

## VRML-Plug-ins auf dem Prüfstand Fluchtpunkt

Frank Decker,  
Christoph Runde

Die Virtual Reality Markup Language wurde zwar speziell für Anwendungen im Internet konzipiert, das schließt jedoch nicht aus, sie für VR-Systeme anderer Art zu verwenden. Ein Test der drei verbreitetsten VRML-Viewer untersucht, ob diese Browser-Plug-ins die erforderliche Leistung bringen.

Für die Entwicklung von VR-Anwendungen sind seit langem umfangreiche Programmpakete wie das World Tool Kit von Sense8 oder Vega von Multigen Paradigm kommerziell erhältlich. Sie erlauben eine flexible Zusammenstellung der mitgelieferten Funktionskomponenten, was sich insbesondere bei der Integration der zum Teil umfangreichen Interaktionstechnik positiv auswirkt. Tracking-Systeme, Datenhandschuhe, Force-Feedback-Gelenkarmsysteme sollen unter Umständen mit mehrkanaligen, stereoskopischen Projektionssystemen kooperieren. Ein solches Zusammenspiel realisieren bereits viele VR-Anwendungen,

die auf Basis der genannten Entwicklungsbibliotheken entstanden sind.

Diesen offensichtlichen Vorteilen der professionellen VR-Softwarepakete stehen einige Nachteile gegenüber. Neben den relativ hohen Anschaffungskosten zählen dazu eine verbesserungswürdige Darstellungsperformanz, die Komplexität in der Bedienung und eine mäßige Unterstützung der Standardbeschreibungssprache für virtuelle Welten, VRML 2.0 (ISO/IEC 14772-1:1997 und ISO/IEC 14772-2:2002).

Letztere, ursprünglich für die Darstellung interaktiver virtueller Umgebungen im Internet entwickelt, umfasst neben der Geometrieinformation Features wie Animation, 3D-Sound, einfache Interaktion über VRML-Sensorik (Touch, Visibility, Proximity et cetera), verschiedene Lichtquellentypen, Sichtlimit, Nebel et cetera. Seit 2002 ist neben der polygonalen Geometriebeschreibung die Darstellung von Nurbs-Modellen (Non-Uniform Rational B-Spline) möglich. Besonders hilfreich sind die Erweiterungsmöglichkeiten von VRML 2.0: Benutzer können mit dem *PROTO*-Statement eigene Elemente definieren. Skriptknoten erlauben die Programmierung beliebiger Objektfunktionen, beispielsweise das Hinterlegen inverser Kinematiken für 3D-Robotermodelle.

Bedingt durch den enormen Markt der PC-Spiele haben die Hersteller von Grafikchips in den letzten Jahren günstige, aber leistungsfähige stereovisionfähige Grafikchips (Z-Buffer-Stereo unter Direct3D-Rendering) auf den

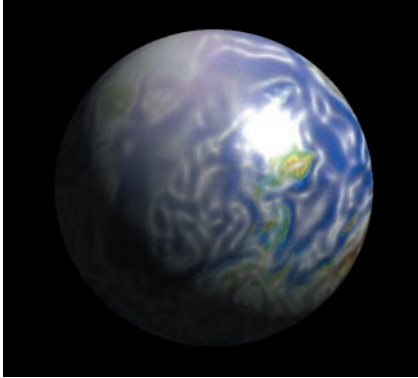
Markt gebracht. Zusammen mit erschwinglichen Shutter-Brillen und kostengünstiger Interaktionstechnik wie dem Datenhandschuh P5 von Essential Reality [1] lassen sich damit relativ einfach VR-Applikationen erstellen – den passenden Viewer vorausgesetzt.

