

REIHE INFORMATIK

6/92

Erweiterung von X für digitale Filme

R. Keller, B. Lamparter und W. Effelsberg

Universität Mannheim

Seminargebäude A5

6800 Mannheim

Erweiterung von X für digitale Filme

Ralf Keller, Bernd Lamparter und Wolfgang Effelsberg
Lehrstuhl für Praktische Informatik IV
Fakultät für Mathematik und Informatik
Universität Mannheim

Kurzfassung

In dieser Arbeit wird eine Erweiterung von X11 für die Bewegbilddarstellung in einer verteilten Umgebung vorgestellt. Sie ermöglicht den benutzerkontrollierten Ablauf von Filmen in Fenstern, die von X zur Verfügung gestellt werden. Nach einer Einführung in die Problematik der Bewegbilddarstellung und -übertragung wird die Erweiterung des X-Window-Systems detailliert beschrieben. Der Schwerpunkt liegt dabei auf der Funktionalität der Benutzerschnittstelle und der Verarbeitung von Filmaufträgen im X-Server. Außerdem werden verschiedene Leistungsmessungen präsentiert, die zur Engpaßanalyse des Systems durchgeführt wurden.

1. Einleitung

In den letzten Jahren hat sich die Erkenntnis verbreitet, daß neben zeitunabhängigen Medien wie Text und Grafik auch zeitabhängige Medien wie Audio und Video in Computersysteme integriert werden können. Dazu wurden und werden neue und innovative Anwendungsszenarien im Bereich multimedialer Systeme entwickelt. Beispiele sind: Gruppenarbeit (CSCW) [CWB91], Lernsysteme [Hua91] und Verlagswesen [PN91].

Allerdings stellt die Verarbeitung multimedialer Daten neue Anforderungen an die Rechner- und Übertragungssysteme [DN91]. Sowohl Isochronität der Datenströme als auch ein hoher Durchsatz müssen bis zur Anwendungsebene sichergestellt sein. Hier ist die Forschung vor allem in den Bereichen Speicherung und Zugriff [SNM+91], Synchronisation [LGC+91], Übertragungsprotokolle [DDK+90] und Durchsatzverbesserung in den oberen Protokollschichten [HEH+92] aktiv.

Das X-Window-System (X, [Jon89]) ist ein mächtiges, netztransparentes, portables Fenstersystem für pixelorientierte Graphikbildschirme. Es läuft auf einer Vielzahl von Systemen und wurde zu einem de-facto Standard aufgrund seiner Herstellerunabhängigkeit, seiner Portabilität, seiner Vielseitigkeit im Umgang mit Farben und seiner Fähigkeit, transparent in einem Netz zu arbeiten. Es ist ein exzellentes Werkzeug für die Forschung, da der Quellcode frei erhältlich ist.

Daher liegt es nahe, eine Integration von X und zeitabhängigen Medien auf leistungsfähigen Arbeitsplatzrechnern anzustreben. Ein erster Schritt in diese Richtung ist die digitale Bewegbilddarstellung in einem Fenster. Dem Benutzer wird es ermöglicht, in seiner gewohnten Rechnerumgebung Filme zu betrachten. Dazu ist es wünschenswert, auf spezielle Hardware zur Speicherung, Übertragung und Darstellung von Bewegbildern zu verzichten, um die leichte Portierbarkeit von X zu erhalten.

Ein Gegenbeispiel für diesen Ansatz ist das DVI-System [Lut89], das spezielle Hardware zur Darstellung und zur Kompression der Audio- und Videodaten benötigt.

Dieser Artikel ist wie folgt strukturiert: In Kapitel 2 werden die Darstellungsformate digitaler Bilder und Bildsequenzen und die Anforderungen der Bewegtbildübertragung an das Netz erläutert. Auf der Basis dieser Anforderungen werden dazu in Kapitel 3 die notwendigen Erweiterungen des X-Window-Systems beschrieben. Auf einige Einzelheiten der Implementierung wird in Kapitel 4 genauer eingegangen. Über Erfahrungen, die bei der Entwicklung und bei verschiedenen Tests gesammelt wurden, wird in Kapitel 5 berichtet. Eine Zusammenfassung und ein Ausblick bilden den Abschluß dieser Arbeit.

2. Digitale Bewegtbildübertragung

2.1 Digitale Bilder und Filme

2.1.1 Formate digitaler Einzelbilder

Moderne Arbeitsplatzrechner können z.B. 1280×1024 Pixel große Bilder darstellen. Dafür werden bei 24 Bit pro Pixel (Echtfarben, [Lut89]) ca. 3,9 MByte pro Bild benötigt. Bei solchen Datenmengen ist Kompression erforderlich, um die Speichermedien und die Übertragungswege zu entlasten. Beide sind knapp und damit teuer.

Um den Speicherverbrauch zu begrenzen, verwenden moderne Arbeitsplatzrechner meist Farbtafeln mit 256 möglichen Einträgen. Dabei wird für jedes Pixel nur ein Byte benötigt, welches einen Index auf eine Farbzelle in der Farbtafel enthält (siehe Abb. 1). Jede dieser Farbzellen beinhaltet die eigentlich darzustellende Farbe. Damit können zwar pro Bild nur 256 verschiedene Farben gleichzeitig verwendet werden, die aber aus einer Palette von mehr als 16 Millionen ausgewählt werden können. Durch die Farbtabellentechnik wird der Speicherbedarf pro Bild auf ein Drittel gesenkt. Zu jedem Bild muß aber zusätzlich die Farbtafel mit abgespeichert werden. Treten in einem Originalbild mehr als 256 verschiedene Farben auf, so entsteht in der digitalen Darstellung ein Qualitätsverlust. Dies ist jedoch für viele Anwendungen akzeptierbar.

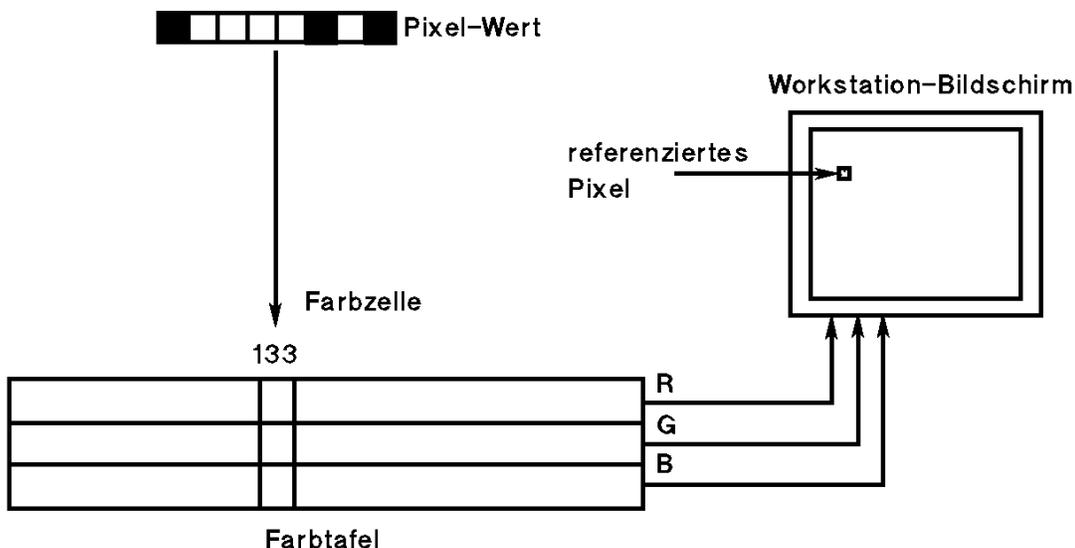


Abb. 1: Farbtafel mit 256 Einträgen

2.1.2 Bildkompression

Aufgrund des hohen Speicherverbrauchs und der Durchsatzanforderungen bei der Übertragung der Bilddateien müssen diese komprimiert werden. Kompression ist immer dann möglich, wenn Redundanzen vorhanden sind oder wenn ein Informationsverlust akzeptabel ist. Kompressionsalgorithmen lassen sich nach zwei Kriterien einteilen. Zum einen werden verlustfreie und verlustbehaftete Verfahren unterschieden. Bei der verlustfreien Kompression werden die Bildinformationen so verdichtet, daß das komprimierte und anschließend dekomprimierte Bild mit dem Originalbild identisch ist. Bei der verlustbehafteten Kompression werden kleinere Abweichungen hingenommen.

Ein anderes Unterscheidungskriterium ist das Zeitverhalten. Während bei symmetrischen Verfahren für Kompression und Dekompression ungefähr derselbe Zeitaufwand benötigt wird, benötigen asymmetrische Verfahren für die Dekompression viel weniger Zeit als für die Kompression. Damit wird erreicht, daß ein großer Teil des Gesamtaufwands schon bei der Kompression entsteht, um die Dekompression möglichst in Echtzeit ausführen zu können. Dadurch kann ein einmal komprimiertes Bild abgespeichert und mit einem geringen Rechenaufwand abgerufen werden. Durch das asymmetrische Zeitverhalten wird somit der Gesamtaufwand bei mehrfachem Abruf minimiert.

