Entwicklung von Algorithmen und Programmen für ein Archivierungs- und Kommunikationssystem zur internetbasierten Verwaltung medizinischer Bilder

Dissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
am Fachbereich IV der Universität Trier

Vorgelegt von
Diplom-Ingenieur (Elektrotechnik),
Kandidat der technischen Wissenschaften (Rußland)
Sergey Khludov

August 1999

Cutachter: Prof. Dr. sc. Christoph Meinel (Universität Trier)
Prof. Dr.-Ing. habil. Heinz-Ulrich Seidel (Universität Ilmenau)

Datum der Disputation: 27. April 2000

Danksagung

Die vorliegende Doktorarbeit wäre in dieser Form nicht denkbar gewesen, wenn mir nicht einige Menschen mit Rat und Tat beistanden hätten.

Ich bedanke mich bei Herrn Dr.-Ing. Lutz Grünwoldt für die Unterstützung bei technischen Fragen sowie bei Herrn Maik Jarre fürs Korrekturlesen des Manuskriptes und der Beratung bei medizinischen Fragestellungen.

Ferner danke ich dem Gutachter meiner Doktorarbeit, Herrn Prof. Dr. habil. H.-U. Seidel.

Mein besonderer Dank gilt meinem Doktorvater Herrn Prof. Dr. sc. Chr. Meinel, Direktor des Instituts für Telematik in Trier.

Sergey Khludov Trier, August 1999

Inhaltsverzeichnis

1. Einleitung **1**

2. Überblick über bekannte Visualisierungs-, Bildbearbeitungs- Archivierungs- und Telekommunikationssysteme für die Radiologie **5**

2.1 Die GE Advantage Windows Workstation **5**

2.2 TIM (Tomografic Imaging) **6**

2.3 MEDORA-PACS **6**

2.4 Das Projekt KAMEDIN **7**

2.5 SIENET Magic Viewer Workstation **7**

2.6 Das PACS vom Klinikum der Technischen Universität München **8**

2.7 CHILI **8**

2.8 DACHS **9**

2.9 DOIA **10**

2.10 Der Hipax DICOM Viewer **10**

2.11 ISIS (Interactive System for Image Selektion) **11**

2.12 Schlußfolgerungen und die Ziele der vorliegende Dissertation **12**

3. Entwicklung eines Konzeptes für ein Intranet/Internet-orientiertes, DICOM-basiertes radiologisches Informationssystems **14**

3.1 Einleitung **14**

3.2 Der DICOM-Standard und Entwicklung eines Datenmodells für eine DICOM-Datenbank **15**

3.2.1 Der DICOM-Standard 15

3.2.2 DICOM-Informationsmodell und –Bildformat 16

3.2.3 Probleme beim Umgang mit medizinischen Daten und Datenbankkonzepte 19

3.2.4 Aktuelle DICOM- Datenbankmodelle 22

3.2.5 Struktur für eine relationale DICOM-Datenbank..... 23

3.2.6 Entwicklung einer java-basierten objektorientierten DICOM-Datenbank 24

3.3 Konzeption eines DICOM-basierten Picture Archiving and Communication System (PACS) für Krankenhäuser **26**

3.4 Wartungsfreundlichkeit, Erweiterbarkeit und Flexibilität des PACS **29**

3.5 Schneller und einfacher Zugriff auf DICOM-Bilder aus dem PACS-Archiv über das Intranet/Internet **30**

3.6 Konzeption des Intranet-/Internet-basierten radiologischen Informationssystems **31**

3.7 Schlußfolgerungen **35**

4. Algorithmen für die Bildkompression **37**

4.1 Einleitung **37**

4.2 Digitale Darstellung des Bildsignals und gebräuchliche Bildformate **38**

4.3 Probleme der Kompression von medizinischen Bildern **40**

4.4 Überblick über bekannte Bildkompressionverfahren **41**

4.4.1 Verfahren zur verlustfreien Kompression 41

4.4.2 Verfahren zur verlustbehafteten Kompression 43

4.5 Parameter zur Beschreibung und Klassifikation medizinischer Bilder **46**

4.6 Kriterien zur Bewertung des Unterschieds zweier Bilder **49**

4.7 Zwei Methoden zur Bewertung und Filterung des Rauschens auf DICOM-Bildern **51**

4.8 Adaptive Kompressionsverfahren **53**

4.8.1 Adaptive Entropie-Kompression digitaler medizinischer Bilder 53

4.8.2 Die LZW-JPEG Kompression 57

4.9 Schlußfolgerungen **61**

5. Implementierung von JAVA- basierte Übertragungs- und Visualisierungs-verfahren für DICOM-Bilder **64**

5.1 Einleitung **64**

5.2 JAVA Client-Server Paradigma und JDBC-Technik **64**

5.3 Servlet Technik **66**

5.4 DICOM-Viewer-Anwendung **68**

5.5 DICOM-Viewer-Applet **71**

5.6 Datenaustausch zwischen Internet/Intranet-Browser und Klinikarchiv **73**

5.7 Schlußfolgerungen **75**

6. Schlußbetrachtungen **77**

7. Anhang **82**

7.1 A **82**

7.2 B **83**

7.3 C **85**

7.4 D **87**

8. Literatur **95**

1. Einleitung

Telemedizin - diese Form des Einsatzes von Informations- und Telekommunikationstechnik (multimediale elektronische Datenverarbeitung und/oder –übertragung) findet im Gesundheitssektor heute immer breiteren Zuspruch. Ohne die Einführung moderner Kommmunikationstechnologie kann in einer Klinik der Qualitätsstandard für die Patientenversorgung nicht dauerhaft gesichert werden.

Die Vernetzung von Kliniken, Rehabilitationszentren, Pflegestationen, Krankenkassen, Apotheken und Arztpraxen bietet neue Möglichkeiten der Anpassung von Arbeitsabläufen in medizinischen Einrichtungen an den allgemeinen Trend der Nutzung von Telekommunikationstechnik. Fast überall sind Arbeitsplatzrechner verfügbar und erleichtern im Zuge der Implementierung moderner Krankenhaus- (KIS), Radiologie- (RIS), Krankenkassen- (HIIS), Pharmazieinformationssysteme (PIS), Informationssysteme für Praxen (MPIS) und anderen ähnlichen Systemen den Zugriff auf medizinische Daten. Fachkollegen an unterschiedlichen Orten können über die Computernetze miteinander kommunizieren, Daten austauschen und mächtige verteilte Datenbanken nutzen, um schnell an notwendige Informationen zu gelangen und so die therapeutische Reaktionszeit drastisch zu verkürzen.

Telemedizininsche Anwendungen sind unter anderem bekannt aus den Bereichen Dermatalogie, Pathologie, Pädiatrie, Chirurgie und insbesondere der Radiologie. Oftmals geht es hier darum, das mit unterschiedlichen Bildquellen erstellte Bildmaterial zu begutachten, zu bearbeiten oder mit Fachkollegen auszutauschen. Da die meisten bildgebenden Verfahren heute unter der Radiologie zusammengefaßt werden, ist auch die Radiologie der Bereich der Medizin, der am meisten von den telemedizinischen Erkenntnissen betroffen ist.

Radiologie-Abteilungen gibt es in vielen Krankenhäusern und fast jeder Arzt verwendet während der Behandlung eines Patienten radiologische Informationen. Die Diagnoseerstellung basiert auf einer möglichst vollständigen Beschreibung des Patientenzustandes. In vielen Fällen wird dazu auf eine „radiologische Krankenakte“ zurückgegriffen. Die darin enhaltenen Informationen sind sowohl heute als noch in 10 Jahren von großer Bedeutung.

Ein problemloser, einfacher und schneller Zugriff auf diese „radiologische Krankenakte“ war bis vor nicht allzu langer Zeit ein seltener Idealfall. Und noch heute wird vielerorts noch ein enormer Aufwand an Organisation und Verwaltung betrieben, um die Hunderte von Bildern und Aufzeichnungen zu ordnen, zu archivieren und bei Bedarf wieder aus den Archiven herauszusuchen. Solche arbeits-, zeit- und kostenintensiven Vorgehensweisen gehen nicht nur zu Lasten des Gesundheitssystems, sondern oft auch zu Lasten des Patienten. Radiologieinformationssysteme bieten komfortable Auswege aus diesem schwer zu koordinierenden Datenauflauf.

Ein modernes und durchdachtes Radiologie-Informationssystem eröffnet klare Vorteile:

> Die große Menge an Röntgen-, CT-, Ultraschall- oder NMR-Bilder, die täglich in Kliniken anfällt, kann *effektiv, platzsparend* und *in einem System* organisiert und

archiviert werden. In Krankenhäusern, die mit modernen radiologischen Systemen ausgestattet sind, fallen pro Tag zwischen 50 und 1000 Radiologiebilder an, was einem digitalen Datenvolumen von 50 bis 1000 Megabyte entspricht. Mechanismen zur Verwaltung und Archivierung von radiologischen Daten sind demzufolge wünschenswert, insbesondere im Zusammenhang mit einem krankenhausweiten Informationssystem, durch das jede autorisierte Person von einem angeschlossenem Terminal aus Zugriff auf die gewünschten Daten hat.
Ein gutes RIS soll also die Daten effektiv, zeit- und platzsparend in einem System organisieren und archivieren können.

Auch niedergelassene Ärzte können von einem Anschluß an ein RIS profitieren, da sie oft ihre Patienten ins Krankenhaus schicken müssen um die benötigten Aufnahmen für die Diagnose dort anfertigen zu lassen. Diese Bilder fallen den Ärzten zur Aufbewahrung an und können das Praxis-Archiv vor schwer lösbare Probleme stellen. Durch ein RIS würden niedergelassene Ärzte bei der Verwaltung der Bildbefunde ihrer Patienten entlastet. Die elektronische Archivierung übernimmt das RIS der radiologischen Abteilung des jeweiligen Krankenhauses, in der die Aufnahmen angefertigt wurden; trotzdem sind die Informationen für den Arzt nur einen Maus-Klick weit entfernt.
Ein gutes RIS soll nach Möglichkeit ein offenes Netzwerk sein, daß auch die Option der Übertragung an andere, nicht Krankenhaus-interne Ärzte offen läßt.

Im allgemeinen existiert in den Krankenhäusern in Deutschland eine stark heterogene Hardwarelandschaft von verschiedenen Computersystemen (PCs, Apple Macintosh Rechner und vereinzelt auch Workstations). Desweiteren stammen im Normalfall die verschiedenen bildgebenden Geräte in einem Krankenhaus nicht von nur einem Hersteller. Der hohe Anteil an älteren, analogen Untersuchungsgeräten im Gerätepark der Krankenhäuser hemmt die Einführung eines digitalen Netzwerkes genauso stark, wie es die vielen unterschiedlichen Gerätetypen tun. Die Bearbeitung und Speicherung unterschiedlichster Daten von unterschiedlichen bildgebenden Geräten sollte am besten in *einem* System mit *einem* Datenformat und *einer* Software, die auf jedem *beliebigen Rechner* unter jedem *bliebigen Betriebssystem* läuft, erfolgen. Bildgebende Geräte von beliebigen Herstellern könnten dann problemlos zusammen geschaltet werden und die Bilder könnten auf jedem beliebigen, angeschlossenen Computer visualisiert und bearbeitet werden. Kompatibilitätsprobleme gehörten somit der Vergangenheit an. Den Benutzern des Systems stände es vollkommen frei, wie sie ihren Gerätepark zusammenstellen, denn das System wäre offen für jede Erweiterung und jede beliebige Hardware.
Ein gutes RIS muß ein System sein, das fähig ist, unterschiedliche Computer-Hardware, unterschiedliche Betriebssysteme und unterschiedliche bildgebende Modalitäten zu integrieren.

Darüberhinaus ist die Analyse eines Bildes am Bildschirm vielfach komfortabler und effizienter, da hier Möglichkeiten zur Bildbetrachtung (Lupe, Markierung, Invertierung, Überlagerung etc.) gegeben sind, die bei der normalen Version des Bildes nicht in der Form und Leistungsfähigkeit zur Verfügung ständen.
Ein gutes RIS muß eine leistungsfähige Bildbetrachtungsfunktion (Viewer-Funktion) besitzen, um die Bildbetrachtung und –beurteilung zu verbessern.

Schließlich können die Anforderungszeiten für medizinisches Bildmaterial minimiert werden (da ja der Verwaltungs- und Organisationsaufwand teilweise entfällt), wodurch eine nicht unbeträchtliche Zeitersparnis erzielt wird, die dem Patienten zugute kommt. Auch kann die Kommunikation zwischen Fachkollegen und Spezialisten effektiver gestaltet werden, da nun beide Seiten quasi-gleichzeitig auf dieselben Informationen zugreifen können. Dabei muß die Informationsübermittlung hinreichend gegen Mißbrauch und Manipulation durch Fremde geschützt sein.
Ein gutes RIS muß die schnelle, sichere und qualitativ hochwertige Übermittlung der Daten gewährleisten.

Diese Arbeit stellt ein RIS vor, daß all die o.g. Möglichkeiten eröffnet:

- Das System bietet eine effektive und erweiterbare Kurzeitspeicherung in relationalen und auch objektorientierten Datenbanken an und sichert platzsparend eine Archivierungszeit von bis zu 20 Jahren für die Daten.
- Durch Standard-Internet/Intranet-Technologie ermöglicht dieses System die kostengünstige, plattformunabhängige und erweiterbare Vernetzung von Rechnern in Krankenhäusern und den Praxen niedergelassener Ärzte zu einem RIS mit Zugang über Internet.
- Der herstellerunabhängige DICOM-Standard erlaubt es, bildgebende Geräte unterschiedlicher Hersteller ins System einzubinden. Durch plattformunabhängige Java-Software kann jeder Computer mit einem handelsüblichen Internet-Browser als leistungsfähige Viewer-Station eingesetzt werden. Es ist also beliebig, welche Art Ausgabe von welchem Gerät erzeugt und auf welcher Art Computer angezeigt wird.
- Das System enthält eine kleine, aber leistungsstarke Viewer-Software, die alle gängigen Funktionen zur Bildbetrachtung bietet.
- Durch neuartige Prinzipien bei der Bildkomprimierung werden die Übertragungszeiten für medizinisches Bildmaterial verkürzt. Die relevanten Daten können dadurch in knapp 10 Sekunden aus dem Archiv geladen werden. Das System ist gegen Datenmißbrauch abgesichert, da die Daten ausschließlich kodiert übertragen werden.

Die vorliegende Arbeit betrachtet und entwirft die Konzeption, die Algorithmen und die Software dieses, auf der Internet-Technologie basierenden, PACS und führt in seine Realisierung ein.

Um ein System mit diesen Anforderungen zu entwickeln, mußten mehrere Neuentwicklungen vollzogen werden :

- Entwicklung eines Konzeptes für den Aufbau eines Web-orientierten und Java-basierten (und damit plattformunabhängigen) RIS;
- Entwicklung eines neuartigen Datenmodells für relationale DICOM-Datenbanken, das gemäß den Anforderungen und Vorstellungen von praktisch tätigen Ärzten modelliert wurde;
- Verwendung des DICOM-Standards zur Einbindung diverser Bildaufnahmegeräte unterschiedlicher Hersteller;

- Entwicklung neuartiger Verfahren zur Bildinhaltsbeurteilung für medizinische Bilder zur effektiveren Auswahl geeigneter, Web-orientierter, verlustfreier, adaptiver Bildkompressionsverfahren;
- Entwicklung der Kommunikationsszenarien für den Datenaustausch zwischen Klinik-Server/DICOM-Archiv und Internet-Clients.

Die Lösung für diese Aufgaben zu erarbeiten, ist der Kernbereich dieser Arbeit gewesen.

2.Überblick über bekannte Visualisierungs-, Bildbearbeitungs-, Archivierungs- und Telekommunikationssysteme für die Radiologie

Heute arbeiten viele Firmen in dem Bereich der Herstellung von Hardware- und Softwaresystemen zur Visualisierung, Bearbeitung, Archivierung und Übertragung von medizinischem Bildmaterial. Dieses Kapitel ist der Betrachtung von bekannten telemedizinischen Systemen von unterschiedlichen Herstellern radiologischer Geräte gewidmet. Einige der bekannteren telemedizinischen Systeme und Software sind :

- Advantage Windows Workstation und Medical Systems PACS (General Electric - GE) [1-3]
- Tomografic Imaging und IVS-DICOM-Server (IVS- Software Engineering, Chemnitz) [4]
- MEDORA-PACS Viewing-Station (Innomed GmbH) [5]
- Kamedin – Telekommunikationssystem (ZGDV, Lübeck) [6-8]
- Magic View Workstation (Siemens) [9]
- Osiris – Software (Universitätskrankenhaus Genf) [10-12]
- PACS (Institut für Röntgendiagnostik, Klinikum Technische Universität München) [13]
- CHILI (DKFZ, Heidelberg) [17-19]
- DACHS - Database and Communications in Health Systems für radiologische Bilder (RTB-Bayern, München) [20-21]
- Bilddatenbank (DOIA- Dermatology Online Image Atlas) [22]
- Hipax DICOM Viewer (Steinhart Medizinsysteme und elektronische Datenverarbeitung, Glottertal) [23]
- Interactive System for Image Selektion (Universitätskrankenhaus Genf) [24-25]

2.1 Die GE Advantage Windows Workstation

Die GE Advantage Windows Workstation stellt ein sehr gutes Beispiel für die Gestaltung eines aktuellen digitalen Bildarbeitsplatzes für den Radiologen dar. Sie stellt alle im Alltag gebräuchlichen und notwendigen Funktionen einer Bildverwaltungs- und Bildbearbeitungs-station in Form einer übersichtlichen, praktisch gestalteten und einfach zu bedienenden Benutzeroberfläche zur Verfügung. Darüberhinaus bietet sie vielfältige 3-D-Anwendungen (z.B. Volumendatensätze, Oberflächenrekonstruktionen, Maximum Intensity Projektion, Fläche-Distanz-Messungen) und kleine, im Alltag nützliche Hilfen an (z.B. Vergrößerungs-glas, Filmformatwahl mit Drag-and-Drop-Funktion), verzichtet aber auf unnötige, verwirrende Funktionen. Hervorzuheben sind ihre Möglichkeiten als Kommunikations-Workstation. Durch Einbindung des DICOM-3-Protokolls sowie des DICOM-3-Datei–Formats kann sie direkt oder mittels Umsetzung durch ein Konvertierungssystem fast alle netzwerkfähigen radiologischen Geräte in ihrem Managementsystem integrieren. Da sie auf einer Standard-UNIX-Plattform installiert ist, können weitere UNIX-Programme sowie Hardware-Peripherie mit Standard-Interfaces (z.B. CD-ROM-Laufwerk/-brenner, ISDN–Karte, Röntgen-filmscanner) implementiert werden. Die GE Medical Systems PACS besteht aus der Advantage Windows Workstation, RAID mit verlustfrei Kompression 2:1, einer HL-7 Schnittstelle zu einem

kommerziellen Krankenhaus- und Radiologie- Informationssystem (KIS, RIS) , sowie dem WEB-Server und dem Archiv auf der Basis einer CD-Jukebox mit CD-Recorder.

2.2 TIM (Tomografic Imaging)

TIM ist ein Softwarepaket (Programmiersprache : Delphi 3) zur 3-dimensionalen Bildverarbeitung von Volumendaten. Das Hauptanwendungsgebiet ist die Arbeit mit Daten von CT und MRT, doch können generell alle Arten von Schichtbilddaten verarbeitet werden. Das Softwarepaket zeichnet sich durch geringe Hardwareansprüche und hohe Geschwindigkeit aus. Dadurch ist ein effektives und ökonomisches Arbeiten gewährleistet. Die Benutzeroberfläche wurde speziell für den Einsatz in der Medizin entwickelt und besitzt eine automatische kontextabhängige Benutzerführung.

Zur Integration der Software TIM in die bereits vorhandene Umgebung der bildgebenden Systeme wurden umfangreiche Applikationen zur Datenkonvertierung und Aufbereitung entwickelt. Damit ist eine Nutzung der verschiedensten Transfermedien, wie z.B. Diskette, magneto-optische Platte oder Netzwerkanbindung, gewährleistet. Der IVS-DICOM-Server gestattet es, beliebige bildgebende Systeme durch die TIM-Software anzusprechen.

Durch die Entwicklung des DICOM-Servers als 32-bit-Applikation, die sowohl unter Windows NT als Background-Service ausgeführt werden kann, ist kein zusätzlicher Server-PC erforderlich. Damit werden die Kosten der Netzwerkanbindung deutlich reduziert. Eine Dienstleistung der IVS stellt die Konvertierung medizinischer Bilddaten in beliebige Standard-Grafikformate (z.B. TIFF) für Präsentationen oder Dokumentation dar. Somit besteht die Möglichkeit der maßstabgetreuen Bildüberlagerung zur erweiterten Diagnostik.

2.3 MEDORA-PACS

Das Produkt MEDORA-PACS Viewing-Station (MPVS) ermöglicht es, medizinische Bilddaten zu visualisieren, zu verteilen und zu bearbeiten. Die MPVS-Software setzt auf dem Betriebssystem Windows NT auf und kann mit den gängigen PC-Softwareprodukten wie Textverarbeitung, Präsentationsprogramme und Multimediadarstellung eingesetzt werden. Die MPVS-Software erlaubt eine Bildverteilung innerhalb der Klinik bis zum Arbeitsplatz des Arztes und sogar bis zum Heimarbeitsplatz und führt somit zu einer effizienteren Nutzung der vorhandenen Ressourcen und einer schnelleren Diagnosefindung. Die MPVS-Software ist für den Einsatz im Netzwerk konzipiert und unterstützt den DICOM-3-Standard für den Austausch von Bild- und Textdaten mit anderen DICOM-fähigen Systemen verschiedenster Hersteller. Durch die vollständige Unterstützung des DICOM-Standards kann die MPVS-Software sehr flexibel in einem Netzwerk mit medizinischen Bilddaten eingesetzt werden. Darüber hinaus unterstützt sie den Bilddatenaustausch in einem WAN (wide area network) über ISDN-Leitungen. Die MPVS-Software unterstützt die Darstellung von Bildern verschiedenster Modalitäten in einer integrierenden Darstellung mit einer graphischen Benutzeroberfläche. Selbst auf einem einfachen Arbeitsplatzrechner bietet die MPVS-Software mehr Funktionalitäten als ein einfaches Bilddarstellungsprogramm. Zusammen mit der Funktionalität zur lokalen Abspeicherung von Bildern bietet die MPVS-Software die derzeit beste Lösung zur Darstellung von medizinischen Bilddaten im Bereich Arbeits-platzcomputer.

Neben der Funktionalität zum Betrachten von Bildern bietet Innomed ebenfalls Softwareprodukte in der MEDORA-PACS-Serie zur Digitalisierung von Bilddaten von Geräten mit analogem Ausgang. Dieses Produkt wird Image Capture Station (MPICS) genannt. Da die Visualisierungs-, Verarbeitungs- und Kommunikationsfunktionen der MPICS-Software auf der MPS-Software basieren, werden diese als Erweiterung der MPVS-Basissoftware behandelt.

Obwohl die Funktionalität der MPVS-Software einen großen Bereich der typischen Anforderungen aus der diagnostischen Bildbeurteilung erfüllt, ist sie immer noch modular aufgebaut. So kann die Software für verschiedene Hardwareressourcen und Applikationsanforderungen konfiguriert werden. Der betrachtete Bereich erstreckt sich von einer simplen Bildbetrachtungsworkstation bis hin zu einer hochleistungsfähigen Bildbearbeitungsworkstation mit diagnostischer Qualität.

2.4 Das Projekt KAMEDIN

Das Projekt KAMEDIN ist ein Verbundprojekt im Bereich der Teleradiologie. Im Rahmen des KAMEDIN-Projektes werden neue Methoden zum kooperativen Arbeiten und zur ferndiagnostischen Unterstützung über ISDN-Netze entwickelt. Durch die Technik des kooperativen Arbeitens wird es möglich, daß alle lokal ausgeführten Bildverarbeitungsoperationen wie z.B. Zoom, Fensterung usw. während einer kooperativen Sitzung quasiparallel und synchronisiert auf den Monitoren der beteiligten, über ISDN verbundenen Workstations ablaufen. Bei der Implementierung wurden insbesondere Aspekte der anwenderorientierten Benutzerschnittstelle, der an radiologischen Arbeitsplätzen benötigten Funktionalität, sowie der problemspezifischen Aufbereitung und Visualisierung medizinischer Bilddaten berücksichtigt.

Mit Hilfe der Funktionen der entwickelte Bildverarbeitungssoftware können unter Einbindung eines externen Superrechners die Möglichkeiten zur automatischen Segmentierung und Klassifikation von medizinischen Bilddaten unter Verwendung künstlicher neuronaler Netze untersucht werden. Die auf Basis repräsentativer Falldatensammlungen erzielten Segmentierungs- und Klassifikationsergebnisse können nach der Analyse zur ferndiagnostischen Unterstützung über Netze abgerufen werden. Neben einer Workstation Version, die unter dem Betriebssystem UNIX arbeitet, ist im Jahr 1995 eine PC-Version des KAMEDIN-Systems unter Windows NT realisiert worden. Im Rahmen eines Feldversuches wurde das Telemedizinsystem KAMEDIN in 15 medizinischen Einrichtungen in ganz Deutschland installiert und praktisch eingesetzt

2.5 SIENET Magic Viewer Workstation

SIENET Magic Viewer Workstation ist ein UNIX-basiertes Radiologie-Kommunikationssystem. Die SIENET Workstation kommuniziert mittels DICOM-Protokol mit beliebigen bildgebenden Geräten von Siemens und bietet die Möglichkeit des Visualisierens, Bearbeitens und Weitersendens dieser DICOM-Bilder über das TCP/IP Protokoll. Die PC Magic View-Software ermöglicht das Betrachten und Bearbeiten digitaler Bilder auf PCs. Sie kann als vollwertiger dezentraler Arbeitsplatz in einem Krankenhaus (OP, Ambulanz, . . .), aber auch im Rahmen eines Teleradiologiesystem unter

Windows 95 oder Windows NT eingesetzt werden. Neben einem integriertem DICOM-Interface und den erforderlichen Query- und Retrieve-Funktionen, gibt es eine Fülle von Möglichkeiten zur Bildpräsentationen, Bildbearbeitung (Bildausschnitt verschieben, Invertierung, Spiegelung, Drehung, Bildveränderungen rückgängig machen, Bildvergrößerungen und –verkleinerungen, Lupenfunktion, Grauwertefenster-Einstellung), systemseitige Funktionen (Menüsteuerung und Funktionsleisten, hierarchische Zugriff auf Funktionen für Personengruppen, Druckfunktion), Bilddokumentation und für administrative Funktionen (Organisation der Bilder in logischen Gruppen (Folder), Zugriff auf lokale Datenbank, Abfrage des zentralen Speicher- und Archivierungssystems nach Patienten über DICOM Abruf von Bildern aus dem zentralen Speicher- und Archivierungssystem).

