

Multimedia im Internet

Oliver Vormberger

Das Internet hat sich zu einem weltumspannenden Kommunikationssystem entwickelt, welches für den Informations-Anbieter und den Informations-Suchenden gleichermaßen von Interesse ist. Dieser Beitrag streift kurz die historische Entwicklung des Internets mit seinen textbasierten Protokollen wie *email*, *news*, *ftp* und *telnet*. Danach werden moderne Konzepte zur Internetpräsenz vorgestellt und anhand einiger Beispiele demonstriert, wie der Informationsgehalt von Web-Seiten durch multimediale Komponenten wie Grafik, Animation, Audio, Video und Virtual Reality verbessert werden kann.

Eine Online-Fassung dieses Beitrags findet sich unter der Adresse <http://www-lehre.informatik.uni-osnabrueck.de/mmii/paper/index.html>

Die zum Betrachten der multimedialen Effekte erforderliche Software gibt es dort im Plugin-Verzeichnis und auch bei den in Tabelle 1 gezeigten Herstellern.

DjVu	http://djvu.research.att.com
Fraktale Kompression	http://www.iterated.com
FlashPix	http://www.livepicture.com
Flash	http://www.macromedia.com
Midi	http://www.liveupdate.com
RealVideo	http://www.real.com
VRML	http://cosmosoftware.com

Tabelle 1: Downloadadressen der Hersteller

1 Geschichte

Das Internet ist ein weltweiter Zusammenschluß von Computernetzen:

- 100 Länder sind beteiligt.
- 100.000 Netzwerke sind miteinander verbunden.
- 100.000.000 Nutzer wurden für 1998 geschätzt.
- 100 neue Nutzer kommen pro Minute hinzu.

Der Ursprung liegt in einem Ende der 60er Jahre von der *Advanced Research Projects Agency* geförderten Forschungsnetz des US-Verteidigungsministeriums. Ziel war es, nach einem Atomschlag die Kommunikationsfähigkeit zwischen den amerikanischen Regierungsbehörden aufrechtzuhalten. Daraus entstand das *ARPA-NET*, ein Netzwerk mit autonomen Knoten, Paketvermittlungen, Fehlertoleranz und alternativen Übertragungswegen. Abbildung 1 zeigt die prinzipielle Vernetzungsstruktur.

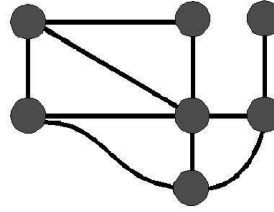


Abbildung 1: Vernetzung von autonomen Knoten

Internetdienste

Unter Verwendung von *TCP/IP* (*Transmission Control Protocol/Internet Protocol*) stehen auf den im Internet beteiligten Hosts verschiedene Dienste zur Verfügung, mit denen Klienten vom Server Daten abrufen können. Zur eindeutigen Kennzeichnung verwendet man einen *Uniform Resource Locator (URL)* mit folgendem Aufbau:

⟨Protokoll⟩ : // ⟨Computername⟩ : ⟨Port⟩ / ⟨Verzeichnis⟩ / ⟨Dokument⟩

Tabelle 2 faßt die wichtigsten Dienste zusammen.

Dienst	Zweck	Beispiel-URL
EMAIL	Nachricht verschicken	mailto:oliver@informatik.uos.de
NEWS	mit Newsgroup diskutieren	news://news.rzrn.uni-hannover.de
FTP	Dateien transferieren	ftp://ftp.rz.uos.de
TELNET	mit Rechner verbinden	telnet://plch.lib.oh.us
HTTP	durch Hypertext navigieren	http://www.informatik.uos.de

Tabelle 2: Internetdienste

HTTP-Links

Neben den rein textbasierten Protokollen *email*, *news*, *ftp* und *telnet* avancierte schließlich zum populärsten Dienst das *Hypertext Transfer Protokoll (HTTP)*. Es wurde 1989 am *CERN - European Laboratory for particle physics* entwickelt und ist ein zustandsloses, verbindungsorientiertes Protokoll für ein verteiltes Hypermedia-Informationssystem basierend auf dem Client/Server-Prinzip unter Verwendung des *Request/Response-Paradigma*. Es gibt vier Operationen:

Connection: Der Klient initiiert eine Verbindung zum Server, die von diesem bestätigt wird.

Request: Der Klient stellt eine Anfrage.

Response: Der Server sendet eine Antwort.

Close: Der Verbindungsabbau erfolgt entweder durch den Server nach der Übertragung der Daten oder durch den Klienten durch Abbruch.

Nach Kontaktaufnahme mit dem Server verlangt der Klient mit der Methode GET eine bestimmte Seite und teilt gleichzeitig mittels accept mit, welche Dateiformate er bereit ist, anzunehmen. Der Inhalt der Seite wird mit der *Hypertext Markup Language (HTML)* formuliert.

In vielen Fällen verbirgt sich hinter der HTTP-Adresse ein Informationsanbieter, der Anfragen zu bestimmten Themen beantworten kann. Tabelle 3 zeigt einige typische Adressen.

Was steht montags im Spiegel?	www.spiegel.de
Wer wohnt wo?	www.switchboard.com
Wer liefert was?	www.wlw.de
Wer verkauft einen Audi A6?	www.jahreswagen.wolfsburg.de
Wann fährt die Bahn?	bahn.hafas.de
Wer fliegt wohin?	www.aer.de
Welche Bücher gibt es?	www.buchweb.de
Wie stehen meine Aktien?	www.exchange.de
Wie finde ich wohin?	www.reiseplanung.de
Wo liegt mein Hotel?	www.varta-guide.de
Welcher Server liefert was	www.altavista.com

Tabelle 3: Informationsanbieter und ihre HTTP-Adressen

Typischerweise sind in dem Text der gelieferten HTML-Seite audiovisuelle Ergänzungen eingebettet, die der Server dem Klienten unter Nennung ihres *MIME*-Typen *Multi-Purpose Internet Mail Extension* ankündigt. Verfügt der Browser über die geeigneten Plugins, so ist er in der Lage, diese Inhalte zu visualisieren. In den folgenden Kapiteln sollen nun die wichtigsten Formate zur Darstellung multimedialer Inhalte vorgestellt werden.

GIF

Bei den auf WWW-Seiten verwendeten Grafikformaten nimmt GIF den Spitzenplatz in punkto Häufigkeit ein. Insbesondere bei künstlich erzeugten Bildern mit einheitlich gefärbten Farbflächen ist es an Kompaktheit nicht zu schlagen. Abbildung 2 zeigt eine nur 2K große Datei.