### Diverse Szenen testen Performance

Damit rücken VRML-Browser-Plug-ins in den Fokus näherer Betrachtung. Das Fraunhofer IPA hat diesen Umstand zum Anlass für einen aktuellen Vergleich der momentan verbreiteten VRML-Viewer genommen: BS Contact VRML der Bitmanagement GmbH ([www.bitmanagement.de](http://www.bitmanagement.de)), Cortona VRML Client von Parallelgraphics ([www.parallelgraphics.com/products/cortona/](http://www.parallelgraphics.com/products/cortona/)) und Cosmo Player. Während die Irix-Version über SGIs Website erhältlich ist ([www.sgi.com/software/cosmo/player.html](http://www.sgi.com/software/cosmo/player.html)), gilt für die Windows-Version nach wie vor die amerikanische Website von Computer Associates als Bezugsquelle ([www.cai.com/cosmo/](http://www.cai.com/cosmo/)), obwohl die Firma nach Auskunft der deutschen Niederlassung nicht mehr im Besitz der Software ist. An wen Cosmo Player verkauft wurde, war nicht in Erfahrung zu bringen. Da BS Contact eine Weiterentwicklung des Viewers von Blaxxun ist, entfällt die zusätzliche Analyse des Blaxxun-Plug-ins in diesem Test. Alle Viewer lassen sich als ActiveX-Elemente in eigene Applikationen einbinden (beispielsweise in C++).

#### ABLAUF DER BERGWERKSSZENE

Zeit [s]	Aktion
Bei Initialisierung der Szene	Förderband (links im Bild) läuft
TouchSensor wird aktiviert: t=0	Messung der Framerate beginnt
t=1	Hydraulik (rechts im Bild) senkt sich ab
t=4	Animation Ende
t=5	Maschinen (beige, rechts im Bild) werden an das Förderband gerückt
t=8	Animation Ende
t=9	Hydraulik (rechts im Bild) hebt sich
t=12	Animation Ende
t=14	Messung der Framerate endet



**Zum Test der Plug-ins dienten zwei VRML-Dateien: eine Kugel und eine Bergwerkszene (siehe Aufmacher) mit jeweils unterschiedlich vielen Polygonen und diversen Texturauflösungen (Abb. 1).**

Zum Testen der Darstellungsgeschwindigkeit diente ein Multiprozessorrechner mit zwei Intel Xeon 2,4 GHz und 1024 MByte Speicher, der unter Windows 2000 (Service Pack 4) lief. Als Grafikumgebung kam die Version 7a von DirectX zum Einsatz (bei neueren Versionen gab es Probleme mit den Treibern im Zusammenspiel der Viewer mit der Wildcat-Grafikkarte). Die Farbtiefe betrug 32 Bit, die Auflösung 1024 × 768.

Das Rendering erfolgte sowohl mit hardwarebeschleunigtem OpenGL als auch mit Direct3D jeweils unter Einsatz folgender Grafikkarten:

- ATIs Radeon 9000 Pro mit 128 MByte DDR
  - Nvidias Quadro4 900 XGL (Treiber Nvidia Version 3.1.0.0)
  - 3Dlabs' Wildcat III 6210 mit 256 MByte Texturspeicher
- Aufgabe der Plug-ins war es, folgende VRML-Dateien darzustellen:
- Rotierende Kugel (Abbildung 1)
  - mit 100 000 (circa 9 MByte Dateigröße) und 200 000 Polygonen (circa 18 MByte),
  - mit Texturen im GIF-Format mit den Pixelauflösungen 512×512 (164 KByte), 1024×1024 (594 KByte), 2048 × 2048 (2,219 MByte) und 4096 × 4096 (8,519 MByte),

- mit und ohne Shading des kompletten Objekts,
- mit einer und mit acht Punktlichtquellen sowie
- mit einem Script, das die Framerate ermittelt und anzeigt.

Das zweite Motiv war eine größere 3D-Szene mit Maschinenteknikdarstellung, die mehrere LODs (Level of Detail), Animationen, Texturen, Sounds, Skripts, PROTOs und Inlines enthält (siehe Aufmacher).

