erweiterbar.

Es gibt verschiedene Denkansätze für die Weiterentwicklung des AR PreViz Systems:

- Verwendung anderer Trackingsysteme, z.B. nicht markerbasiertes Tracking etc.
- Motioncapture und/oder Import von Characteranimation
- Kombinierte Marker zur Verbesserung des Tracking, Verwendung

mehrere Weltmarker etc.

- Verwendung mehrerer AR Cameras für genauere Trackingdaten
- Erstellung einer Bibliothek mit mehreren vordefinierten Objekten wie z.B. Filmscheinwerfer, Kameras, 3D Modelle
- Optimieren der Userinterfaces

Das AR PreViz Tool soll beim nächsten Kurzfilm von Andreas zum praktischen Einsatz kommen. Bei dem Projekt soll die Planung von Szenenbild, Kameraarbeit und VFX durch das PreViz Tool unterstützt werden. Benutzerfreundlichkeit und Stabilität sollen weiter getestet und verbessert werden.

Abbildungsverzeichnis

Abb. 1: Videoanbindung

Abb. 2: Beziehung zwischen Markern und Kamera [Bru07]

Abb. 3: Die Abbildung zeigt Channel Operatoren, 3D und 2D Compositing

Abb. 4: Beispiel einer Benutzungsoberfläche mit Schaltflächen und
Reglern zur Positionsbestimmung von Lichtquellen

Abb. 5: Echtzeitshading [Joe04]

Abb. 6: Network socket Interface

Abb. 7: OpenGL Videodarstellung und Ausgabe der Trackingparameter

Abb. 8: Shared Memory In Top, Tracking Daten und augmented Videobild

Abb. 9: AR PreViz Tool Graphic User Interface mit 3D Library

Abb.10: Interne Verbindung Marker Node – 3D Objekt

Abb.11: Programm Ablauf Diagramm [GW07]

Abb.12: AR Camer Anwendung

Abb.13: AR Camera und virtuelle Kamera

Abb.14: AR Camera und virtuelle Kamera in Maya

Literaturverzeichnis

[DSVL] <http://www.sourceforge.net/projects/dsvideolib>.

[ART+] *AR ToolKit Plus Documentation*
http://studierstube.icg.tu-graz.ac.at/handheld_ar/artoolkitplus_docu/index.html
[Online, Oktober 2006]

[ART] *Dokumentation des ARToolkit*.
<http://www.hitl.washington.edu/artoolkit/documentation/>
[Online, Oktober 2006]

[GNU] *GNU Free Documentation Licenses*
<http://www.gnu.org/licenses/fdl.html>
[Online, März 2007]

[Hiro] Kato Hirokazu. *Inside ARToolKit*.
<http://www.hitl.washington.edu/artoolkit/Papers/ART02-Tutorial.pdf>
[Online, Dezember 2006]

[Wag06] Wagner Daniel. *Handheld augmented reality-ARToolkitPlus*.
<http://studierstube.icg.tu-graz.ac.at/handheld---ar/artoolkitplus.php>
[Online, Dezember 2006]

[Lam06] Philip Lamp. *Webseite des ARToolkit*
<http://www.hitl.washington.edu/artoolkit/>
[Online, Oktober 2006]

[GW07] Claus Weymann, Andreas Gutsche *AR-Prävisualisierung für Filmproduktion als Maya Plug-In* Studienarbeit im Studiengang Computervisualistik, Koblenz, Mai 2007

[Bru07] Martina Brümmer *Realisierung einer eigenständigen Trackingbibliothek auf Basis des ARToolkit und des ARToolkitPlus* Studienarbeit im Studiengang Computervisualistik, Koblenz, April 2007

[Azu04] Ronald Azuma *Overview of augmented reality* In ACM SIGGRAPH 2004 courses notes, SIGGRAPH 2004

[BR05] Oliver Bimber, Ramesh Raskar *Spatial augmented reality: Modern approaches to augmented reality* ACM SIGGRAPH 2005 courses notes, SIGGRAPH 2005, S. 102 -138

[Ohs01] T. Ohshima, T. Kuroki, T. Kobayashi, H. Yamamoto, and H. Tamura
2001: An MR-Space Odyssey": Applying mixed reality technology to VFX in filmmaking In SIGGRAPH 2001 Conference Abstracts and

Applications, p. 142, 2001

[Par05] Mideum Shin, Byung-soo Kim, Jun Park *AR Storyboard: An Augmented Reality based Storyboard Authoring Tool* IEEE and ACM International Symposium on Mixed and Augmented Reality, Oct. 6-8, 2005

[Ich06] Ryosuke Ichikari, Keisuke Kawano, Asako Kimura, Fumihisa Shibata, and Hideyuki Tamura *Mixed Reality Pre-visualization and Camera-Work Authoring in Filmmaking* In Proc. 5th Int. Symp. on Mixed and Augmented Reality, pp.239-240, 2006

[Mar04] Fiala Mark. *ARTag Revision 1. A Fiducial Marker System Using Digital Techniques* published as NRC/ERB-1117. November 2004, 46 Pages. NRC 47419
<http://www.artag.net> [Online, Dezember 2006, aktualisiert Februar 2007]

[SW07] Wagner Daniel, Schmalstieg Dieter. *ARToolkitPlus for Pose Tracking on Mobile Devices* In Proceeding of 12th Computer Vision Winter Workshop, CVWW 2007)
<http://www.icg.tu-graz.ac.at/Members/daniel/Publications/ARToolkitPlus>
[Online, February 2007]

[DART04] Blair MacIntyre, Maribeth Gandy, Steven Dow and Jay David Bolter *DART: A Toolkit for Rapid Design Exploration of Augmented Reality Experiences* In Proceedings of the 17th annual ACM symposium on User interface software and technology UIST 04, October 24–27, 2004, Santa Fe, New Mexico, USA.

[Ber04] Rodney Berry, Naoto Hikawa, Mao Makino, Masami Suzuki, Takashi Furuya
Authoring augmented reality: a code-free approach ACM SIGGRAPH 2004 Posters SIGGRAPH '04

[Ber06] Rodney Berry, Masahide Naemura, Yuichi Kobayashi, Masahiro Tada, Naomi Inoue, Yusuf Pisan, Ernest Edmonds *An interface test-bed for 'Kansei' filters using the touch designer visual programming environment* In Proceedings of the 7th Australasian User interface conference - Volume 50 AUIC '06, January 2006