Abbildung 2: bauarbeiter.gif, Dateigröße: 2K

Zwei Kompressionsideen tragen zur Datenreduktion bei:

Farbpalette: Statt in jedem Pixel das komplette RGB-Tripel mit 3 Byte = 24 Bit Farbinformation zu speichern, werden für geeignetes p die 2^p wichtigsten Farben in einer Tabelle, genannt Farbpalette, gehalten und über einen p -Bit langen Index referiert. Für $p = 8$ schrumpft der Platzbedarf daher auf ein Drittel.

LZW: Das von *Lempel, Ziv* und *Welch* entwickelte und als Patent geschützte Verfahren zur Kompression beliebiger Zeichenfolgen basiert auf der Idee, in einer sogenannten Präfix-Tabelle die Anfangsstücke bereits gelesener Strings zu speichern und wiederholtes Auftauchen derselben Strings durch Verweise in die Tabelle zu kodieren.

Beide Ansätze zahlen sich insbesondere dann aus, wenn die Vorlage weite Bereiche mit identischer Information enthält, wie es bei computergenerierten Grafiken, wie z. B. Logos, der Fall ist. Zum einen enthält das Bild dann gar nicht die theoretisch verfügbare Zahl von $256^3 = 16$ Millionen Farben, sondern nur wenige Dutzend, und kann daher völlig verlustfrei durch eine Palette mit 256 Einträgen dargestellt werden. Zum anderen führen Folgen von identischen Pixelindizes zu kompakten Einträgen in der Präfixtabelle.

Bei der Umwandlung eines True-Color-Bildes in ein Palettenbild mit Hilfe eines Bildbearbeitungsprogramms gibt es zwei Ansätze: Standard oder individuell. Standard bedeutet die Verwendung einer mit 216 repräsentativ gewählten Farben vorbesetzten Farbtabelle (z. B. vom Netscape-Browser). Diese wird eventuell dem gegebenen Bild nicht optimal gerecht, aber die Auswirkungen bei der Präsentation sind vorhersehbar. Individuell bedeutet, eine für die jeweilige Bildvorlage maßgeschneiderte Farbpalette zu vorgegebener Farbzahl 2^p zu konstruieren. Hierzu werden im Median-Cut-Algorithmus die beobachteten Farbhäufigkeiten im RGB-Würfel angeordnet und dieser dann längs eines orthogonal verlaufenden Schnittes (Median) in zwei Subwürfel geteilt (Cut), mit etwa gleichmächtiger Pixelzahl. Diese Partitionierung wird solange fortgeführt, bis 2^p Würfel entstanden sind, die dann jeweils durch einen Index der Farbpalette repräsentiert werden.

Durch ein sogenanntes *Floyd-Steinberg-Dithering* kann die ohnehin kaum wahrnehmbare Differenz zum Originalbild weiter reduziert werden. Hierzu wird der beim Quantisieren längs einer Scanline verursachte Fehler (d. h. die Differenz zwischen wahren und approximiertem Farbwert) den nachfolgenden Pixeln vor ihrer Quantisierung zugeschlagen, so daß auf größere Flächen bezogen die Summe der konstruierten Farbwerte mit dem Ausgangsmaterial übereinstimmt. GIF ist streamingfähig; durch ein sogenanntes *Interlacing* wird zunächst nur jede 8. Zeile übertragen und danach jeweils die mittlere Zeile der verbleibenden Lücken.

Das GIF-Format erlaubt auch die Abspeicherung mehrerer Bilder in einer Datei, welche zudem mit Timing-Information versehen werden können (animated GIF). Hierdurch ist die Erstellung einfacher Animationen möglich, bei denen wenige kleine Grafiken in schneller Abfolge die Illusion eines Films verursachen.

JPG

Wie soeben erläutert, liegen die Stärken von GIF in der Wiedergabe von computergenerierten gleichfarbigen Flächen. Geht es dagegen um die Reproduktion einer fotorealistischen Vorlage, reichen 256 Tabelleneinträge oft nicht aus, sanfte Farbverläufe zu beschreiben. Hier kommt JPEG zu Hilfe, ein Verfahren benannt nach der Gruppe, die es entwickelt hat, der *Joint Photographic Expert Group*, Mitglieder der Standardisierungsgremien *CCITT* und *ISO*.

Zunächst wird das RGB-Bild in den YUV-Raum transformiert, d. h. die Farbinformation wird verlustfrei durch einen Helligkeitsanteil Y und zwei Farbdifferenzen U und V kodiert. Da das Auge für Helligkeitssprünge sensitiver ist als für Farbdifferenzen, kann man nun die Y -Matrix in der vollen Auflösung belassen und in den U , V -Matrizen jeweils 4 Pixel mitteln (4:1:1 Subsampling).

Für je 4 Originalpixel mit insgesamt 12 Bytes werden nun $4 + 1 + 1 = 6$ Bytes benötigt (pro Bildpunkt also 6 mal $8/4 = 12$ Bit). Die Reduktion beträgt 50%.

Nun werden die drei Matrizen in Blöcke mit 8×8 Abtastwerten aufgeteilt. Anschließend durchlaufen die Blöcke folgende Schritte:

Diskrete Cosinus Transformation: Hierdurch wird die 8×8 Ortsmatrix verlustfrei in eine 8×8 Frequenzmatrix umgewandelt. Die Helligkeitsinformation einer Fläche ist nun kodiert als Überlagerung von 64 zweidimensionalen Schwingungen.

Quantisierung: Die errechnete Matrix hat längs einer im Zickzack laufenden Scanline von links oben nach rechts unten Werte abnehmender Größe. Da die Werte rechts unten den hohen, für das menschliche Auge eher unwichtigen Frequenzen entsprechen, werden alle Einträge längs dieser Scanline durch Faktoren zunehmender Größe dividiert.

Komprimierung: Die Folge der quantisierten Koeffizienten wird einer Lauflängenkompromierung unterzogen und anschließend durch eine empirisch ermittelte Huffman-Tabelle kodiert.

Um aus dem komprimierten Bild das Original zu rekonstruieren, werden die Schritte in umgekehrter Reihenfolge und inverser Funktionalität durchlaufen. Abbildung 3 zeigt die Auswirkungen der drei Phasen angewendet auf eine 8×8 Grauwertmatrix.

Durch die Wahl der Rundungstabelle läßt sich der Tradeoff zwischen Qualität und Kompression beliebig steuern. Ein typisches Farbbild läßt sich auf etwa 5% seiner Originalgröße reduzieren, ohne daß ein menschlicher Betrachter die Detailfehler bemerken könnte. Bei 2% und weniger entstehen deutlich sichtbare Artefakte in Form von einfarbig gefärbten Klötzchen. Abbildung 4 vergleicht die Auswirkungen zweier Kompressionsgrade an einer Vorlage mit fein abgestuftem Helligkeitsverlauf.