Eine einfache, verlustbehaftete Kompression stellt die oben beschriebene Bilddarstellung mit einer Farbtabelle dar. Andere Verfahren, mit denen die Bilder noch stärker komprimiert werden können, erfordern auch einen größeren Aufwand an Rechenleistung zur Dekompression, der in der Regel zur Laufzeit der Bilddarstellung erbracht werden muß.

Ein von der Softwareseite her interessantes Verfahren wird in [GM91] vorgeschlagen. Dabei wird ein Bild in Blöcke der Größe 16×16 Pixel aufgeteilt. Für jeden Block werden die Varianz und der Mittelwert berechnet. Falls die Varianz kleiner als ein bestimmter Schwellwert ist, wird der gesamte Block mit seinem Mittelwert kodiert. Andernfalls wird der Block in vier rechteckige Unterblöcke aufgeteilt und das Verfahren wird wiederholt. Ein Block von 2×2 Pixel wird nicht weiter unterteilt. Diese Methode erfordert keine Gleitkommaberechnungen, und da der Algorithmus mit Blöcken arbeitet, lassen sich Kompression und Dekompression gut parallelisieren. Das Verfahren ist verlustbehaftet und symmetrisch, und der Kompressionsfaktor läßt sich durch den Schwellwert einstellen.

Zur Kompression von Einzelbildern wird zur Zeit JPEG (*Joint Photographic Experts Group*) von der ISO genormt [Wal91]. Es ist ein verlustbehaftetes und symmetrisches Verfahren. Mittels einer vorangestellten Cosinustransformation werden sehr gute Kompressionsraten erzielt. Der Aufwand an Rechenzeit ist allerdings beachtlich. Daher wurden für JPEG bereits spezielle Chips entwickelt (z.B. von C-Cube Microsystems).

2.1.3 Formate digitaler Filme

Bei der Speicherung von digitalen Filmen bieten sich prinzipiell dieselben Möglichkeiten wie bei Einzelbildern, d.h. entweder mit 24 Bit/Pixel in Echtfarben oder mit 8 Bit/Pixel als Index auf eine Farbtafel. Dabei entsteht in erheblich verschärfter Form das Problem der großen Datenmenge. Die Verwendung einer Farbtafel für jedes Bild stellt zwar schon eine beträchtliche Einsparung dar, doch kann durch die Anwendung geeigneter Kompressionsverfahren das Datenvolumen weiter vermindert werden.

2.1.4 Kompressionsverfahren für Filme

Viele Kompressionsverfahren für Filme beruhen auf dem Prinzip, zuerst die Differenz zweier aufeinanderfolgender Bilder zu berechnen und dann statt dem Originalbild nur die Differenz zum Vorgänger zu komprimieren. Dieser Ansatz ermöglicht eine effiziente, verlustfreie

Kompression bei Filmen, bei denen sich von Bild zu Bild nur wenig ändert. Oft werden auch zweidimensionale Verfahren, die für Standbilder entwickelt wurden, für Bildfolgen erweitert, wobei als dritte Achse die Zeitachse hinzukommt.

Ergänzend zu JPEG normiert die ISO ein Verfahren zur Kompression von Bewegtbildern: MPEG (*Moving Picture Experts Group*, [Gal91]). Bei diesem verlustbehafteten und symmetrischen Verfahren wird wie bei JPEG eine Cosinustransformation vorangestellt, um die Kompressionsrate zu verbessern. Der Algorithmus von MPEG versucht, einen Ausgleich zwischen zwei gegensätzlichen Anforderungen zu schaffen: Einerseits soll der Film möglichst stark komprimiert werden; dies läßt sich mit Einzelbildkompression allein nicht erreichen, da dadurch die Redundanzen im zeitlichen Ablauf nicht ausgenutzt werden (z.B. bei gleichbleibendem Hintergrund). Andererseits soll ein wahlfreier Zugriff ermöglicht werden, so daß ein Filmlauf an beliebiger Stelle gestartet werden kann; dies wird am besten durch die Kompression von Einzelbildern erreicht.

2.1.5 Filme und Farbtabelle

Werden die Einzelbilder eines Films mit einer Farbtabelle kodiert, so zeigt sich bei der Übertragung folgendes Problem: Besitzt jedes Bild eine eigene Farbtabelle, so muß diese vor den Bilddaten geladen werden. Das zuvor angezeigte Bild wird dadurch kurzfristig in Falschfarben sichtbar. Selbst wenn die beiden Bilder ähnliche Farben besitzen, bleibt dieser Effekt für den Betrachter unangenehm, da sich die Indizes der Farben unterscheiden können.

Eine Lösung dieses Problems besteht darin, alle Bilder eines Films auf eine einzige Farbtabelle umzurechnen. Dann können allerdings im ganzen Film nicht mehr als 256 Farben verwendet werden. Die Qualität leidet dadurch erheblich, aber dafür muß die Farbtabelle nur einmal übertragen und geladen werden, und es treten keine Falschfarben mehr auf.

Um einem Film beiden Anforderungen gerecht zu werden, d.h. dem Film mehr als 256 Farben zur Verfügung zu stellen und dabei dennoch Fehlfarben zu vermeiden, wird in [LEM92] eine dynamische Anpassung der Farbtabelle vorgeschlagen: Zur Vermeidung von Falschfarben bleibt jeweils ein kleiner Teil der Farbtabelle unbenutzt; es werden z.B. 32 Indizes freigelassen. Die Farben für diese 32 Indizes können dann vor dem Laden des nächsten Bildes eingetragen werden, ohne das aktuelle Bild zu verfälschen. Im nächsten Bild werden diese Indizes benutzt und dafür andere Plätze für das übernächste Bild freigelassen usw. Auf diese Weise paßt sich die Farbtabelle während des Filmlaufs dynamisch an den Farbinhalt der Filmszenen an.

2.2 Übertragungsprotokolle für Bilder und Filme

Die Übertragung von Filmen muß so gestaltet werden, daß ein isochroner Datenfluß und ein hoher Durchsatz bis zur Anwendungsebene erreicht wird. Dazu muß einerseits das verwendete Transportprotokoll sowohl eine hohe Bandbreite als auch eine konstante Verzögerung garantieren. Andererseits müssen auch die höheren Schichten diese Dienstgüte sicherstellen, die in heutigen Systemen einen erheblichen Engpaß darstellen [CT90].

Obwohl die heutigen Arbeitsplatzrechner oft mit Mehrbenutzerbetriebssystemen wie UNIX ausgestattet sind, werden sie meistens nur von einer Person benutzt. Jedoch läßt ein Benutzer meistens mehrere Prozesse gleichzeitig auf seinem Rechner laufen. Bisherige UNIX-Scheduler nehmen aber keine Rücksicht auf Prozesse mit Echtzeitanforderungen und konstanter Verzögerung [ADH91].

Schon beim Netzzugangsprotokoll zeigt sich, daß das Problem bei der Übertragung von kontinuierlichen Multimedia-Datenströmen noch einmal schwieriger wird. Das in lokalen Netzen weit verbreitete CSMA/CD-Protokoll kann weder eine konstante Verzögerung noch

eine konstante Datenrate garantieren, da sowohl beim *Carrier Sensing* als auch bei der Kollisionsbehandlung unterschiedliche Wartezeiten für aufeinanderfolgende Pakete entstehen können. Die Zugangsprotokolle in Tokenring-Netzen und im Hochgeschwindigkeitsnetz FDDI [RHF91] garantieren immerhin eine maximale Verzögerung. Innerhalb bestimmter Grenzen kann mit diesen Protokollen ein isochroner Dienst durch Puffern verfrühter Datenpakete simuliert werden [Cou91]¹.

Doch auch die heutigen Transportprotokolle sind für isochrone Datenübertragung ungeeignet. Sowohl TCP [Com88] als auch ISO TP4 [Ros90] benutzen eine Fehlerkorrektur durch Zeitschranken und Wiederholung. Dadurch kann natürlich nicht garantiert werden, daß alle Pakete mit einer konstanten Verzögerung übertragen werden. Eine isochrone Datenübertragung wird erst mit dem zukünftigen DQDB-Protokoll oder dem kanalvermittelten Breitband-ISDN auf der Basis von ATM [HH91] für Glasfaser-Netze zur Verfügung stehen [Kil91]. Für diese Hochgeschwindigkeitsnetze werden auch neue Transportprotokolle entwickelt [HS91, DDK+90].

2.3 Vom Standbild zum Bewegbild in Fenstersystemen

Fenstersysteme wie X erlauben neben der Darstellung von einfachen Graphiken (z.B. Linien, Kreise, Polygone) auch die Einzelbildübertragung aus einer Anwendung heraus an das Fenstersystem. Der Schritt vom Stand- zum Bewegbild wurde daher zunächst mittels Anwendungen versucht, die ihre Bilder schnell über das Netz beziehen und an den X-Server weiterreichen, der diese dann darstellt. Für die notwendigen Mechanismen zur Bewegbildübertragung ist die Anwendung jedoch selbst zuständig [Lof90].