Ziel der Messungen waren die Framerate und der Speicherbedarf des Internet Explorers 5, in dem die Plug-ins liefen. Die Diagramme in Abbildung 2 zeigen das Ergebnis der Framerate-Messungen. Die Wildcat-Karte verfügt nicht über hardwarebeschleunigtes Direct3D-Rendering, daher taucht sie in diesem Diagramm nicht auf.

Da sich die Größe der Texturen bei der Framerate fast nicht bemerkbar machte, sind die Messergebnisse nicht nach diesem Kriterium aufgeschlüsselt. Lediglich der Cortona konnte mit der Wildcat-Grafikkarte die Kugeln mit der größten Textur (4096 × 4096 Pixel) nicht mehr darstellen. Der Bedarf des Hauptspeichers stieg in dem Fall auf zum Teil über 300 MByte, und der Viewer stürzte ab.

Beim OpenGL-Rendering punktete häufig BS Contact gegenüber Cortona, der lediglich bei vielen Lichtquellen meist schneller war. Auf dem letzten Platz landete Cosmo Player, konnte jedoch bei der Wildcat den Abstand gering halten.

Überragend schnitt BS Contact beim Direct3D-Rendering auf der Nvidia-Karte ab, auf der ATI Radeon konnte hingegen Cortona leichte Vorteile für sich verbuchen. Cosmo Player lag bei beiden Grafikkarten weit hinten, was leicht mit dem Entwicklungs-rückstand dieses Plug-ins zu erklären

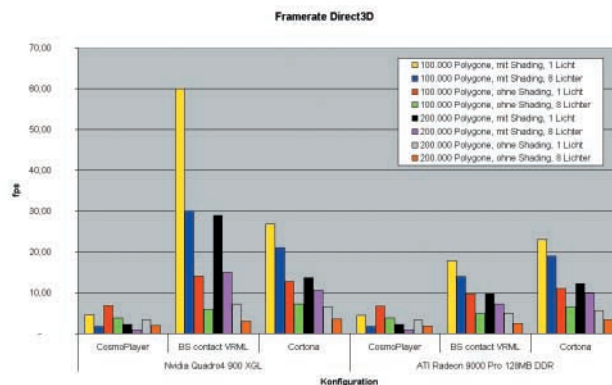
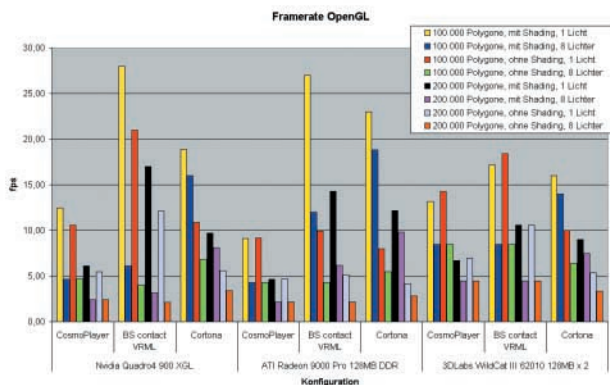
ist, da es neuere Direct3D-Funktionen nicht nutzen kann.

Die Nutzung des Hauptspeichers zeigte, dass die Viewer bei größeren Texturen nicht mit dem Texturspeicher der Grafikkarte auskommen, sondern zusätzliche Ressourcen erfordern. Am wenigsten beanspruchen sie den Hauptspeicher beim Einsatz der Wildcat-Karte, die über den größten Texturspeicher der drei beim Test eingesetzten Beschleuniger verfügt. Generell benötigte Cortona den meisten zusätzlichen Platz, während Cosmo Player mit wenig zurechtkommt.