DjVu

In den Forschungslaboratorien des amerikanischen Kommunikationsspezialisten AT&T wurde im Jahre 1998 ein neues Bildformat ausgebrütet, welches die Grafikindustrie revolutionieren könnte: DjVu, gesprochen *deschah wü*. Hiermit ist der aus dem französischen stammende Begriff *deja vu* (schon mal gesehen) gemeint, und

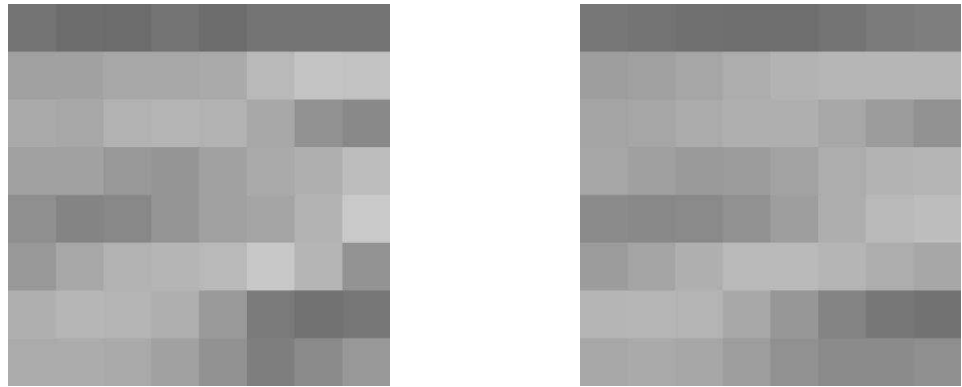


Abbildung 3: Transformation eines 8×8 Bildausschnittes: Ausgangsbildmatrix (links) und rekonstruierte Bildmatrix.



Abbildung 4: JPEG-Kompression eines 200×279 True-Color-Bildes, unkomprimierter Speicherbedarf: 163 KB. **links:** Dateigröße: 7,3 KB, Kompression 22:1. **rechts:** Dateigröße 2,5 KB, Kompression 65:1

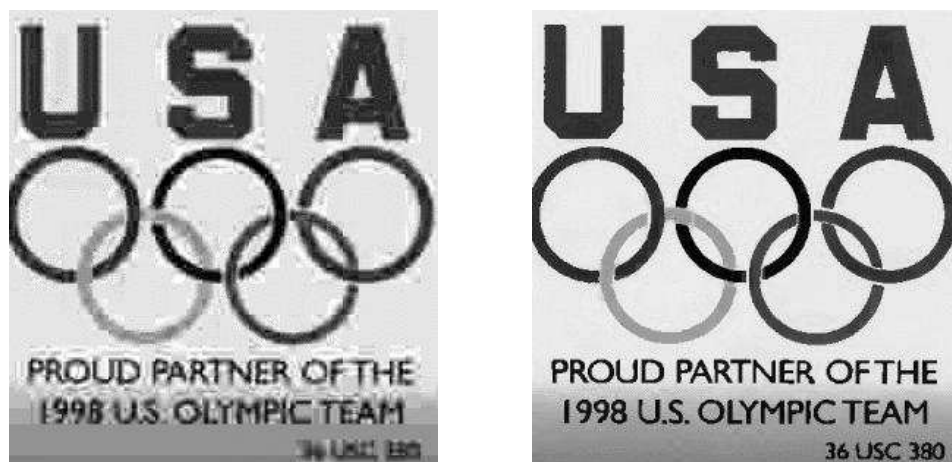


Abbildung 5: Vergleich JPEG versus DJVU anhand eines Posterscans auf 2460×3252 Pixel, Platzbedarf der unkomprimierten Datei: 22,8 MB. **links:** Ausschnitt aus JPEG-Bild 45 KB, Kompression 532 : 1. **rechts:** Ausschnitt aus DJVU-Bild 41 KB, Kompression 584:1.

dieser Name kennzeichnet das typische Einsatzgebiet dieses Formats: Abspeichern von bereits existierenden Postern, Büchern, Landkarten, etc. Die Überlegenheit von DjVu zeigt sich insbesondere bei Vorlagen, in denen Schrift und Zeichnungen auf farbigem Hintergrund platziert sind, wie dies bei historischen Dokumenten der Fall ist. Mit dieser Mischung von hochfrequenten Mustern innerhalb von weichen Farbverläufen hat JPEG immer besonders zu kämpfen, denn die diskrete Cosinus-Transformation kann scharfe Kontrastsprünge nur dann in erträglicher Qualität abbilden, wenn die Koeffizienten der hochfrequenten Anteile nicht zu sehr quantisiert wurden, was aber nur einen geringen Kompressionsgrad bedeutet.

DjVu löst dieses Problem durch eine Separierung der Vorlage in Hintergrund- und Vordergrundinformation sowie durch spezielle darauf angewandte Kompressionsverfahren. Für den Hintergrund mit seinen kontinuierlich und meistens unscharf verlaufenden Farben wird eine sogenannte *IW44 wavelet* Kompression eingesetzt, die ähnlich wie JPEG durch Überführung der Ortsmatritzen in den Frequenzraum versucht, die Bildinformation als Überlagerung weniger Grundschwingungen darzustellen. Die Kodierung der scharfkantigen und meist einfarbigen Vordergrundanteile geschieht separat davon. Insgesamt kommen drei patentierte Verfahren zum Einsatz: *JB2*, eine Zwei-Ebenen-Bildkompression, *ZP-coder*, ein adaptiver arithmetischer Entropie-Kodierer sowie die *IW44* Maskierungstechnik zum Einsparen von Bits auf dem Teil des Hintergrunds, der durch Text abgedeckt ist.