Der X-Client übernimmt dabei die Aufgabe, Bilder von einem Movie-Server anzufordern und an den X-Server weiterzuleiten. Der Movie-Server liest auf Anforderung die gespeicherten Bildinformationen von einem digitalen Massenspeicher und sendet sie an den X-Client. Er benutzt dabei ein Übertragungsprotokoll, welches besonders auf die Belange der Bewegbildübertragung ausgerichtet ist. Diese Vorgehensweise hat mehrere Nachteile:

- Die Bildinformationen werden unter Umständen zweimal über das Netz gesendet, und zwar genau dann, wenn jeweils X-Server und X-Client bzw. X-Client und Movie-Server auf verschiedenen Rechnern ablaufen.
- Der Code zur Bewegbilddarstellung wird zu jedem X-Client gebunden. Dadurch werden die ausführbaren Programme vergrößert.
- Der Kommunikationsaufwand zwischen X-Server und X-Client ist durch die Bewegbildübertragung sehr hoch. In einer relativ kurzen Zeit muß eine große Datenmenge übertragen und der Filmablauf gesteuert werden. Die Kommunikation zwischen X-Server und X-Client basiert jedoch auf einem verbindungsorientierten und zuverlässigen Transportprotokoll (z.B. TCP/IP oder DECnet). Dabei werden Fehler wie verlorengegangene und aus der Reihenfolge geratene Pakete erkannt und behoben, wodurch der Durchsatz der Verbindung verringert wird.

Andere Forschungsarbeiten haben sich darauf konzentriert, die Integration der Bewegbildübertragung in ein Fenstersystem zunächst theoretisch aufzuarbeiten. Das Ergebnis einer solchen Arbeit ist ACME (*Abstractions for Continous Media*). Es beschreibt eine Menge von Abstraktionen für die Ein- und Ausgabe von kontinuierlichen Medien (Audio und Vi-

¹ Ein FDDI-Ring wird oft als Backbone für lokale Netze benützt. Die dabei zur Zeit verwendete Technik der Einkapselung bringt allerdings ernüchternde Ergebnisse in Bezug auf den Durchsatz [Spa91].

deo) mit einem netzwerkbasiereten System zur Fensterverwaltung. Das Ziel ist die Erweiterung eines Standard-Fenstersystems wie X oder NeWS [AGH90].

Eine andere Erweiterung von X ist VEX (*Video Extension to X*, [Bru90]). Diese ermöglicht die Steuerung von Spezialhardware, um z.B. Live-Video in Fenstern anzuzeigen und Graphiken aus Fenstern zu entnehmen und in ein Videosignal umzusetzen.

2.4 Das XMovie-System

Das System XMovie ist eine verteilte Testumgebung für digitale Bewegbilddarstellung in einem Netz aus UNIX-Rechnern an der Universität Mannheim. Es kommt ohne spezielle Hardware aus und benutzt herkömmliche Netztechnik zur Übertragung und handelsübliche Graphik-Adapterkarten zur Darstellung von Bewegbildsequenzen [LE91].

In XMovie wird das X-Window-System aufgrund dreierlei Tatsachen benutzt. Erstens, weil es sich als ein de-facto-Standard auf Arbeitsplatzrechnern etabliert hat, zweitens, weil der Quellcode vollständig zur Verfügung steht und drittens, weil es eine Vielzahl von Anwendungen für X gibt, die von einer Möglichkeit zur Filmdarstellung profitieren könnten. Der X-Server übernimmt dabei die Rolle eines Movie-Client. Die Architektur des XMovie-Systems zeigt Abb. 2.

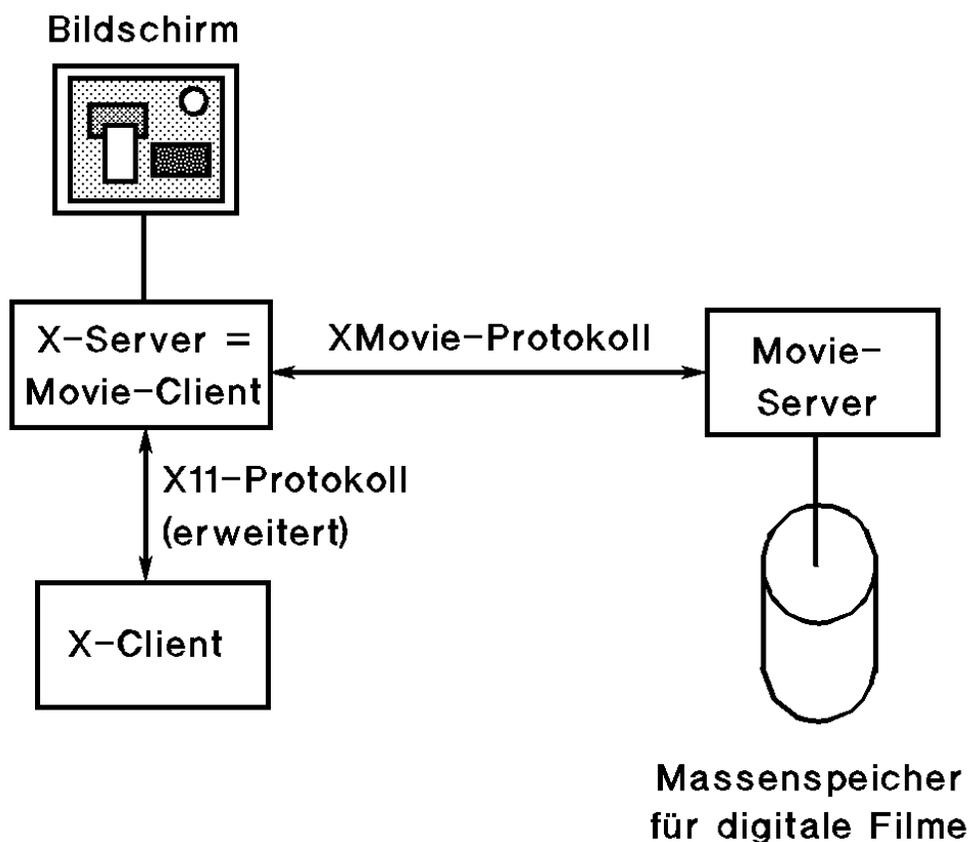


Abb. 2: Die Architektur des XMovie-Systems

Der Movie-Client kommuniziert mit dem Movie-Server, der die Bilddateien bereithält, und stellt die empfangenen Bilder in einem Fenster dar. Das XMovie-Protokoll ermöglicht es, mit mehreren Servern gleichzeitig Verbindungen zu unterhalten. Da sowohl der Übertragungs- als auch der Darstellungsaufwand bei der Bewegbildübertragung aufgrund der großen Datenmenge beachtlich ist, muß die Implementierung vor allem an Effizienzkriterien gemessen werden.

Bei Experimenten wurde festgestellt, daß die mit TCP erreichbaren Datenraten völlig unzureichend sind [LE91]. Daher wurde die Entscheidung getroffen, die Kommunikation mit dem Movie-Server über eine UDP-Verbindung zu betreiben (siehe auch [GM91]). Das Aufsetzen auf eine Transportschnittstelle (Schicht 4) schirmt zugleich die Implementierung des XMovie-Systems von den Details der tieferen Schichten ab, so daß neue Netztechnologien wie FDDI, DQDB und später auch ATM dem XMovie-System mit minimalen Anpassungsarbeiten unmittelbar zugute kommen.

3. Bewegtbilder mit X

3.1 X-Client

3.1.1 Erweiterte Xlib-Schnittstelle

Um den Bewegtbildservice für den Anwendungsprogrammierer zugänglich zu machen, wurde eine Anfrageschnittstelle zum X-Server mit den folgenden Befehlen entworfen:

- *<list movies>*
- *<list movies with info>*
- *<init>*
- *<open>*
- *<play>*
- *<show picture>*
- *<stop>*
- *<update>*
- *<destroy>*

Auf die Funktionalität von *<list movies>* und *<list movies with info>* wird im weiteren nicht näher eingegangen, da beide zur eigentlichen Bewegtbildübertragung nicht benötigt werden. Sie ermöglichen Anfragen an einen Filmdatenbank-Server [Kel91].

3.1.2 Befehlsfolgen

Vorbereitung

Die Arbeitsweise der Programmierschnittstelle des erweiterten X-Clients läßt sich mit Hilfe des Zustands-/Übergangendiagramms in Abb. 3 beschreiben; dabei wird auf die in Abb. 2 vorgestellte Architektur zurückgegriffen. Der X-Client befindet sich vor der Bewegtbild-darstellung im Zustand *<Idle>*. Er muß zunächst ein *<init>* durchführen, um das Filmfenster vorzubereiten. Dazu wird eine eigene Farbtafel erzeugt und dem Fenster zugeordnet. Diese Vorgehensweise ist notwendig, damit die Farbtabelle des Films abgespeichert werden kann, ohne die Standard-Farbtafel von X zu überschreiben (diese wird normalerweise einem Fenster bei der Erzeugung zugeordnet).

Beim Erzeugen einer neuen Farbtafel sind die Farbeinträge noch nicht belegt. Damit nicht alle Farben aus den anderen Fenstern des Bildschirms verschwinden, wenn die neue Farbtafel vor dem Film lauf geladen wird, werden aus der Standard-Farbtafel von X einige Farbeinträge in die neue Farbtafel kopiert. Dabei werden speziell die Farben ausgewählt, die von

den meisten anderen X-Anwendungen, z.B. für Menüs und Rollbalken, ebenfalls benutzt werden.