2.6 Das PACS vom Klinikum der Technischen Universität München

Das PACS beinhaltet: Archivrechner, Archiv, RAID, Befundungskonsole (Hard- und Software), Viewingsoftware (auf Standardhardware), Demonstrationseinheit (Hard- und Software) sowie den Intranetserver. Das PACS ermöglicht so ein Vorladen von Bildern aus dem Langzeitarchiv in das RAID, Laden der Bilder aus dem Archiv an den Arbeitsplatz / die Befundungskonsole, automatische Verteilung der neuen Bilder aufgrund der Modalität, Sicherheit und des Datenschutzes und die Identifikation des Benutzers. Das Archiv des PACS besteht aus dem Archivrechner (Doppelrechnersystem), RAID: 600 - 1000 GB für kurze Zeit (1 bis 3 Monate) speichert, Langzeitarchiv (permanente Speichermedien). Der Inhalt des Archivs setzt sich aus Verwaltungsdaten, Bildern sowie allen Befunden (im RAID) zusammen. Das dabei verwendete Datenformat ist standardmäßig ein internes Format, jedoch kann das DICOM-Format alternativ verwendet werden; neben einem Export-Format für DICOMs besteht auch eine Exportmöglichkeit im TIFF- und JPEG-Format (ggf. MPEG, Wavelet-Kompression).

2.7 CHILI

CHILI ist Software für den Gebrauch in der radiologischen Praxis, die neben den klassischen Funktionen einer reinen Viewing-Station auch auf die Teleradiologie spezialisiert ist. Die Aufgaben der Systementwicklung beruht auf den zweijährigen Erfahrungen des erfolgreichen Teleradiologie-Projekts MEDICUS-2 [14-16]. Ein kompletter Neuanfang in Konzept und Programmierung erlaubte es, alle gemachten Erfahrungen mit dem System zu verwerten.

In CHILI können radiologische Bilder aus verschiedenen Quellen integriert werden, wie etwa CT, MRT, Videokamera, Ultraschallgerät oder Röntgenfilm-Scanner. Der Transfer der Bilddaten von den bildgebenden Geräten (Modalitäten) erfolgt weitgehend automatisch. In der Regel besteht die Übertragung der Daten aus einem Export-Befehl auf der bildgebenden Modalität. Der Transferprozeß basiert idealerweise auf dem radiologischen Kommunikationsstandard DICOM und auch auf der Basis der Kommunikationsprotokolle DECnet oder TCP/IP. Die Datenbank basiert auf dem SQL-Standard, unterstützt aber auch ODBC- und JDBC-Datenbanken. Bei der Speicherung werden alle personenbezogenen Daten mit einem symmetrischen Verschlüsselungsverfahren geschützt. Komplette Untersuchungen können mit drei Mausklicks an eine andere CHILI-Workstation oder per DICOM an Viewing-Stations

anderer Hersteller geschickt werden. Der Bildversand erfolgt über lokale Netze, das Internet oder per ISDN. Um Bilddaten zu verschicken, wählt der Absender zunächst die radiologische Untersuchung über die Studien-Nummer oder den Patientennamen aus, wählt den Adressaten und aktiviert den Versand.

CHILI ist modular aufgebaut. Dadurch kann die Software den individuellen Anforderungen und Wünschen der Kunden angepaßt werden. Die Heidelberger Firma hat inzwischen über 20 Software-Module im Angebot, die zu verschiedenen Paketen zusammengestellt werden können. Der sog. Plug-In-Mechanismus erlaubt es, zusätzliche Software-Komponenten nahtlos in das System zu integrieren. Mit dem Software Developers Kit (SDK) kann jeder, der technisch dazu in der Lage ist, eigene Zusatzfunktionen schreiben. Der Kunde profitiert von diesem System, da er nur noch ein Zusatzmodul zu kaufen braucht und nicht noch eine zusätzliche Workstation mit spezieller Auswertungssoftware erwerben muß - oder der Kunde schreibt sich seine Auswertungssoftware mit Hilfe des SDK sogar selbst.

Die entwickelte Software stellt die Funktionalität einer (preiswerten) Viewing-Station zur Verfügung. Mehrere Workstations in einem Krankenhaus können einen gemeinsamen Datenpool haben und somit ein preiswertes Mini-PACS realisieren. Dabei wird in der Regel eine UNIX-Workstation von Silicon Graphics als zentraler Server benutzt, der den Datentransfer mit den bildgebenden Geräten übernimmt, die Bilddaten in einer zentralen Datenbank speichert, druckt und auch die Versandfunktionen an andere Systeme erledigt. An den zentralen UNIX-Server, der übrigens kaum teurer als ein entsprechend ausgerüsteter PC ist, lassen sich sog. Viewer auf PCs mit Windows-NT anschließen, die die Datenbank des zentralen Servers abfragen können und alle Funktionen zur Bilddarstellung und Befundung bereitstellen. Es lassen sich (fast) beliebig viele Viewer im lokalen Netz anschließen.

2.8 DACHS

DACHS - Database And Communication in Health Systems für radiologische Bilder. Moderne Bild- und Signalumgebung wie Kernspintomographie (MRT), Positronenemissions-tomographie (PET) und Elektroencephalographie (EEG) sind heute eine wichtige Grundlage neurologischer Diagnostik und Therapieverlaufskontrolle. Der Neurologe kann sich durch das Zusammenspiel dieser Methoden ein komplettes Bild von der Anatomie (MRT), dem Stoffwechsel (PET) und der Physiologie (EEG) im Gehirn des Patienten oder Probanden verschaffen. Im Rahmen des Projekts wurde der Prototyp eines Neurologenarbeitsplatzes geschaffen, der den Online-Zugriff auf die dezentral beim Erzeuger gespeicherten Daten sowie deren Integration und Bearbeitung ermöglicht. Dies geschieht durch die Bereitstellung der EEG-Signale, MR- und PET-Volumen auf jeweils einem eigenen Datenserver in der Neurologischen Klinik, dem Institut für Röntgendiagnostik und der Nuklearmedizinischen Klinik der TU München. Die DACHS-Anwendung basiert auf einer im Netz verteilten Datenbank. Sie wird über eine benutzerfreundliche WWW-Benutzeroberfläche bedient. Die Anwendung ist offen, d. h., es können jederzeit weitere Datenserver, z. B. für andere bildgebende Untersuchungsmethoden, in den Verbund mit aufgenommen werden.

2.9 DOIA

DOIA ist eine WEB-basierende verteilte Bilddatenbank, die über 50.000 digitalisierte Diapositive zum Thema Dermatologie enthält. DOIA soll als verteiltes Hypermedia-Lehrbuch der Dermatologie dienen. Diese Datenbank wurde an das World Wide Web (WWW) angebunden und steht seit Herbst 1994 online zur Verfügung. Der WWW-Server befindet sich außerhalb des Medizinnetzes, durch eine "Firewall" vom Wissenschaftsnetz (WIN) getrennt.

Klinische und histologische Bilder werden ausgewählt, auf einem Scanner digitalisiert und auf einer Photo-CD gespeichert. Die aufgenommenen Bilder sind True-Color-Bilder (24bit) mit einer Auflösung von 768x512 Bildpunkten. Nach der Bildmanipulation werden die korrespondierenden Verkleinerungen auf einer magneto-optische Platte gespeichert und im JPEG-Format komprimiert (ca. 1:35) auf der Festplatte abgelegt.

Die Daten werden unter Verwendung eines Visual Basic front-ends in die Datenbank eingegeben danach wird die Bildbeschreibung in der Datenbank gespeichert. Die Bilder dagegen bleiben im Dateisystem, da MS Access, das als lokale Datenbank verwendet wird, einige Probleme mit dem Speichern von Bildern hatte. Darüber hinaus können Bilder in dieser Art und Weise durch die Datenbank und durch lokale HTML-Dokumente genutzt werden.

2.10 Der Hipax DICOM Viewer

Der Hipax DICOM Viewer stellt sich als einen einfach zu bedienenden Bildbetrachter innerhalb eines PACS-Projektes dar. Der Hipax DICOM Viewer ermöglicht den Zugriff auf ein DICOM-PACS-Archiv über die dafür notwendigen DICOM-Befehle (Find, Move). Außerdem besteht die Möglichkeit, daß die Radiologie-Abteilung Bilder direkt an die DICOM–Viewer schickt (Store).

Als weitere Möglichkeit kann man kleinere lokale Netze (beispielsweise abteilungs- oder stationsbezogen) gestalten, in denen die relevanten Bilder temporär abgelegt werden.
Die Möglichkeiten der DICOM-Kommuniation sind im Hipax-DICOM-Conformance - Statement beschrieben.

Das Hipax-Stationsnetz ermöglicht es, innerhalb eines Kliniknetztes abteilungs- oder stationsbezogen Bilder zu verwalten. So können beispielsweise auf einer Klinikstation die Bilder der Patienten abgelegt sein, die gerade stationär aufgenommen wurden. Dabei sendet die radiologische Abteilung die entsprechenden Bilder im DICOM an das Hipax-Stationsnetz. Dort werden die Bilder auf dem Bildserver abgelegt. Der Zugriff auf diese Daten innerhalb der Station ist sehr einfach über PCs möglich (Clients), die auf diesen Bildserver zugreifen. Der sehr einfach zu bedienenden Bild-Viewer lassen sich auch in ein Patientenver-waltungssystem integrieren.

Hipax bietet verschiedene Möglichkeiten, in ein Patientenverwaltungssystem integriert zu werden. Ziel ist es, einer beliebigen anderen Applikation die Möglichkeit zu geben, die wesentlichen Funktionen von Hipax selbst zu steuern. So kann beispielsweise realisiert werden, daß aus der elektronischen Karteikarte die Bilder des aktuellen Patienten angezeigt werden (einzelne Bilder, mehrere Bilder, Anzeigen der Bildübersicht).

Heute existieren zwei Möglichkeiten, Hipax zu integrieren. Die sehr einfach und schnell zu realisierende Möglichkeit verwendet Ini-Dateien (bzw. Windows-Botschaften) für die Kommunikation. Beim Start von Hipax wird eine Ini-Datei als Parameter übergeben. In dieser Ini-Datei sind dann weitere Kommandos enthalten (Patientendaten, Bilder usw.). Zusätzlich kann Hipax über Windows-Botschaften aufgefordert werden, die Ini-Datei erneut auszuwerten, so daß zum Anzeigen der Bilder Hipax nicht jedesmal neu gestartet werden muß.

2.11 ISIS (Interactive System for Image Selektion)

ISIS stellt ein PACS dar, das als krankenhausweites Bild-Management-System für radiologische und nichtradiologische Bilder dienen soll. Es basiert auf einer offenen Architektur und auf einer Menge weitverbreiterter Quasi-Industriestandards (UNIX, TCP/IP, SQL). Das System ist unmittelbarer Bestandteil des Krankenhausinformationssystem DIOGENE. Es basiert auf einer verteilten Architektur von Archiv-Server-Clustern, die mit großen optischen Plattenbibliotheken, und über das Krankenhaus verteilten Display-Servern ausgestattet sind.

Die PACS-Datenbank selbst ist eine eigenständige Einheit, die nur diejenigen Daten verwaltet, die sich auf die Bilder und radiologischen Untersuchungen beziehen. Um mit dem restlichen Krankenhausinformationssystem kompatibel zu sein, wurde die PACS-Datenbank auch mit dem relationalen Datenbank-Management-System (INGRES) entwickelt. Die PACS-Datenbank selbst enthält nur eine minimale Datenuntermenge, die die Bilddateien beschreibt. Sie besteht aus mehreren relationalen „Tabellen“, in denen bildbezogene Daten gespeichert werden. Der Userzugriff auf die PACS-Datenbank erfolgt über eine generische Applikation, die ISIS genannt wird und die wie ein Browser und Transferoperator für Bildfiles wirkt. Der Anwender hat Zugriff auf die vollständige Liste radiologischer Untersuchungen eines gegebenen Patienten. Das Bildformat basiert auf dem DICOM-3.0 Standard.

Als User-Interface zur Bildanzeige und –manipulation wird OSIRIS (verfügbarfür UNIX, X-Windows-, Macintosh-, Windowsoberfläche) verwendet. OSIRIS wurde in C++ geschrieben und ist eine allgemeine Manipulations- und Analysesoftware für medizinische Daten. OSIRIS ermöglicht folgende Funktionen:

- Anzeige der Bildmenge: Möglichkeit der Veränderung des vordefinierten Layouts durch Auswahl der Anzahl von Zeilen und Spalten;
- Region of Interest (ROI): Erweiterung auf Volumen-ROI, automatisches Auszeichnen (farbiges Überdecken) der markierten ROI’s;
- Bilddarstellung: Zoom, Verbesserungen der Einstellungen von Kontrast und Helligkeit;
- Segmentieren, Mehrebenenrekonstruktion, Farbmaske, Polygonglättung, Präferenzen, Filterverarbeitung.

Zum Vergleich der oben beschriebenen Systeme und Software sind in der Tabelle (Anhang A) ihre wichtige Parameter zusammengestellt.

2.12 Schlußfolgerungen und die Ziele der vorliegende Dissertation

Die oben beschriebenen Systeme und Software geben einen knappen Überblick über einen Teil der heute existierenden KIS, RIS, PACS und deren Elemente. Aufgrund der Betrachtung der oben beschriebenen Systeme und Software bemerkt man, daß die momentan erhältlichen Systeme einige gemeinsame Nachteile haben. Erstens, sind Sie für Internet-Nutzer nicht zugänglich. Zweitens kann man diese Systeme „Krankenhaus-Hardware-orientiert“ nennen. D. h., damit PACS-Software vom Klinikum 1 im Klinikum 2 angewandt werden kann, muß man diese Software zuerst komplett umarbeiten. Drittens unterstützt die Software dieser Systeme nur jeweils ein Betriebssystem; allerdings ist die Ausgangssituation in Bezug auf die Geräteausstattung an den einzelnen Krankenhäusern sowie in den radiologischen Praxen sehr unterschiedlich. Im allgemeinen existiert in den Krankenhäusern in Deutschland eine stark heterogene Hardwarelandschaft von verschiedenen Computersystemen (PCs, Apple Macintosh Rechner und vereinzelt auch Workstations). Auch die verschiedenen bildgebenden Geräte in einem Krankenhaus stammen im Normalfall nicht von nur einem Hersteller. Der hohe Anteil an älteren, analogen Untersuchungsgeräten im Gerätepark der Krankenhäuser hemmt die Einführung eines digitalen Netzwerkes genauso stark, wie es die vielen unterschiedliche Gerätetypen tun; in diesem Zusammenhang ist insbesondere die Problematik von Netzwerkanschluß und Datenlesbarkeit zu nennen. Sich auf einen Hersteller festzulegen, ist keine wirkliche Lösung, denn das System wird dadurch teuer und ist nach außen praktisch nicht mehr offen, da es keine Kommunikation zu anderen Systemen unterstützt und die rasante Weiterentwicklung auf dem Computer- und Softwaremarkt nicht angemessen schnell nachvollzieht.

Damit ein Maximum an Plattformunabhängigkeit und ein Minimum an Wartungsaufwand beim schon beim Entwurf von KIS, RIS, PACS erreicht werden kann, ist es notwendig, die Benutzerschnittstelle und Software der Systeme in einer plattformunabhängigen, intranet/internet-orientierten Netzprogrammiersprache Java zu realisieren . Eine andere Grundvoraussetzung ist es, daß die verschiedenen Komponenten der Informationssysteme ein herstellerunabhängiges Bildformat unterstützen. Seit geraumer Zeit hat der herstellerunabhängige DICOM-Standard (Digital Image and Communication in Medicine) als Bildformat und Kommunikationsprotokoll große Bedeutung gewonnen. Er ermöglicht die Kommunikation von Komponenten verschiedener Hersteller. Mit der Einführung des DICOM-Standards wurde die Basis für eine einfache Verbindung und problemlosen Datenaustausch zwischen medizinischen Geräten gelegt. Dementsprechend ist das erste Ziel dieser Arbeit: ***Die Erarbeitung der Konzeption, Struktur und Software eines Intranet/-Internet–orientierten, JAVA-DICOM-basierten radiologischen Informations-systems zur Daten- und Bildübertragung und zur Archivierung zum Gebrauch in Kliniken und in Praxen niedergelassener Ärzte.***

Die projektierten radiologischen Informationssysteme für die Bildarchivierung müssen eine verteilte Bilddatenbank und ein Archiv beinhalten. Die verteilte Bilddatenbank soll die Möglichkeit offerieren (und zwar soll die Bilddatenbank auf heterogener Hardware im Klinikumfeld und unter verschiedenen gängigen Betriebsystemen laufen) relationale Datenbanken (Oracle, Sybase, Access) und andere Datenbanken einzubinden. Daraus resultierend ergibt sich die zweite Aufgabe dieser Arbeit : ***Die Entwicklung eines verteilten objekt-orientierten Datenbanksystems zur Speicherung und Verwaltung der DICOM-Daten.***

Das große Problem internet-basierter medizinischer Informationssysteme besteht in der sicheren und schnellen Übertragung medizinischer Bilder über das Netzwerk. Das zweite Problem ist die effektive Archivierung großer Datenmengen (Bilder). Eine mögliche Lösung beider Problemen besteht in der Kompression der medizinischen Bilder. Dabei ist es auf der einen Seite wichtig, die Komprimierung mit einem großen Koeffizienten zu realisieren und auf der anderen Seite darf der Verlust an Information für den Radiologen visuell nicht bemerkbar sein. Diese positiven Ergebnisse kann man bei Benutzung adaptiver Komplexitätsalgorithmen erreichen, die aus verschiedenen verlustfreien und verlustbehafteten Kompressionsalgorithmen bestehen. Aus der Sicht des Autors ergibt sich daraus eine dritte Aufgabe: ***Die Erarbeitung internet-orientierter, adaptiver Bildkompressionsverfahren.***

Die zentrale Rolle in der radiologischen Diagnostik oder der klinischen Präsentation medizinischer Bilder spielt die Bildbearbeitungs- und Visualisierungssoftware. Visualisierungssysteme zur radiologischen Befundung haben bereits einen hohen Standard erreicht. In einem Krankenhaus, das normalerweise eine heterogene Hardware hat, wird in einem ersten Schritt einfache, plattformunabhängige Visualisierungs- und Bildbearbeitungssoftware auf den Arztworkstations oder PCs eingesetzt, die eine einfache radiologische Befundung erlaubt. Diese Software muß beliebige Bilddaten nach dem DICOM 3.0 und ACR/NEMA 2.0 Standard interpretieren können und soll auf jedem Arztarbeitsplatzrechner innerhalb des KIS, RIS oder PACS unabhängig von der Betriebsplattform laufen. Die Anwendung der plattform- und betriebsystemunabhängigen Java-Technologie ermöglicht es, keine spezielle Softwareinstallation auf dem Rechner der beteiligten Kommunikationspartner machen zu müssen. Der Datenfluß wird mittels einer Intranet/-Internetkommunikation in einem Internetbrowser gewährleistet. Dies ermöglicht die freie Wahl eines Kommunikationspartners ohne vorinstallierte, lokale Programmressourcen.
In diesem Zusammenhang heißt die vierte Aufgabe dieser Arbeit ***die Konzeption und Entwicklung eines plattformunabhängigen Java-basierten DICOM-Viewer und die Implementation intranet/internet-basierten Kommunikationsszenerien von DICOM-Daten zwischen dem Client, Server und DICOM-Archiv.***

3. Entwicklung eines Konzeptes für ein Intranet/Internet-orientiertes, DICOM-basiertes radiologisches Informationssystems

3.1 Einleitung

Heutzutage verwenden die meisten Krankenhäuser neben bewährten, traditionellen Untersuchungsmethoden, hochmoderne digitale Untersuchungsgeräte (CT, MRT, DR, etc.) in ihren Radiologieabteilungen. Normalerweise werden in den Krankenhäusern pro Tag hunderte unterschiedliche radiologische Bilder erzeugt, die so schnell wie möglich bearbeitet werden müssen.

Eine große Zahl der Hersteller von Bildgebergeräten liefert eigene PACS oder einzelne Komponenten solcher Systeme (Workstation, leistungsfähige PC, CD-Jukeboxes, CD-Recorder, etc.), mit entsprechender Software. Die momentan erhältlichen radiologischen Informationssysteme haben eine Reihe von ernsten technischen und politisch-strategischen Nachteilen, die eine Einführung und Nutzung der Telemedizin in der Radiologie stark bremst [26]. Diese sind

- Verwendung proprietärer Bilddatendarstellungs-, Bildübertragungs- und Bildarchi-vierungsstandards in den PACS.
- Damit verbunden ist eine vollkommene Abhängigkeit vom Hersteller. Das PACS kann nicht mit den radiologischen Geräten von anderen Firmen betrieben werden.

Als Grundlage für eine effiziente Bildübertragung, -bearbeitung und -archivierung im Krankenhaus ist ein leistungsfähiges Netzwerk und ein Archivierungssystem notwendig. Dieses Netzwerk soll sowohl für den Datenaustausch innerhalb des Krankenhauses als auch für weltweite Verbindungen über das Internet genutzt werden. Somit ist ein Datentransfer von und hin zu allen Einrichtungen über Internet-Verbindung möglich. Diese Strategie ist ein wesentliches Merkmal der weiteren Entwicklung heutiger Systeme zur Archivierung und Übertragung radiologischer Bilder (PACS) im Krankenhaus. Ein PACS ist ein vernetztes System, in dem bildgebende Geräte, eine Datenbank mit angeschlossenem Archiv sowie eine Vielzahl von Arbeitsplatzrechnern miteinander kommunizieren. Eine Grundvoraussetzung dabei ist , daß die verschiedenen Komponenten alle die gleiche “Sprache” sprechen, d.h. alle Komponenten unterstützen einheitliche Archivierungs- und Datenübertragungsstandards. Dieses, in dieser Hinsicht, nächste wesentliche Merkmal der weiteren Entwicklung der PACS ist die Unterstützung des DICOM-Standards. Dieser Standard ermöglicht die problemlose Kommunikation der PACS-Komponenten und der bildgebenden Geräte sowie die einfache Visualisierung von verschiedenen Modalitäten der Bilddaten unterschiedlicher Geräteher-steller. Daraus ergeben sich folgende, wünschenswerte Anforderungen an ein zu projektie-rendes PACS [27,51]:

- *Standardisierung:* Anwendung von in der Medizin gebräuchlichen Bildformaten, Bildübertragungs- und Archivierungsstandards (DICOM-Standard);
- *Bedienerfreundlichkeit*: Schneller, einfacher und autorisierbarer Zugriff auf alle Informationen;

- *Sicherheit*: Sichere und geschützte Speicherung und Übermittlung von DICOM-Bildern und –Daten;
- *Flexibilität*: Unabhängigkeit des PACS von der krankenhausinternen EDV-Infrastruktur, den Rechnern und Betriebsystemen;
- *Erweiterbarkeit*: Möglichkeit zur Integration neuer Hardware und von alten oder fremden Datenbeständen in das PACS;
- *Wartungsfreundlichkeitkeit*: Einfache und offene Schnittstellen zwischen den Bildgeber-geräten, dem PACS und den Arbeitsplatzrechnern ermöglichen einfache Wartung;
- *INTERNET-basiert*: Schneller, einfacher und autorisierbarer Zugriff auf Informationen aus dem PACS-Archiv über das World Wide Web.

Das Ziel dieses Kapitels ist die Entwicklung eines kostengünstigen DICOM-basierten Informationssystems zur Übertragung und Archivierung von Daten in Kliniken und in den Praxen niedergelassener Ärzte, das den oben genannten Anforderungen gerecht wird. Dazu wird in dieser Arbeit ein Konzept eines Informationssystems zur Datenübertragung und -archivierung vorgeschlagen und analysiert, das auf dem DICOM-Standard, den modernen plattformunabhängigen JAVA-Programmen und der Intranet/Internet-Technologie basiert.

3.2 Der DICOM-Standard und Entwicklung eines Datenmodells für eine DICOM-Datenbank

In diesem Abschnitt wird der DICOM-Standard und verschiedene Datenbankkonzepte kurz beleuchtet. Es wird eine Analyse des DICOM-Informationsmodells und vorhandener DICOM-Datenbanklösungen durchgeführt und eine eigene Konzeption für den Aufbau einer DICOM-Datenbank vorgeschlagen.

3.2.1 Der DICOM-Standard

Auf dem Weg in die Informationsgesellschaft stehen Krankenhäuser vor der Aufgabe, das täglich anfallende umfangreiche Bildmaterial, das mit digitalen Modalitäten aufgenommen wurde, auch digital zu speichern. Als einheitlicher Standard für das Bildformat hat sich in den vergangenen Jahren der, in den USA ursprünglich vom American College of Radiology entwickelte, DICOM-Standard [29,30] etabliert. Derzeitig gibt es ein unabhängiges DICOM- Komitee, in dem 15 Arbeitsgruppen an der Weiterentwicklung des Standards (Integration der sog. Supplements) arbeiten. Alle führenden Hersteller radiologischer Modalitäten haben ihre neuen Geräte mit DICOM-Schnittstellen ausgerüstet. Ihre DICOM-Konformität muß in entsprechenden Conformance-Statements offengelegt werden, die selbst ein Teil des DICOM-Standards darstellen.

Für die Krankenhäuser ergibt sich damit die Möglichkeit, Geräte unterschiedlicher Hersteller in einem Netz zu integrieren. Der DICOM-Standard definiert aber nicht nur das Bildformat, sondern legt auch fest, wie DICOM-kompatible Geräte miteinander über elektronische Netzwerke kommunizieren. DICOM unterstützt derzeit eine ganze Reihe radiologischer Modalitäten wie CT und MR, Ultraschall, Computerradiographie, Röntgen-Angiographie, Röntgen-Radiofluoreskopie, digitalisierte Filme und Video.

Im DICOM-Standard werden zwei grundlegende Fähigkeiten zur Netzwerkkommunikation, die DICOM-Service-Klassen (Storage- und Query/ Retrieve-Service-Klassen), definiert. Bei der Implementierung der Storage-Service-Klasse kann eine sendende DICOM-Anwendung ein Bild an eine empfangende Anwendung versenden („push“). Natürlich können nicht nur Einzelbilder, sondern auch komplette radiologische Untersuchungen (Study) über ein Netzwerk übertragen werden. Der DICOM-Standard erlaubt einen Datenaustausch über elektronische Netzwerke oder andere physikalische Speichermedien. DICOM-Datenaustauschmechanismen basieren auf der Client-Server-Konzeption. Client und Server müssen in der Lage sein, Anforderungen an die Dienste der unteren Schichten (low level services) des Übertragungsprotokolls zu senden.

Die Low-Level-Services behandeln den Informationsaustausch und werden vor dem Applikationsbereich des Clients oder Servers verborgen. Der Partner, der einen Dienst anfordert, ist der „service user“, sein Gegenspieler ist der „service provider“. Der Service-Provider muß erkennen, in welchem Format die Information übertragen wurde und sie in die Darstellung umwandeln, die vom Applikationsbereich erwartet wird.