Abbildung 5 zeigt das Potential von DjVu im direkten Vergleich mit der JPEG-Technik anhand einer eingescannten Vorlage aus einem Versandhauskatalog. Deutlich zu sehen sind die scharfkantigen Schriftzüge, die JPEG bei diesen Kompressionsraten nicht mehr auflösen kann. Im Mittel sind folgende DjVu-Dateigrößen zu erwarten: Das Digitalisieren einer typischen farbigen DIN-A-4-Magazin-Seite mit 300 DPI (dots per inch) ergibt 25 MB unkomprimierte Daten und dagegen nur

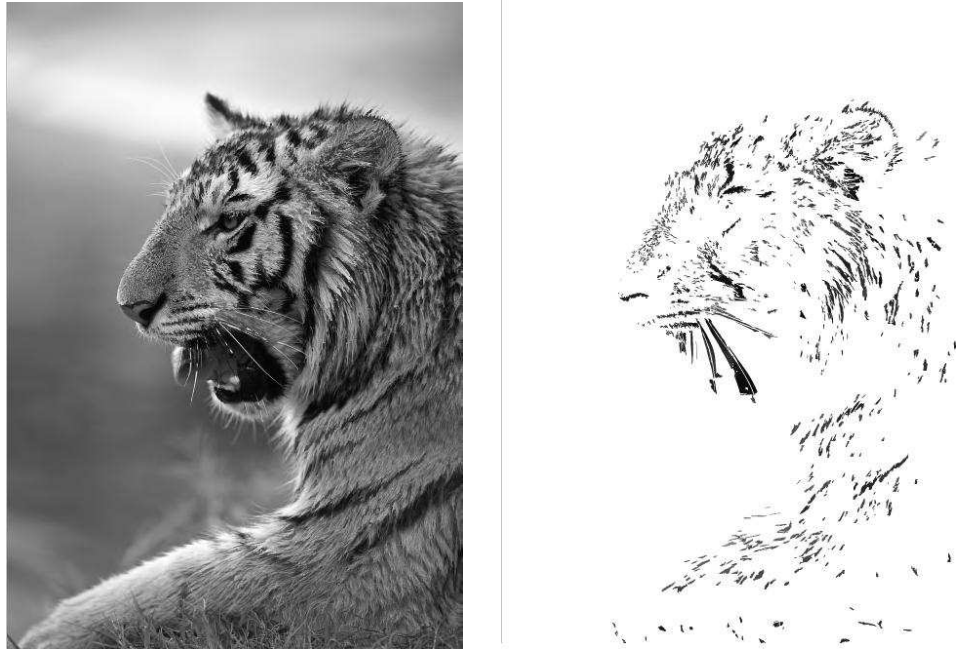


Abbildung 6: Strukturierte Bildinformation durch Ebenen unterschiedlicher Auflösung: Gesamtinformation des DjVu-Bildes (links) und Vordergrundanteil des DjVu-Bildes.

zwischen 40 und 70 KB DjVu-Daten. Dies ist etwa 5–8 mal besser als JPEG. Beim Digitalisieren einer schwarz/weißen Faxvorlage entstehen um etwa 3–8 mal kleinere Dateien als mit dem bekannten *CCITT-G4*-Algorithmus.

Abbildung 6 zeigt, daß es nicht nur auf Bildern mit aufgedrucktem Text gelingen kann, feine detailreiche Strukturen von einem eher groben Hintergrund zu trennen.

Bei Verwendung des Plugins ist der Benutzer in der Lage, Vordergrund und Hintergrund separat anzeigen zu lassen. Außerdem besteht die Möglichkeit eines interaktiven Zooms, da durch die AT&T-Technologie immer nur der in Realzeit errechnete relevante Ausschnitt für den Betrachter angezeigt wird.

Unter der Adresse <http://www.djvu.att.com/examples/dl.html> finden sich mehrere Beispiele für Bilder im DjVu-Format.

Fraktale Kompression

Viele in der Natur vorkommende Strukturen weisen eine starke Selbstähnlichkeit auf. Beispiele dafür sind Gebirgsformationen, Meeresküsten oder Blätter von Pflanzen. Solche in sich wiederholende Muster, die quasi beim Hereinzoomen immer wieder zutage treten, lassen sich ausnutzen, wenn man ein umfangreiches Gebilde durch ein kleines Erzeugendensystem darstellen möchte. Sogenannte *Iterierte Funktionensysteme* sind z. B. in der Lage, mit wenigen Regeln komplexe und natürlich



Abbildung 7: Farn mit Selbstähnlichkeit.

aussehende Bilder zu generieren. Abbildung 7 zeigt das Blatt eines Farns, welches durch die wiederholte Anwendung von vier affinen Abbildungen entstanden ist.

Nach dieser Vorlage ist die Idee der fraktalen Kompression entstanden. Ausgehend von einer gescannten Bildvorlage werden Regelmäßigkeiten und Selbstähnlichkeiten gesucht und dadurch der Bildinhalt als Fixpunkt einer iterierten Anwendung von Transformationsmatrizen definiert.

In stark vereinfachter Form läuft das Verfahren folgendermaßen ab: Zunächst wird das Bild in sogenannte *Domainblöcke* partitioniert. Zu jedem Domainblock wird nun ein *Rangblock* gesucht, dessen Abbild unter einer affinen Abbildung dem Inhalt des Domainblocks möglichst nahe kommt. Erforderlich für diesen Schritt ist natürlich die Wahl einer geeigneten Metrik, die den Abstand von zwei Pixelmengen definiert. Affine Abbildungen erlauben einfache geometrische Transformationen wie Drehung, Skalierung und Translation. Je nach zugestandener Zeitvorgabe kann der Suchalgorithmus mehr oder weniger viele Rangblockpositionen mit mehr oder weniger vielen affinen Abbildungen ausprobieren. Zur Vereinfachung beschränkt er sich auf eine feste Zahl möglicher Transformationsmatrizen sowie feste Rangblockgrößen, die an den Eckpunkten eines 2-dimensionalen Gitters fester Gittergröße positioniert sind.

Abbildung 8 zeigt, wie die Vorlage G zunächst mit 4 Domainblöcken überdeckt wird und dann der Inhalt jedes Domainblocks durch eine affine Transformation des zugehörigen Rangblocks dargestellt wird (in diesem Beispiel sind zufällig alle Rangblöcke identisch).

Bei der Dekompression eines fraktalen Bildes werden nun, beginnend mit einem zufälligen Bildinhalt, die Transformationsmatrizen solange angewendet, bis sich keine Änderung gegenüber der Vorgängeriteration mehr ergibt. Gewisse mathematische Randbedingungen, die bei der Wahl der Matrizen einzuhalten sind, garantieren, daß es einen Fixpunkt gibt, auf den das Verfahren konvergiert.

Das Verfahren sieht auf den ersten Blick wie eine typisch akademische Spie-

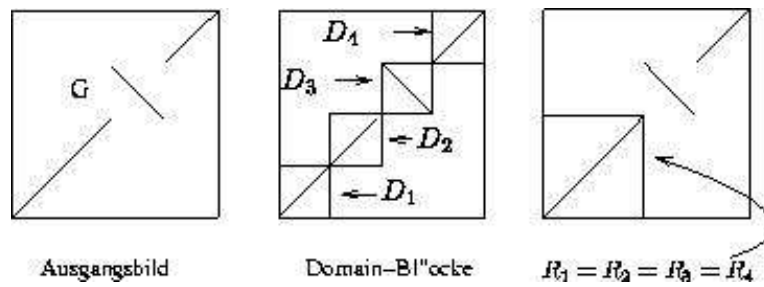


Abbildung 8: Beschreibung der Vorlage G durch 4 affine Transformationen.