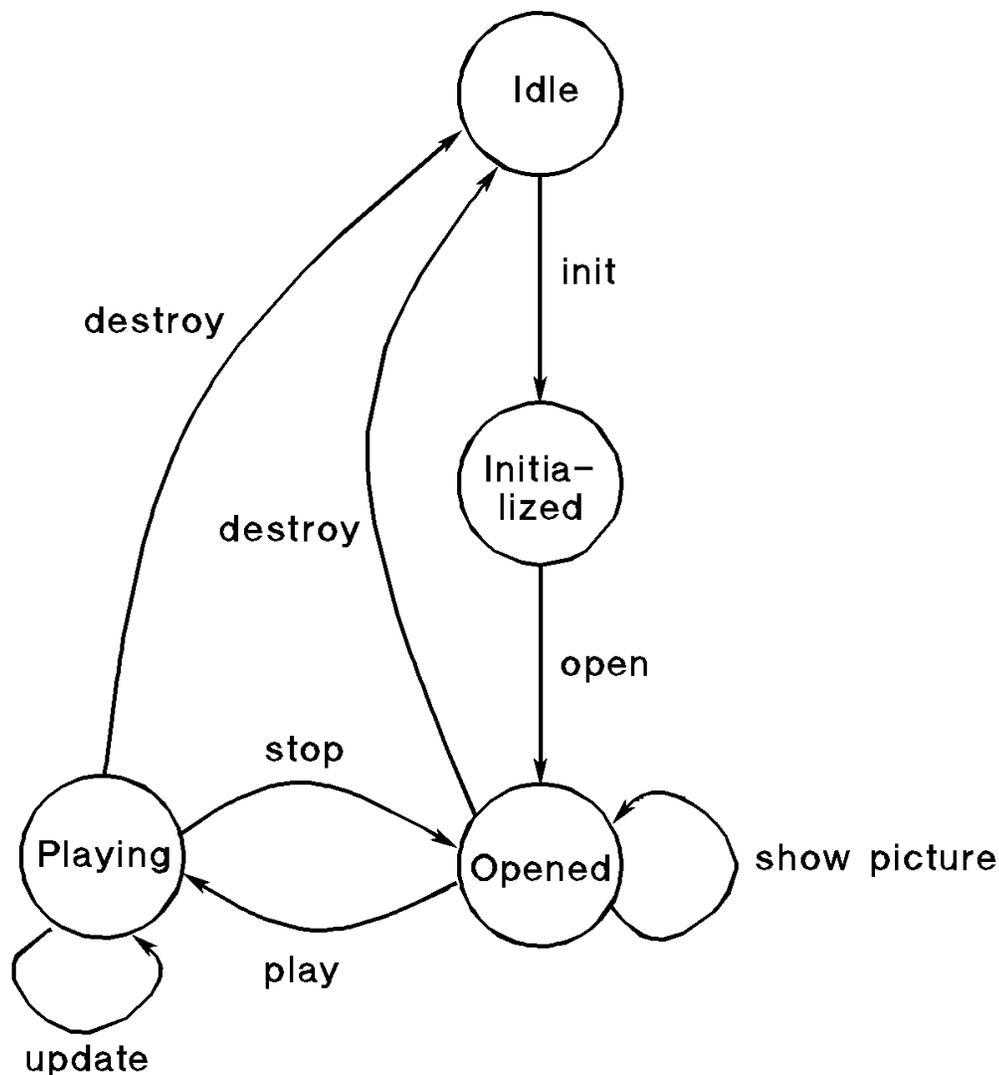


Abb. 3: Zustands-/Übergangsdiagramm der Xlib-Erweiterung

Im Zustand `<Initialized>` angekommen, kann ein Filmauftrag mittels `<open>` abgesetzt werden. Damit wird der X-Server angewiesen, alle Ressourcen und Informationen für den Film zur Verfügung zu stellen. Dazu wird dem X-Server mitgeteilt, von welchem Movie-Server der Film angefordert werden soll. Falls kein Fehler auftritt, ist nach diesem Aufruf der Auftrag eröffnet, und die Bewegbilddarstellung kann beginnen (im Zustand `<Opened>`).

Durch ein `<open>` erhält die Anwendung die Identifikationsnummer des Films (Movie-ID). Nur mit dieser können alle weiteren Funktionen aufgerufen werden. Die Movie-IDs werden von der Xlib vergeben und sind für jede Kombination von Film und Anzeigefenster eindeutig. Falls also derselbe Film in einem anderen Fenster angezeigt werden soll, wird diesem Auftrag eine neue Movie-ID zugeordnet.

Filmlauf

Im Zustand `<Opened>` stehen zwei Bilddarstellungsoptionen zur Verfügung: `<play>` und `<show picture>`. Durch ein `<play>` wird die Darstellung des durch die Movie-ID angegebenen Films gestartet. Dabei können verschiedene Parameter gesetzt werden, die den genauen Ablauf bestimmen. Es kann der ganze Film vom ersten bis zum letzten Bild angefordert

werden oder nur eine bestimmte Bildfolge. Sowohl die Laufrichtung (vorwärts/ rückwärts) als auch die Bildwiederholfrequenz lassen sich angeben. Diese Parameterwerte lassen sich beliebig kombinieren. Es kann z.B. der ganze Film langsam rückwärts oder nur ein Teil des Films schnell vorwärts abgespielt werden. Schnell und langsam beziehen sich dabei auf die Bildwiederholfrequenz, die mit "Bilder pro Sekunde" spezifiziert wird. In einer komfortablen Anwendung lassen sich diese Optionen bequem über Schaltknöpfe und Menüs auswählen.

Durch ein `<play>` gelangt die Anwendung in den Zustand `<Playing>`, aus dem sie entweder mit einem expliziten `<stop>` oder durch das Ende des Films wieder in den Zustand `<Opened>` zurückkehrt.

Alternativ zu einem `<play>` kann im Zustand `<Opened>` auch ein `<show picture>` verlangt werden. Dieses bewirkt die Darstellung genau eines Bildes. Dabei wird darauf geachtet, daß dieses, ebenso wie das letzte Bild eines normalen Filmablaufs, möglichst komplett und fehlerfrei dargestellt wird.

Während eines Filmablaufs können die beim Start gesetzten Parameter durch ein `<update>` geändert werden. So läßt sich die Laufrichtung umkehren, die gewählte Bildfolge verlängern oder verkürzen und die Geschwindigkeit erhöhen oder erniedrigen.

Der Filmablauf kann jederzeit durch ein `<stop>` angehalten werden. Dabei wird die Nummer des zuletzt angezeigten Bildes durch eine Antwort zurückgegeben. Läuft der Film dagegen bis zum letzten Bild ohne Unterbrechung, so wird das Filmende durch eine Ereignismeldung angezeigt, welche ebenfalls die Nummer des letzten Bildes als Information mitführt.

Mit `<destroy>` wird der X-Server aufgefordert, alle Ressourcen freizugeben, die für diesen Filmauftrag von ihm verwaltet werden. Dazu gehören die Informationen über den Movie-Server und die Movie-ID. Läuft der Film noch (d.h. die Anwendung befindet sich im Zustand `<Playing>`), so wird er gestoppt. Dies geschieht analog zu einem `<stop>`, mit dem Unterschied, daß die Nummer des letzten angezeigten Bildes nicht zurückgegeben wird. Nach einem `<destroy>` befindet sich die Anwendung wieder im Zustand `<Idle>`.

Fehlermeldungen

Durch den Bewegtbildservice kann es zu Fehlern kommen, die sich durch die Semantik der im X-Protokoll definierten Fehlermeldungen (siehe [SGN88]) nur ungenügend beschreiben lassen. Daher wurde es notwendig, für diese Erweiterung eigene Fehlermeldungen zu definieren, um auf fehlerhafte Verbindungen zum Movie-Server und falsche Parameter in den Anforderungen hinzuweisen. Einzelheiten sind in [Kel91] zu finden.

3.2 X-Server

3.2.1 Filmaufträge

Der X-Server wurde dahingehend erweitert, daß er Filmaufträge verwalten kann. Ein Filmauftrag (*Order*) spielt die Vermittlerrolle zwischen den Wünschen des X-Client, die über das X-Protokoll an den X-Server herangetragen werden, und der Abarbeitung dieser Wünsche durch die Kommunikation mit dem Movie-Server über das XMovie-Protokoll (siehe Abb. 4).

Der X-Server verwaltet eine Auftragsliste. Jeder dieser einzelnen Aufträge enthält Informationen über den jeweils darzustellenden Film. Diese umfassen den Auftraggeber (den X-Client), den Filmtitel und den Movie-Server, von dem der Film angefordert werden soll. Darüber hinaus werden das Filmfenster, genaue Daten über die darzustellenden Bilder

(Breite, Höhe, Anzahl der Bilder im Film, Größe eines Pixels) sowie verschiedenen Hilfs- und Zustandsinformationen gespeichert.