Die Beziehung zwischen beiden Partnern wird durch die Service-Klassen-Beschreibung definiert, bei der eine Einteilung in Service Class User (SCU bzw. Client) und Service Class Provider (SCP bzw. Server) vorgenommen wird. Dieses Wechselspiel kann im Krankenhausumfeld durch verschiedene Modalitäten im Umfeld eines PACS, oder über das Internet zwischen entfernteren Institutionen, implementiert werden.

Die Beschreibung der Information (Informationsobjektdefinitionen) und der Operationen (Services) bilden im Zusammenhang mit der Klassendefinition eine Service-Object-Pair-Klasse (SOP). Eine Service-Klasse beschreibt die Beziehungen von Informationen. SOP-Klassen werden durch die Partner ausgewählt, um eine von beiden Seiten akzeptierte Menge von Merkmalen zur Unterstützung der Interaktion zu definieren. Der verwendete Mechanis-mus hängt vom Typ des Informationsaustausches (Netzwerk oder Speichermedium) ab.

3.2.2 DICOM-Informationsmodell und -Bildformat

Das DICOM-Format ist TAG-basiert [30]. Das einzelne Datenelement (TAG) kann über ein Nummernpaar in Form von standardisierten Gruppen- und Elementnummern (sozusagen die Adresse) angesprochen werden. Die miteinander verwandten Datenelemente sind in Gruppen zusammengefaßt. So enthält z.B. die Gruppe 10 die Patientendaten oder die Gruppe 18 die Aufnahmedaten. Jede Gruppe besteht aus mehreren Elementen, welche die im Teil 3 des DICOM-Standards definierten Informationsobjekte repräsentieren.

Bevor DICOM-Datenobjekte über ein Netzwerk gesendet werden können, müssen die Informationen in geeigneter Weise in einem Datensatz arrangiert werden. Die Regeln für den Aufbau eines Datensatzes, sind im Teil 5 des DICOM-Standards definiert. Abbildung 3.1 zeigt den Struktur eines Datensatzes und eines Datenelements. Wie aus der Abbildung zu ersehen ist, besteht das Datenelement aus fünf Feldern. Das erste und zweite Feld entspricht der Gruppennummer und der Elementnummer. Das dritte Feld beinhaltet den Datentyp (er bestimmt das Format des Elementwertes). Das vierte Feld ist die Länge des

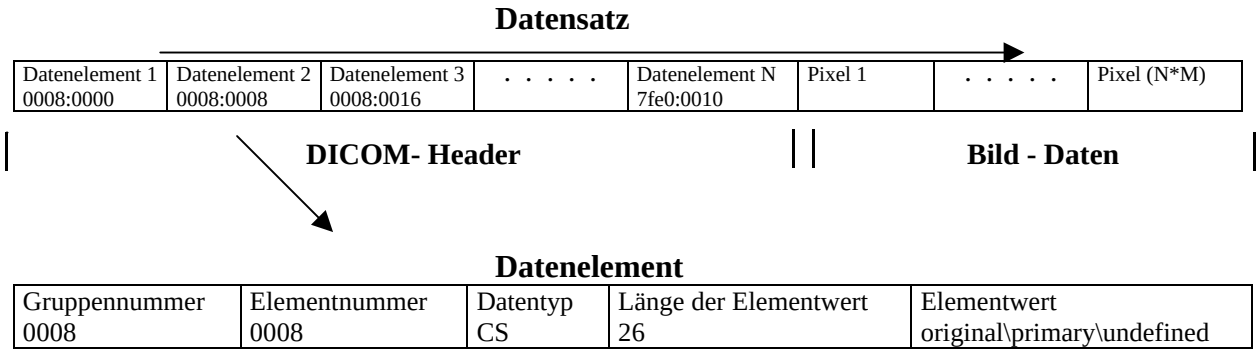


Abb. 3.1: Der Struktur eines Datensatzes und eines Datenelements

Elementwertes im Byte und das fünfte Feld beinhaltet den Wert des Datenelementes. Man unterschiedet zwei Teilstrukturen im Datenelement. Beide Teilstrukturen haben die gleiche Anzahl Felder. Der Unterschied besteht lediglich in der Länge dieser Felder.

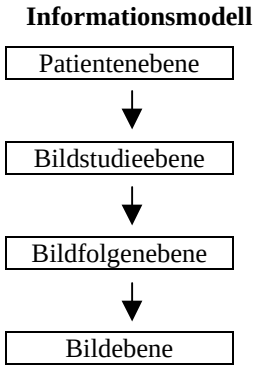
Struktur 1

Gruppennummer 2 Byte	Elementnummer 2 Byte	Datentyp 2 Byte	Länge des Elementwerts 2 Byte	Elementwert

Struktur 2

Gruppennummer 2 Byte	Elementnummer 2 Byte	Datentyp 2 Byte	Länge des Elementwerts 4 Byte	Elementwert

Als Beispiel ist im Anhang B der DICOM-Header eines realen MR-Bildes angegeben. Zur wichtigsten Gruppe von Informationen gehören die Patienten-, die Untersuchungs- und die Bilddaten. Attribute der Patientengruppe (PAT) sind der Name, die Identifikationsnummer (ID), das Geburtsdatum, das Geschlecht des Patienten etc. Im Informationsmodell unterschiedet man die Patientenebene, Bildstudienebene, Bildfolgenebene und Bildebene.



Die Patientenebene (0010-Gruppe) enthält die Identifikationsnummer und demographische Daten des zu einer Bildstudie (study) gehörenden Patienten. Die wichtigste Ebene repräsentiert die Ebene der Bildstudie. Eine Studie ist das Ergebnis einer Anforderung für einen bestimmten Untersuchungstyp. Diese Ebene kann ebenfalls Referenzen auf Informationen, die sich auf dieselbe Studie in einem Administrationssystem (Befunde etc.) beziehen, enthalten. Im allgemeinen kann eine Anforderung Untersuchungsprozeduren auf unterschiedlichen Modalitäten einschließen. Als Ergebnis entstehen Serien von einem oder mehreren Bildern (Bildfolgen). Alle Bilddaten werden dann zusammen mit derselben Bildstudie als Root (Wurzelverzeichnis) gespeichert. Zu einem Patienten können mehrere

Bildstudien als Ergebnis anderer (früherer) Anforderungen für eine Untersuchungsprozedur zugeordnet werden. Unterhalb der Ebene der Bildstudie im Informationsmodell werden alle Bildfolgen (series) eingeordnet. Die Ebene der Bildfolgen enthält den Modalitätstyp (Bildquelle), mit dem die Bilder erzeugt wurden, den Aufnahmezeitpunkt (date/time) und Einzelheiten über den Untersuchungstyp und verwendete Geräte. Die Art und Weise wie Bilder in Bildfolgen eingruppiert werden, hängt von ihrer klinischen Verwendung ab. Die unterste Ebene des Informationsmodells ist die Bildebene. Die beschreibenden Bildparameter sind in der Gruppe 0028 im DICOM-Header dokumentiert, die Bilddaten beginnen ab Tag 7FE0.

Werden Bilder in einem Archivsystem gespeichert, müssen einheitliche und abgestimmte Regeln der Informationsidentifikation für alle Systeme (Modalitäten, Speichersysteme, Arbeitsplatzrechner etc.) bestehen, die auch festlegen, wie Informationen zu behandeln sind. Als Schlüssel werden in DICOM UID's (Unique Identifier) verwendet. Jede Informationseinheit im Informationsmodell, außer der patientenbezogenen Informationseinheit, hat ihre eigene UID. Unter anderem werden für DICOM-Bildserver UID's zur Identifikation von Bildern bei der Übertragung zwischen DICOM-Modalitäten verwendet. Zur Identifikation von Patienten in Krankenhaus- bzw. Radiologie-informationssystemen oder in Systemen der elektronischen Krankenakte benutzt man die ID-Nummer des Patienten.

Die Ebene der Bildstudie spielt bei der Identifikation von Information eine besondere Rolle. Wird die Study Instance UID in konsistenter Weise von allen beteiligten Systemen

Bild-SOP-Instanz

SOP Class UID **SOP Instance UID**
Patients' Name Patient ID Patients' Birth Date Patient Sex
Study UID Study Date Study Time Study ID Referring Physician Accession Number
Series UID Series Number ***Modality*** Manufacturer Institution Name
Image UID Image Number Image Type ---------- **Bits Allocated, Bits Stored** **High Bit, Rows, Columns,** **Samples per Pixel** **Planar Configuration** **Pixel representation** **Photometric Interpretation** ---------- Pixel Data
Window Width Window Center

Abb.3.2: Gemeinsame Basismenge von Attributen für die Bild-SOP-Klasseninstanzen

verwendet, ist es nicht schwer, die Informationsteile mit den Bilddaten in der DICOM-SOP-Instanz zu verknüpfen. Am sinnvollsten würde es sein, wenn die Study-UID's durch das RIS oder das KIS vergeben werden würden.

In Abhängigkeit von der Modalität, welche die Bilddaten erzeugt, definiert der DICOM-Standard unterschiedliche Bild-SOP-Klassen. Die Bild-SOP-Klasseninstanzen haben eine gemeinsame Basismenge von Attributen (Abbildung 3.2).

Diese oben beschriebene Bildformat wird in jeder Bild-SOP-Klassendefinition verwendet. Allerdings wird in Abhängigkeit vom Modalitätstyp die eine oder andere Zusatzinformation bzgl. der Aufnahme etc. hinzugefügt.

Für das Extrahieren der Pixel-Matrix aus den DICOM-Bildern (aus dem Pixel-Zellen-Strom), die über Netze übertragen werden, wird der Tag 7FE0 (die Bilddaten beginnen ab Tag 7FE0) verwendet.

3.2.3 Archivierungsprobleme beim Umgang mit medizinischen Daten und Datenbankkonzepte

Bilddatenbanken in der Größenordnung von mehreren hundert Gigabyte sind heute für die Radiologie nicht ungewöhnlich und auch Terabyte-Datenbanken existieren bereits. Beispielsweise werden täglich in der Radiologie eines Klinikums 1-2 Gigabyte an Bildmaterial erzeugt, das heißt pro Jahr etwa 350-700 Gigabyte. Das führt zu hohen Speicherkosten sowie zu Problemen bei der Datenbankadministration und der Performance. Ein anderer Typ von Problemen ist der heterogene Charakter des Bildmaterials, weil der Klinik normalerweise bildgebende Geräte von verschiedenen Herstellern zur Verfügung stehen. Die Lösung dieses Problems besteht in der Standardisierung aller Bilder in den DICOM-Standard und der anschließenden Abspeicherung der DICOM-Bilder in der Datenbank .

Eine Datenbank ist eine sich selbst beschreibende Sammlung von Daten, die untereinander in einer logischen Beziehung stehen und in einem eigenen Datenbankverwaltungssystem (Database Management System (DBMS)) verwaltet werden [34]. Eine Datenbank besteht aus den integrierten Daten und der Software (DBMS). Das DBMS ist die Software, die zwischen den spezifischen Anwendungen und den Daten vermittelt. Die wichtigsten heute üblichen Datenbankmodelle sind:

- das hierarchische (hierarchical) Datenbankmodell;
- das Netzwerk-Modell (network);
- das relationale Datenbankmodell;
- das objektorientierte Datenbankmodell;

Eine hierarchische Datenbank ordnet verschiedene Datentypen verschiedenen Ebenen einer Datenstruktur zu (Abbildung 3.3). Die Verbindungen zwischen den Daten verschiedener Ebenen sind einfach. Die Struktur ist klar und übersichtlich. Ein Nachteil des hierarchischen Modells ist, daß ein einzelner Dateneintrag u.U. mehrfach erscheinen muß, da er in verschiedenen hierarchischen Zusammenhängen steht. Ein weiterer Nachteil dieses Modells ist es, daß es schwierig ist, im Nachhinein die Struktur der Datenbank wesentlich zu verändern.

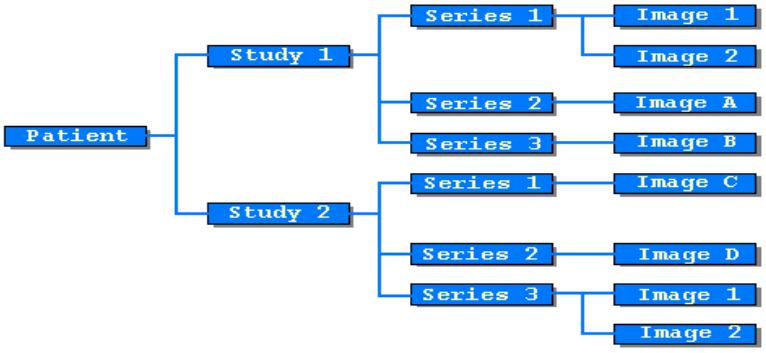


Abb. 3.3: Beispiel einer Datenbank nach hierarchischem Modell

In einer Datenbank nach dem Netzwerkmodell hat prinzipiell jeder Datenknoten unmittelbaren Zugang zu jedem anderen (Abbildung 3.4). Deswegen muß kein Knoten mehrfach vorhanden sein. Der wichtigste Nachteil des Netzwerkmodells ist, daß die Struktur unübersichtlich und ein hoher Aufwand bei der Aktualisierung von Verweisen nach Datenänderung unvermeidbar ist.

Das Hauptmerkmal einer relationalen Datenbank ist, daß die Daten in Tabellen, üblicherweise Relationen genannt, organisiert sind, die voneinander weitgehend unabhängig sein können. Alle Einträge in einer bestimmten Spalte haben dieselbe Bedeutung (z.B. Titel, Untertitel ...). Jede Spalte repräsentiert ein bestimmtes Attribut der durch die Tabelle repräsentierten Objekte. Jede Zeile enthält die Attributs eines einzigen Objektes der

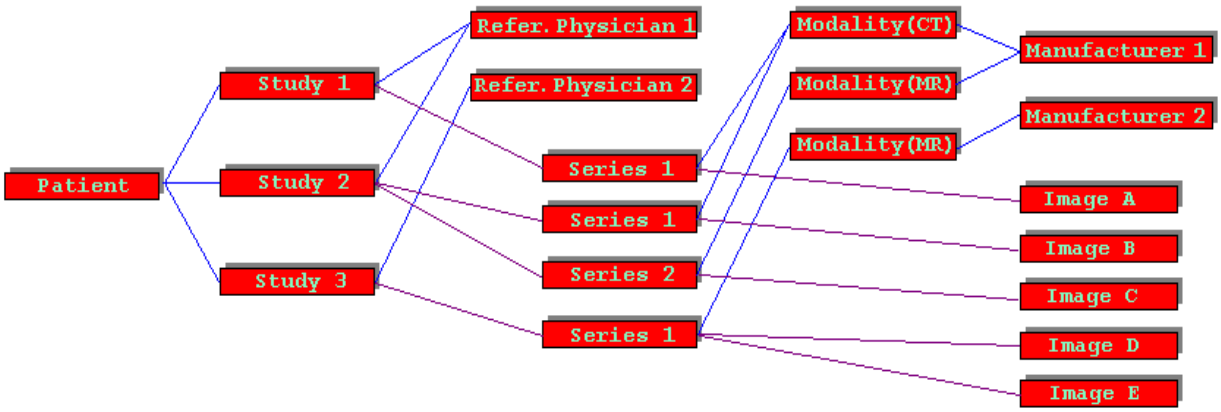


Abb. 3.4: Beispiel einer Datenbank nach Netzwerk- Modell

Tabelle. Die Reihenfolge der Zeile und Spalten in einer Tabelle ist ohne Bedeutung. Verschiedene Tabelle sind dadurch direkt verknüpft, daß sie jeweils mindestens eine Spalte mit demselben Attribut enthalten. Über direkt verknüpfte Tabelle lassen sich indirekte Verknüpfungen herstellen.

Hauptvorteil des relationalen Datenbankmodells ist, daß die Struktur der Datenbank verändert werden kann (indem man Tabellen hinzufügt oder entfernt), ohne daß deswegen

Anwendungen geändert werden müssen, die sich auf die ältere Struktur bezogen. Ein weiterer Vorteil des relationalen Modells ist, daß man beliebig viele Sichtweisen oder virtuelle Tabellen der Daten mit unterschiedlichster logischer Struktur schaffen kann, indem verschiedene Tabellen bzw. Teile von Tabellen kombiniert werden. Dazu muß die Datenbank physisch nicht verändert werden. Darüberhinaus existiert bereits eine weitverbreitete Standardabfragesprache für relationale Datenbanken. Die Nachteile des relationalen Datenbankmodells ist der hohe Aufwand bei Aktualisierung von Schlüsselfeldern nach Datenänderungen und die Redundanz durch mehrfache Speicherung der Schlüssel. Für n:m Beziehungen müssen zusätzliche Tabelle erzeugt werden.

Die Entwicklung objektorientierter Datenbanken stellt den bisher letzten Schritt in der Datenbank-Evolution dar und schafft die Grundlage für die objektorientierte Datenverarbeitung. In objektorientierten Datenbanken befinden sich die Informationen in Objekten, die aus ihren Objektklassen instanziiert werden. Die Daten sind innerhalb eines Objektes verkapselt, so daß der Zugriff nur über die Operatoren der jeweiligen Klasse erfolgen kann. Bei den Operatoren handelt es sich um Methoden, die sich auf Objekte einer Klasse anwenden lassen. Die Klassen ordnen sich in eine Generalisierungs-/Spezialisierungshierarchie ein. Eine Klasse kann Daten und Methoden von ihren Oberklassen erben. Objektorientierte Datenbanksysteme sollten daher die Objektidentität unterstützen, bei der jedes Objekt seine unveränderbare Identität hat, die unabhängig von den Attributwerten ist und vom System vergeben und verwaltet wird. Zur Realisierung der Objektidentität kann zum Beispiel die physikalische Adresse des Objekts genommen werden. Ein Beispiel des abstrakten Objekts „Image“ für eine objektorientierte Datenbank ist in der Abbildungen 3.5,3.6 dargestellt. Zum Beispiel: reales Objekt „Image 01“ besteht aus realen Objekten Patient, Study 1, Series 1, Image A.

Die wichtigsten Vorteile von objektorientierten Datenbanken sind eine übersichtliche und einfach zu wartende Struktur durch Zusammenfassung und Kapselung der Daten.

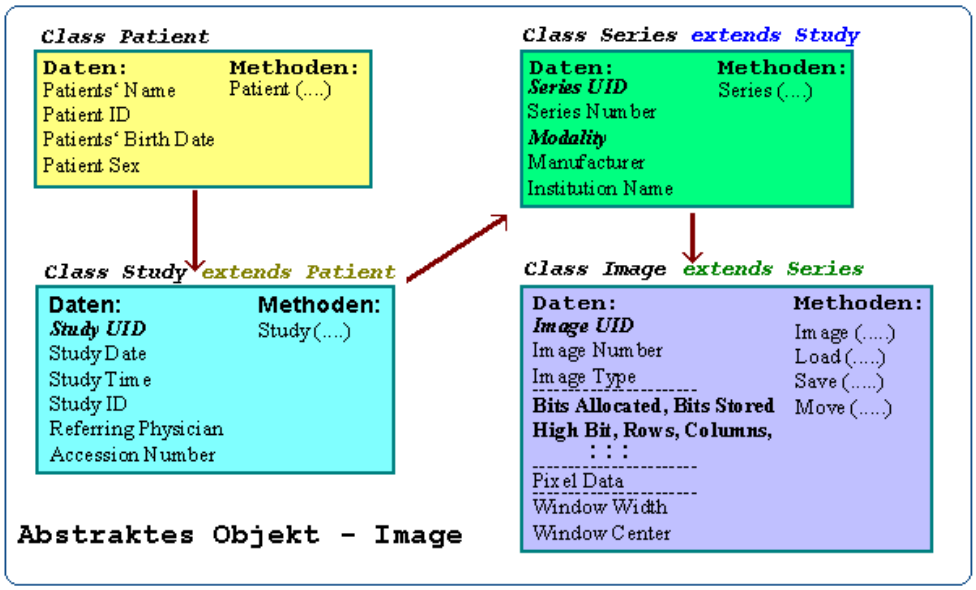


Abb.3.5: Ein Beispiel des abstraktes Objekts „Image“

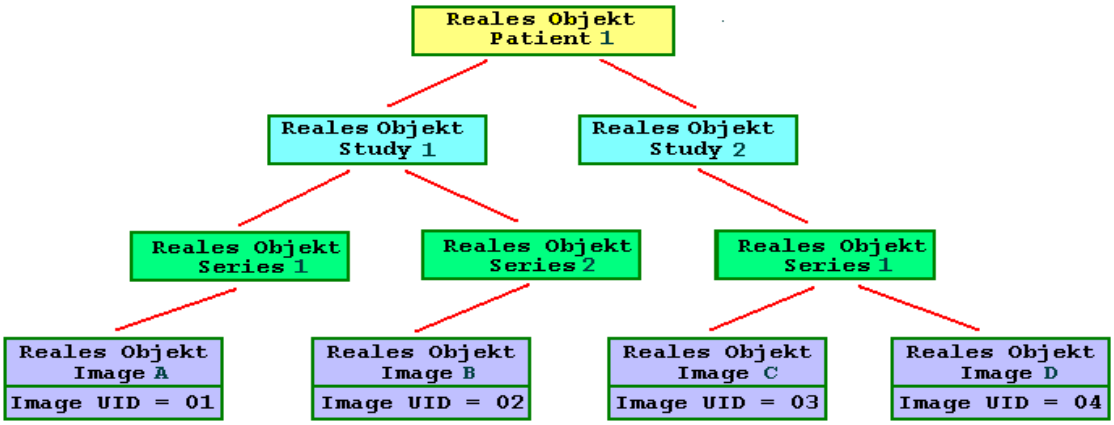


Abb.3.6: Ein Beispiel die realen Objekten: Image A, Image B, Image C, Image D

Der Nachteil von objektorientierten Datenbanken ist, daß noch keine standardisierte Abfragesprache existiert.

3.2.4. Aktuelle DICOM-Datenbankmodelle

Wie aus dem obigen Abschnitt zu ersehen ist, gibt es vier Datenbankmodelle. Welche Datenbankmodelle auf dem Gebiet der DICOM-Datenarchivierung heute aktuell sind und warum das so ist, beantwortet der folgende Abschnitt. Dieser Abschnitt gibt einen kurzen Überblick über die DICOM-Datenbanken, die heute in den vorhandenen radiologischen Systemen für Archivierung von DICOM-Bildern verwendet werden.

Das Lösungsmodell nach CHILI (Heidelberg, DKFZ)[17]. CHILI ist eine Software, die auf die Teleradiologie spezialisiert ist. Auf der Basis der CHILI-Software in Verbindung mit einer UNIX-Workstation, die als zentraler DICOM-Server benutzt wird, kann man ein PACS realisieren. Der Server übernimmt den Datentransfer und speichert die Bilddaten in einer zentralen Datenbank. Die Datenbank basiert auf dem SQL-Standard, jedoch besteht auch die Möglichkeit die Daten in einem Filesystem statt in der SQL-Datenbank abzulegen . Die in den Bilddateien enthaltenen alphanumerischen Daten über Patientennamen, Identifikations-nummer, Geburtsdatum, Untersuchungsart, Datum, Seriennummer usw. werden in der SQL- Patientendatenbank von CHILI gespeichert. Der Benutzer wählt die Bilddaten über die Studiennummer oder den Patientennamen aus.

Die Lösung gemäß MEDOS-Krankenhaus-Kommunikationssystem (MEDOS GmbH)[48]. Eine Erweiterung dieses System ist das PACS MEDOS-X unter Windows NT. Die MEDOS-Datenbank ist eine SQL-Datenbank. Die Auswertungen erfolgen über eine integrierte ODBC-Schnittstelle zur Ergebnisübernahme. MEDOS-X speichert die Headerdaten der DICOM-Bilder in einer separaten SQL-Datenbank und kennzeichnet die als relevant markierten Einzelbilder. Dabei wird kein neues Verzeichnis physikalisch erzeugt, sondern ein "virtueller" Folder aller diagnostisch relevanten Bilder mit voller Auflösung angelegt.

Das Dokumentverwaltungs- und Archivierungssystem – (Pegasos) von der Marabu EDV-Beratung und -Service GmbH [50] ermöglicht es, die Dokumente eines Krankenblattes

(Archivdatenbank), sowie Röntgenbilder, Aufzeichnungen der Funktionsdiagnostik, CT s etc. (Objektdatenbank) zu archivieren sowie hochauflösende Bilder für den normalen Arztarbeitsplatz in übliche PC-Formate umzuwandeln, oder diese in bester Qualität für die Befundung bereitzustellen.. Die Datenbank für Bilder ist eine SQL-Anwendung. Die Funktionen des TC-Moduls (Transfer/Cache) erlauben schnellste und effektivste Bereitstellung der notwendigen Daten. Hauptkomponenten des PACS sind ein DICOM 3.0 Server unter WINDOWS NT 4.0 und eine relationale Datenbank (Oracle, Informix, Interbase etc.).

Das PACS- Datenbank von ISIS (Interactive System for Image Selektion) [24] wurde mit dem relationalen Datenbank-Management-System (INGRES) entwickelt. Die Datenbank selbst enthält nur eine minimale Datenuntermenge, die die Bildfiles beschreibt. Sie besteht aus mehreren relationalen „Tabellen“, in denen bildbezogene Daten gespeichert werden. Die PACS-Datenbank enthält nicht alle klinischen oder technischen Daten. Diese Daten werden entweder durch das RIS kontrolliert oder sind in den Bildfiles enthalten. Der Hauptzweck der PACS-Datenbank besteht in der Aufrechterhaltung eines Verzeichnisses aller Bildfiles, die zu einem bestimmten Patienten gehören.

Die WEB-basierte Bilddatenbank der Dermatologie (DOIA) [22] stellt eine SQL-Datenbank (MS Access, INFORMIX) und ein Dateisystem dar. Dabei wird die Bildbeschreibung unter Verwendung eines Visual Basic Programms in die Datenbank eingegeben und die Bilder werden im Dateisystem gespeichert.

Das DACHS-Projekt [20] realisiert eine verteilte Datenbanksystem nach dem Client-Server-Prinzip für die Speicherung medizinischer Bilder. Als Datenbank stehen die Netzwerk–Datenbank WAIS und eine Multimedia-Datenbank zur Wahl. WAIS-Server fragen die in strukturierten Verzeichnissen abgelegten DICOM-Bilder und andere medizinische Bilder ab.

Aufgrund dieser Übersicht und der Tabelle (Anhang A) kann man folgendes erkennen: obwohl DICOM ein objektorientierter Datenspeicherungs- und Datenkommunikations-standard ist, verwenden die meisten vorhandenen PACS und RIS das relationale Datenbankmodell. Dafür gibt es wichtige Gründe:

- Es existiert eine weitverbreitete Standardabfragesprache für relationale Datenbanken;
- Heute stehen viele kommerzielle relationale Datenbanken für alle Betriebsysteme zur Verfügung;
- Entwurf und Erweiterung der Struktur einer relationale Datenbank sind einfach.