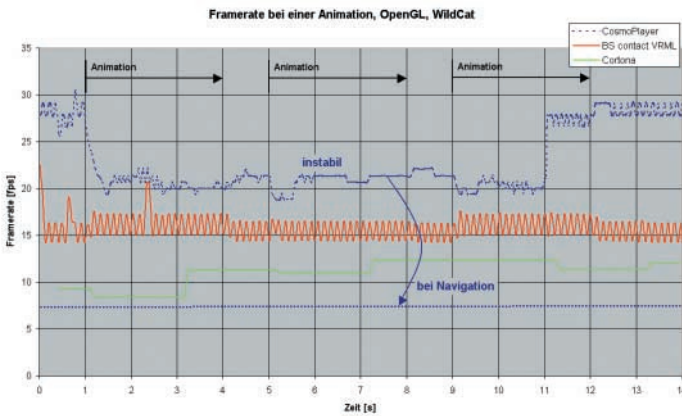
## Messergebnisse differieren stark

Als Nächstes erfolgte ein Test mit einer (kompletten) animierten, texturierten und vertonten 3D-Szene, die einen Ausschnitt aus der Maschinenteknik eines Bergwerks unter Tage darstellt (siehe Aufmacher). Sobald der Benutzer einen Touchsensor auslöst, beginnt die Messung der Framerate, während Animationen laufen und Sound erklingt. Die gleichnamige Tabelle (S. 68) zeigt den Ablauf der Bergwerksszene.

Die Messungen ergaben wiederum recht unterschiedliche Ergebnisse, abhängig nicht nur vom getesteten Viewer, sondern sowohl von der eingesetzten Grafikkarte als auch von der Rendering-Methode. Unter Direct3D dominierte klar BS Contact und erbrachte bessere Ergebnisse als unter OpenGL. Die meisten Schwierigkeiten mit dieser Szene hatte anscheinend Cortona. Cosmo Player zeigte ein unterschiedliches Verhalten, je nachdem ob der Betrachter die Szene ruhig betrachtete oder navigierte. Die in Abbildung 3 im oberen Teil gestrichelt eingezeichnete Framerate dieses Plug-ins galt auf der Wild-



**Während die Performance der Viewer bei unterschiedlichen Texturgrößen nicht stark differierte, führte unter anderem die Anzahl der Lichtquellen zu deutlich unterschiedlichen Ergebnissen (Abb. 2).**



**Bei einer animierten Szene gibt es deutliche Unterschiede in der Framerate der getesteten Viewer, unter anderem abhängig von den eingesetzten Grafikfunktionen (Abb. 3).**

cat-Karte nur bei stehendem Betrachter: Sobald er mit der Navigation begann, brach die Framerate auf einen Wert zusammen, den die untere gestrichelte Linie in der Abbildung andeutet. Andere Kombinationen zeigten dieses Verhalten nicht. Erstaunlich an den Ergebnissen ist einerseits, dass abgesehen von diesen Performanceeinbrüchen Cosmo Player unter OpenGL zu guten Ergebnissen kommt. (Weitere Diagramme mit Messergebnissen sind über den iX-Listingsservice erhältlich.)

Unterschiede zu den ersten Kugel-Tests sind in erster Linie darin zu sehen, dass die Bergwerksszene im Gegensatz zu den Kugeln viel mit LOD,

Skripting, verschiedenen Texturen und Sound arbeitet. Die Struktur ihres Szenengraphs ist zudem wesentlich komplexer als die des Kugel-Szenengraphs, die sich fast nur aus der geometrischen Beschreibung zusammensetzt.

Dass Cosmo Player als eines der be- tagtesten Plug-ins (Stand von 1997) nach wie vor zu den relevanten Viewern gehört, zeigt die Qualität der damaligen SGI-Entwicklung. Zwischenzeitig war er in Gestalt des Pivoron Player der Firma Nexternet wieder aufgetaucht (fast unverändert, nur die Möglichkeit zum Vollbild war neu), ist jedoch ge- nau so rasch wieder verschwunden. Ähnlich verhält es sich mit Intervista World- view, der zwar noch über die Website von Computer Associates erhältlich ist, sich aber technisch auf einem Stand von 1998 befindet. Relativ neu ist Octagon Player der Firma Octaga aus Norwegen, der allerdings (noch) wenig Verbreitung gefunden hat. Erste Tests zeigten hier gute Framerates.