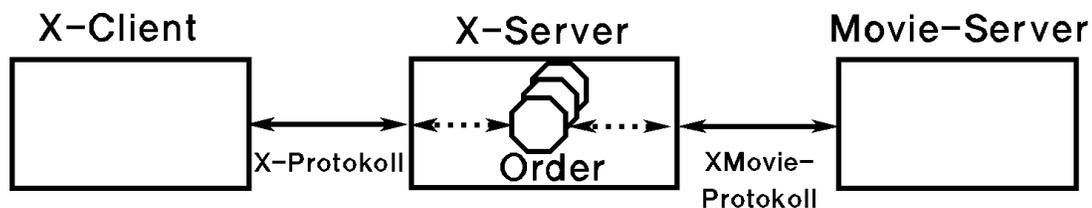


Abb. 4: Funktion eines Auftrags

Insbesondere auf diese Zustandsinformationen soll im weiteren vertieft eingegangen werden, um das Verständnis für die Zusammenhänge der einzelnen Abläufe im X-Server zu wecken.

3.2.2 Verarbeitung eines Auftrags

Trifft eine *<open>*-Anforderung beim X-Server ein, so wird ein neuer Auftrag eröffnet. Dieser neue Auftrag befindet sich im Zustand *<New Order>*. Der Zustand *<Idle>* ist gleichbedeutend mit nicht existent, d.h. erst beim Eröffnen wird der benötigte Speicherplatz belegt (siehe Abb. 5).

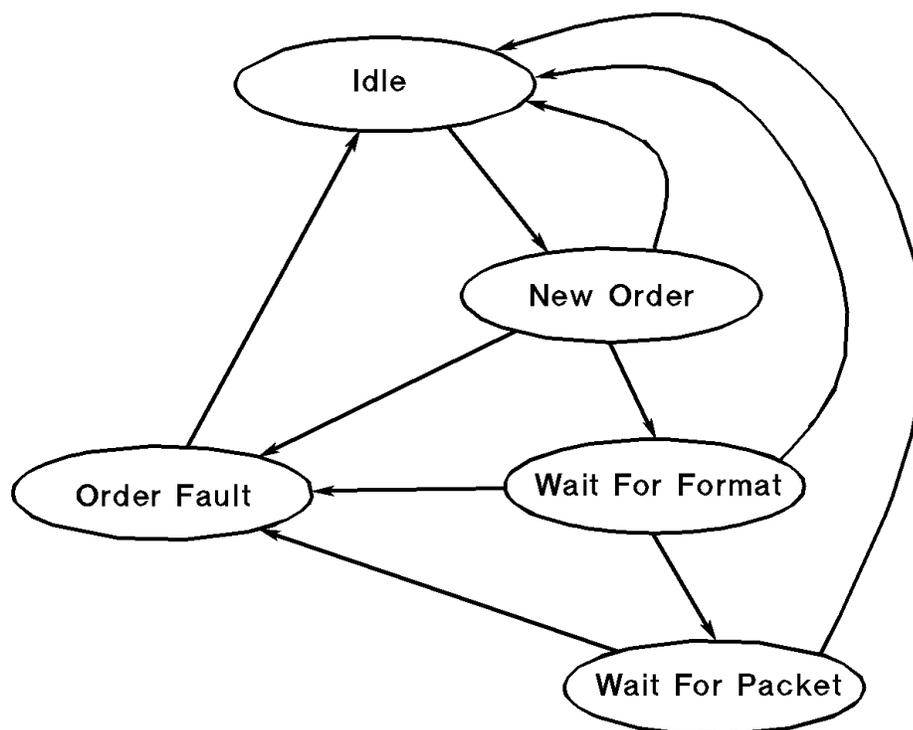


Abb. 5: Zustände eines Auftrags

Im neuen Auftrag werden die Informationen abgelegt, die in der Anforderung enthalten sind: die Movie-Identifikationsnummer, der Name des Movie-Servers, der Filmmame und das Filmfenster. Außerdem wird ein neues UDP-Socket für den XMovie-Service eröffnet. Gelingt das nicht, so wird eine Fehlermeldung an den X-Client gesandt, und der Auftrag geht in den Zustand *<Order Fault>* über. Andernfalls wechselt er in den Zustand *<Wait For Format>*.

Hier wird versucht, eine Verbindung zum Movie-Server aufzubauen. Für den X-Server ist ein Verbindungsaufbau erfolgreich, sobald er vom Movie-Server ein Format-Paket empfängt. Dieses enthält Daten über den angeforderten Film, die für die Übertragung und Dar-

stellung wichtig sind. Dazu gehören die Breite und die Höhe eines Bildes, die Größe eines Pixels und die Zahl der im Film verfügbaren Bilder. Weiterhin sind in diesem Paket Informationen über das Format enthalten, in dem die Bilder über das Netz fließen, d.h. in wieviele Pakete ein Einzelbild zur Übertragung zerlegt werden muß und wieviele Pixel in einem Paket abgelegt sind.

Schon ab einer relativ geringen Größe muß ein Bild auf mehrere Pakete verteilt werden. Durch Experimente wurde festgestellt, daß das UDP-Protokoll auf den verwendeten IBM PS/2 unter AIX nur eine maximale Paketgröße von 2 KByte ermöglicht. Damit könnten beispielsweise Vollbilder mit einer Seitenlänge von 40×50 Pixeln auf einmal übertragen werden, also sehr kleine Bilder.

Schlägt der Verbindungsaufbau fehl, weil der Movie-Server sich entweder nicht meldet oder den Verbindungsaufbau durch ein Fehler-Paket ablehnt, so wechselt der Auftrag in den Zustand *<Order Fault>*. Andernfalls befindet er sich im Zustand *<Wait For Packet>*.

In diesem Zustand kann zu jeder Zeit eines der folgenden Pakete vom Movie-Server eintreffen:

- Bildpaket: Dieses enthält einen Teil des angeforderten Bildes, der sofort für die Ausgabe im Filmfenster weiterverarbeitet wird.
- Farbtabelle-Paket: Hiermit können bis zu 128 Farbtabelleinträge übertragen werden, die sofort in die dem Filmfenster zugeordnete Farbtafel eingetragen werden. Dadurch kann die Farbtabelle während des Filmlaufs geändert werden.
- Fehler-Paket: Mit diesem signalisiert der Movie-Server dem X-Server, daß er keine weiteren Bilder mehr senden kann und die Verbindung als abgebrochen betrachtet. Daraufhin wird eine Fehlermeldung an den X-Client geschickt, und der Auftrag geht in den Zustand *<Order Fault>* über.

Falls kein Paket eingetroffen ist, wird entschieden, ob eine Bildanforderung an den Movie-Server gesendet werden muß. Dies ist dann der Fall, wenn zuvor die Ablaufparameter durch eine *<play>*-Anforderung entsprechend gesetzt wurden. Aus diesen wird jeweils die Nummer des nächsten darzustellenden Bildes errechnet, wenn ein eventuell zuvor angefordertes Bild entweder vollständig eingetroffen ist oder die zum Empfang und zur Darstellung verfügbare Zeit abgelaufen ist (siehe unten). Diese Nummer errechnet sich aus den Ablaufparametern. Soll z.B. der ganze Film vorwärts ablaufen und wurde als letztes Bild Nummer 10 empfangen, so fordert der X-Server als nächstes Nummer 11 an.

Während sich ein Auftrag im Zustand *<Wait For Packet>* befindet, kann der X-Client die Ablaufparameter durch eine *<update>*-Anforderung ändern oder den Film stoppen und ihn anschließend wieder starten. Ebenso kann er durch eine *<destroy>*-Anforderung anzeigen, daß dieser Auftrag nicht mehr benötigt wird. Daraufhin wird der Auftrag aus der Auftragskette entfernt und der belegte Speicher freigegeben, wodurch der Zustand *<Idle>* eingenommen wird. Eventuell noch eintreffende Pakete vom Movie-Server für diesen Filmauftrag werden ignoriert.

Aus den drei Zuständen *<New Order>*, *<Wait For Format>* und *<Wait For Packet>* gelangt der Auftrag beim Eintreten eines Fehlers in den Zustand *<Order Fault>*. Aus diesem kann er entweder durch eine *<destroy>*-Anforderung in den Zustand *<Idle>* übergehen, oder der Auftrag wird vom Server automatisch gelöscht, wenn er entdeckt, daß der X-Client nicht mehr existiert. In diesem Fall gibt der X-Server alle Ressourcen frei, die dieser Anwendung zugeordnet sind (z.B. Fenster, Farbtafeln, Film-Aufträge).

3.2.3 Realisierung der Bildwiederholffrequenz

Der X-Server versucht, für jeden Filmauftrag die geforderte Bildwiederholffrequenz so weit wie möglich zu erreichen, ohne daß die Bildqualität zu stark absinkt. Dies wird durch die Verwaltung von Timern und mehreren Zustandsvariablen erreicht.

Einer dieser Timer legt z.B. fest, wieviel Zeit für den Empfang und die Darstellung eines Bildes verbraucht werden darf. Diese berechnet sich aus der Anzahl Bilder pro Sekunde, die vom X-Client als Wiederholffrequenz gefordert werden. Wenn z.B. 10 Bilder pro Sekunde dargestellt werden sollen, dann bleibt für ein Bild nur eine Verarbeitungszeit von 0,1 Sekunden.