Aus diesem Grund kann man voraussetzen, daß auch noch während der nächsten 2-3 Jahre das relationale Datenbankmodell in der DICOM-Archivierung dominieren wird [52].

3.2.5 Eine Struktur für eine relationale DICOM-Datenbank

Der Zugriff auf die relationale DICOM-Datenbank wird über den DICOM-Java-Server durch JDBC-ODBC-Schnittstelle realisiert. Die Header-Informationen aus den DICOM-Bildern legt der Server in den Tabellen der relationalen DICOM-Datenbank ab [47,53].

Die Struktur der relationalen DICOM-Datenbank wird durch die Auflistung aller Tabellen in der Datenbank mit ihren Beschreibungen sowie den Beziehungen zwischen den

Tabellen dargestellt, die die Verbindung zwischen den zu verschiedenen Tabelle gehörenden Elementen herstellen. Die verwendeten Attribute sind nur eine Teilmenge der Standardattribute, die das DICOM-Format zur Verfügung stellt. In diesem Modell wurde jedoch absichtlich auf die komplette Übernahme aller Attribute verzichtet. Intensive Gespräche mit praktisch tätigen Ärzten ergaben, daß die meisten der Standardattribute für den behandelnden Arzt von keinerlei praktisch nutzbaren Wert sind und deswegen auch nicht übertragen werden müßten. Deswegen ist es aus zweierlei Gründen sinnvoll, die übertragenen Attribute auf die für den behandelnden Arzt relevante Informationen zu reduzieren : erstens reicht die verminderte Attributmenge völlig aus, um die gewünschten Daten in der Datenbank zu suchen und zweitens wird die Übertragungsgeschwindigkeit durch die verminderte zu übermittelnde Datenmenge verkürzt. Werden doch einmal alle Header-Informationen benötigt, so kann man sich auch den kompletten Header aus dem Dateisystem herunterladen.

Den Kern der Datenbank bilden vier Tabellen.

Tabelle - Patient

Patient ID	Patients‘ Name	Patients‘ Birth Date	Patient Sex

Tabelle - Study

Study UID	Study Date	Study Time	Study ID	Referring Physician	Accession Number	Patient ID

Tabelle - Series

Series UID	Series Number	Modality	Manufacturer	Institution Name	SOP Class UID	Patient ID	Study UID

Tabelle - Image, Teil 1

Image UID	Image Number	Image Type	Bits Allocated	Bits Stored	High Bit	Planar Configuration	Photometric Interpretation

Tabelle - Image, Teil 2

Rows	Columns	Pixel represen-tation	Samples per Pixel	Window Width	Window Center	Patient ID	Study UID	Series UID	File Name

Die Zusammenhänge zwischen der Tabelle Patient und anderen Tabellen werden über die Schlüsselfelder (Beziehungen) „Patient ID“ hergestellt. Diese Beziehungen (Relationen) sind in den Tabellen mit abgespeichert.

Die Zusammenhänge zwischen der Tabelle Study und den Tabellen Series und Image4 werden über die Schlüsselfelder „Patient ID“ oder „Study UID“ hergestellt. Die Zusammenhänge zwischen Tabelle Series und Tabelle Image werden über die Schlüsselfelder „Patient ID“ oder „Study UID“ oder „Series UID“ hergestellt.

Die Bilddaten werden durch den DICOM-Java-Server in einem bestimmtem Verzeichnis des Dateisystems gespeichert. Dieses Verzeichnis besteht aus drei Unterverzeichnissen: das erste Unterverzeichnis beinhaltet das Original-DICOM-Dokument, das zweite die DICOM-Header und die komprimierten Bilddaten (nach einer Kompression von 5:1 bis 15:1 ohne visuellen Unterschied zum Original). Das dritte Verzeichnis beinhaltet die skalierte Version des Originalbildes. Die Adresse jeder Datei (Original-DICOM-Dokument, DICOM-Header, die komprimierten Bilddaten und die skalierte Version) entsprechend dem Hauptverzeichnis des Dateisystems, sind in der Spalte „File Name“ (Tabelle-Image) abgelegt.

3.2.6 Entwicklung einer java-basierten objektorientierten DICOM-Datenbank

Die Konzeption des relationalen Datenbankmodells berücksichtigt den objektorientierten Charakter des DICOM-Formats nicht. Daraus folgt ein unnötig hoher Aufwand beim

Speichern der Schlüssel, sowie die Notwendigkeit zusätzlicher Tabellen zur Darstellung der Beziehungen und bei der Aktualisierung von Schlüsselfeldern nach Datenänderungen.

Dieser Abschnitt ist der Idee für eine konkrete Implementierung einer objektorientierten DICOM-Datenbank gewidmet [27]. Für die Implementierung der objektorientierten DICOM-Datenbank (OODDB) wurde die objektorientierte Programmiersprache Java benutzt [54].

Die abstrakten Bauelemente des objektorientierten DICOM-Datenbankmodells sind Klassen und Objekte. Ein abstraktes Objekt besteht aus einer Menge von Attributen, die verschiedene Datentypen haben können und einer Sammlung von Methoden, mit denen diese Daten bearbeitet und verändert werden können. Diese Menge von Attributen und Methoden werden durch eine oder mehrere Klassen beschrieben. Die Attributzuweisung und die Methodenimplementierung erzeugen ein reales Objekt, das man Instanz nennt. Jede Instanz befindet sich zu jedem konkreten Zeitpunkt in einem bestimmten Zustand. Der Zustand einer Instanz wird durch seine aktuellen Attributwerte beschrieben. Eine Instanz bewahrt während ihrer Lebensdauer ihren eigenen, unveränderlichen Identifikator, der unabhängig vom jeweiligen Zustand der Instanz ist.

Eine Klassenhierarchie für eine objektorientierte DICOM-Datenbank wird in Abbildung 3.7 dargestellt. Die allgemeinste Klasse ist die Klasse „Activ Element“, die von einem TCP/IP-Server abgeleitet ist. Die Klasse „Server DBMS“ implementiert die Objekte und Methoden, die die allgemeinen SOP-Objekte des DICOM-Server unterstützen. Die Klassen „Create IO“, „Select IO“, „Delete IO“ und „Move IO“ erzeugen spezielle SOP-Objekte des DICOM-Servers und bilden die Schnittstelle zwischen standardisierten SOP-Objekten des DICOM-Servers und den Daten in der OODDB. Die Klasse „Create IO“ erzeugt ein neues reales Objekt „Image“ (IO) in der OODDB. Die Klasse „Select IO“ sucht und extrahiert aus der OODDB die Attribute der IO, die den verwendeten Suchparametern gerecht werden. Die Klasse„Move IO“ extrahiert aus der OODDB die Attribute und Rohdaten bestimmter realer IO und sendet sie zum Client. Die Klasse „Delete IO“ löscht ein bestimmtes IO in der OODDB. Während ein neues reales IO erzeugt wird, werden die Identifikatoren dieses Objekts generiert und in einer speziellen Tabelle gespeichert. Diese Operationen führt die Klasse „Identität“ aus. Die Klasse „Sicherheit“ macht eine Sicherheitskopie von den Identitätstabellen und den Attributen aller realen IO, aus der man alles rekonstruieren kann. Die Klasse „Benutzerverwaltung der Datenbank“ beinhaltet Methoden für die Zugriffssicherheit und zur Klassenhierarchieveränderung der Datenbank.
Die Objekte „Server DBMS“, „Create IO“, „Selekt IO“, „Delete IO“, „Move IO“, „Identität“, „Sicherheit“ und „Benutzerverwaltung der Datenbank“ bilden das DBMS (Datenbank- Management-System) der OODDB. Diese Objekte verwirklichen eine zentrale Koordination.

Das Objekt „Schnittstelle des IO“ bildet eine Schnittstelle für die Manipulation und Kommunikation der Attributen und Rohdaten, die in der Image-Instanz gespeichert sind. Instanzen dieses Objekts haben über das TCP/IP-Protokoll Kontakt mit den Instanzen der Objekte „Select IO“, „Create IO“, „Delete IO“, „Move IO“. Außerdem können sie mit den anderen Clients im Netz über das TCP/IP-Protokoll kommunizieren. Das Objekt „Schnittstelle des IO“ beinhaltet Methoden für die Speicherung, das Lesen und Suchen von Attributen und Rohdaten auf der Festplatte. Die Klasse „Patient“ erbt die Daten und Methoden der Klasse „Aktiv Element“ und „Schnittstelle des IO“. Objekte der Klasse

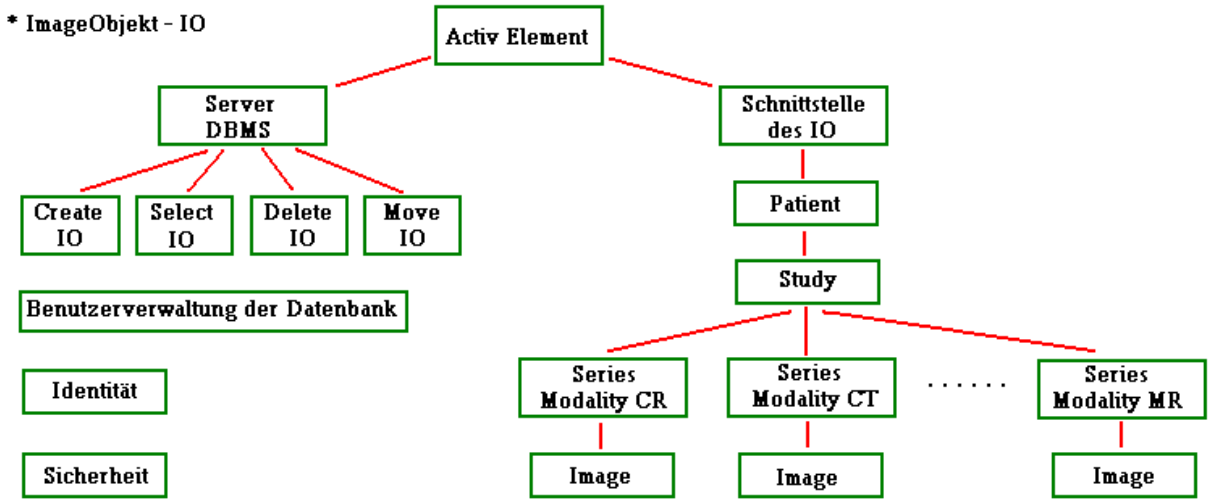


Abb.3.7: Eine Klassenhierarchie für eine objektorientierte DICOM-Datenbank

„Patient“ enthalten patientenbezogene Attribute. Objekte der Klasse „Study“ verwalten Studydaten. Die Modalitäten werden über die Objekte der Modalitätenklasse bestimmt. Die Image-Klasse erbt die Attribute und Methode der darüberliegende Klassen und die Objekte enthalten die bildbezogenen Attribute und Rohdaten. Alle Instanzen dieser Objekte („Image“, „Series“, „Study“, „Patient“ und „Activ Element“) haben einen Identifikator („Image UID“, „Series UID“, „Study UID“, „Patient ID“, „Host:Port“). Der Identifikator „Host:Port“ beinhaltet die IP-Adresse und den Socket-Port. Die Nummer des Socket-Port ist identisch mit dem Attribut „Patient ID“. Alle Identifikatoren der vorhandenen Image-Instanzen (5 pro Instanz) werden in einer speziellen „Identität“-Instanz verwaltet. Wenn alle Identifikatoren eines neuen DICOM-Bildes nicht mit den Identifikatoren der Identität-Instanz übereinstimmen, werden fünf neue Instanzen „Patient“, „Study“, „Series“, „Image“, „Server IO“ in der OODDB erzeugt. In jedem anderen Fall werden nur die fehlenden Instanzen neu erzeugt. Die beschriebene Klassenhierarchie des Image-Objekt berücksichtigt den hierarchischen Charakter des DICOM-Informationsmodells. Dies reduziert den Speicheraufwand und die Zeit für die Datenmanipulation in der OODDB.

Das vorgeschlagene Klassen-Modell der OODDB ermöglicht eine verteilte Implementierung. DBMS-Instanzen (Datenbank-Management-System) und Image-Instanzen können auf verschiedenen Rechnern laufen und miteinander durch das TCP/IP-Protokoll über das Netz kommunizieren. Das Kriterium für die Datenverteilung zwischen den Rechner der OODDB ist die mögliche Datenvolumenspeicherung (die Größe des Festplatte) für den speziellen Rechner. Die wichtigsten Vorteile des OODB-Modells sind eine bessere Skalierbarkeit, eine gute Verteilbarkeit und ein geringeres Datenaufkommen als im relationalen Datenbankmodell. Durch die Java-Implementation der Klassen der OODDB wird die Unabhängigkeit vom Betriebssystem gewährleistet.

3.3 Konzeption eines DICOM-basierten PACS für Krankenhäuser

Den einfachen, modularen Aufbau eines DICOM-basierten Systems zur automatischen Archivierung und Übertragung von Bildinformationen in der Radiologie-Abteilung eines Krankenhauses zeigt Abbildung 3.8 [28]. Der modulare Systemaufbau dient der Übersichtlichkeit, der Wartungsfreundlichkeitkeit und der Anpassungsfähigkeit an

wachsende Anforderungen. Entsprechend den Anforderungen können die integrierten Rechner mit mehr oder weniger Leistung konzipiert werden. Allgemein wird hier von einem Rechner gesprochen, wenn im einzelnen ein PC als Arbeitsplatzrechner, ein hochwertiger PC als Server oder sogar eine klassische Workstation gemeint ist. Dieses System besteht aus DICOM-fähigen Bildgebergeräten und einem System zur Archivierung und Übertragung von DICOM-Bildern (PACS). Wenn ein Bildgebergerät keine DICOM-Schnittstellen hat, wird ein Modul aktiviert, das die Bilddaten digitalisiert und in das DICOM-Format umwandelt. Bilddaten nach dem DICOM-Standard können über Datennetze unabhängig von der Bildgeberhardware übertragen oder auf einem Speichermedium archiviert werden.

Die Elemente des Systems zur automatischen Archivierung und Übertragung von DICOM-Bildern kann man entsprechend ihrer Funktionalität drei Hauptgruppen zuordnen. Die erste Gruppe realisiert die Schnittstelle zu den radiologischen Untersuchungsgeräten. Die zweite Gruppe bilden der Server, die DICOM-Datenbank und das DICOM-Archiv. Zur dritten Gruppe gehören die Befundungsstationen und Arbeitsplatzrechner der Ärzte.

Ein Bildgebergerät erzeugt ein DICOM-Bild und transferiert es auf den Rechner, der die Schnittstelle zum PACS bildet. Auf diesem Rechner läuft der DICOM-Server, der mit DICOM-fähigen Untersuchungsgeräten über das DICOM-Protokoll kommuniziert. Er reagiert auf entsprechende Anfragen, ein Bild anzunehmen und abzuspeichern. Wenn das Untersuchungsgerät kein DICOM-Format unterstützt, transferiert es das Bild in ein bestimmtes Verzeichnis auf dem Schnittstellenrechner des PACS‘. Dieser DICOM-Server-Rechner kann bei der Aufgabe der Digitalisierung und Formatumwandlung durch einen weiteren, für diesen Zweck vorbehaltenen Rechner unterstützt werden, der bei Bedarf und Vorliegen analoger radiologischer Daten die Aufgabe der automatischen Digitalisierung und Konvertierung in das DICOM-Format übernehmen kann. Ein solches Subsystem beinhaltet auf der Hardwareseite zusätzlich einen Framegrabber und eventuell einen Scanner oder eine Digitalkamera. Damit können dann analoge Bildsignale oder auf Film vorliegendes Bildmaterial digitalisiert werden. Anschließend transformiert ein Softwaremodul das Digitalbild oder Video in ein DICOM-Bilddokument.

Ein Anschluß an das Krankenhausinformationssystem oder Radiologieinformationssystem ist von Vorteil, da die Integration der dort vorgehaltenen Informationen in das DICOM-Dokument ermöglicht wird. Wie auch immer im einzelnen ein DICOM-Dokument erzeugt wurde, im Ergebnis fügt man die diagnostisch relevanten Bildinformationen im integrierten DICOM-Header mit den Stamminformationen zu einem Patienten (wie zum Beispiel Name, Geschlecht, Alter und Patientenidentifikationsnummer etc.) zu einer Behandlungsakte zu-sammen.

Der Server und die DICOM-Datenbank dienen zur mittelfristigen Speicherung der vom Schnittstellenrechner erhaltenen DICOM-Dokumente. Hardware-Plattform dafür kann ein leistungsfähiger PC oder eine Workstation sein, woran ein Speichermedium über einen Controller angeschlossen wird. DICOM-Bilder sollten 1 bis 3 Wochen lang gespeichert werden. Dazu wird eine Kapazität von 20 bis 40 Gigabyte benötigt. Das DICOM-Archiv ist eine CD-Jukebox mit einem CD-Recorder. Die CD-Jukebox ist ein Robotersystem mit einer Kapazität von etwa 600 CD’s (das entspricht etwa 400 GB). Der Server der DICOM-Datenbank realisiert eine Reihe von Aufgaben. Er reagiert auf die Anfragen des Schnittstellenrechners des PACS und auf Anfragen der Befundungsstationen, mit denen er

über das lokale Netz verbunden ist. Zweitens nimmt der Server aus der DICOM-Datenbank neue DICOM-Bilder entgegen und speichert sie im DICOM-Archiv. Drittens reagiert er auf Anfragen von den Befundungsstationen und Arbeitsplatzrechnern der Ärzte. Das Steuerprogramm für die CD-Jukebox und für den CD-Recorder analysiert das Datenvolumen der neuen DICOM-Bilder zur Erstellung einer neuen CD.

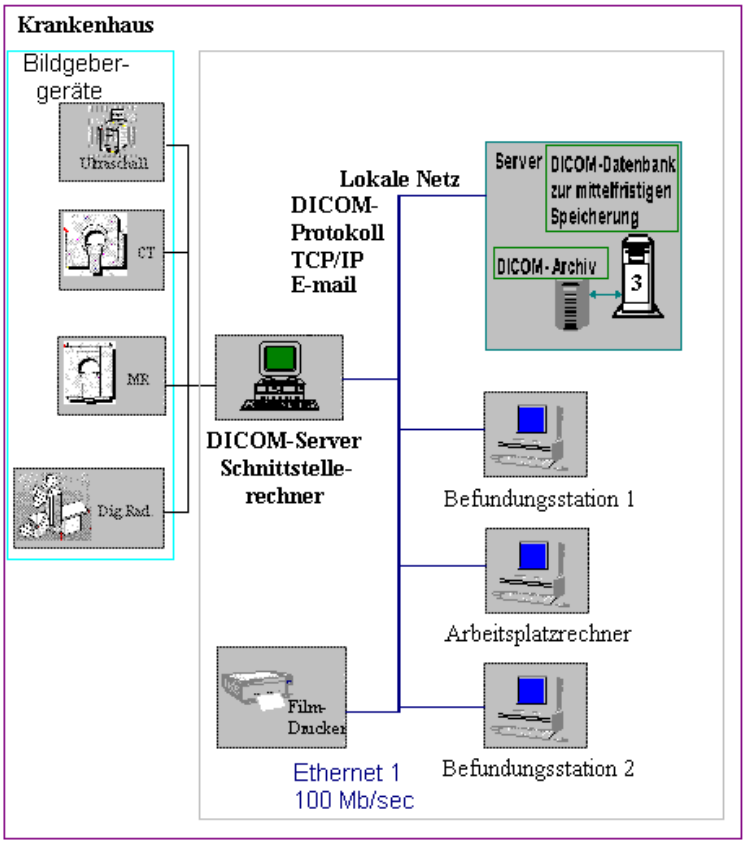


Abb.3.8: Struktur eines PACS

Die Kommunikation zwischen “Schnittstellenrechner”, dem Server der DICOM-Datenbank und den Befundungsstationen kann durch zwei Verfahren verwirklich werden. Im ersten Fall findet die Verbindung durch das DICOM-Protokoll statt, wenn die DICOM-Datenbank und das DICOM-Archiv entsprechend dem DICOM-Standard implementiert sind. Im zweiten Fall findet die Verbindung durch SMTP-, FTP- oder TCP/IP-Protokoll statt, wenn die DICOM-Datenbank und das DICOM-Archiv als File-System implementiert sind. Für die Adressierung der Komponenten des Systems (DICOM-Datenbankserver, “Schnittstellenrechner“, Befundungsstationen) werden E-mail-Adresse oder IP-Adresse und Portnummer verwendet.

Als Befundungsstationen dienen normalerweise die Arbeitsplatzrechner der Ärzte, i.a. also PC-Rechner mit hochauflösender Graphikkarte und entsprechendem Monitor. Auf diesen Rechnern ist eine DICOM-Viewer-Software installiert. Jeder Arbeitsplatzrechner ist über das lokale Netz mit der DICOM-Datenbank und dem Archiv verbunden, so daß der Arzt von seinem Rechner aus direkt auf jedes Bild aus der Datenbank zugreifen kann und dieses mit Hilfe des DICOM-Viewers analysieren kann. Nach der Bildanalyse entscheidet der Arzt, ob das Bild gelöscht werden kann oder im Archiv zu speichern ist.

Der Schnittstellenrechner und alle Befundungsrechner sind an Multi-Format-Laserdruckern angeschlossen. Jedes DICOM-Bild kann aus dem DICOM-Viewers heraus gedruckt werden.

3.4 Wartungsfreundlichkeit, Erweiterbarkeit und Flexibilität des PACS

Damit das oben beschriebenen System zur automatischen Archivierung und Übertragung von DICOM-Bildern evaluiert werden kann, muß es folgenden Anforderungen an ein modernes RIS gerecht werden:

- Offene und standardisierte Schnittstellen zwischen Bildgebergeräten, Datenbanken und Arbeitsplatzrechnern;
- Möglichkeit zur Integration neuer Hardware und von alten oder fremden Datenbeständen in das PACS;
- Unabhängigkeit des PACS von der Infrastruktur, den Rechnern und Betriebsystemen.

Die Implementierung sichert die Erfüllung dieser Anforderungen durch:

- Anwendung von DICOM-Protokoll und Intranet/-Internet Protokollen (HTTP und TCP/IP);
- Einsatz eines verteilten Datenbanksystems;
- Verwendung skalierbarer Standard-Hardwarekomponenten;
- Verwendung plattformunabhängiger Software und moderner Programmiersprachen.

Kernstück des Kommunikationskonzepts des PACS ist die Unterstützung des DICOM 3.0-Protokoll und die Implementierung als JAVA-Client-Server- und Web-Technologie. Dabei wird das DICOM-3.0-Protokoll für die Übertragung der Bild- und Textdaten von den DICOM-fähigen Bildgebersystemen verschiedenster Hersteller nach den Schnittstellen des PACS verwendet. Durch das DICOM-Protokoll bestimmt die Schnittstelle des PACS automatisch welche Daten, in welcher Menge und Folge, von welchem Bildgebergeräte zu erwarten hat und wohin dieses weiter zu versenden ist. Außerdem soll die Software des Systems das Intranet-Kommunikationsprotokoll für die Übertragung der DICOM-Daten als Datei unterstützen. Dafür werden das TCP/IP- und das HTTP-Protokoll verwendet. Zugriff auf die DICOM-Datenbanken und das DICOM-Archiv des PACS über das Internet wird durch Web-Kommunikationstechnologie ermöglicht. Die Implementierung solcher Kombinationen, die standardisierte Kommunikationsprotokolle verwenden, ermöglicht sichere und offene Schnittstellen zwischen den Bildgebergeräten und dem PACS, den Komponenten des PACS untereinander, sowie zwischen dem PACS und den Internet-Kunden. Durch einfache und offene Schnittstellen kann man leicht neue Hardware-Komponente in das PACS integrieren oder neue Bildgebergeräte anschließen.

Die Bildarchivierung erfordert ein hohes Maß an Zuverlässigkeit und Sicherheit und muß im Hinblick auf den Investitionsschutz auf einer zukunftsorientierten Technologie basieren, die ausreichende Erweiterungs- und Kommunikationsmöglichkeiten garantiert. Deshalb wird die Verwendung eines verteilten Datenbanksystems vorgeschlagen. Verteilte Datenbanksysteme können gleichzeitig aus relationalen DICOM-Datenbanken, objektorientierten DICOM-Datenbanken, und Datei-Systemen bestehen, die alle die

verschiedenen Objekte des Datenbanksystems darstellen [31-34]. Die Hardware dieses Datenbanksystems kann aus heterogenen Komponenten (PCs, Workstations) bestehen. Dabei kann man leicht neue Hardware in das Datenbanksystem integrieren. Die einzelnen Rechner des Datenbanksystems können sich in unterschiedlichen Abteilungen der Radiologie im Krankenhaus befinden. Alle Rechner des Datenbanksystems sind am Netzwerk des PACS angeschlossen. Auf den Rechnern des Datenbanksystems können verschiedene Betriebsysteme (UNIX, OS/2, DOS, Windows95/98/NT) und relationale Datenbanken (Oracle, Sybase, Access) laufen. Jeder Rechner ist mit preiswerten optischen Speichersystemen (CD-Jukebox mit einer Kapazität von etwa 400GB) ausgestattet. Die Verwaltungssoftware der DICOM–Datenbank und des DICOM-Archivs ermöglicht es, zuerst nur einen Rechner mit einer Jukebox zu verwenden und bei Bedarf das DICOM-Archiv zu vergrößern, indem ein neuer Rechner mit installierter relationaler Datenbank und eigener Jukebox integriert wird. Der Vorteil dieser Technik ist die Möglichkeit der einfachen Vergrößerung des Datenbanksystems durch die über das angeschlossenen neuen Rechner. Da jeder Rechner mit seiner eigenen Datenbank unabhängig von den anderen Rechnern ist, kann das System eine gute Datensicherheit unterstellt werden. Die Such- und Datentransfergeschwindigkeiten sind hoch und unabhängig vom gesamten Datenvolumen im Archiv, da der Suchvorgang gleichzeitig auf allen Rechnern des Datenbanksystems verwirklicht wird.

Die Software des PACS muß in einer plattformunabhängigen Programmiersprache geschrieben sein [27,28,35]. Dazu wird die Programmiersprache Java verwendet [36-38,54]. Java ist eine objektorientierte Sprache und eins der innovativsten Werkzeuge, die im Internet zur Verfügung stehen. Das entscheidende zusätzliche Feature der Sprache Java, und damit der Grund für das weltweite Interesse, ist die Möglichkeit, Programme in compilierter Form über das Internet zu verschicken und auf dem Client-Rechner ohne aufwendige Installationsprozesse sofort auszuführen. Java ermöglicht unter anderem die Definition systemunabhängiger Schnittstellen für Datenbanken mittels JDBC [39,40], die Entwicklung von Methoden zur optimierten Bildübertragung. Auf diese Weise kann man durch Java-basierte Software die Unabhängigkeit des PACS von der Infrastruktur, den Rechnern, Betriebsystemen und dem Netzwerk erreichen.