Abschließend gehen die folgen- den Ausführungen auf Stärken und Schwächen der getesteten Viewer ein.

### Vorteile von VRML 2.0

Im Vergleich zu anderen 3D-Formaten, insbesondere gegenüber proprietären Formaten von VR-Softwareherstellern verfügt die Virtual Reality Markup Lan- guage über folgende Vorteile:

- Exportfilter gängiger Softwarelösungen im CAD- und Modellierungs- bereich sind nutzbar.
- Softwarepakete wie 3DS Max von Dis- creet und Cinema 4D von Maxon beinhalten VRML-Modellierwerkzeuge.
- Einige Hersteller bieten spezielle VRML-Programmierwerkzeuge an (bei- spielsweise VRML-Pad von Parallelgraphics).
- Auch VRML-Code-Optimierwerkzeu- ge sind verfügbar (etwa Chisel von Tra- pezium).
- Texturen und Sounds erhöhen den Im- mersionseffekt.
- VRML-Sensoren erlauben eine um- fangreiche Interaktion.
- VRML-Interpolatoren sorgen für viel Dynamik.
- Die GeoVRML-Erweiterung ermög- licht die Darstellung von Geologiedaten.
- Javascript erlaubt eine beliebige Ob- jektmodellierung (Logik, Abhängigkei- ten, Kinematik).

### Cosmo Player

Der Cosmo Player hat einige Mängel, die infolge unterbliebener Weiterent- wicklung nie korrigiert wurden. So schneidet die Sichtweitenbegrenzung bei der Navigation zwar die Welt kor- rekt ab, das hat jedoch keinen Einfluss auf die Darstellungsgeschwindigkeit. Knoten, die *CreateVrmlFromXXX* dyna- misch in einer Szene erstellt, lassen sich über das EAI-Interface nicht anspre- chen, wodurch es unmöglich ist, über eine Anwendung 3D-Objekte zu impor- tieren und anschließend über die EAI- Schnittstelle zu steuern. Wer mit Cosmo Player demnach Szenen mit veränderba- ren 3D-Modellen betrachten möchte, muss wohl oder übel zunächst eine VRML-Datei erzeugen, die die aktuelle

Konfiguration enthält. Erst dann lädt der Viewer diese Szene und kann *eventIns* an alle Empfänger senden.

Bei größeren Szenen erweisen sich die Animationen als empfindlich insta- bil, da bei hoher Auslastung des Vie- wers Animationen einfrieren können und sich danach nie mehr ansprechen lassen. Weniger gravierend, aber nicht schön ist das Ignorieren der Radiusfel- der der Punkt- und Spotscheinwerfer unter OpenGL sowie eine fehlerhafte Einstellung des Rotationszentrums bei Gruppenknoten. Wenn MPEG1-Filme als Textur dienen, kann ihr Abspielen zum Einbruch der Framerate führen. Bildschirmfüllende Darstellungen (unter Windows 2000 und Windows XP mit aktuellen Stereotreibern, notwendig zum Umschalten auf den Stereomodus) sind mit dem Cosmo Player nicht möglich.

### Cortona

Auch Cortona geht mit der Sichtwei- tenbegrenzung nicht so um, wie man es sich wünscht: Sie funktioniert rein visuell, bringt aber ebenfalls keinen Geschwindigkeitsvorteil. Störend ist zudem die feste Kopplung zwischen Variablen. Weist eine Variable in meh- reren *IS*-Anweisungen anderen Variab- len Werte zu, wirken sich Manipulationen an der zuweisenden Variablen uner- wünschterweise auch auf die anderen aus. Auf der anderen Seite hat Cortona einige nette Features wie Fit Scene (Ausfüllen des Sichtfensters mit der Szene), Bump Mapping (Einfluss der Lichteffekte auf die plastische Darstel- lung eines Objekts) und Unterstützung für Microsofts Sidewinder Joystick.