Treffen innerhalb dieses Intervalls keine Bildpakete ein, so fordert der X-Server ein neues Bild beim Movie-Server an. Andernfalls werden zunächst die eingetroffenen Pakete verarbeitet und die Bildteile dargestellt. Erst nachdem keine neuen Bildteile mehr eintreffen, wird das nächste Bild angefordert². Es kann also passieren, daß bei einer geforderten Bildwiederholffrequenz von z.B. 10 Bildern pro Sekunde der X-Server zwar 0,1 Sekunden für die Übertragung, den Empfang und die Darstellung eines Bildes zur Verfügung hat, jedoch diese Arbeit 0,2 Sekunden beansprucht. Dann beträgt die realisierte Wiederholffrequenz auch nur 5 Bilder pro Sekunde. Genauso kann der Fall eintreten, daß die Bilder schneller zur Verfügung stehen, z.B. 0,08 Sekunden nach Anforderung. Dann wartet der X-Server mit der Anforderung des nächsten Bildes.

3.2.4 Bildaufbau

Das verwendete UDP-Protokoll für die Kommunikation mit dem Movie-Server ist ein Datagramm-Protokoll; es garantiert nicht die Ablieferung aller gesendeten Pakete. Dies ist für die Bewegbilddarstellung aber nicht relevant, da fehlende Bildteile beim nächsten Bild überschrieben werden. Wie schon erwähnt, stellt der X-Server jeden ankommenden Bildteil sofort dar. In jedem Paket befindet sich ein rechteckiger Bildausschnitt in voller Bildbreite und eine Information darüber, an welche Stelle des Gesamtbildes dieser Teil gehört. Der X-Server erstellt das Gesamtbild aus den Einzelteilen gemäß Abb. 6. Das Bild besteht in diesem Beispiel aus fünf Teilen. Es wird von oben nach unten aufgebaut.

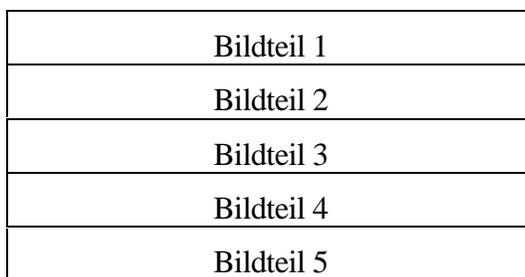


Abb. 6: Aufbau eines Bildes aus einzelnen Teilen

Dabei werden die Pixelwerte des zuvor dargestellten Bildes überschrieben. Fehlt ein Teil, so bleibt der entsprechende Ausschnitt des vorherigen Bildes sichtbar. Dabei treten keine Fehlfarben auf, falls die in Kapitel 2.1.5 vorgestellte dynamische Anpassung der Farbtafel angewendet wird. Bei 25 Bildern pro Sekunde wird ein fehlender Bildteil bereits nach 1/25s

² Die Entscheidung, jedes Bild einzeln anzufordern, wurde bei einer Prototyp-Implementierung getroffen, um das Protokoll möglichst einfach zu halten. Da aber der Kommunikationsaufwand dabei sehr hoch ist, befindet sich zur Zeit ein verbessertes Protokoll in der Entwicklung.

durch einen neuen überschrieben, und die Erfahrung zeigt, daß dies vom Betrachter nicht wahrgenommen wird.

3.3 Der Movie-Server

Die Aufgabe des Movie-Servers ist es, die vom X-Server angeforderten Einzelbilder von der Platte zu lesen, vorzuerarbeiten und an den X-Server zu senden. Das für die Kommunikation zwischen X-Server und Movie-Server entwickelte hybride Protokoll (XMovie-Protokoll, siehe Abb. 2) basiert, wie erwähnt, auf UDP. Es ermöglicht sowohl die gesicherte als auch die ungesicherte Übertragung von Daten. Das XMovie-Protokoll wird im folgenden kurz beschrieben.

Nach dem Erhalt des Bilddateinamens vom Movie-Client prüft der Movie-Server, ob eine Datei mit diesem Namen vorhanden ist. Falls dies zutrifft, werden die Daten über das Bildformat an den Movie-Client gesendet. Dieses besteht aus der Anzahl der Pixel pro Zeile sowie die Anzahl der Zeilen pro Bild. Außerdem werden noch die Pixelhöhe und die Pixelbreite angegeben sowie die Zahl der Bilder, die in der betreffenden Bilddatei gespeichert sind. Ohne auf eine Bestätigung für diese Informationen zu warten, überträgt der Server danach die Farbtafel in zwei Teilen. Nach jedem Teil wartet der Server auf eine Empfangsbestätigung vom Client (gesicherte Übertragung der Farbtafel). Nach dem Senden der Farbtafel geht der Server in den Zustand über, in dem er Bildanforderungen bearbeiten kann.

Der Movie-Server ist nicht dafür ausgelegt, eine Bewegtbildübertragung im engeren Sinn zu ermöglichen. Dies bedeutet, daß er keine Anforderungen der Art "Sende aus Bilddatei <movie> fortlaufend die Bilder von Nummer <n> bis <m>" bearbeiten kann. Stattdessen erhält er für jedes Bild eine eigene Anforderung, aufgrund derer er die im folgenden beschriebenen Aktionen ausführt (also ein einfaches "Stop-and-Wait"-Protokoll). Diese Vorgehensweise ist nicht optimal. Um allerdings Bildfolgen zwischen Movie-Client und Movie-Server zu übertragen, muß erst eine verbesserte Flußkontrolle implementiert werden.

Der Server liest jeweils einen Paketteil ein und sendet ihn an den Client, ohne auf eine Bestätigung zu warten. Dies wird solange wiederholt, bis das ganze Bild sequentiell abgearbeitet ist. Danach wartet er auf eine erneute Anforderung.

4. Implementierung

4.1 Implementierungsumgebung

Die Implementierung erfolgte zunächst auf einem IBM PS/2 Modell 80 (80386 CISC-Prozessor) mit 12 MByte Hauptspeicher und dem Betriebssystem AIX. Der PS/2-Rechner war mit einem Bildschirmadapter 8514a ausgestattet. Als Software-Entwicklungsumgebung diente X11 Rel. 4, und als Programmiersprache wurde C verwendet. Die benutzte Arbeitsstation besaß einen Ethernet-Anschluß, über den u.a. mit einem DECSystem 5400 unter dem Betriebssystem Ultrix kommuniziert werden konnte. Dadurch konnten verschiedene Testreihen mit lokaler und entfernter Übertragung der Bilddaten zwischen X-Server und Movie-Server durchgeführt werden (siehe Kapitel 5).

Inzwischen erfolgte die Portierung auf eine DECstation 5000/120 CX (R3000 RISC-Prozessor) mit 16 MB Hauptspeicher, dem Betriebssystem ULTRIX und X11 Rel. 5.

4.2 Einbettung der Erweiterung in X

Um die in Kapitel 3 beschriebene Funktionalität in das X-Window-System einzufügen, waren Erweiterungen in der Xlib, im X-Protokoll und im X-Server nötig (siehe Abb. 7 sowie [Kel91]).

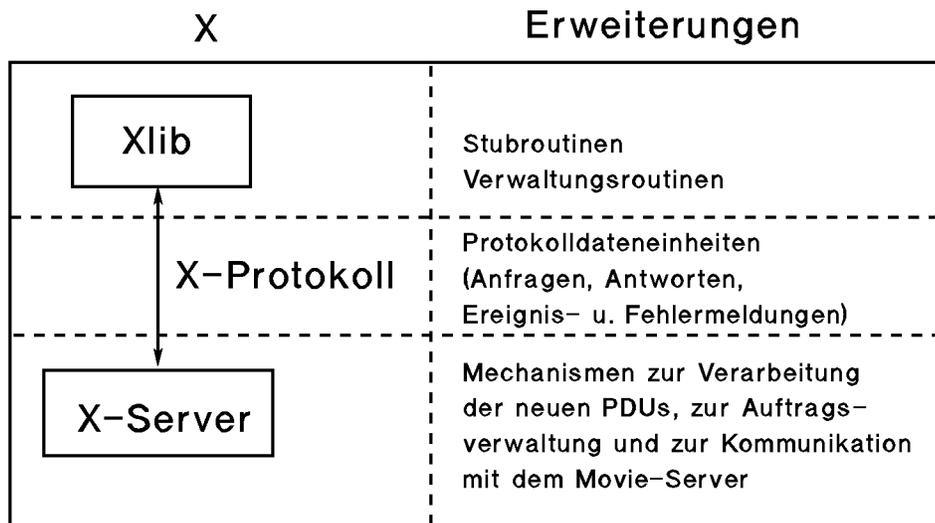


Abb. 7: Erweiterung von X

In die Xlib wurden neue Stub-Routinen eingefügt, um die Anfragen an den X-Server zu senden und Antworten zu verarbeiten. Für die Verwaltung der Erweiterung, insbesondere für die neuen Ereignis- und Fehlermeldungen, mußte ebenfalls gesorgt werden [SGN88].