3.5 Schneller und einfacher Zugriff auf DICOM-Bilder aus dem PACS-Archiv über das Intranet/Internet

Das Konzept des Informationssystems zur Darstellung, Verarbeitung und Übertragung medizinischer Bilddaten im Intra- und Internet ist in Abbildung 3.9 dargestellt. Diese Systemkonzept baut auf der Kombination von Java-basierter Web-Technologie durch Java Anwendungen, Applet und Servlets auf [41-46]. Typischerweise werden die Bilddaten der unterschiedlichen diagnostischen Systeme (CT, MRT usw.) auf dem Befundungsrechner (Arztworkstation) ausgewertet und in einem zentralen Bildarchiv gespeichert. Im dargestellten Systemkonzept verbindet ein spezieller Web-Server das zentrale Datenarchiv der radiologischen Abteilung (PACS) mit den dezentralen Arbeitsplätzen (Clients). Die Kopplung zwischen Web-Server und Clients erfolgt über Intra- oder Internet. Die Clients können sowohl in den verschiedenen Klinikbereichen (Intranet-Kopplung), als auch in den Praxen niedergelassener Ärzte (Internet-Kopplung) plaziert sein. Auf dem Web-Server ist

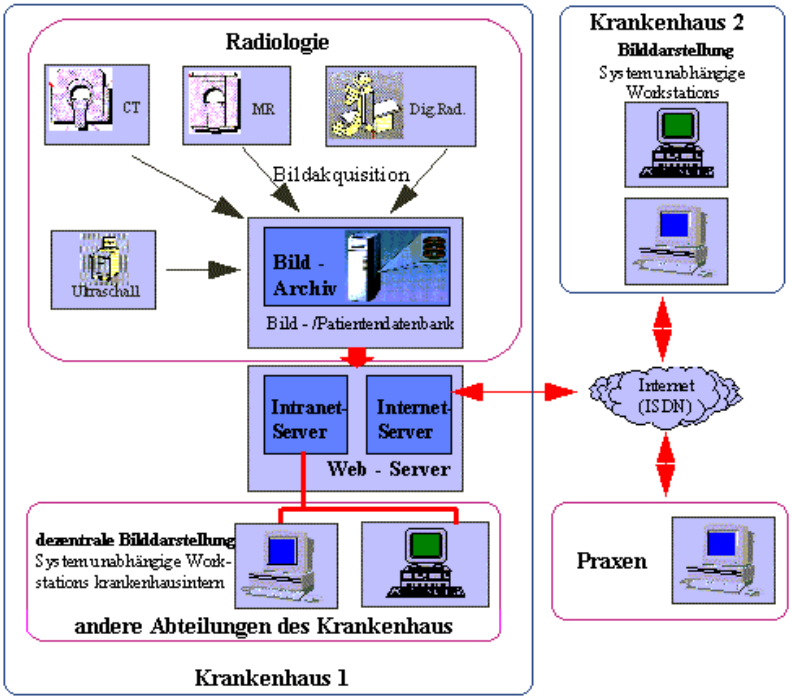


Abb.3.9: Systemkonzept eines verteilten, multimedialen Informationssystems im Bereich der bildgebenden Diagnostik.

eine Schnittstelle des verteiltes Informationssystems installiert, das auf den Datenbestand des Zentralarchivs zugreift.

Der Systembenutzer benötigt auf dem Client-Rechner mit Ausnahme eines Web-Browsers und einer Internetverbindung keine spezielle Software. Durch die HTTP-Servlet-Software und die Applets, die auf dem Web-Server installiert sind, greift der Client direkt auf die Daten aus dem Datenbanksystem des PACS zu. Das Java-Applet wird automatisch in seinem Browser gestartet. Medizinische Bilddaten werden in originärer Qualität via ISDN und dem Internet an den beteiligten Client übertragen.

Das Java-Framework garantiert eine von der Hardware und von dem Betriebssystem des Client-Rechners unabhängige Anwendung. Diese Plattformunabhängigkeit stellt eines der generellen Vorteile der Java-Programmen dar. Durch das Java-Applet wählt der Benutzer die gewünschte medizinische Akte eines Patienten aus und aktiviert die Übertragung der Bilddaten und der zugehörigen Befunde und Patienteninformationen vom Server auf den Client. Ebenfalls werden sämtliche Java-Komponenten zum Einlesen, Darstellen und Verarbeiten der Daten auf den Client übertragen und ausgeführt.

3.6 Konzeption des Intranet-/Internet-basierten radiologischen Informationssystems

Dieser Abschnitt präsentiert die Konzeption des gesamten Aufbaus des intranet/internet-basierten Radiologieinformationssystems für Kliniken und niedergelassene Ärzte, der die Struktur, die Komponenten, die Kommunikation zwischen Komponenten und die Software des System bestimmt. Die Abbildung 3.10 zeigt die Struktur des Systems [47]. Das

vorgeschlagene Informationssystem hat eine modulare Struktur und enthält grundsätzlich folgende Komponenten:

- Untersuchungsstationen/ Bildgebergeräte;
- Befundungsstationen / Arbeitsplatzrechner;
- Filmbelichter / Multi-Format-Laserdrucker;
- Intranet: lokales Netz;
- einen oder mehrere DICOM-Schnittstellenrechner, die zwischen Bildgebergeräten und PACS vermitteln;
- einen Java-Server, relationale DICOM-Datenbank und ein Datei-System, die die Daten für kurze Zeit (1 bis 3 Wochen) speichert;
- einen Java-Server, eine DICOM-Datenbank (relationale oder objektorientierte), ein Datei-System und ein DICOM-Archiv, das die Daten für lange Zeit archiviert;
- Zentralrechner und Intranet-Server;
- Transmissions-Kontrollrechner zur Steuerung des Datenaustausches zwischen Krankenhausnetz und Internet (Verschlüsselung-/ Identifikations Server (Security Server) und Web-Server);
- Internet: globales Netz;
- Internet-Client-Rechner (Praxis, andere Krankenhäuser und Klinik).

Die Elemente dieses System kann man entsprechend ihrer Funktionalität vier Hauptgruppen zuordnen. Die erste Gruppe beinhaltet die bildgebenden Geräte. Die zweite Gruppe besteht aus Elementen des PACS. Die dritte Gruppe bilden die Netzwerkkomponenten (Internet, Transmissions-Kontrollrechner, Verschlüsselung/ Identifikation Server, Web-Server) und zur vierten Gruppe gehören die Intranet/Internet Kunden.

Die Elemente eines PACS bilden fünf Hauptgruppen. Die erste Gruppe realisiert die Schnittstelle zu den radiologischen Untersuchungsgeräten. Die zweite Gruppe bildet der Server und die DICOM-Datenbank. Die dritte Gruppe bildet der Server, die DICOM-Datenbank und das DICOM-Archiv für Langzeitspeicherung. Zur vierten Gruppe gehören die Befundungsstationen und Arbeitsplatzrechner der Ärzte und die fünfte Gruppe bilden der Zentralrechner und der Intranet-Server.

Bildgebergeräte erzeugen DICOM-Bilder und transferieren diese zum Schnittstellenrechner des PACS. Auf dem Schnittstellenrechner ist ein DICOM-Java-Server installiert, der das DICOM-Protokoll für den Datenaustausch unterstützt. Dieser Rechner kommuniziert mit DICOM-fähigen Untersuchungsgeräten über das DICOM-Protokoll. Das Programm DICOM-Java-Server wurde in der Programmiersprache JAVA realisiert, so daß es unter jedem Betriebssystem (UNIX, WINDOWS NT, WINDOWS 95) lauffähig ist. Seine Funktionen können wie folgt beschrieben werden:

- Das Programm reagiert auf Anfragen der Bildgeberstationen und nimmt von den Bildgeberstationen neue DICOM-Bilder entgegen;
- Das Programm schickt neue DICOM-Bilder, die es direkt von den Bild gebenden Stationen erhalten hat, zum Java-Server der DICOM-Datenbank zur Kurzzeitspeicherung.

DICOM-Datenbank für Kurzzeitspeicherung. Die DICOM-Datenbank für die Kurzzeitspeicherung dient zur Speicherung der vom Schnittstellenrechner erhaltenen DICOM-Dokumente für einen Zeitraum von 1 bis 3 Wochen. Die DICOM-Datenbank zur

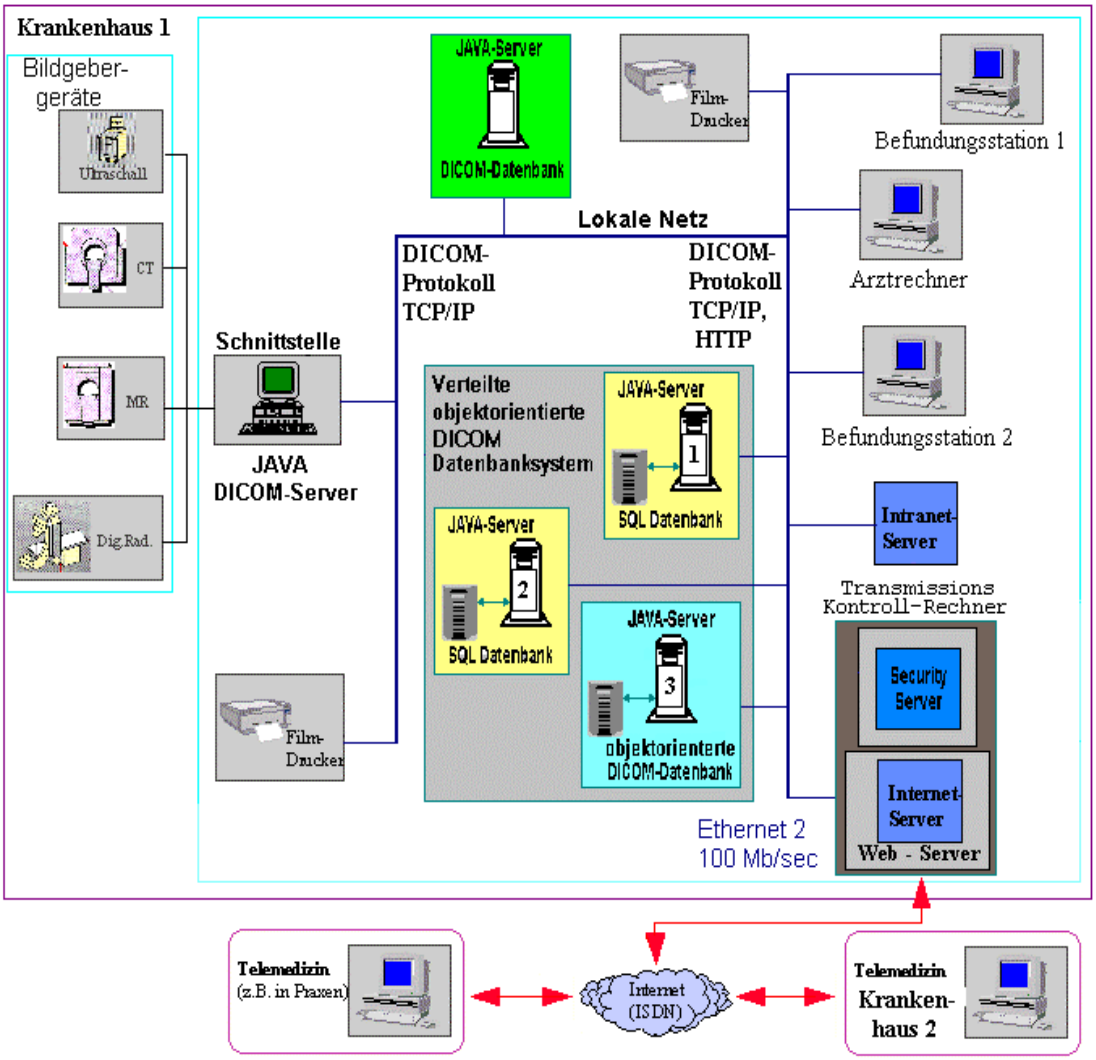


Abb.3.10: Struktur des intranet/internet-basierten Radiologieinformationssystems für Kliniken und niedergelassenen Ärzte

Kurzzeitspeicherung besteht aus zwei Java-Servern, der Java-basierten SQL-Schnittstelle für die DICOM-Datenbank (JDBC), einer relationalen DICOM-Datenbank und einem Datei-System für die Speicherung der DICOM-Bilder. Die Hardware-Plattform für beide Realisierungslösungen kann ein leistungsfähiger PC oder eine Workstation sein, an die ein skalierbares Speichermedium (Kapazität von 20 bis 40 Gigabyte) angeschlossen wird. Der Rechner der Datenbank für die Kurzzeitspeicherung hat zwei IP-Adressen. Die erste IP-Adresse ist für das erste lokale Netz, die zweite IP-Adresse ist für das zweite lokale Netz. Der erste Server reagiert auf die Anfrage der Schnittstelle des PACS und nimmt neue Bilder entgegen. Die Kommunikation zwischen dem Schnittstellenrechner und dem ersten Java-Server basiert auf dem DICOM-Protokoll. Der zweite Server kommuniziert mit dem DICOM-Java-Server oder dem Java-Server der zentralen DICOM-Datenbank für

Langzeitspeicherung, mit denen er über das zweite lokale Netz verbunden ist. Außerdem reagiert der zweite Server
auf die Anfrage der Befundungsstation, mit denen er ebenfalls über das zweite lokale Netz verbunden ist.

Die zentrale DICOM-Datenbank und das DICOM-Archiv für Langzeitspeicherung. Die zentrale DICOM-Datenbank ist ein verteiltes objektorientiertes Datenbanksystem (VODBS), das aus vielen heterogenen Objekten bestehen kann. Unter Objekt verstehe ich im diesem Kapitel einen PCs oder Workstation, die mit einem CD-Recorder und einer CD-Jukebox ausgestattet sind. Die minimale Konfiguration der Objektssoftware der zentralen DICOM-Datenbank beinhaltet einen DICOM-Java-Server oder Java-Server, eine relationale DICOM-Datenbank oder eine objektorientierte DICOM-Datenbank mit angeschlossenem Archiv, der Java-basierten SQL-Schnittstelle für die relationale DICOM-Datenbank, einem Dateisystem, der Archivsteuerungssoftware. Jeder Objektrechner ist an das zweite lokale Netz angeschlossen. Der Objektserver reagiert auf die Anfrage der DICOM–Datenbank für Kurzzeitspeicherung, nimmt neue Bilder entgegen und reagiert auf die Anfragen der Befundungsstationen, sowie der Clients aus dem Intranet und des Internet-WEB-Server. Die Kommunikation zwischen der Datenbank für Kurzzeitspeicherung, dem VODBS und den Befundungsstationen basiert auf dem DICOM- und dem TCP/IP-Protokoll. Das Steuerprogramm für die CD-Jukebox und für den CD-Recorder analysiert das Datenvolumen der neuen DICOM-Bilder zur Erstellung einer neuen CD.

Als Befundungsstationen dienen PC mit hochauflösenden Graphikkarten und entsprechenden Monitoren. Jede Befundungsstation ist über das zweite lokale Netz (Intranet) mit den beiden DICOM-Datenbanken verbunden, so daß der Arzt von seinem Rechner direkt auf jedes Bild aus der Datenbank zugreifen und dieses mit Hilfe des DICOM-Viewers analysieren kann. Nach der Bildanalyse entscheidet der Arzt, ob das Bild gelöscht werden kann oder in der zentralen Datenbank zu speichern ist. Zum Zweck einer orts- und plattformunabhängigen Möglichkeit der Visualisierung von DICOM 3.0-Daten wurde ein Java-Applet-Viewer erstellt, welcher in jede Intranet- und Internetanwendung eingebunden werden kann. Dieses Java-Applet interpretiert beliebige Bilddaten nach dem DICOM 3.0- Format.

Die Kommunikationsverwaltung zwischen der Datenbank für Kurzzeitspeicherung, dem VODBS, den Befundungsstationen (die sich auf den Krankenhausstationen und Arztambulanzen befinden) über das zweite lokale Netz übernimmt der lokale Intranet-Server. Der Intranet-Server ist auf einem leistungsfähigen PC-Rechner oder einer Workstation installiert. Außerdem ist auf diesem Rechner die Java-DICOM-Viewer Software und eine Java-Software für die Unterstützung des VODBS installiert. DICOM-Dokumente werden zwischen Komponenten des PACS mittels DICOM-, TCP/IP- und HTTP-Protokoll übertragen.

Das letzte Element des PACS ist der Transmissions-Kontroll-Rechner (TKR). Er erzeugt die Verbindung zwischen beiden Datenbanken des PACS und einem autorisierten Benutzer im Internet. Auf dem TKR ist ein WEB-Server installiert. Die Bildübertragung zwischen dem Web-Server und dem Web-Browser wird über das TCP/IP Protokoll und über das HTTP-Protokoll abgewickelt. Die auf dem Client hergestellten Bilder, die aus der Bilddatenbank ausgewählt wurden, werden auf den Web-Server übertragen. Sie werden

durch eine Java-Anwendung empfangen und durch ein Java-Servlet verarbeitet. Die Ergebnisse werden zum Client mittels JAVA-DICOM-Viewer-Applet zurückübertragen, bzw. auf dem Web-Browser am Bildschirm angezeigt. Die Verschlüsselung und Identifikation der DICOM-Daten vor und nach Verbindung mit dem Internet-Kunden werde durch eine spezielle JAVA-Software des Security-Server implementiert.

Aufgrund des Java-Sicherheitskonzepts für Applets wurde für das serverseitige Speichern der Overlay- und Kommentierungsdaten der Weg über ein Java-Servlet gewählt, das seinerseits neue Dateien anlegen kann und für diese Dateien schreibrechte hat. Damit verbunden ist die fortlaufende Überprüfung der Zugangsberechtigung des anfordernden Applets. Das Speichern der Overlay- und Kommentardaten übernimmt ein Java-Servlet nach dem Senden der Daten an den Web-Server.

3.7 Schlußfolgerungen

In diesem Kapitel wurde ein neues Konzept für ein intranet-/internet-basiertes, modulares, radiologisches Informationssystem für Kliniken und niedergelassene Ärzte vorgestellt. Das Ziel, das mit diesem System verfolgt wird, ist der interne/externe Zugriff auf alle DICOM-Daten, die von den verschiedenen bildgebenden Abteilungen produziert und archiviert werden. Dies wurde durch die Verwendung der DICOM-Standardisierung, der modernen plattformunabhängigen Java-Programierung- und Intranet/Internet-Technologie erreicht.

Zum Kern des Systems gehören die digitalen, bildgebenden Modalitäten (Bildgebergeräte), das Archivierungssystem, der Security Rechner, die Befundungsstationen, die Praxenrechner und das Netzwerk. Dabei ist die Integration aller digitalen Modalitäten über den DICOM-Standard eine wesentliche Voraussetzung des Systemkonzepts.

Das Archivierungssystem besteht aus einer DICOM-Datenbank und einem verteilten Datenbanksystem. Das verteilte Datenbanksystem dient der langfristigen Speicherung aller akzeptierten, zur langfristigen Speicherung vorgesehenen DICOM-Bilder. Das verteilte Datenbanksystem besteht aus den objektorientierten DICOM-Datenbanken und den relationalen Datenbanken als selbständigen Objekten. Eine solche Architektur gibt dem Datenbanksystem Skalierbarkeit, gute Verteilbarkeit und eine hohe Geschwindigkeit bei der Datenmanipulation.

In bezug auf den Objektaufbau für das verteilte Datenbanksystem wurden in diesem Kapitel die praktischen Grundlagen des Informationsmodells für DICOM-Daten sowie bekannte Datenbankmodelle betrachtet. Die kurze Übersicht über heute vorhandene DICOM-Datenbanken hat gezeigt, daß die Betonung beim Aufbau von DICOM-Datenbanken auf kommerziellen relationalen Datenbanken liegt. Vor diesem Hintergrund wurde neue Datenstruktur einer relationalen DICOM- Datenbank zur Speicherung von Header-Attributen und einem Dateisystem für die Speicherung von Rohdaten als Original-, komprimierte und skalierte Bilder vorgeschlagen. Ferner wurde neue Klassenhierarchie für eine OODDB vorgeschlagen. Die vorgeschlagene Struktur der OODDB eignet sich sehr gut für verteilte Netzwerkimplementierung. Die Verwendung der JAVA-Technologie verleiht der Software der OODDB Unabhängigkeit vom Betriebsystem und große Flexibilität bei der Einführung im Klinikum und in der Praxis.

Alle DICOM-Daten archivierenden und bearbeitenden Komponenten des vorgeschlagenes intranet/internet-basiertes, radiologisches Informationssystem sind über das lokale (Intranet) und globale (Internet) Netzwerk verbinden. Dabei können durch den Einsatz moderner Java- und Web-Technologie (Web-Browser) beliebige Informationsquellen auf den PC- und Stationsarbeitsplätzen der Ärzte und Praxen verfügbar gemacht werden.

4. Algorithmen für die Bildkompression

4.1 Einleitung

Die Größe der medizinischen Bilder wächst heute schnell mit steigender Bild- und Grauwertauflösung der bildgebenden Geräte. Das Computer-Radiologie (CR)-Bilder eine Auflösung von ca. 4000x4000 Pixel und eine Grauwertauflösung von 12 Bit haben, ist keine Neuigkeit. Die effektive Archivierung und die schnelle Übertragung medizinischer Bilder zwischen dem Client und dem Server über das Telefonnetz sind strategisch wichtige Fragen für die Zukunft der Teleradiologie. Die Frage stellt sich, ob sich die Teleradiologie genau so schnell wie das Internet entwickelt oder lediglich auf das Krankenhaus beschränkt bleibt.

Das Problem besteht dabei in der Übertragung großer Datenmengen in kurzer Zeit. Im günstigsten Fall erfordert die Übertragung eines CR-Bildes bei einer mittleren Übertragungsgeschwindigkeit von 128 Kbit pro Sekunde (ISDN) bis zu 40 Minuten (Abbildung 4.1). Für die Speicherung eines derartigen Bildes sind etwa 24 MByte erforderlich.

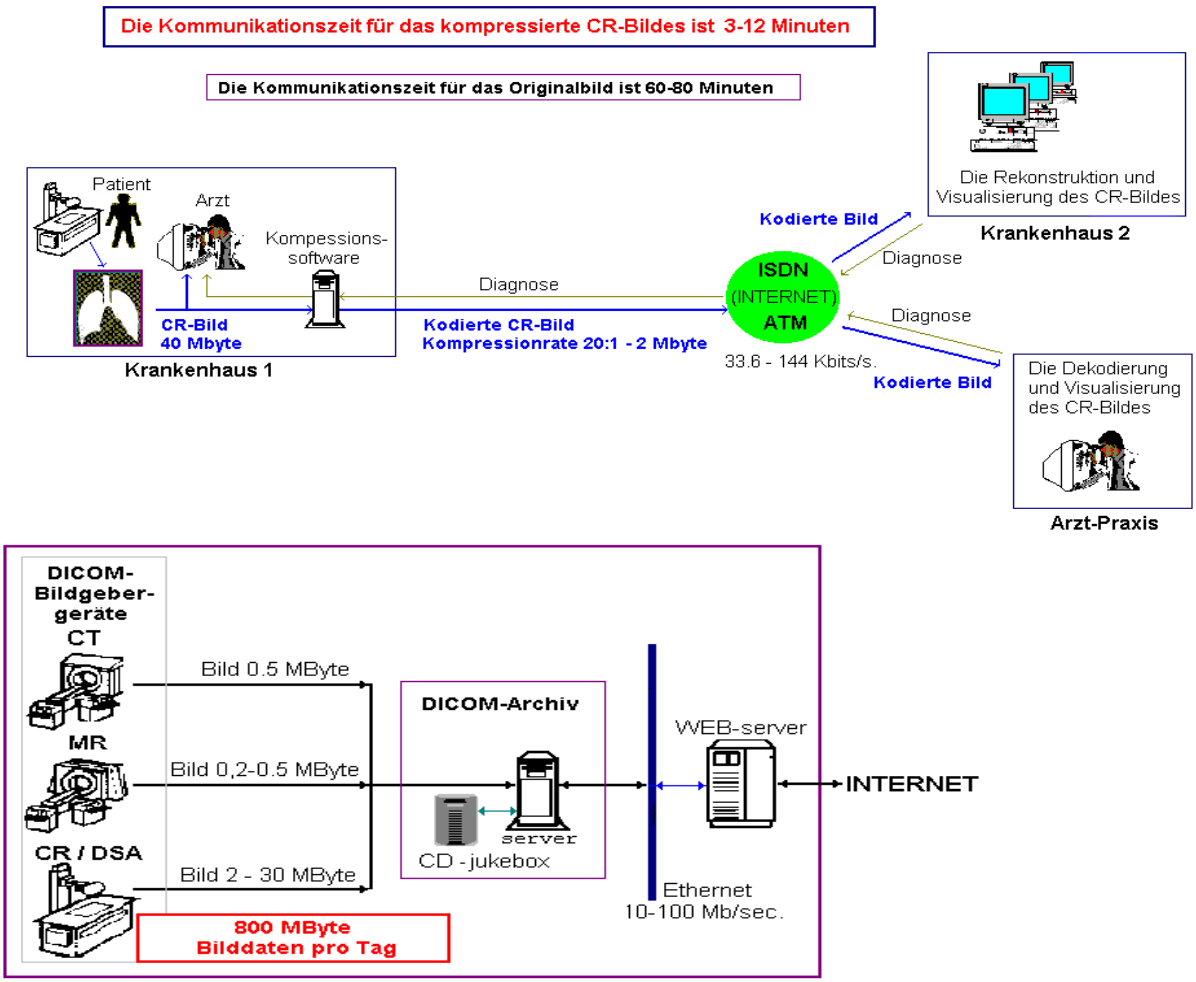


Abb. 4.1: Bildübertragungs - und Archivierungsprobleme in der Telemedizin

Ein möglicher Weg zur Lösung dieses Problems ist die Kompression von Bildern auf der Server-Seite, bevor diese kodierten Bilder über ein Datennetz übertragen und auf der Client-Seite wieder rekonstruiert werden. Dabei ist es auf der einen Seite wichtig, die Komprimierung mit einem großen Koeffizienten zu realisieren und auf der anderen Seite darf der Verlust an Information für den Radiologen visuell nicht bemerkbar sein.

Aus der Literatur sind zahlreiche Methoden zur Kompression von Digitalbildern bekannt [55-98]. Man unterscheidet zwischen Algorithmen zur verlustfreien und verlustbehafteten Kompression. Hohe Kompressionsraten lassen sich dabei nur mit den letztgenannten Methoden erreichen. Die Komprimierung mit einem relevanten großen Koeffizienten und die unsichtbare Qualitätsminderung des rekonstruierten Bildes können bei der Benutzung des Komplexitätsalgorithmus erreicht werden. Dieser Algorithmus besteht aus verschiedenen verlustfreien und verlustbehafteten Kompressionsalgorithmen. Die Kombination dieser Algorithmen ist von den Eigenschaften des Originalbildes abhängig.