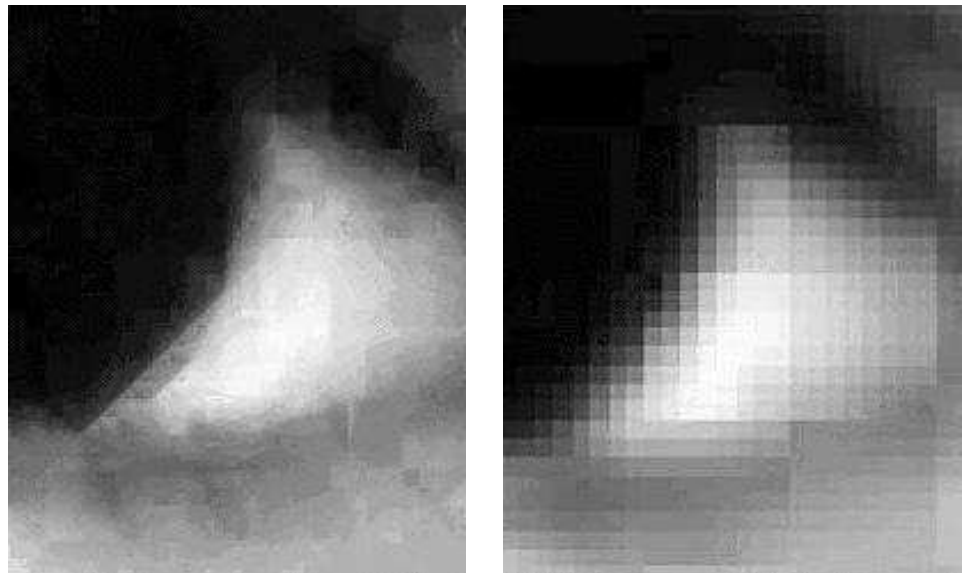


Abbildung 9: Vergleich Fraktale Kompression versus JPEG anhand eines vergrößerten Ausschnitts (Augenpartie) einer True-Color-Vorlage, eingescannt auf 696×924 Pixel. **links:** Fraktal-Bild, Kompressionsrate 72 : 1. **rechts:** JPEG-Bild, Kompressionsrate 71 : 1

lerei aus, bei der ein recht eleganter Algorithmus zwar (verblüffenderweise) eine gegebene Vorlage reproduzieren kann, aber ohne praktischen Nutzen scheint. Weit gefehlt! Die entscheidende Überlegenheit zu den konventionellen Bildkompressionsverfahren liegt in der Auflösungsunabhängigkeit. Nach dem Digitalisieren der Vorlage ist der Bildinhalt in eine Menge von Transformationsmatritzen gegossen. Diese können bei ihrer iterierten Anwendung Bilder jeder gewünschten Auflösung erzeugen, da sie jedes Bildstück nicht durch endlich viele Pixel beschreiben, sondern durch mathematische Beziehungen. Daher läßt sich aus demselben fraktalen Bild sowohl ein briefmarkengroßes Thumbnail als auch ein hochwertiges Poster generieren. Insbesondere kann man in ein fraktales Bild beliebig hereinzoomen, ohne daß die gefürchteten Klötzchen entstehen. Abbildung 9 zeigt, daß das generierte Bild qualitativ besser sein kann als die eingescannte Vorlage, weil die durch das Digitalisieren entstandenen Rasterpunkte im Fixpunkt der Transformationen weggemittelt werden.

Fraktal komprimierte Bilder haben sich in den Web-Browsern bisher nicht durchsetzen können. Zum einen, weil der Komprimierungsaufwand deutlich höher ist als bei GIF und JPEG. Zum andern kann der Vorteil der Auflösungsunabhängigkeit erst dann voll ausgespielt werden, wenn ein analoges Foto (hinter einer analogen Linse), ohne den Umweg über das digitale Abtasten direkt in die beschreibenden Transformationsmatritzen umgewandelt werden könnte.

FlashPix

Es wird noch einige Jahre dauern, bis die Zukunftsvisionen der fraktalen Kompression Wirklichkeit werden. Solange wollte die kalifornische Firma *LivePicture, Inc.* nicht warten. Ihre Lösung der Auflösungsunabhängigkeit heißt FlashPix und wurde mit den heute möglichen Mitteln auf völlig unspektakuläre Weise gelöst. Die Idee geht zurück auf eine gemeinsame Anstrengung der Firmen Kodak, Microsoft und Hewlett Packard bei der Suche nach einem neuen Bildformat und ist recht einfach: Das im Web-Browser zu präsentierende Bild wird in mehrfachen Versionen unterschiedlicher Auflösung auf dem Server gehalten und jeweils der vom Klienten angeforderte Teil in passender Größe übertragen. Typischer Einsatzbereich sind Online Shops, die ihr Warenangebot auf Web-Seiten in einer Weise präsentieren wollen, daß der Kunde ausgehend von einer Totalansicht des Objekts fast beliebig nah heranzoomen und somit alle Details inspizieren kann. Wegen der Verfügbarkeit der abzubildenden Waren für den Shop-Anbieter ist auch die Gelegenheit für die Erstellung hochauflösender Bilder durch die Verwendung einer professioneller Kameraausrüstung gegeben.

In der einfachsten Form erfordert dies beim Klienten keine weiteren Vorkehrungen, da der LivePicture Image Server den angeforderten Bildteil als JPEG-Bild liefert. Welcher Teil eines bereits im Browser geladenen Bildes in erhöhter Auflösung angefordert wird, kann durch die Koordinaten des Mausclicks bestimmt werden. Auf diese Weise führt der Betrachter einen Zoomvorgang aus, der ohne die übliche Rasterung abläuft. Abbildung 10 demonstriert diesen Effekt. Daß auf dem Server ggf. große Datenmengen vorgehalten werden müssen, kümmert den Klienten



Abbildung 10: Auswirkung der FlashPix-Technologie in einem Online-Shop (Gesamtansicht, JPEG-Ausschnitt, FlashPix-Ausschnitt)

wenig; er profitiert vielmehr von kurzen Kommunikationszeiten, da immer nur der Teil übertragen wird, den er benötigt.