In das X-Protokoll wurden die neuen Protokolldateneinheiten (PDUs) hinzugefügt, die zwischen dem X-Server und einem X-Client, der die XMovie-Erweiterung anspricht, ausgetauscht werden.

Den größten Implementierungsaufwand erforderte der Ausbau des X-Servers. Hier mußte dafür gesorgt werden, daß die neuen PDUs des X-Protokolls richtig empfangen, verarbeitet bzw. gesendet werden [ADK+88]. Außerdem mußte die Verwaltung der Aufträge geregelt und insbesondere die zur Kommunikation mit dem Movie-Server notwendige Funktionalität eingefügt werden.

4.3 Der Movie-Server

Der Movie-Server wurde parallel auf IBM PS/2 und DECSystem 5400 entwickelt, um Bildübertragungen von unterschiedlich schnellen Maschinen zu testen. Von der Universität Karlsruhe wurden uns dankenswerterweise Filmausschnitte zur Verfügung gestellt, die mit dem VERA-Raytracing-System [LMS91] erzeugt wurden. Diese Filmausschnitte wurden vorab in das oben beschriebene Farbtabelleformat umgewandelt und auf der Festplatte abgelegt. Der Movie-Server kann auf jeden Film mit normalen Dateioperationen zugreifen.

5. Erfahrungen

5.1 Leistung des Bewegbildservices

In der ersten Implementierung standen für Tests zwei Filme zur Verfügung. Der erste Film mit Namen *Schwarm* hatte das Bildformat 128×128 und bestand aus 50 Bildern, die in je 9 Paketen übertragen wurden. Im zweiten Film (*DNA*) waren mehr (72) und größere

(220 × 220) Bilder abgespeichert, von denen jedes in 25 Paketen übertragen wurde³. Es fanden mehrere verschiedene Testreihen statt:

1. Ein lokaler Movie-Server sendete den Schwarm-Film, d.h. der X-Server und der Movie-Server liefen beide auf demselben IBM PS/2 Modell 80.
2. Ein lokaler Movie-Server sendete den DNA-Film.
3. Ein Movie-Server auf einer schnelleren Maschine (DEC 5400) sendete den Schwarm-Film über das Netz.
4. Ein Movie-Server auf einer gleich schnellen Maschine sendete den Schwarm- oder den DNA-Film über das Netz.

Bei diesen Tests wurde u.a. gemessen, wieviel Zeit zwischen den Bildanforderungen und der Ankunft der einzelnen Bildpakete verging. In jeder Testreihe wurde der Film mit unterschiedlichen Bildwiederhol frequenzen abgespielt. Dabei wurde festgestellt, daß zwischen dem Eintreffen zweier Bildpakete mindestens 20 ms vergingen, selbst wenn nur ein Auftrag zu bearbeiten war. Wurde der X-Server jedoch mit mehr Arbeit belastet, so wuchs diese Zeitspanne auf 40 bis 60 ms oder mehr.

Diese Messungen bestätigten auch einige vorherige Experimente, bei denen sukzessive die Bildwiederhol frequenz erhöht wurde. Dabei stellten wir fest, daß sich der Schwarm-Film nicht schneller als mit 6 und der DNA-Film nicht schneller als mit 2 Bildern pro Sekunde abspielen ließ. Durch eine einfache Rechnung wurden diese Beobachtungen verifiziert: Da zwischen jedem Bildpaket mindestens 20 ms vergehen, können nicht mehr als 50 Pakete pro Sekunde verarbeitet und folglich bei dem kleinen Format (9 Teilpakete pro Bild) 5,6 und bei dem großen Format (25 Teilpakete pro Bild) 2 Bilder pro Sekunde dargestellt werden.

Neben der Bildwiederhol frequenz interessierte uns natürlich auch die Filmqualität. Dabei wurde festgestellt, daß bei einer lokalen Übertragung fast keine Bildpakete verloren gingen. Bei der Testreihe 3 jedoch mußte zuerst die Geschwindigkeit des Movie-Servers gebremst werden, mit der er die Bildpakete auf das Netz schickt, um den Verlust von Datenpaketen zu minimieren. Doch selbst dann gingen teilweise mehr als die Hälfte der Pakete verloren. Dieses Ergebnis war nicht befriedigend.

Wie oben erwähnt portierten wir das XMovie-System auf einen schnelleren Rechner (DECstation 5000/120). Bei der Portierung auf ULTRIX konnte neben der CPU ein weiterer Engpaß beseitigt werden: unter ULTRIX können größere UDP-Pakete versendet werden, so daß beim Senden über das Netz jedes Bild des Schwarm-Films nur noch in 6 Paketen übertragen werden mußte. Dadurch wurde der X-Server beim Lesen der Bildpakete von der Netzschnittstelle entlastet und die Anzahl der IP-Pakete auf dem Netz wurde durch die bessere Aufteilung der Bildinformationen reduziert. Da ein Großteil des Software-Overheads einmal pro Paket entsteht, unabhängig von der Paketgröße, wurde das Kommunikationssystem erheblich entlastet. Durch diese Maßnahme (und durch die schnellere CPU) wurde die maximale Bildwiederhol frequenz erheblich gesteigert. Der Schwarm-Film ließ sich jetzt mit max. 17 Bildern pro Sekunde und der DNA-Film mit max. 7 Bildern pro Sekunde abspielen.

Die geringste Anzahl verlorengangener Pakete wurde erreicht, wenn Movie-Client und Movie-Server auf verschiedenen Maschinen (beides DECstations) liefen, da dann für jede

³ Die Namen der Filme ergaben sich aus ihren Inhalten. Im ersten Film kreiste ein Schwarm Schmetterlinge um eine Lampe, im zweiten Film rotierte eine DNA-Doppelhelix über einem Schachbrett.

der beiden Seiten mehr CPU-Zeit zur Verfügung stand; im Schnitt ging weniger als ein Paket pro Bild verloren.

5.2 Engpässe

Die vom X-Server erreichbare Bildwiederholfrequenz hängt von der Auslastung der beteiligten Systemkomponenten ab. Dies hat zwei Ursachen: Erstens verwendet der X-Server bei der Abarbeitung eintreffender Ereignisse eine *Round-Robin*-Strategie ohne Prioritäten. Die XMovie-Erweiterung fügt sich in diesen Mechanismus ein, um Portierungen einfach zu machen. Der X-Server verwaltet in der Regel aber mehrere Verbindungen zu X-Clients, z.B. eine pro Fenster auf dem Bildschirm. Je nachdem, wieviele Anforderungen auf diesen gesendet werden, hat der Server mehr oder weniger Arbeit. Das führt dazu, daß zwischen den einzelnen Nachfragen, ob inzwischen Pakete vom Movie-Server eingetroffen sind, eine nicht bestimmbare Zeitspanne verrinnt. Zusätzlich kann der X-Server mehrere Filmaufträge gleichzeitig verarbeiten, wodurch ebenfalls einiges an Rechenzeit verbraucht wird.

Die andere Ursache ist, daß neben dem X-Server und einigen X-Clients auch noch andere Programme auf demselben Rechner ablaufen (z.B. der Movie-Server) und die Prozessorzeit gleichmäßig auf diese Prozesse verteilt wird (siehe auch Kapitel 2.2).

Daher sinkt sowohl mit steigender Prozeßzahl als auch mit mehr Bildaufträgen die erreichbare Bildwiederholfrequenz, da einfach nicht genügend Zeit vorhanden ist, die Bilder schnell genug anzufordern, zu empfangen und darzustellen.

Bei weiteren Leistungstests wurde festgestellt, daß das langsamste Glied in der Kette der physikalischen Übertragungsmittel (siehe Abb. 8) die verwendete Netzadapterkarte darstellte. Obwohl auf unserem Ethernet theoretisch eine Übertragungsrate von 10 MB/s möglich ist, konnte die zunächst eingesetzte Karte nicht mehr als 2 MB/s verarbeiten [LE91].

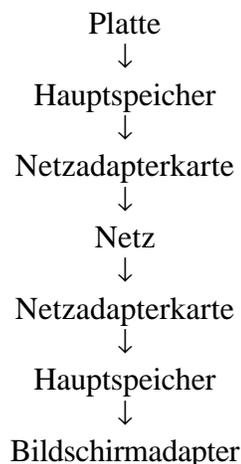


Abb. 8: Kette der physikalischen Übertragungsmittel

Daraus folgerten wir, daß sich die Leistung des XMovie-Systems durch schnellere Hardware für CPU und Netzadapterkarten erheblich verbessern läßt. Dies wurde auch durch die Portierung auf eine schnellere Hardware nachgewiesen. Die Netzadapterkarte der DECstation stellt zwar immer noch den Engpaß dar, sie kann aber immerhin 4 MB/s verarbeiten. Dies führte zu der oben erwähnten Steigerung der Bildwiederholfrequenz.

5.3 Portabilität

Die vorgenommene Erweiterung des X-Servers ist zum größten Teil portabel, da beim Benutzen der Systemeigenschaften von X auf Routinen zurückgegriffen wurde, die ihrerseits portabel sind. Dazu gehören vor allem die Aufrufe zum Abspeichern der Farbtableneinträge und zum Darstellen der Bildteile. Einzig die Erweiterungen, die für die Kommunikation mit dem Movie-Server eingebracht wurden, sind nicht ohne weiteres auf einen anderen Rechner übertragbar, da diese zu den vom Betriebssystem abhängigen Teilen von X gehören.