Dieses Kapitel ist der Erarbeitung neuer Verfahren zur Beschreibung der Ortstruktur von radiologischen Bildern und neuer Komplexitätsalgorithmen zur fast „verlustfreien“ Bildkompression gewidmet.

4.2 Digitale Darstellung des Bildsignals und gebräuchliche Bildformate

Jedes digitale Grauwert-Bild - L (Abbildung 4.2) ist ein endliches zweidimensionales (2D) Array von beschränkten nicht-negativen ganzzahligen Zahlenwerten $l(x, y)$ wobei

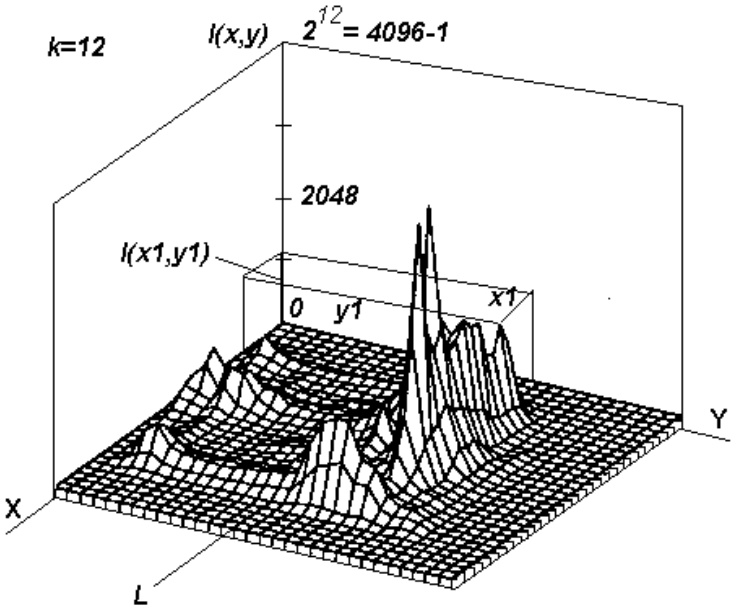


Abb. 4.2: Die digitale Darstellung des Bildsignal

die Position der einzelnen Zahlenwerte durch die Koordinaten x und y im Array gegeben sind. Es gilt $1 \le x \le X$ und $1 \le y \le Y$. Digitale radiologische Bilder werden von unterschiedlichen Modalitäten (Computer-Tomographie (CT), Ultrasound (US), Magnetic-Resonance-Tomographie (MR), Digitales-Röntgen (DR), Nuclear-Magnetic-Resonance (NM) etc.) erzeugt. Die Zahl der Graustufen des Bildes ist abhängig von der Modalität des

Bildgebergerätes und kann von 8 Bit pro Pixel bis 16 Bit pro Pixel variieren. Einige Bildgebergeräte (US, NM) erzeugen Farbbilder mit einer Farbauflösung von 24 Bit pro Pixel. Das ist auch die Art, wie Bilder im Computer gespeichert werden. Außerdem spielt außer der Bildortsauflösung und der Grauwertauflösung noch ein physikalischer Parameter eine große Rolle bei der Charakterisierung des Bildes - das Signal–Rausch-Verhältnis (signal to noise ratio - *SNR*). Dieser Parameter definiert die Größe des auf dem Originalbild vorhandenen Rauschens mit der Schätzung der Variation D_R und dem Mittelwert 0. Für das digitale Grauwertbild existieren folgende Grundbegriffe [55]:

- Bildauflösung --> Matrix $X * Y$ Pixel
- Grauwertauflösung - Anzahl der Grauwerte pro Pixel
- zeitliche Auflösung (Zahl der Bilder pro Sekunde oder Zahl der Bildschichten pro Zentimeter)
- Anzahl der Bitebenen – K
- Signal–Rausch-Verhältnis - *SNR*

Insgesamt hat ein zweidimensionales radiologisches Bild die Größe $X * Y * K$ Bit, wobei 2^K - die Zahl der Graustufen ist. Die Speicherung eines 9-,10-,11-,12-Bit Bildes erfordert zwei Byte pro Pixel im Speicher des Archivs, für Farbbilder benötigt man drei Byte pro Pixel.

Im wesentlichen wird zwischen drei verschiedenen Klassen von Bildformaten unterschieden: Bitmap-, Vektor- und Meta-Bilddateien [56,57,58].

Bitmap-Formate:

- Die Datei enthält Informationen über Ort, Helligkeit und Farbe jedes Punktes;
- Die Punkte (Pixel) sind voneinander unabhängig;
- Es kann ohne großen Rechenaufwand gespeichert werden; die Datei nimmt aber unkomprimiert viel Speicherplatz ein;
- Auch komprimierte Formate oder nachträglich gepackte Dateien nehmen oft noch mehrere hundert Kilobyte ein;
- Veränderungen von Größe, Farbe und sonstigen Eigenschaften können nicht immer ohne Qualitätsverluste vorgenommen werden.

Vektor-Formate:

- Die Datei besteht aus mathematischen Beschreibungen der dargestellten Bildelemente
- (Rechtecke, Linien, Kurven, Flächen, Körper...);
- Für die Ausgabe auf einen Bildschirm oder Drucker ist mehr oder weniger Rechenaufwand erforderlich. Dafür ist eine Vektor-Datei oft wesentlich kompakter als eine Bitmap-Datei, kann aber auch um einiges größer sein, abhängig von der Komplexität der mathematischen Beschreibungen;
- Veränderungen von Größe, Farbe und sonstigen Eigenschaften können ohne Qualitätsverluste vorgenommen werden.

Meta-Formate:

- Die Datei kann sowohl aus Bitmap-Anteilen als auch aus Vektor-Grafiken bestehen;
- Zur Speicherung auf Bildplatte oder zur Erzeugung eines Videoclips muß die Datei zunächst in eine Bitmap-Datei umgewandelt werden;
- Das Format ist häufig kompakter als ein Bitmap-Format;
- Veränderungen der Eigenschaften erfolgen ohne Qualitätsverluste (Vektorielle Anteile).

In telemedizinischen Anwendung werden heute Bitmap-Files verwendet [56]. Im Anhang B werden die momentan gebräuchliche Bitmap-Formate betrachtet.

4.3 Probleme der Kompression von medizinischen Bildern

Der Algorithmus der radiologischen Bildkomprimierung hängt von der Kompressionsrate und der Qualität des rekonstruierten Bildes ab. Die Kompressionsrate ist das Verhältnis aus Dateigröße des ursprünglichen und des komprimierten Bildes. Die Qualität des rekonstruierten Bildes wird durch die Differenz zwischen Originalbild und rekonstruiertem Bild ermittelt. Je höher die Qualität des rekonstruierten Bildes desto weniger Differenz besteht zwischen den Bildern. Die hochqualitativen medizinischen Bilder zeigen die feine Struktur- und Funktionalinformation menschlicher Organe und unterstützen eine sichere Diagnoseerstellung.

In den verschiedenen Gebieten medizinischer Untersuchungen ist die erforderliche Qualität der Bilder unterschiedlich. Zum Beispiel reicht für eine US-Untersuchung in den meisten Fällen die Bildauflösung 320x240 oder 640x480 Pixel und die Grauwertauflösung von 8 Bit pro Pixel aus, während andererseits für CR (Computer Radiography)-Untersuchungen mindestens die Bildauflösung 2048x2048 Pixel und die Grauwertauflösung 10 Bit pro Pixel notwendig ist.

Fast alle medizinischen Bilder (verschiedene Modalitäten) haben feine Strukturen mit vielen kleinen Details, die eine wichtige Rolle bei der Diagnose spielen. In dieser Hinsicht muß das Rekonstrukt dem Originalbild so ähnlich sein, daß sichtbare Fehler unterbleiben. Bei einer verlustbehafteten Komprimierung geht jedoch immer Information verloren. Dabei gilt : je höher die Kompressionsrate desto niedriger die Qualität des rekonstruierten Bildes. Die meisten Kompressionsverfahren erlauben es, die Kompressionsrate und damit den Verlust an Bildqualität zu verändern. Welche Kompressionsverfahren und Methoden man für die adaptive Kompressionsteuerung überhaupt wählt, entscheidet sich je nach Bildmaterial und Randbedingungen, wie z.B. die Größe der kodierten Bilddatei, die Qualität des rekonstruierten Bildes und der Kompressions- / Dekompressionsgeschwindigkeit.

Berücksichtigt man das oben Gesagte, ergeben sich die folgenden Schwerpunkte bei der medizinischen Bildkomprimierung [56, 64-67,96-98]:

- Der ausgewählte Kompressionsalgorithmus soll einen höheren Wert als Kompressionsfaktor liefern, als andere Algorithmen bei gleichen Bedingungen für die Qualität des rekonstruierten Bildes.

- Der Algorithmus soll keine sichtbare Verzerrung zum Originalbild ergeben, d.h. das *Peak Signal to Noise Ratio* (PSNR) für die acht höherwertigen Bitebenen des Originalbildes und des dekodierten Bildes soll größer als 40dB sein.
- Die benötigte Zeit für die Kompression und besonders für die Dekompression soll weniger als 1-2 Sekunden sein.

Für die Telemedizin ist also einerseits die Komprimierung mit einem großen Koeffizienten wichtig und andererseits muß sich der Verlust der Information auf Bits beschränken, die im Sinne der medizinischen Diagnose irrelevant sind. Ergänzend werden kompakte Algorithmen entwickelt, die eine schnelle Dekodierung auf der Client-Seite ermöglichen.

4.4 Überblick über bekannte Bildkompressionverfahren

Dieser Teil des Kapitel ist bekannten Verfahren zur Bildkompression gewidmet [84-94]. Alle Kompressionsalgorithmen kann man in zwei Gruppe aufgeteilen. Die erste Gruppe beinhaltet Algorithmen ohne Informationsverlust (Huffman-Kodierung, RLE**,** LZW-Komprimierung), die zweite Gruppe besteht aus verlustbehafteten Algorithmen (JPEG, Wavelet-Codering, Vektor Quantisation). Die Auswahl eines Algorithmus ist abhängig von den jeweiligen Bildeigenschaften. Wenn die Bildeigenschaften den Eigenschaften des Bildmodells entsprechen, die im Kompressionsalgorithmus verwendet werden, dann liefert dieser Algorithmus größere Kompressionsraten als andere Algorithmen, bei gleichen Auflagen für die Qualität des rekonstruierten Bildes.

4.4.1 Verfahren zur verlustfreien Kompression

Huffman-Kodierung

Die Idee der Huffman-Kodierung [55,68] geht auf das Prinzip des Morsealphabets zurück. Dort werden den häufig vorkommenden Symbolen kürzere Codes zugeordnet als den seltener vorkommenden. Nach Huffman ordnet man alle Symbole zunächst nach ihrer Häufigkeit in einer Tabelle. Die beiden seltensten erhalten als letzte Codeziffer eine 0 und eine 1. Beide werden in der Tabelle mit der Summe ihrer Wahrscheinlichkeiten zusammengefaßt. Die Häufigkeitstabelle hat damit ein Element weniger. Wieder sucht man die beiden seltensten Elemente und stellt ihrem Code eine 0 und eine 1 voran. Nach Zusammenfassen ihrer Wahrscheinlichkeiten beginnt das Spiel von vorn. Alle Symbole sind kodiert, wenn nur noch ein Element in der Tabelle übrig ist.

Diese Methode wird auch als Entropie-Kodierung bezeichnet. Entropie-Kodierungen kodieren nicht die zu übertragenden Daten, sondern die Symbole der Quelle. Entropie-Kodierungen arbeiten verlustfrei. Die Kompressionsrate des Algorithmus hängt von der Grauwertentropie des Originalbildes ab. Je geringer die Entropie ist, desto höher die Kompressionsrate. Diese Eigenschaft kann man zur Abschätzung der Effektivität der Algorithmen für jedes zu kodierende medizinische Bild benutzen. Durch Anwendung dieses Verfahrens kann man medizinische Bilder mit Kompressionsraten bis zu 2:1 komprimieren.

RLE

Das RLE-Verfahren (Run Lenght Encoding) [55,69] ist das einfachste unter den hier vorgestellten. Bei diesem Verfahren werden Folgen gleicher Bytes durch ein Paar (Zeichen, Anzahl) dargestellt: AAAAAABBBCCDDDDD also durch A6B3C2D5. In diesem Beispiel ergibt sich eine Reduktion von 50% (16:8 Byte). Run Length Encoding eignet sich besonders für Dateien mit langen Folgen gleicher Zeichen, z.B. Schwarz/Weiß-Grafiken. Aus diesem Grund wird dieses Verfahren auch häufig für Fax-Formate verwendet, in denen sehr große weiße Flächen nur gelegentlich von schwarzen Buchstaben unterbrochen werden. Dateien mit häufig wechselnden Bytes sind dagegen nicht gut geeignet. Im schlechtesten Fall wird die neue Datei doppelt so groß wie das Original. Die Effektivität von RLE-Verfahren hängt vom Mittelwert der Streckenlänge der Bildzeilen ab. Je größer der Mittelwert ist, desto höher wird die Kompressionsrate. Die Kompressionsrate für die medizinische Bildkodierung mittels RLE-Verfahren beträgt bis zu 1.8:1.

LZW-Komprimierung

Das LZW-Verfahren [59,60,70] wurde nach seinen Entwicklern Lempel, Ziv und Welch benannt. Diese Kompressionsmethode wird von vielen Grafikformaten (z. B. GIF) genutzt. Dahinter steckt die Idee, häufiger vorkommende Symbolfolgen in Teilketten zu zerlegen. Bei dieser Kodierung wird nach Folgen einzelner Bytes gesucht und diese durch eine Angabe von Anzahl und Wert ersetzt. Das LZW-Verfahren sucht nach sich wiederholenden Zeichenketten und ersetzt diese durch einen Verweis auf die identische Zeichenfolge. Jede Teilkette erhält einen Index, der in einer Tabelle abgelegt wird. Der Ausgabecode besteht nur aus den Indizes dieser Tabelle.

Ein Beispiel: Der Zeichnenstrom "whoiswho" zerlegt man geschickterweise in die Teilketten "who" und "is". In der Zuordnungstabelle erhält "who" den Index 0 und "is" den Index 1. Das Codewort lautet dann 101. Dieses Verfahren arbeitet mit einer Zuordnungstabelle in der Verweise auf die einzelnen Zeichenketten gespeichert werden. Von den ursprünglich acht Zeichnen sind nur noch drei übrig. In natura ist es allerdings doch etwas komplizierter, da die Tabelle nicht mit abgespeichert wird und bei jeder Kompression und Dekompression erst erzeugt werden muß. Hinzu kommt, daß die Tabelle nicht beliebig lang werden darf, weshalb ihre Größe bei konkreten Implementierungen beschränkt wird. Im Gegensatz zum Huffman-Verfahren, in dem jedes Quellensymbol codiert wird, werden bei dem LZW-Verfahren ganze Symbolfolgen zu jeweils einem neuem Symbol zusammengefaßt. Tests des LZW-Verfahren auf dem Gebiet der Komprimierung medizinischer Bilder ergab Kompressionsraten bis zu 2.2:1, wobei die Effektivität des Verfahrens vom Mittelwert der Streckenlänge der Bildzeilen abhängt. Je größer der Mittelwert ist, desto höher wird die Kompressionsrate.

Für den Vergleich der oben beschriebenen verlustfreien Algorithmen sind in der Tabelle 4.1 die Ergebnisse einer statistischen Modellierung dargestellt, bei der mehr als 500 DICOM-Bilder komprimiert wurden. Tabelle 4.1 beinhaltet die Werte für die Kompressionsraten bei verschiedenen Bildmodalitäten.

Insgesamt ergeben alle verlustfreien Kompressionsmethoden kleine Kompressionsraten. Oft ist die Kompressionsrate nach verlustfreier Komprimierung von medizinischen Bildern geringer als 2:1, was in den meisten Fällen für medizinische Anwendungen nicht ausreicht.

Die verlustfreien Kompressionsmethoden geben gute Ergebnisse beim Kodieren der höheren Bitebenen des Originalbildes. In diesem Fall erreicht die Kompressionsrate den

Tabelle 4.1: Kompressionsraten für verlustfreie Kompressionsalgorithmen.

	CT	**MR**	**US**	**NUK**	**CR**	**DS**
Huffman - Kodierung	1.4 : 1	1.7 : 1	1.6 : 1	1.5 : 1	1.7 : 1	1.7 : 1
LZW - Kodierung	1.3 : 1	1.5 : 1	1.8 : 1	1.5 : 1	1.4 : 1	1.4 : 1
Arithmetische Kodierung	1.4 : 1	1.6 : 1	1.5 : 1	1.4 : 1	1.6 : 1	1.6 : 1

Faktor 10-15. Erst die Anwendung verlustbehafteter Verfahren ermöglicht es, die Kompressionsrate zu steigern. Der Preis dafür ist eine Qualitätsminderung des rekonstruierten Bildes. Aber durch Manipulation der Kompressionsrate kann man die Differenz zwischen dem Original- und dem decodierten Bild auf visuell nicht unterscheidbarem Niveau halten.

4.4.2 Verfahren zur verlustbehafteten Kompression

JPEG-Kompressionsverfahren

JPEG [59,61,62] ist ein standardisiertes Bildkompressionsverfahren, das von der Joint Photographic Experts Group entwickelt wurde. Es ist ein aus mehreren Verfahrensschnitten zusammengesetztes Verfahren. Die Komprimierung der Bilder erfolgt in den folgenden Schritten:

- Teilen des Originalbildes in Blöcke von 8x8 Pixel
- Diskrete Cosinus-Transformation (DCT) der 8x8 Pixel großen Blöcke
- Quantisierung
- Lauflängen-Kodierung
- Huffman-Kodierung

Die Kompressionsrate der JPEG-Kompression ist abhängig vom Parameter Q, der auf dem Schritt „Quantisierung“ den Informationsverlust bestimmt. Je größer der Parameter Q ist, desto geringer ist der Informationsverlust, d.h. desto besser ist die Qualität des rekonstruierten Bildes und entsprechend geringer ist die Kompressionsrate. Aufgrund dieser Abhängigkeit wird der Parameter Q für JPEG-Verfahren festgelegt.

Das JPEG-Verfahren ist ein „Standard“ für die Komprimierung der Bilddaten in DICOM-Dateien.

Wavelet-Kompression

Eine Wavelet-Transformation wandelt das Orginalbild in Wavelet-Koeffizienten um, die im Prinzip verkleinerte Kopien des Ausgangsbildes beschreiben [71-77]. Zusätzlich bestehen die Koeffizienten aus hoch- und tiefpaßgefilterten Versionen dieser Bilder und sind in hohem Maße redundant. Die eigentliche Kompression erfolgt im sogenannten Quantisierungsschritt. Verschiedene Strategien fassen dabei unwichtige Bilddetails zusammen, sofern man nicht gleich auf sie verzichtet. Nach einer anschließenden Kodierung (z. B. Huffman-Verfahren) liegt das Ergebnis vor. Dieses Verfahren arbeitet global, es verwendet also keine Blockbildung. Um aus den Daten wieder ein Bild zu erhalten, werden alle Schritte in umgekehrter Reihenfolge ausgeführt.

Vektorquantisierung

Vektorquantisierung (VQ) [78-83]- Die Bildkompression basiert auf der Aufteilung des Originalbildes in Blöcke der Größe $M'*N'$. Jeder Block ist ein Vektor der Dimension $F = M'*N'$ (normalerweise von 4x4 bis 8x8 Pixel). Da die Werte des Originalbildes korreliert sind, wird für jede Bildmodalitätsart ein eigenes Codebuch erzeugt. Wird bei der Erzeugung des Codebuches eine aus zu wenigen Bildvorlagen bestehenden Trainingsfolge verwendet, so besteht die Gefahr, daß das Codebuch sich stark an die Statistik dieser speziellen Bilder anpaßt; es ist dann übertrainiert. Je mehr Vektoren das Codebuch enthält, desto besser wird die Qualität des rekonstruierten Bildes und desto geringer wird die Kompressionsrate sein. Abbildung 4.3 stellt die Vektoren eines Codebuches (256 Vektoren mit je $F = 4*4 = 16$ Bildpunkten) dar, welches an einer Gruppe von 15 verschiedenen Bildern trainiert wurde. Die Etappe der Codierung besteht aus drei Schritten.

Blöckbildung. Das Bild wird in kleine Blöcke (z.B. 4x4 Pixel) zerlegt.

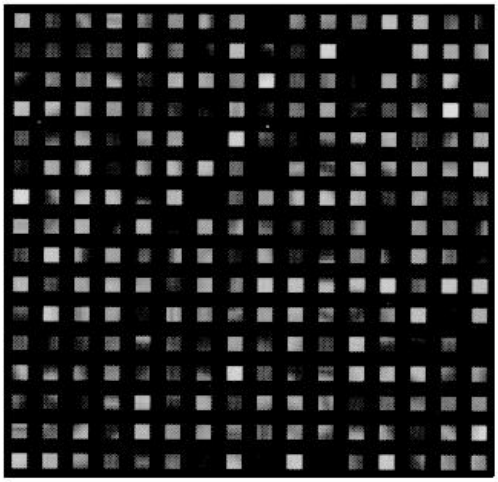


Abb. 4.3: 256 Vektoren mit Größe 4x4 Pixel eines Codesbuches

Codierung. Für jeden Block wird der Vektor aus dem Codebuch gesucht, der das folgende Kriterium auf der Menge C der Vektoren des Codebuches minimiert

$$J_{m,n} = \sum_{x=0}^{3} \sum_{y=0}^{3} (l_{m,n}(x,y) - C_u(x,y))^2 \qquad \boldsymbol{u = 1,...,B} \qquad \textbf{(4.1)}$$

wobei $l_{m,n}(x,y)$ der Grauwert des Pixels x, y von Block m, n des Originalbildes L ist; C_u- ist der Grauwert des Pixel x, y von Vektor u des Codebuchs; B- ist die Anzahl der Vektoren im Codebuch. Der ausgewählte Block des Originalbildes wird mit der Nummer des gefundenen Vektors codiert.

Huffman-Kodierung. Diese Kodierung bildet den Schluß der Bearbeitung. Das Verfahren nutzt die unterschiedliche statistische Wahrscheinlichkeit des Auftretens eines Vektors aus dem Codebuch für des konkrete Originalbild.

Bei der Decodierung wird vorausgesetzt, daß auf der Client-Seite das bei Codierung verwendete Codebuch existiert.

Für den Vergleich der oben beschriebenen verlustbehafteten drei Algorithmen wurde eine statistische Modellierung durchgeführt, bei der mehr als 500 DICOM-Bilder komprimiert wurden [66]. Die Tabelle 4.2 beinhaltet die höheren Werte der Kompressionsraten für die verschiedenen Bildmodalitäten bei gleichen Anforderungen hinsichtlich der Qualität des rekonstruierten Bildes. Es dürfen keine visuellen Unterschiede zwischen dem rekonstruierten Bild und dem Originalbild existieren, d.h. die PSNR muß größer als 40 dB sein.

Tabelle 4.2: Kompressionsratio für verlustfreie Kompressionsalgorithmen

DICOM-Bildmodalität	**CT**	**MR**	**US**	**NUK**	**CR**	**DS**
JPEG – Kodierung	4 : 1	4 : 1	6 : 1	3 : 1	7 : 1	7 : 1
Wavelet – Kompression	5 : 1	4 : 1	6 : 1	3 : 1	8 : 1	7 : 1
VQ – Kodierung	5 : 1	6 : 1	4 : 1	4 : 1	5 : 1	5 : 1

JPEG ist außerordentlich beliebt, im Internet gehört es zu den gebräuchlichsten Kompressionsmethoden. Das Verfahren hat den unbestreitbaren Vorteil, daß es standardisiert ist. JPEG eignet sich gut zur Komprimierung von CR-, DS-, US-Bildern mit Größen von 1024x1024 Pixeln und einer Grauwertauflösung von 8 bis 12 Bit. Die Kompressionsrate für diese Bilder, bei denen die PSNR größer als 40 (dB) ist, reicht von 4:1 bis zu 8:1. Zur Kompression der MR-, CT-, NM-Bilder mit Größen bis zu 512x512 Pixel und einer Grauwertauflösung von 8 bis 12 Bit eignet sich JPEG weniger, denn „harte“ Kanten und feine Bilddetails zeigen besonders deutlich störende Artefakte der JPEG-Kompression.

Wavelets arbeiten bei hohen Kompressionsraten besser als JPEG. Zum Teil sind sie auch der VQ-Kompression überlegen. Die Wavelet-Kompression zeigt die besten Ergebnisse mit einer niedrigeren Fehlerrate als JPEG- und VQ-Kompression bei größtmöglicher Kompression der CR-, DS-, US-Bilder und bei einer PSNR > 40dB. Die Wavelet-basierten Verfahren ermöglichen Kompressionsraten, die in der Größenordnung von 4:1 bis 9:1 liegen. Wann ein Wavelet-Verfahren geeignet ist, läßt sich nicht eindeutig sagen. Es gibt eine Vielzahl von Basisfunktionen, die für verschiedene Bildtypen unterschiedlich gut geeignet sind. Die größte Stärke der Wavelets liegt zur Zeit in der Vielzahl der verwendeten Basisfunktionen. Anderseits steht diese Eigenschaft einer Standardisierung im Wege.

VQ-Verfahren sind stark bei der Komprimierung hochkontrastreicher Bilder mit vielen Details (MR-, CT-, NUM- Bilder mit Größen bis zu 512x512 Pixel und einer Grauwertauflösung von 8 bis 12 Bit). Die Modellierungsergebnisse des VQ-Algorithmus für MR und CT Bilder mit 12 Bit Auflösung zeigt eine Kompressionsrate von 4:1 bis 6:1 bei einer Begrenzung des PSNR auf >40dB. Die schwache Seite der VQ-Verfahren ist die fehlende Standardisierung.

Allgemein sind nicht alle Ergebnisse direkt miteinander vergleichbar – schon gar nicht bei Verwendung verschiedener Bilder. Die Stärken und Schwächen der einzelnen Verfahren treten aber oft klar zu Tage und lassen Schlüsse zu, bei welchem Bildtyp welche Methode am besten geeignet ist. Auf diese Weise können positive Ergebnisse bei der Benutzung des Komplexitätsalgorithmus erreicht werden. Dieser Algorithmus besteht aus verschiedenen verlustfreien und verlustbehafteten Komressionsalgorithmen. Die Kombination dieser

Algorithmen ist von den Bildeigenschaften abhängig. Welche Bildeigenschaften man benutzt, um Bilder zu klassifizieren, wie man bei der Klassifikation vorgeht, und welche Kompressionsalgorithmen in welcher Kombination verwendet werden, beantworten die folgenden Abschnitte.

4.5 Parameter zur Beschreibung und Klassifikation medizinischer Bilder

Wie stark sich ein Digitalbild ohne Verlust nützlicher Information komprimieren läßt, hängt im wesentlichen von dessen Bildinhalten ab. Das heißt, daß für ein Bild mit verschiedenen Kompressionsalgorithmen unterschiedliche Kompressionsraten erzielt werden können. Somit besteht die zu lösende Aufgabe darin, für jedes Bild durch eine automatische adaptive Auswahl einen geeigneten Kompressionsalgorithmus aufgrund einer vorläufigen Analyse des Bildinhaltes und der Anforderungsspezifikation zu finden.