Die komfortablere Version von FlashPix verwendet ein Plugin oder setzt einen Java-fähigen Browser voraus, der ein leichtgewichtiges Applet vorweggeschickt bekommt. Vorteil dieser beiden Ansätze ist eine höhere Funktionalität beim Betrachten der Bilder wie z. B. Vor- und Zurückzoomen sowie Verschieben des Betrachterausschnitts.

Neben der multiplen Auflösung, die ja bereits auf der Kodak Foto CD Verwendung fand, gehören folgende Features zum FlashPix-Format:

Kompression: Für jede Auflösungsstufe wird das Bild nach dem JPEG-Verfahren kodiert. Es besteht auch die Möglichkeit, ohne Kompression zu speichern.

Text: Textuelle Zusatzinformation kann gespeichert werden. Diese Daten sind strukturiert anhand der Schlüssel, wie z.B: Quelle, Urheber, Inhalt, Kameraeinstellungen, Filmcharakteristika, Scancharakteristika.

Farbräume: Neben unkalibrierten RGB-Werten gibt es monochrom und Photo YCC

Transformation: FlashPix-Bilder halten neben den Rohdaten einige wenige Filter- und Transformationsmatrizen vor, die jeweils nur auf den zur Zeit sichtbaren Teil angewendet werden müssen. Beispiele sind Translation, Rotation, Skalierung, Schärfe, Kontrast, Helligkeit und Farbbalance. Hierdurch reduziert sich der Hauptspeicherbedarf eines Bildverarbeitungsprogramms und gleichzeitig erhöht sich seine Bearbeitungsgeschwindigkeit, da die Masse der Bilddaten unbearbeitet auf der Platte verbleiben können.

LivePicture eröffnet mit seinem FlashPix-Plugin auch die Möglichkeit zum interaktiven Betrachten von 360-Grad Panoramen. Bei der Herstellung solcher Bilder ist der Ausgangspunkt eine Folge von Einzelschnappschüssen, die mit einer konventionellen Kamera im Kreis erstellt wurden. Nach dem Digitalisieren werden mit dem Werkzeug *LivePicture PhotoVista* die Aufnahmen, unter automatischer



Abbildung 11: Vorlage für 360-Grad-Projektion (Kreißaal des Marienhospitals Osnabrück)

Berücksichtigung der überlappenden Bereiche, zu einem länglichen Schlauch im JPEG-Format zusammengeschweißt. Abbildung 11 zeigt die aus einer Sequenz von 12 Einzelbildern generierte JPEG-Datei. Vorteil der Verwendung dieses gängigen Bildformats ist natürlich die Möglichkeit, mit jeder Bildverarbeitungssoftware weitere Manipulationen an dem Rundblick durchführen zu können.

PhotoVista erzeugt neben dem JPEG-Bild noch eine gleichnamige Datei mit der Endung *ivr*. Die dort gespeicherte Information benutzt das Plugin zur perspektivisch richtigen Umrechnung des JPEG-Bildes in ein interaktiv drehbares Panorama. Als multimediale Zugabe können der Datei noch Klanginformation und Animationen beigefügt werden.

Unter der Adresse

<http://www.livepicture.com/zr/solutions/advanced/content.html>

finden sich mehrere Beispiele für Bilder im FlashPix-Format.

Flash

Alle bisher vorgestellten Formate bauen auf pixelorientierten Vorlagen auf. Das bedeutet, die dargestellten Bilder sind ursprünglich durch Einscannen einer Vorlage mit fest vorgegebener Auflösung entstanden und das Ziel von kleinen Dateigrößen wurde durch trickreiche Verfahren erreicht, die Pixelinformationen reduzieren, ohne den visuellen Eindruck wesentlich zu verändern.

Eine völlig andere Ausgangslage ergibt sich, wenn die zu präsentierenden Bilder computergenerierte Daten darstellen, da nun eine kompakte Repräsentation im Rechner gewählt werden kann, aus der das externe Erscheinungsbild erst errechnet wird. Für geometrisch strukturierte Vorgaben bietet sich das Vektorformat an, da hier mit einer Handvoll von Koordinatenangaben die Platzierung von elementaren Objekten wie Linien, Rechtecken und Kreisen möglich ist. Auch Kurven sind durch Bezier- und Splines-Techniken über wenige Parameter zu spezifizieren. Als Resultat kann sich der Betrachter in eine Vektorgrafik beliebig hineinzoomen, ohne daß Rastereffekte auftreten.

Die Firma *Macromedia* hat im Jahre 1996 das Format *Flash* auf den Markt gebracht und zur Unterstützung einer schnellen Verbreitung kurz darauf auch veröffentlicht. Damit sind Entwickler von Grafikprogrammen in der Lage, in ihre Applikationen das Format zu integrieren. Flash wird im Bundle mit Netscape Communicator und Microsoft Internet Explorer ausgeliefert und pro Tag über 100.000 mal per Download von der Macromedia Homepage geholt.

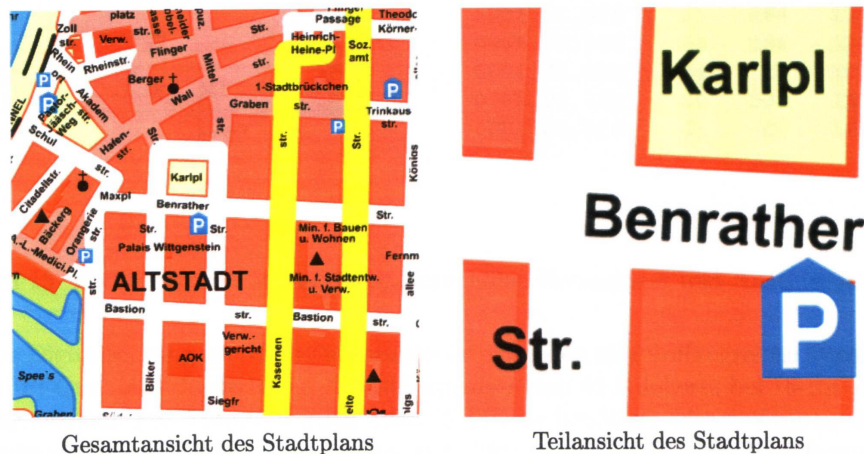


Abbildung 12: Rasterfreies Zoomen bei Dateien im Flash-Format durch Vektorgrafik

Abbildung 12 zeigt eine sinnvolle Anwendung des Flash-Formats, gezogen von der Homepage <http://www.stadtplan.net> des Verwaltungsverlages München. Ein Stadtplan ist sowohl in der Totalen als auch bis auf Straßengröße runtergebrochen vom selben Datensatz präsentierbar. Der an diesem Beispiel beobachtete Skalierungsunterschied von etwa 1:6 könnte durch ein pixelorientiertes Programm nur unter Bereitstellung riesiger Dateien bewerkstelligt werden.