### BS Contact

Als einziger der drei Viewer unter- stützt BS Contact die Sichtweitenbe- grenzung auf korrekte Weise, sodass mit ihm wesentlich größere Szenen dar- stellbar sind. Bei großen Szenen kann es allerdings vorkommen, dass Anima- tionen stocken, wenn der Anwender mit der Maus über die Bildschirmober- fläche fährt (ohne zu navigieren). Wie Cosmo Player verträgt BS Contact das Abspielen von MPEG1-Texturen nicht immer klaglos. Fügt man mit dem Be- fehl *addChilden* einer Szene neue View- points oder Lichtquellen hinzu, lassen sich diese nicht wieder mit dem Befehl *removeChildren* entfernen. Umlaute in Kommentarzeilen der Skriptknoten füh-



ren zu Fehlermeldungen. Weiterhin empfiehlt es sich, innerhalb der Skriptknoten die *directOutput*-Werte auf *TRUE* zu setzen, da es sonst vorkommen kann, dass die definierten *eventIn*-Felder keine Nachrichten empfangen können. Schließlich war zu beobachten, dass das Kommando *saveUrl* keine lauffähigen VRML-Dateien erzeugt, falls zuvor der Befehl *addChilden* die Szene um ein *PROTO* erweitert hat.

Auf der Haben-Seite unterstützt dieser Viewer in seiner aktuellen Version Multitexturing (Verknüpfen mehrerer Texturen beim Rendering), Bump Mapping und Environment Mapping (Textur mit der Reflexion einer Umgebung auf einem 3D-Objekt). Zudem lassen sich DirectX-Eingabegeräte unmittelbar verwenden. Auf der Website der Bitmanagement GmbH steht dazu eine VRML-Datei zum Download zur Verfügung, die beispielhaft die von diesem Viewer ausgelesenen Gerätekanäle anzeigt. Weiterhin sind durch die Nutzung bereits angebotener Optimierungsmethoden für größere Szenen wie Portals & Cells, Quadtree-, Octree- oder Occlusion-Culling-Verfahren (Rendering-

Verfahren für die Darstellung komplexer Datenstrukturen), die die obigen Tests nicht angewendet haben, Performancesteigerungen zu erwarten.

## Fazit

Festzuhalten bleibt, dass sich eine gute Abstimmung zwischen VRML-Viewer, Grafikkarte und Rendering-Methode auszahlt. Die Vor- und Nachteile der einzelnen Viewer müssen zudem unter dem Licht verschiedener Lizenzkosten und -bedingungen gesehen werden. Für nicht kommerzielle Anwendungen dürfen alle zumindest auf Zeit verwendet werden. Hinsichtlich Cosmo Player galt dies bisher auch bei kommerzieller Nutzung. Die beiden anderen Viewer sind bei kommerzieller Anwendung kostenpflichtig. Beide Firmen erstellen kundenspezifische Angebote. Die Preise liegen in den Regionen wenige Euro (Cortona) und einige hundert Euro (BS Contact, inklusive deutschen Support).

Generell stellen die VRML-Viewer eine interessante Alternative zu den

eingangs erwähnten Profi-VR-Systemen dar, deren Preise bei circa einigen tausend Euro beginnen. Nachteilig sind ihre Kapselung sowie das EAI-Interface, das einen Bottleneck darstellt. Andererseits zeichnen sich die Plug-ins durch eine gute Performance und einfach zu erstellende, interaktive Umgebungen aus. (ka)

### FRANK DECKER

ist Dipl.-Ing. (FH) und arbeitet seit 2000 am Fraunhofer IPA in den Themenfeldern Digitale Fabrik und Virtuelle Realität.

### CHRISTOPH RUNDE

ist Dipl.-Ing. Dipl.-Kfm. und arbeitet seit 1999 am Fraunhofer IPA in den Themenfeldern Digitale Fabrik und Virtuelle Realität.

## Literatur

- [1] Jörn Loviscach; Aus der Luft greifen; Datenhandschuh P5 von Essential Reality; c't 4/03, S. 74 