Daher erforderte es nur einen geringen Aufwand (etwa einen Mann-Monat), die XMovie-Erweiterung von X11 Rel. 4 auf einem IBM PS/2 Modell 80 mit dem Betriebssystem AIX auf X11 Rel. 5 auf einer DECstation 5000/120 mit dem Betriebssystem ULTRIX zu portieren und das XMovie-System an die neue und schnellere Umgebung anzupassen.

6. Zusammenfassung und Ausblick

Die Erweiterung des X11-Servers zur Bewegbilddarstellung ermöglicht den benutzerkontrollierten Ablauf von Filmen in Fenstern, die von X zur Verfügung gestellt werden. Diese neue Funktionalität wurde in das System eingefügt, ohne dessen Handhabung und Erscheinungsbild ("Look and Feel") zu verändern. Für den Programmierer wurde eine komfortable Programmierschnittstelle auf Xlib-Basis bereitgestellt, die es ihm ermöglicht, einen oder mehrere Filme von einem Movie-Server anzufordern und den Filmablauf zu steuern.

Nach einer Einführung in die Problematik der Bewegbildübertragung wurde die Erweiterung des X-Window-Systems vertieft vorgestellt. Dabei wurde besonderer Wert auf die Beschreibung der Funktionalität der Benutzerschnittstelle und die Verarbeitung von Filmaufträgen im X-Server gelegt. Es zeigte sich deutlich, daß eine signifikante Leistungsverbesserung des XMovie-Systems durch schnellere Hardware für die CPU und die Netzadapterkarten zu erwarten ist. Diese Erwartung wurde auch durch die Portierung auf eine DECstation 5000/120 bestätigt.

Als nächstes ist der Anschluß von zwei DECStations an einen bereits vorhandenen FDDI-Ring geplant, um eine schnellere und zuverlässigere Übertragung von Bewegbilder zu erreichen.

Im weiteren Ausbau des Systems wird die Einbeziehung der Tonübertragung und -ausgabe angestrebt. Dazu muß zusätzliche Audio-Hardware beschafft werden. Außerdem müssen annähernd isochrone Datenströme über das Netz übertragen werden können. Wünschenswert wäre in fernerer Zukunft ein Anschluß an ein DQDB- oder B-ISDN-Netz, da in diesen Netzen Bitströme isochron und schnell übertragen werden können.

Literaturverzeichnis

- [ADH91] D. P. Anderson, L. Degrossi und R. G. Herrtwich. Process Structure and Scheduling in Real-Time Protocol Implementations. In *Kommunikation in Verteilten Systemen*, Mannheim. Informatik-Fachbericht 267, Springer-Verlag, 1991, S. 83-95
- [ADK+88] S. Angebranntt, R. Drewry, P. Karlton und T. Newman. *Definition of the Porting Layer for the X v11 Sample Server*. Digital Equipment Corporation, März 1988

- [AGH90] D. P. Anderson, R. Govindan und G. Homsy. *Abstractions for Continuous Media in a Network Window System*. University of California at Berkeley, Report No. UCB/CSD 90/596, September 1990
- [Bru90] T. Brunhoff. *VEX: Video Extension to X, Version 5.9*. Textronic, Inc., 1990
- [Com88] D. Comer. *Internetworking with TCP/IP*. Prentice Hall, Englewood Cliffs, 1988
- [Cou91] B. Cousin. Digital Video Transmission and the FDDI Token Ring Protocol. In *Proc. 2nd International Workshop on Network and Operating System Support for Digital Audio and Video*, Springer-Verlag, Heidelberg, 1991, S. 375-387
- [CT90] D. C. Clark and D. L. Tennenhouse. Architectural Considerations for a New Generation of Protocols. *Computer Communication Review*, 20(4): 200-208, September 1990
- [CWB91] G. Coulson, N. Williams und G. S. Blair. Group Presentation of Multimedia Applications in IXION. *Computer Communications*, 14(3): 205-215, Mai 1991
- [DDK+90] T. W. Doeringer, D. Dykeman, M. Kaiserswerth, B. Meister, H. Rudin und R. Williamson. A Survey of Light-Weight Transport Protocols for High-Speed Networks. *IEEE Transactions on Communications*, 38(11): 2025-2039, November 1990
- [DN91] N. A. Davies und J. R. Nicol. Technological Perspective on Multimedia Computing. *Computer Communications*, 14(5): 260-272, Juni 1991
- [Gal91] D. Le Gall. MPEG: A Video Compression Standard for Multimedia Applications. *Communications of the ACM*, 34(4): 46-58, April 1991
- [GM91] R. Gusella und M. Maresca. Design Considerations for a Multimedia Network Distribution Center. In *Proc. 2nd International Workshop on Network and Operating System Support for Digital Audio and Video*, Springer-Verlag, Heidelberg, 1991, S. 185-196
- [HEH+92] B. Hofmann, W. Effelsberg, T. Held und H. König. On the Parallel Implementation of OSI Protocols. In *Proc. IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems (HPSC'92)*, Tucson, Arizona, Feb. 1992
- [HH91] R. Händler und M. N. Huber. *Integrated Broadband Networks: An Introduction to ATM-Based Networks*. Addison-Wesley, Wokingham, England, 1991
- [HS91] L. Henkel und H. Stüttgen. Transportdienste in Breitbandnetzen. In *Kommunikation in verteilten Systemen*, Mannheim. Informatik-Fachbericht 267, Springer-Verlag, Berlin Heidelberg, 1991, S. 96-111
- [Hua91] K.-T. Huang. Multimedia Classroom of the Future. *IEEE Data Engineering*, 14(3): 46-51, September 1991
- [Jon89] O. Jones. *Introduction to the X Window System*. Prentice Hall, Englewood Cliffs, 1989
- [Kel91] R. Keller. *Erweiterung des X11-Servers zur Bewegtbilddarstellung*. Diplomarbeit, Lehrstuhl für Praktische Informatik IV, Universität Mannheim, 1991

- [Kil91] U. Killat. B-ISDN und MAN: Konkurrierende Netztechnologien. In *Kommunikation in Verteilten Systemen (Tutorial)*, Mannheim. Deutsche Informatik-Akademie, 1991, S. 1-34
- [LE91] B. Lamparter und W. Effelsberg. X-MOVIE: Digitale Filmübertragung und Darstellung im X-Window-System. In *Telekommunikation und multimediale Anwendungen der Informatik, Tagungsband GI-21. Jahrestagung, Darmstadt, 1991*, Informatik-Fachbericht 293, Springer-Verlag, Berlin Heidelberg, 1991, S. 343-352
- [LEM92] B. Lamparter, W. Effelsberg und N. Michl. MTP: A Movie Transmission Protocol for Multimedia Applications. In *Proc. 4th IEEE ComSoc International Workshop on Multimedia Communications*, Monterey, California, USA, April 1992
- [LGC+91] T. D. C. Little, A. Ghafoor, C. Y. R. Chen und P. B. Berra. Multimedia Synchronisation. *IEEE Data Engineering*, 14(3): 26-35, September 1991
- [LMS91] W. Leister, H. Müller und A. Stößer. *Fotorealistische Computeranimation*. Springer-Verlag, Heidelberg, Berlin, 1991
- [Lof90] G. Loff. *Konzeption, Entwurf und Implementierung eines netzweiten Videodienstes*. Diplomarbeit, Universität Karlsruhe, 1990
- [Lut89] A. C. Luther. *Digital Video in the PC Environment*. McGraw-Hill Book Company, New York, 1989
- [PN91] W. Putz und E. J. Neuhold. is-News, a Multimedia Information System. *IEEE Data Engineering*, 14(3): 16-25, September 1991
- [RHF91] F. E. Ross, J. R. Hamstra und R. L. Fink. FDDI - A LAN among MANs. *Computer Communications Review*, 20(3): 16-31, July 1991
- [Ros90] M. T. Rose. *The Open Book: A Practical Perspective on OSI*. Prentice Hall, Englewood Cliffs, 1990
- [SGN88] R. W. Scheifler, J. Gettys und R. Newman. *X Window System: C Library and Protocol Reference*. Murray Printing Company, 1988
- [Spa91] O. Spaniol. Betriebserfahrungen und Messungen an einem großem FDDI-Netz ... und sich daraus ergebende Konsequenzen. In *Telekommunikation und multimediale Anwendungen der Informatik, Tagungsband GI-21. Jahrestagung, Darmstadt, 1991*, Informatik-Fachbericht 293, Springer-Verlag, Berlin Heidelberg, 1991, S. 22-24
- [SNM+91] C. Stavros, A. Natassa, F. Manolis, K. Yorgos und K. Leonidas. A Object Oriented Architecture for Multimedia Information Systems. *IEEE Data Engineering*, 14(3): 4-15, September 1991
- [Wal91] G. K. Wallace. The JPEG Stillpicture Compression Standard. *Communications of the ACM*, 34(4): 30-44, April 1991