Zu den gebräuchliche Bildparametern gehören statistische Parameter erster und zweiter Ordnung [55,56]:

- Mittelwert

$$m_0 = \frac{1}{X * Y} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} l(x, y) \tag{4.2}$$

- Varianz

$$m_1 = \frac{1}{X * Y} \sum_{x=0}^{X-1} \sum_{y=0}^{Y-1} (l(x, y) - m_l)^2 \tag{4.3}$$

- Grauwertentropie

$$H_k = -\sum_{k=0}^{Q-1} p(k) * \log_2(p(k)) \tag{4.4}$$

wobei $p(k)$ die Auftretenswahrscheinlichkeit für Grauwert k im Bild ist; Q Anzahl der Grauwerte pro Pixel

- Grauwertentropie des Differenzbildes B, das aus dem Originalbild L erzeugt wurde, wobei

$$b(x, y) = \left| l(x+1, y) - l(x, y) \right| \tag{5.5}$$

Eine andere Art von Bildparametern sind spektrale Parameter (DFT, diskrete Cosinus-Transformation (DCT), diskrete Sinus-Transformation (DST), diskrete Wavelet-Transformation (DWT):

- Spektralverteilung
- Energie der Spektralverteilung
- Entropie der Spektralverteilung

Eine wichtige Bedeutung für die Beschreibung des Bildinhalts spielen struktur- und inhaltsorientierte Bildparameter:

- Texturen der lokalen Bildbereiche (periodischer oder nicht-periodischer Texturen, die Verteilung der lokalen Maxima und Minima der Helligkeit, deren Abstände und Ausrichtungen);
- Größe, Form, Orientierung, Grauwert oder Mittelwert vom Grauwert der Bildsegmente;
- statistische Eigenschaften obengenannter Parameter der Bildsegmente.

Es gibt noch viele, wesentlich unterschiedlichere Methoden zur Bildbeschreibung. Allerdings findet man in der Literatur zur Zeit kein Verfahren zur Analyse des Bildinhaltes, das es ermöglicht, die Ortsstruktur eines Bildes (Anzahl, Größe und Orientierung grober, mittlerer und feiner Details des Bildes, usw.) unter Verwendung nur weniger Parameter zu fixieren. Im folgenden Text werden zwei neue Methoden zur Beschreibung der Bilder vorgeschlagen, die durch den Autor entwickelt wurden.

Die erste Methode [65] verwendet zur Analyse des Bildes zwölf Parameter. Diese Parameter werden durch vier Iterationsschritte der Wavelettransformation berechnet. Sie repräsentieren die relative Anzahl an Waveletkoeffizienten für jeden Quadrant einer Waveletiteration, deren absolute Amplitudenwerte höher als ein bestimmter Schwellwert sind. Die Parameter werden mit folgendem Ausdruck berechnet:

$$k_{i,v} = \frac{n_{i,v}}{N_{i,v}} \tag{4.6a}$$

wobei $k_{i,v}$: Parameter für die Waveletiteration i und den Quadranten u für i =0,1,2,3 und u = 1,2,3 (Abbildung 4.4). Quadrant 1 besteht aus Hochfrequenzelementen der Horizontalrichtung, Quadrant 2 besteht aus Hochfrequenzelementen der Vertikalrichtung, Quadrant 3 besteht aus Hochfrequenzelementen beider Richtungen; $N_{i,v}$: Gesamtzahl der Waveletkoeffizienten der Iteration i in Quadrant u; $n_{i,v}$: Zahl der Waveletkoeffizienten der Iteration i in Quadrant u, deren absoluten Amplitudenwerte höher als ein Schwellwert $R_{i,v}$ sind. $R_{i,v}$ wird wie folgt berechnet:

$$R_{i,v} = 2^i * \sqrt{4 * D_R} \qquad u = 1,2 \tag{4.6b}$$

$$R_{i,v} = 2^i * \sqrt{2 * D_R} \qquad u = 3 \tag{4.6c}$$

wobei D_R: Schätzung der Variation des das Bild überlagernden Rauschens. Dabei muß man berücksichtigen, daß das Rauschen unkorreliert im Ortsbereich ist und eine Gauß'sche Wahrscheinlichkeitsfunktion mit dem Mittelwert 0 und der Variation D_R darstellt. Für die Wavelettransformation wird das Basiswavelet nach Daubechies mit 4 Koeffizienten verwendet. Der Grund für dieses Vorgehen liegt in der Tatsache begründet, daß der Zeitaufwand für die Berechnung der Wavelettransformation mit einem Basiswavelets mit 4 Koeffizienten geringer ist, als wenn man mehr Koeffizienten verwenden würde.

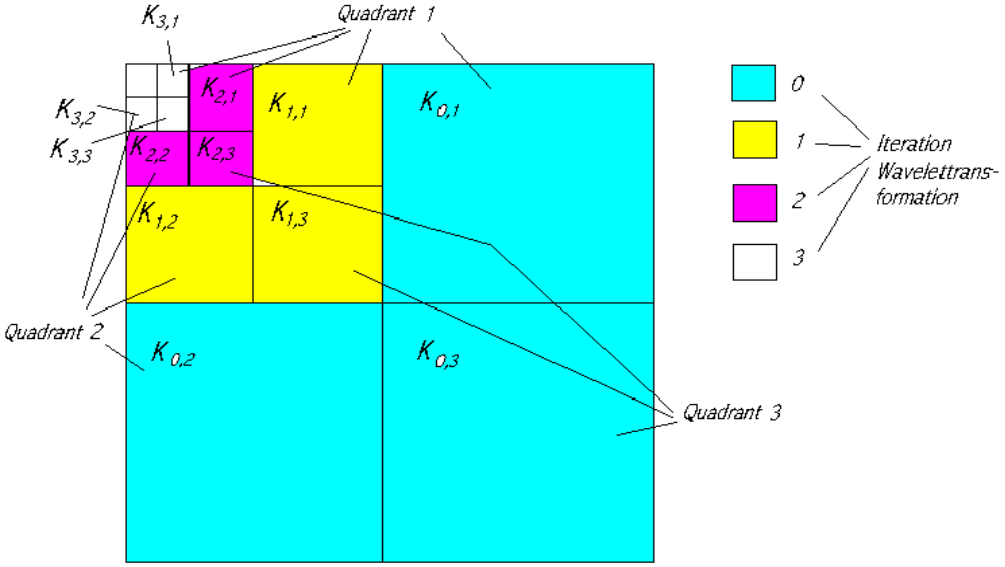


Abb. 4.4: Ablaufschema der Wavelettransformation

Experimente, in denen viele verschiedenartige medizinische Bilder mit obigem Verfahren analysiert wurden, ergaben folgende Ergebnisse. Die Parameter für den Iterationsschritt i=0 liefern Informationen über Anzahl und Orientierung kleiner Details und Konturen auf dem Bild. Die Elemente beider Ortsrichtungen sind nicht größer als 2-3 Pixel. Ebenfalls

Bild 1

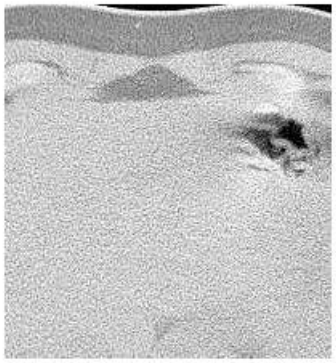


Bild 2

Abb. 4.5: Beispiel zum Beschreibungsverfahren von medizinischen Bilder auf dem Basis Waveletkoeffizienten

Tabelle 4.3: 12-Waveletparameter für das Bild 1

u /i	0	1	2	3
1	18,3	19,9	15,6	3,5
2	23,9	22,6	17,9	5,4
3	18,9	23,3	26,3	9,3

Tabelle 4.4: 12-Waveletparameter für das Bild 2

u /i	0	1	2	3
1	28,1	4,5	3,2	1,9
2	29,2	2,1	1,4	1,1
3	34,2	19,2	2,1	0,8

repräsentieren die Parameter für den Iterationsschritt i=1 Informationen über Bildobjekte, die nicht größer als 4-5 Pixel sind. Wenn die Dimension der Bilder mehr als 200x200 Pixel beträgt, beschreiben diese Parameter die Struktur des Bildes im Ortsbereich, die nicht sichtbar sind. Im Gegensatz dazu repräsentieren die Parameter für die Iterationsschritte i=2,i=3 die Anzahl und Orientierung mittlerer und großer Details auf dem Bild. Die Größe dieser Details beträgt mehr als 7-8 Pixel. Die Differenz zwischen Parametern, die zum selben Iterationsschritt gehören, ermöglicht es, die Orientierung der Details und Konturen auf dem Bild zu schätzen. Je größer ein Parameter in Bezug auf einen anderen Parameter ist, desto mehr geht die Orientierung der Objektkonturen in die Richtung, die durch den größeren Parameter bestimmt wird. Dabei repräsentiert Quadrant 1 die Horizontalrichtung, Quadrant 2 repräsentiert die Vertikalrichtung. Ein Beispiel für das vorgeschlagene Beschreibungsverfahren von medizinischen Bildern auf der Basis von Waveletkoeffizienten ist in Abbildung 4.5 dargestellt.

Das zweite von Autor neu entwickelte Verfahren zur Beschreibung der Ortstruktur radiologischer Bilder basiert auf einer segmentorientierten Beschreibung des Bildes [64,95]. Dafür wird jedes Bild segmentiert und für jedes Segment Fläche, Grauwert, Breite und Höhe, Ortskoordinaten, Umfang, Extremwert und Formkoeffizient berechnet. Danach wird jedes Bild in gleichgroße Blöcke aufgeteilt und für jeden Block sowie für das gesamte Bild der Mittelwert der Segmentflächen, die Segmentgrauwertvariation, der Segmentformkoeffizient, die Segmentbreite und Segmenthöhe, sowie die Anzahl der Lokalensegmentminima, Lokalensegmentmaxima und die Grauwertvariation der Extremsegmente berechnet. Das Segment ist ein beschränkter Bildbereich von beliebiger Form, dessen Pixel alle den gleichen Grauwert haben (Abbildung 4.6).

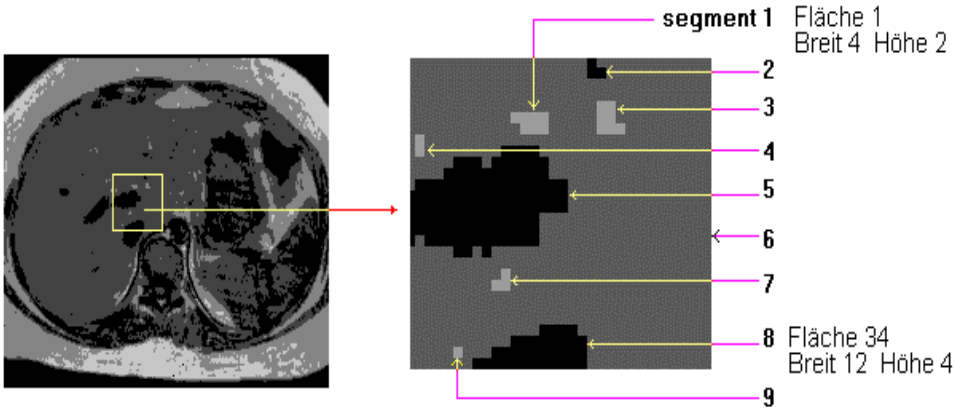


Abb.4.6: Das Bild besteht aus verschiedenen nicht-überlappenden Segmenten.

4.6 Kriterien zur Bewertung des Unterschieds zweier Bilder

Die Hauptkriterien für die Effektivität eines Kompressionsalgorithmus sind die Kompressionsrate und die Qualität des rekonstruierten Bildes. Für die Bewertung der Bildqualität werden verschiedene Methoden und Kriterien verwendet.

Das einfachste Kriterium ist die Wurzel des mittleren quadratischen Fehlers zwischen dem rekonstruierten Bild und dem Originalbild

$$J_1 = \sqrt{\frac{1}{X * Y} \sum_{x,y} (l(x,y) - s(x,y))^2} \tag{4.7}$$

wobei $s(x,y)$ ist der Grauwert des Pixel x, y des rekonstruierten Bildes S ist; $l(x,y)$- der Grauwert des Pixel x, y des Originalbildes L ist; $X * Y$ die Ortsauflösung des Bildes S und L.

Als gebräuchliches Verzerrungsmaß in der Bildkodierung hat sich das Signal-Rausch-Verhältnis (PSNR)

$$J_2[dB] = 10 * \log * \frac{X * Y * A^2}{\sum_{x=0}^{X} \sum_{y=0}^{Y} (l(x,y) - s(x,y))^2} \tag{4.8}$$

etabliert, bei dem die Energie des Fehlers zwischen dem Originalbild L und dem am Decodierer rekonstruierten Bild S zu der maximal möglichen Grauwertamplitude $A = 2^K - 1$ (z.B. $A = 255$ bei $K = 8$ bit Auflösung) ins Verhältnis gesetzt wird. Das PSNR läßt damit eine direkte Aussage über den mittleren quadratischen Fehler zu. In [55] ist die Tabelle 4.3 vorgestellt. Die Tabelle wurde durch einen subjektiven Test aufgebaut. Tests werden im allgemeinen nach einer mittleren psychologischen Bewertung (MOS) durchgeführt, wobei die Urteile von einer Reihe von Betrachtern ausgewertet werden. Die Tabelle gibt einen Überblick über die gebräuchliche MOS-Skala. Die Daten von Tabelle 4.5 und Experimente haben gezeigt, daß bei Werten des Kriteriums (4.8) größer als 40dB keine visuellen Unterschieden zwischen dem rekonstruierten Bild und dem Originalbild wahrgenommen werden.

Tabelle 4.5: MOS-Skala und Approximation der zuzuordnenden PSNR- Werte

Subjektive Kriterien	PSNR- Werte (dB) bei Detailgehalt		
	gering	mittel	hoch
Keine sichtbare Verzerrung zum Original	>40	>37	>35
Gering sichtbare Verzerrung	34-40	31-37	29-35
Mittlere bis starke Verzerrungen	28-34	25-31	24-29
Erkennbarer Bildinhalt, aber schlechte Qualität	22-28	20-25	20-24
Kaum erkennbarer Bildinhalt	<20	<20	<20

Die Kriterien (4.7) und (4.8) erlauben keine unmittelbare Aussage über die tatsächliche visuelle Qualität, weil die Bildstruktur unberücksichtigt bleibt.

Vor diesem Hintergrund werden in dieser Arbeit vom Autor neue Verzerrungskriterien vorgeschlagen. Durch den Differenzwert zwischen den entsprechenden, oben beschriebenen Wavelet-Parametern (Abschnitt 4.4) des Originalbildes und des dekodierten Bildes kann die Größe des visuellen Unterschiedes zweier Bilder geschätzt werden. Die Wavelet-Parameter berücksichtigen den Detailgehalt des Bildes und die Detailorientierung auf dem Bild. Experimente haben gezeigt, daß bei einem Differenzwert zwischen den Parametern des 2. und 3. Iterationsschritts zweier Bilder (original und dekodiert), der nicht mehr als 10% vom Absolutwert der Parameter des ursprünglichen Bildes abweicht, die Bilder visuell als gleich empfunden werden. Aus diesem Grund wird für die Schätzung der Größe

für den Unterschied zwischen zwei Bildern folgendes Kriterium (Differenzfaktor) vorgeschlagen:

$$T = \frac{1}{6} \sum_{i=2}^{3} \sum_{v=1}^{3} \frac{\left| k_{i,v}^{o} - k_{i,v}^{D} \right|}{k_{i,v}^{o}} \tag{4.9}$$

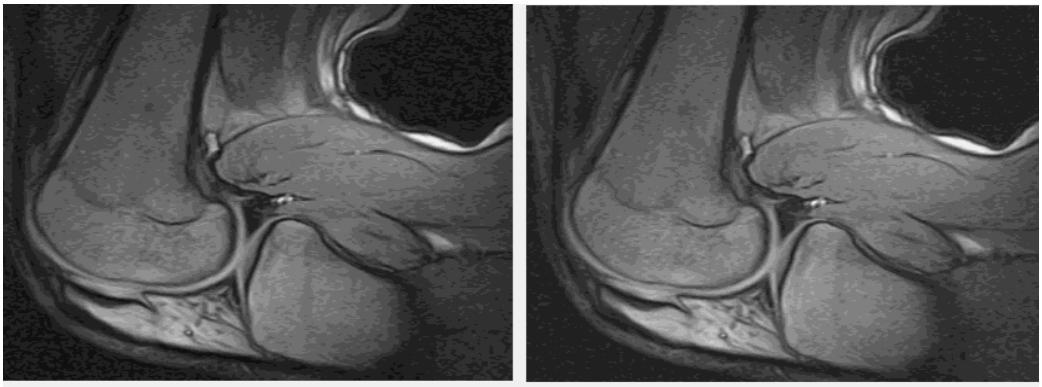
wobei $k_{i,v}^{o}$ Parameter des Originalbildes; $k_{i,v}^{D}$: Parameter des dekodierten Bildes; $k_{i,v}$: Parameter für die Waveletiteration i =2,3 und den Quadranten v = 1,2,3 (Abbildung 4.3).

Experimentell wurde bestätigt, daß es keinen visuellen Unterschied zwischen dem Original und dem dekodierten Bild gibt, wenn der Wert des Kriteriums (4.9) kleiner als 0.1 ist. Die Tabelle 4.6 stellt die T-Werte und die subjektiven Kriterien dar, die sich durch die statistische Modellierung ergeben haben.

Tabelle 4.6: Subjektive Kriterien und T- Werte

Subjektive Kriterien	T - Werte bei Detailgehalt		
	Gering	mittel	hoch
Keine sichtbare Verzerrung zum Original	<0.08	<0.08	<0.1
Gering sichtbare Verzerrung	0.1-0.08	0.1-0.08	0.12-0.1
Mittlere bis starke Verzerrungen	0.1-0.15	0.1-0.15	0.15-0.12
Erkennbarer Bildinhalt, aber schlechte Qualität	0.25-0.15	0.25-0.15	0.25-0.15
Kaum erkennbarer Bildinhalt	>0.25	>0.25	>0.25

Die Abbildung 4.7 zeigt das Originalbild und das dekodierte Bild. Das Kriterium (4.9) hat für die Bilder einen Wert von 0.09. Bei einem Wert von kleiner als 0.1 weist das dekodierte Bild noch eine hohe Qualität und nur einen geringen Unterschied zum Originalbild auf (PSNR >35, für die acht höchstwertigsten Bitebene des Bildes).



a) Originalbild b) Dekodierten Bild

Abb. 4.7

4.7 Zwei Methoden zur Bewertung und Filterung des Rauschens auf DICOM-Bildern

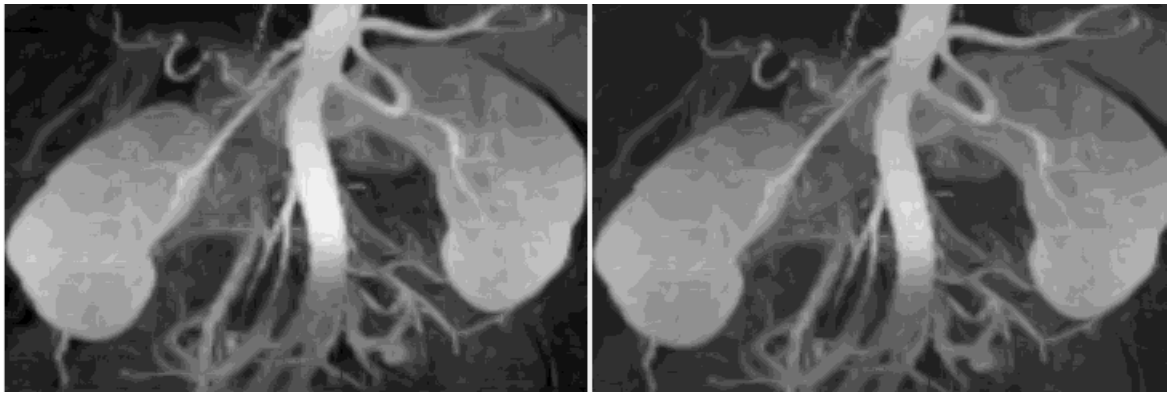
Gegebenheiten 8 bis 16 Bit pro Pixel. Doch normalerweise repräsentieren nur einige Bitebenen des Bildes dessen Inhalt. Die praktische Erfahrung des Autors belegt, daß die zwei bis drei niedrigsten Bitebenen lediglich das Rauschen des Gerätes wiedergeben, mit

dem das Bild erzeugt wurde. Deshalb ist es sinnvoll, nur diejenigen Bitebenen des Bildes zu speichern oder über das Netz zu übertragen, die die Informationen über die Ortsstrukturen des Bildes liefern.

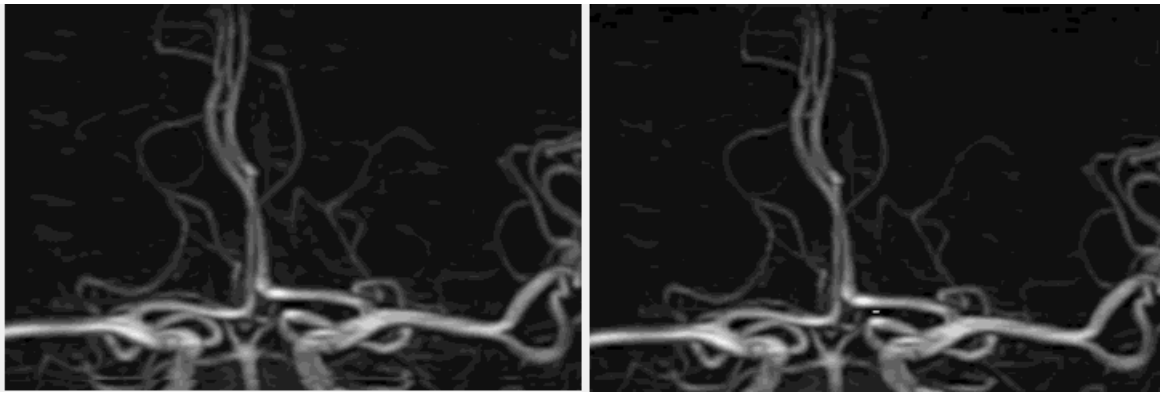
Erste Methode [65]. Mittels der folgenden Bedingung wird die Klassifizierung der Bitebenen in solche, die die Informationen des Bildinhaltes tragen, und solche, die keine Informationen zum Bildinhalt enthalten, vorgenommen:

$$T_j > 0.1 \tag{4.10}$$

wobei T_j : repräsentiert das Kriterium (4.9) für das Originalbild und das aus dem Originalbild abgeleitete Bild, in dem es keine Bitebene j gibt. Wenn die Bedingungen erfüllt sind, wird diese Bitebene als eine Bitebene klassifiziert, die Informationen über den Bildinhalt trägt. Ansonsten enthält sie nur Rauschen. Die Klassifizierung der Bildebenen verschiedener medizinischer Bilder zeigt folgendes: In 90% der Fälle tragen bei Bildern mit 8 Bit pro Pixel nur die Bitebenen 5-7 und bei Bildern mit 10-16 Bit pro Pixel nur die Bitebenen 6-9 Informationen über den Bildinhalt. Abbildung 4.8 zeigt einige Testbilder, auf die der Algorithmus „Klassifikation der Bitebene“ angewendet wurde. Links befinden sich die Originalbilder, rechts liegen Bilder, die sich aus der Klassifikation nach Kriterium (4.10) ergeben.



Originalbild - 12 Bit pro Pixel Bild - 6 Bit pro Pixel



Originalbild - 16 Bit pro Pixel Bild - 5 Bit pro Pixel

Abb. 4.8 Testbilder, vor und nach der Anwendung der ersten Methode

Zweite Methode [64]. Die statistische Analyse von Mittelwerten der Segmentflächen für verschiedene DICOM-Bilder hat folgendes gezeigt : aufgrund des Unterschiedes zwischen dem Mittelwert der Segmentfläche für jede Bitebene und dem Mittelwert der Segmentfläche für die niedrigste Bitebene, kann man Bitebenen klassifizieren als die Bitebenen, die die Informationen über den Bildinhalt (medizinisch relevante Bildobjekte) tragen und jenen, die nur das auf dem Bild vorhandene Rauschen enthalten. Im ersten Fall soll der Unterschied größer als 10% vom Mittelwert der Segmentfläche für die niedrigste Bitebene sein.

Die Verwendung der oben beschriebenen Methoden zur Reduzierung der Bitebenenzahl des Originalbildes vor der Verwendung verlustfreier Kompressionsverfahren erhöht die Kompressionsrate um das 2 bis 3-fache. Zusätzlich zur Erhöhung der Kompressionsrate verbessern die vorgeschlagenen Methoden den visuellen Eindruck des Bildes durch Verminderung des Bildrauschens und Verschieben der Histogrammfunktion.

4.8 Adaptive Kompressionsverfahren

Grundsätzlich gibt es zwei unterschiedliche Arten von Kompressionsmethoden: - *verlustfreie* Methoden, die eine Rekonstruktion des Bildes ohne jeglichen Verlust an Qualität erlauben und *verlustbehaftete* Methoden, die eine höhere Kompression erlauben, wobei sich allerdings das rekonstruierte Bild sich vom Originalbild unterscheiden kann. Selektive Bildkompression ist eine Kompressionstechnik, die signifikante Bereiche, welche explizit markiert werden, verlustfrei speichert und Regionen im Bild, die von geringerer Bedeutung sind verlustbehaftet komprimiert. Diese Techniken stoßen auf großes Interesse in der Telemedizin und anderen medizinischen Anwendungen, welche große Ressourcen an Speicher benötigen. Medizinische Bildkompression leidet unter der Tatsache, daß die meisten Ärzte ihre Diagnose nicht auf einem Bild aufbauen dürfen, das verlustbehaftet gespeichert wurde. Dies ist zum Teil durch rechtliche Umstände begründet (abhängig von der jeweiligen Rechtsprechung) sowie durch die Furcht vor einer Fehldiagnose, die durch verloren gegangene Informationen während des Kompressionsvorgangs hervorgerufen werden. Dadurch können nur verlustfreie Kompressionstechniken verwendet werden, was die mögliche Kompressionsrate auf in etwa 3:1 beschränkt. Dies steht im Gegensatz zu Raten von bis zu 100:1, die durch verlustbehaftete Methoden erreicht werden können. Einen Kompromiß stellen die adaptive Bildkompressionsmethoden dar, die eine Kombination der verschiedenen Arten von Kompressionsalgorithmen sind. Dieser Abschnitt präsentiert zwei neue adaptive Kompressionsverfahren für die verlustfreie Kompression medizinischer Bilder, die vom Autor entwickelt wurden.