MIDI

Durch die Digitalisierung von Klanginformationen entstehen sehr umfangreiche Dateien, wenn die für HiFi-Qualität erforderlichen Abtastfrequenzen von 44 Kilohertz bei 16-Bit Auflösung verwendet werden: Pro Minute werden etwa 10 MByte benötigt. Dieser Vorgang entspricht dem Einscannen einer Bildvorlage. Analog zur Vektorgraphik läßt sich die Klanginformation wesentlich kompakter speichern, wenn die Soundinformation nicht gesampelt werden muß, sondern im Computer entstehen kann. Hierzu ist eine Notation erforderlich, welche im wesentlichen den Verlauf jeder einzelnen Melodiestimme durch eine Sequenz von Ereignissen kodiert. Abbildung 13 zeigt beispielhaft, wie aus einer konventionellen Notenschrift eine Befehlsfolge entstanden ist, welche für jeden Schlag eines jeden Taktes unter Angabe des Zeitpunktes in Millisekunden die Notendauer, die Notenhöhe und die Anschlagstärke angibt.

Die zu diesem Zweck entworfene Sprache MIDI (*Musical Instrument Digital Interface*) entstand 1982 als Ergebnis einer Kooperation der Gerätehersteller Sequential Circuits, Roland, Yamaha, Kory und Kawai und war ursprünglich konzipiert als Kommunikationssystem zur Ansteuerung elektronischer Musikinstrumente. Mit der Verbreitung von Personal-Computern kamen weitere Komponenten hinzu (1987 *Midi Time Code*, 1988 *Midi Files*). Inzwischen ist *General Midi* ein weitverbreite-



The image shows two musical staves on the left and a table of MIDI data on the right. The top staff starts at measure 8 and the bottom at measure 12. The table lists MIDI events with their IDs, durations, and pitch classes.

0001.01.019	145	A#1	127
0001.01.019	92	D4	127
0001.01.022	48	A#4	99
0001.01.022	89	D4	127
0001.01.023	48	A#3	127
0001.01.025	45	A#4	127
0001.01.208	63	D#4	127
0001.01.211	64	C5	101
0001.01.211	60	D#4	127
0001.01.214	61	C4	127

Abbildung 13: Kodierung von Musik durch Notenhöhe, Notendauer und Anschlagstärke

ter Standard zur Spezifikation, Manipulation, Übertragung und Speicherung von elektronisch erzeugter Musik. In Kombination mit einem PC reicht ein sogenanntes *Masterkeyboard*, welches über ein 5-poliges Kabel die Beschreibung des Musikstückes an die Soundkarte schickt, wo sie dann von einem Synthesizer, ggf. mit Unterstützung von digital abgelegten Klangsamples, in Töne umgesetzt wird.

Midinachrichten sind Byte-orientiert und werden über eine serielle Schnittstelle mit 31250 Baud übertragen.

Es wird unterschieden zwischen Status-Bytes und Datenbytes.

Status-Bytes beginnen mit einer 1, und enthalten vier Kanalbits, mit denen 16 angeschlossene Instrumente assoziiert werden sowie drei Befehlsbits, mit denen die in Tabelle 4 gezeigten acht Aktionen angekündigt werden können.

000	Note off	schaltet Ton aus
001	Note on	schaltet Ton an
010	Polyphonic Aftertouch	Anschlagdruck für eine Taste
011	Control Change	Realzeitmeldung, z. B. Lautstärke
100	Program Change	Instrument/Klangfarbe
101	Channel Aftertouch	Anschlagdruck für alle Tasten
110	Pitch Bend	Auslenkung Tonhöhenrad
111	Systembefehl	z. B. zur Synchronisation

Tabelle 4: Befehle im Statusbyte

Einem Statusbyte folgen ggf. mehrere Datenbytes. Z. B. gehören zum Note-On-Befehl zwei Parameter: Notennummer und Anschlaggeschwindigkeit. Mit 128 Werten können mehr als 10 Oktaven abgedeckt werden, eine Klaviertastatur mit 88 Tasten von Subkontra A bis zum fünfgestrichenen C belegt das Intervall 33 bis 120. Zum Note-Off-Befehl gehören wiederum Notennummer und Loslaßgeschwindigkeit.

Um den zeitlichen Aspekt beim Abspielen einer Midi-Datei zu rekonstruieren, wird die Zeitachse zu sogenannten Tics diskretisiert. Z. B. beträgt beim Sequenzer Cubasis die zeitliche Auflösung 384 Tics pro Viertelnote. Beim Abspeichern einer Notensequenz werden die Intervalle zwischen den Events als sogenannte Delta Times abgespeichert. Hierzu werden bis zu 4 Bytes verwendet.



Abbildung 14: Streaming Video vom RealServer gabriel.informatik.uni-osnabrueck.de

RealVideo

Noch stärker als bei Audio-Files wachsen die Dateigrößen bei der Videodigitalisierung. In unkomprimierter Form erzeugen 25 PAL-Fernsehbilder pro Sekunde im Format 768×576 Pixel bei 24 Bit Farbtiefe einen Datenstrom von 32 MBytes/sec. Durch eine Vereinfachung des Videosignals (Subsampling) sowie durch Ausnutzung von räumlicher Redundanz (JPEG-Komprimierung) und zeitlicher Redundanz (MPEG-Komprimierung) entstehen Datenraten, wie sie von einem CD-ROM-Laufwerk mit einfacher Geschwindigkeit geliefert werden können.

Für eine Übertragung im Internet ist dies jedoch immer noch ungeeignet, da der Benutzer bei Anforderung eines Videos zu lange warten müsste, bis dieses komplett im Web-Browser eingetroffen ist. Aus dieser Überlegung heraus wurde das Streaming-Konzept geboren, welches zwischen dem sendenden Media-Server und dem empfangenden Web-Klienten ein Protokoll vorsieht, das den Beginn des Abspielens erlaubt, bevor alle Daten eingetroffen sind.

Marktführer in diesem Segment ist die Firma *RealNetworks*, die für ihr *RealVideo*-Format eine breite Palette an Werkzeugen zum Erstellen, Versenden und Abspielen anbietet. Abbildung 14 zeigt den Screenshot von einem Web-Browser mit *RealVideo*-Plugin, der gerade ein *Video on Demand* abspielt, welches vom einem *RealVideo*-Server auf einem Windows NT Rechner serviert wird.

Der *RealServer* kann auch zur Ausstrahlung von Musik- und Sprachsendungen verwendet werden. Für eine passable Wiedergabequalität reichen dabei Bandbreiten von 40 KBit bzw. 16 KBit aus.

VRML

VRML, sprich Wörmel, ist eine für das WWW entworfene Virtual Reality Modelling Language zur Beschreibung von 3-dimensionalen Szenen mit multimedialen Komponenten und Animation. Analog zum 2D-Vektorformat Flash gelangen Szenen nicht durch Einscannen in den Rechner, sondern durch eine geometrische Beschreibung aller räumlichen 3D-Koordinaten. Die gerenderte Projektion der Szene kann von jedem Web-Browser geliefert werden, der über ein passendes Plugin verfügt.

VRML-Szenen werden beschrieben in ASCII-Dateien mit der Endung *.wrl, welche innerhalb einer HTML-Seite mit dem EMBED-Kommando referiert werden, z. B. <EMBED SRC=quader.wrl WIDTH=400 HEIGHT=300>

Ein entsprechend konfigurierter Web-Server schickt dem anfordernden Klienten als Vorspann dieser Daten den Mime-Typ VRML, worauf das zur Betrachtung installierte Plugin, z. B. Cosmo Player 2.0 von Silicon Graphics, die eingehenden Daten in eine interne Datenstruktur einliest, von wo sie zur Berechnung einer fotorealistischen Projektion verwendet werden. In welcher Weise Blickwinkel und Orientierung in der Szene modifiziert werden können, bleibt dem Plugin überlassen: Mit Mauszeiger und Keyboard Shortcuts wandert der Benutzer durch eine virtuelle Welt, verkörpert im wahrsten Sinne des Wortes durch einen Avatar, seiner geometrischen Repräsentation, beschränkt in seiner Beweglichkeit durch physikalische Restriktionen und durch eine simulierte Schwerkraft.

Wichtigster Bestandteil von VRML-Szenen ist der sogenannte Knoten, der wie ein Programmiersprachenrecord aus Feldern verschiedenen Typs besteht. Diese Felder verweisen entweder auf nicht weiter strukturierte Objektknoten oder andere Gruppenknoten, die wiederum mittels ihrer Felder verzweigen können.

```
#VRML V2.0 utf8
# quader.wrl
Transform {
  translation 0 0 0
  children [
    Shape {
      geometry Box {
        size 2 2 2
      }
      appearance Appearance {
        material Material {
          diffuseColor 1 0 0
          shininess 0.9
        }
      }
    }
  ]
}
```

Abbildung 15: VRML-Datei für einen roten Quader

Das Listing in Abbildung 15 zeigt den Aufbau einer Szene, in der ein Quader mit Kantenlänge 2 im Ursprung des Weltkoordinatensystems plaziert wird. Die x -Richtung entspricht der horizontalen Bewegung, y beschreibt die vertikale Richtung und z wächst auf den Betrachter zu. Der Box-Knoten hat dabei als einziges (optionales) Feld die Kantenlänge. Dieser Quader wird referiert über das geometry-Feld des Shape-Knotens, zuständig für die Gestaltung eines Objekts. über das appearance-Feld wird die Materialbeschaffenheit in Form einer RGB-Farbe und eines Transparenz-Koeffizienten spezifiziert. Der Shape-Knoten wiederum ist als eins der Kinder im Transform-Knoten eingetragen, der über ein translation-Feld für die Verschiebung des Quaders sorgt.

Die daraus gerenderte perspektivische Würfeldarstellung zeigt Abbildung 16. Ein Screenshot einer kompletten Gebäudemodellierung bietet Abbildung 17. Hier wurden in mehreren verzeigerten VRML-Dateien alle Büros des Fachbereich Mathematik/Informatik der Universität Osnabrück kodiert. Die Einstiegsseite für diesen virtuellen Fachbereich lautet:

<http://www-lehre.informatik.uni-osnabrueck.de/~blaxxun/BxxWorlds/avz/html>.

Zusammenfassung

Eins ist sicherlich deutlich geworden: die ideale multimediale Komponente für das Internet gibt es nicht. Dazu sind die Anforderungsprofile der Anwender zu verschiedenen. GIF hat seine Berechtigung für kleine, animierte Logos mit wenigen, klaren Farben. JPEG bietet sich an für Landschafts- und Personenfotos mit wenig Details. Die auflösungsunabhängige Fraktale Kompression ist sicherlich ihrer Zeit noch ein wenig voraus. FlashPix ist eine praktikable Lösung für Online Shops, die auch ihren Kunden mit Browsern der älteren Generation die Illusion des Zooming in Warendetails bieten wollen. DjVu stellt sicherlich einen großen Durchbruch dar insbesondere für die Kompression von Bild/Schrift-Mixturen. Das Vektorformat Flash kann in sehr kompakter Weise geometrische Strukturen kodieren, die sowohl in der Totalen als auch im Detail betrachtet werden sollen. Midi ersetzt gesampelte Klänge durch künstliche Sounds. RealVideo eignet sich für verzögerungsfreie Übertragung von Videoclips von geringer Auflösung und Qualität. VRML schließlich erlaubt die interaktive Navigation durch 3-dimensionale Szenen.

Für welche Variante der Anbieter sich auch entscheidet: Ein Mehrwert an Informationsgehalt gegenüber rein textbasierten Daten ist es allemal. Es bleibt zu hoffen, daß die entwickelten Formate vom Anwender im richtigen Maße und wohlndosiert zur Visualisierung seines Anliegen verwendet werden.

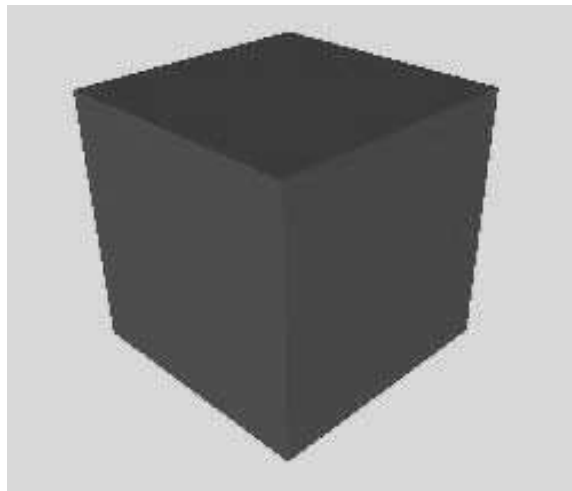


Abbildung 16: Screenshot vom VRML-Würfel



Abbildung 17: Universitätsgebäude am Westerberg, modelliert mit VRML