4.8.1 Adaptive Entropie-Kompression digitaler medizinischer Bilder [65]

Die Kompressionsrate hängt von den Bildinhalten ab. Der vorgeschlagene Komplexitäts-algorithmus besteht aus drei verschiedenen verlustfreien Entropie-Kompressions-algorithmen mit drei unterschiedlichen Konzepten zur Bildkodierung.

- Algorithmus (A1) basiert auf der Huffmancodierung der Pixelgrauwerte für das Differenzbild vom Originalbild mit einem adaptiven Huffmancodebuch.
- Algorithmus (A2) basiert auf der Lauflängencodierung für das Differenzbild vom Originalbild mit adaptiven Huffmancodebüchern.

- Algorithmus (A3) basiert auf der Bitebenencodierung durch Lauflängencodierung der höchstwertigsten Bitebene des Bildes mit adaptiven Huffmancodebüchern.

Die Bedingungen für die Anwendung jedes einzelnen Algorithmus sind von den Bildeigenschaften abhängig. Die Effektivität des Algorithmus‘ (A1) hängt von der Grauwertentropie des Differenzbildes (P1) ab. Je geringer die Entropie ist, desto höher wird die Kompressionsrate sein. Die Kompressionsrate von Algorithmus (A2) hängt vom Mittelwert der Streckenlänge der Bildzeilen (P2) ab. Je größer der Mittelwert ist, desto höher wird die Kompressionsrate. Die Effektivität des Algorithmus‘ (A3) hängt von der Übergangswahrscheinlichkeit des Pixels der k_1 höchstwertigsten Bitebenen aus dem Zustand 0 in den Zustand 1 und zurück (P3) ab. Je größer diese Übergangswahrscheinlichkeit ist, desto niedriger wird die Kompressionsrate für diese Bitebene. Diese Parameter werden als Schätzgrößen für die Effektivität der Algorithmen für jedes zu kodierende Bild benutzt

$$P(0/1\,\&\,1/0) = \sum_{k=0}^{k_1} \frac{n_{0,k}+n_{,k1}}{N} \qquad \textbf{(4.11)}$$

wobei $n_{0,k}$: Zahl der Pixel, die den Wert 0 haben und der nächsten Pixel, in Richtung des Bildscans, die den Wert 1 haben;

$n_{1,k}$: Zahl der Pixel, die den Wert 1 haben und der nächste Pixel, in die Richtung des Bildscans, die den Wert 0 haben;

N : Gesamtzahl der Pixel im Bild; k : Nummer der Bitebene.

Die Modellierungsergebnisse zeigen, daß eine Abhängigkeit von den Mittelwerten der Kompressionsraten in Bezug auf die Größe der obengenannten Bildparameter für jeden Algorithmus besteht. Es existieren Parameterbereiche, in denen ein Algorithmus bessere Kompressionsraten als andere Algorithmen liefert. Die Algorithmen (A2) und (A3) zeigten höhere Kompressionsraten als der Algorithmus (A1) für folgenden Parameterwerte: Grauwertentropie des Differenzbildes < 0.8, Mittelwert der Streckenlänge der Bildzeilen > 2.4, Mittelwert der Übergangswahrscheinlichkeit des Pixels der Bitebene aus Zustand 0 in den Zustand 1 und zurück < 0.4. Der Algorithmus (A3) liefert ungefähr die gleichen Kompressionsraten wie der Algorithmus (A2). Übersteigt der Mittelwert der Streckenlänge der Bildzeilen den Wert 2.8, liefert (A3) etwas höhere Kompressionsraten als (A2).

Während der Modellierung wurde die Fehlergröße für eine falsche Algorithmenauswahl geschätzt. Aufgrund dieser Untersuchungen wurde für jeden Parameter der entsprechende Wertebereich in drei Abschnitte aufgeteilt. Der erste erstreckt sich von 0 bis 0.6. für Parameterwert (P1), von 2.8 bis + ∞ für Parameterwert (P2) und von 0 bis 0.3 für Parameterwert (P3). Dieses Gebiet spezifiziert die Auswahl der Algorithmen (A2) und (A3), da sie hier höhere Kompressionsraten als Algorithmus (A1) liefern. Die Wahrscheinlichkeit der falschen Algorithmenauswahl liegt in diesem Gebiet unter 0.1. Das zweite Gebiet erstreckt sich von 1.5 bis + ∞ für Parameterwert (P1), von 1 bis 1.6 für Parameterwert (P2) und von 0.5 bis 1.0 für Parameterwert (P3). In diesem Gebiet liefert Algorithmus A1 größere Kompressionsraten als die Algorithmen (A2) und (A3). Die Wahrscheinlichkeit einer falschen Algorithmenauswahl liegt auch hier unter 0.1. In dem dritten Gebiet liegt die Wahrscheinlichkeit der falschen Algorithmenauswahl über 0.1.

Abschließend wurde untersucht, welcher der drei Parameter P1-P3 die kleinste Fehlerwahrscheinlichkeit für eine falsche Algorithmenauswahl liefert. Dafür wurde für jeden Parameter einzeln der Prozentsatz der Bilder, die in das dritte Gebiet fallen(Tabelle 4.7), aus der Gesamtheit aller Testbilder berechnet. Die Untersuchung ergab, daß etwa 28% der Bilder für den Parameter P1, weniger als 16% der Bilder für den Parameter P2 und fast 24% der Bilder für den Parameter P3 in dieses Gebiet fallen. Das bedeutet, daß der Parameter P2 nützlichere Bildinformationen hinsichtlich der Klassifikation für die Bildkompression enthält als die anderen Parameter. Außerdem wurde festgestellt, daß für 50% der Testbilder alle drei Parameter den gleichen Algorithmus auswählen. Während der statistischen Modellierung wurden mehr als 200 DICOM-Bilder analysiert und komprimiert.

Tabelle 4.7: Bereiche für die Fehlerwahrscheinlichkeit und Prozentsatz der Bilder, die in das dritte Gebiet fallen

Parameter für Klassifikation	Erstes und zweites Gebiet für eine Fehlerwahrscheinlichkeit <0.1	Drittes Gebiet für eine Fehlerwahrscheinlichkeit >0.1	Prozentsatz der Bilder, die in das dritte Gebiet fallen
P1	[0,0.6], [1.5,+ ∞]	[0.6,1.5]	28%
P2	[1,1.6], [2.8,+ ∞]	[1.6,2.8]	16%
P3	[0,0.3], [0.5,1]	[0.3,0.5]	24%

In Anbetracht der erzielten Ergebnisse wird die folgende Regel zur Bildklassifizierung und zur Auswahl des Kompressionsalgorithmus vorgeschlagen. Das Bild wird in Abhängigkeit vom Parameter P2 der Klasse K1 oder K2 zugeordnet:

P2 > 2.8 $B \subset K1$
P2 < 1.6 $B \subset K2$

Wenn sich der Parameter P2 im Bereich $1.6 \leq P2 \leq 2.8$ befindet, wird für das Bild der Parameter P3 berechnet. Das Bild wird nun nochmals auf der Basis von drei Klassen klassifiziert:

P3 > 0.5 $B \subset K2$
P3 < 0.3 $B \subset K3$

Wenn sich der Parameter P3 im Bereich $0.6 \leq P3 \leq 0.9$ befindet, wird für das Bild der Parameter P1 berechnet.

P1 > 1.5 $B \subset K3$
P1 < 0.6 $B \subset K2$
Sonst $B \subset K3$

Das Block-Schema der oben beschriebenen „adaptiven Kompression digitaler medizinischer Bilder“ ist in Abbildung 4.9 dargestellt. In die Klasse K1 fallen die Bilder, die weniger komplexe Strukturen (Grobstrukturen bis homogene Flächen) aufweisen. Für die Bilder dieser Klasse liefert der Kompessionssalgorithmus A2 die besseren Ergebnisse. In die Klasse K2 fallen die Bilder mit vorrangig feinen Struktur (viele kleine Details bzw.

große Ortsfrequenzanteile). Für diese Bilder ist der Algorithmus A1 adäquat. Die Klasse K3 enthält Bilder, die grobe und feine Strukturen enthalten. In diesem Fall werden beide Algorithmen A2 und A3 auf das Bild angewandt und derjenige mit den besseren Ergebnissen ausgewählt.

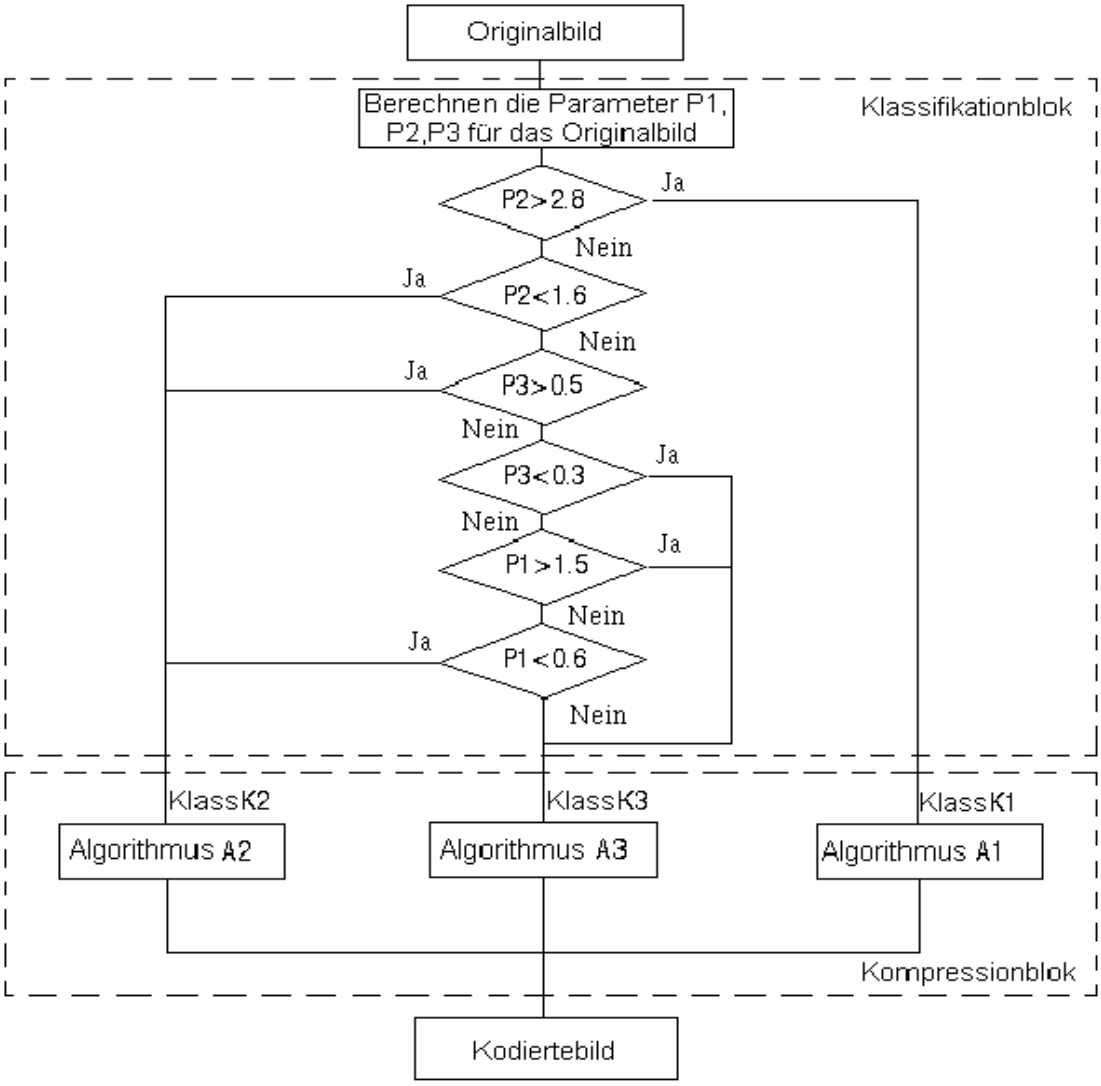


Abb. 4.9: Block-Schema des Algorithmus‘

Die statistische Modellierung, die mit dem Ziel der Überprüfung dieser Klassifizierungsregeln durchgeführt wurde, liefert folgende Aussage. Nach Verwendung des zweiten und dritten Parameters konnten nur 8% der Testbilder nicht eindeutig klassifiziert werden. Der Gesamtgewinn bezüglich der Kompressionsrate beträgt unter Verwendung der adaptiven Kompression der DICOM-Bilder 10% - 70% im Verhältnis zur Kompression ohne vorläufige Klassifizierung der Bilder. Wendet man diese Einteilung vor den oben beschriebenen (Abschnitt 4.6) Methoden für die Reduzierung der Bitebeneanzahl des Originalbildes an, kann man bei geringem Informationverlust eine Erhöhung der Kompressionsrate um ungefähr das 1.3- bis 5-fache erreichen. Die einzelnen Modellierungsergebnisse sind in Tabelle 4.8 dargestellt.

Tabelle 4.8: Kompressionsrate für Entropiealgorithmen und adaptive Algorithmus

	A1	A2	A3	Adaptive Algorithmus	Adap. Algorithmus nach der Bitebenereduzierung
Min-Kompressionrate	1.4 : 1	1.3 : 1	1.3 : 1	1.3 : 1	1.6 : 1
Max-Kompressionrate	1.7 : 1	1.8 : 1	1.8 : 1	2.7 : 1	8 : 1

Alle Algorithmen wurden mit der plattformunabhängigen Programmiersprache Java realisiert. Die Programmcodegröße des Dekompressionsteils, der mit dem komprimierten Bild zum Client übertragen wird, beträgt für die obengenannten Algorithmen 4 - 7 KByte. Damit ist dieser zusätzlich zu übertragende Code etwa 5-10 mal kleiner als das Datenvolumen des kodierten Bildes. Die Zeit für die Dekodierung des Bildes liegt für beide Algorithmen in einem Bereich von 0.5 - 5 Sekunden. Die Rechenzeit für die Klassifizierung eines Bildes der Größe 512x512 Pixel beträgt weniger als eine Sekunde. Diese Zeit wird durch die Größe des Originalbildes bestimmt. Zur Modellierung wurde ein Arbeitsplatz-PC (Pentium200) benutzt.

4.8.2 Die LZW-JPEG Kompression [64,95]

Die Idee des Algorithmus ist einfach und besteht im Aufteilen des Originalbildes in zwei Bilder. Das erste Bild besteht aus den höchsten Bitebenen des Originalbildes und das zweite aus dem Rest, also den niederwertigen Bitebenen des Originalbildes. Im folgenden werden das erste Bild mittels LZW-Verfahren und das zweite mittels JPEG-Verfahren kodiert (Abbildung 4.10).

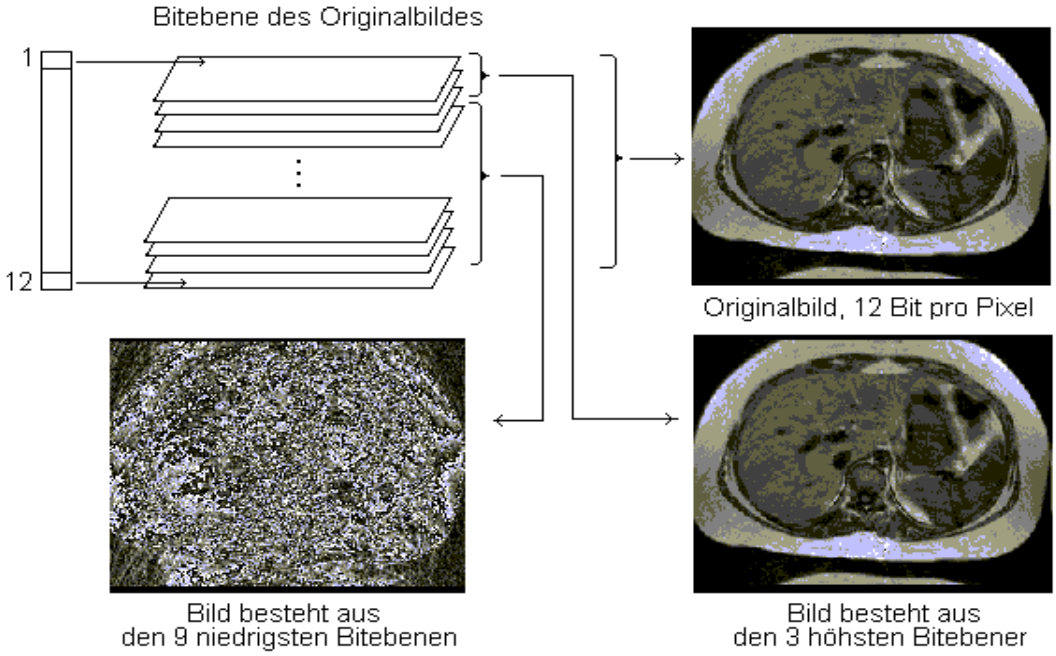


Abb.4.10: Aufteilen des Originalbild in zwei Bilder

Das Problem besteht nun darin, das Originalbild in zwei Bilder aufzuteilen. Dabei soll ein Kompromiß zwischen zwei Bedingungen erfüllt werden und zwar soll der Kompressionsfaktor für die kombinierte LZW-JPEG-Kompression einen höheren Wert erreichen als für die alleinige LZW-Kompression oder für die alleinige JPEG–Kom-

pression. Gleichzeitig sollen LZW-JPEG- und JPEG-Algorithmen keine sichtbare Verzerrung zum Originalbild verursachen, d.h. das PSNR-Kriterium für die acht höherwertigen Bitebenen des Originalbildes und des dekodierten Bildes soll größer als 40dB sein.

Vor der Aufteilung des Originalbildes in zwei Bilder wird das Originalbild in Bitebenen und in Bilder, die aus 2,.....,N-1 höherwertigen Bitebenen bestehen, aufgeteilt (Abbildung 4.11). Danach wird jedes Bild segmentiert und für jedes Bild der Mittelwert der Segmentflächen berechnet. Das Segment ist ein beschränkter Bildbereich von beliebiger Form, dessen Pixel alle den gleichen Grauwert haben. Es wird der Mittelwert der Segmentflächen für alle Bilder berechnet.

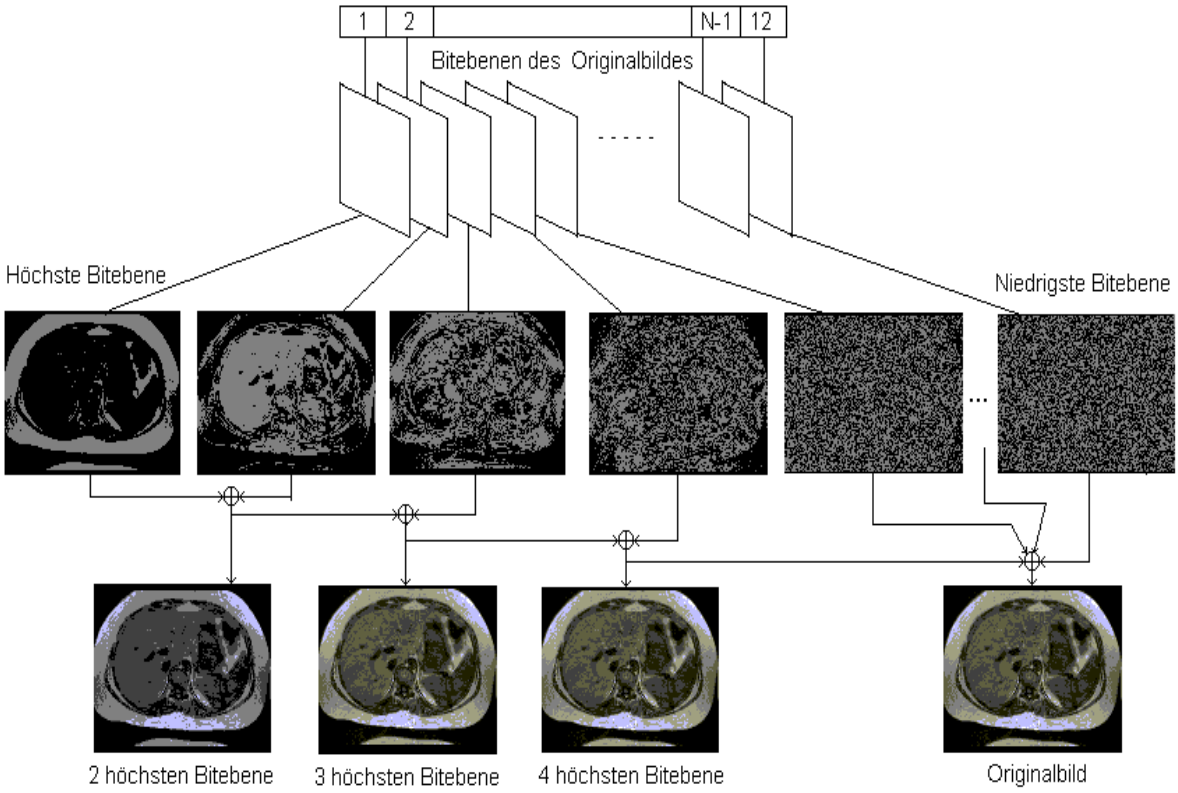


Abb. 4.11. Aufteilen des Originalbildes in Bitebenen und in Bilder, die aus 2,.....,N-1 höheren Bitebenen bestehen

Die statistische Analyse von Mittelwerten der Segmentflächen für verschiedene DICOM-Bilder hat folgendes gezeigt:

- Der Kompressionsfaktor des LZW-Verfahrens ist abhängig vom Mittelwert der Segmentfläche des kodierten Bildes (Abbildung 4.12). Aufgrund dieser Abhängigkeit und der Mittelwerte der Segmentflächen der Bilder, die aus 1, 2, 3 ... , N höherwertigen Bitebenen des Originalbildes bestehen, wird automatisch die Zahl der höherwertigen Bitebene des Originalbildes für die LZW–Kompression bestimmt. Dabei sollen gleichzeitig zwei Bedingungen erfüllt werden. Die Zahl der Bitebenen soll möglichst groß sein und der Kompressionsfaktor soll größer als 10:1 sein.

- Es existiert eine Abhängigkeit zwischen der Anzahl der Bitebenen des zweiten Bildes, die die Informationen über den Bildinhalt tragen, und dem Parameter Q für die JPEG-Kompression, der die Kompressionsrate bestimmt und bei dem die PSNR größer als 40 (dB) ist. Je größer die Anzahl dieser Bitebenen ist, desto geringer soll der Parameter Q sein. Aufgrund dieser Abhängigkeit wird der Parameter Q für JPEG-Verfahren ausgewählt. Diese Abhängigkeit wird durch die statistische Modellierung für jede JPEG–Version berechnet. Die Anzahl der Bitebenen, die die Informationen über den Bildinhalt tragen, kann man durch die oben beschriebene zweite Methode (Abschnitt 5.6) der Klassifizierung der Bitebenen finden.

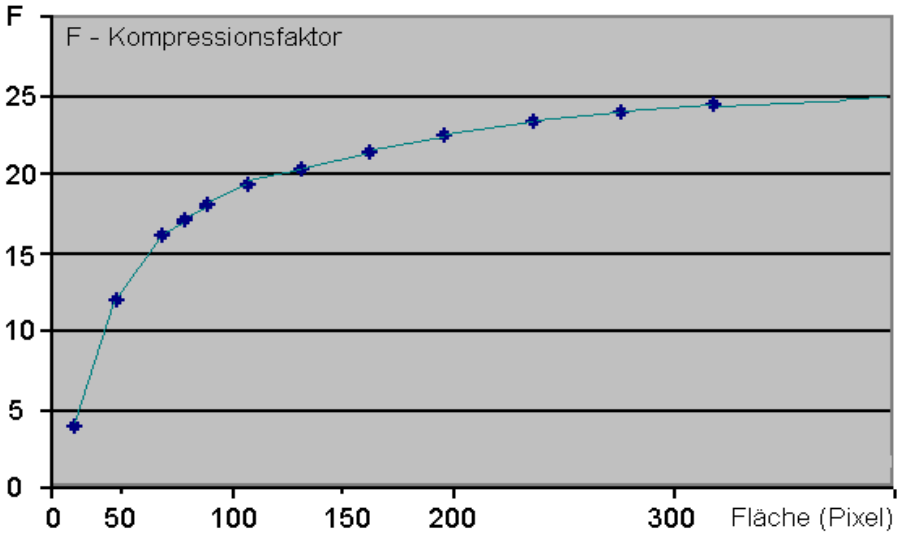


Abb. 4.12**.** Die Abhängigkeit des Kompressionsfaktors vom Mittelwert der Segmentfläche des kodierten Bildes

Sind die oben genannten Ergebnisse berücksichtigt, so kann man folgendes Blockschema für den vorschlagenen LZW-JPEG Algorithmus entwerfen (Abbildung 4.13). Block 1 teilt das Originalbild in Bitebenen und in Bilder auf, die aus 2,.....,N-1 höherwertigen Bitebenen bestehen. Block 2 segmentiert die Bilder und berechnet den Mittelwert der Segmentflächen für jedes Bild (*Si(n), n* gleich Anzahl der Bitebenen, aus denen das Bild besteht) und für jede Bitebene (*Sb(k), k* gleich Nummer der Bitebene). Die Blöcke 3 bis 5 berechnen, aufgrund der Abhängigkeit des Kompressionsfaktors des LZW-Verfahrens vom Mittelwert der Segmentflächen, die Anzahl (n) der Bitebenen, bei denen die Bedingung, daß der Kompressionsfaktor größer als 10 ist, noch richtig ist. Die Blöcke 6 und 7 erzeugen das erste Bild aus den n höchstwertigen Bitebenen und das zweite Bild aus den N-n niederwertigen Bitebenen des Originalbildes. Block 8 bis 10 berechnen die Anzahl (k) der Bitebenen, die die Informationen über den Bildinhalt tragen. Block 11 wählt auf der Basis von k den Parameter Q für das JPEG-Verfahren, bei dem das PSNR größer als 40dB ist. Die Blöcke 12 und 13 implementieren die LZW- und JPEG–Kodierungen. Block 14 erzeugt eine Datei des kodierten Bildes aus zwei Dateien, die die nach LZW- und JPEG-Verfahren kodierten Bilder beinhalten. Der Header der erzeugten Datei beinhaltet Informationen über die Größen der LZW- und JPEG-Dateien und den Header des DICOM-Bildes.

Die Effektivität des oben beschriebenen Algorithmus wird durch eine statistische Modellierung bewertet. Für eine statistische Modellierung der LZW-JPEG - , der LZW- und der JPEG–Kodierungen wurden DICOM-Bilder verschiedener Hersteller von radiologischen Geräten (Siemens, Philips, ...) mit jeweils verschiedenen Modalitäten verwendet, insgesamt zirka 500 Bilder. Die statistische Modellierung wurde mit dem Ziel des Vergleiches der Kompressionsraten bei gleicher Beschränkung für die PSNR durchgeführt. Die einzelnen Modellierungsergebnisse werden in Tabelle 4.9 dargestellt.

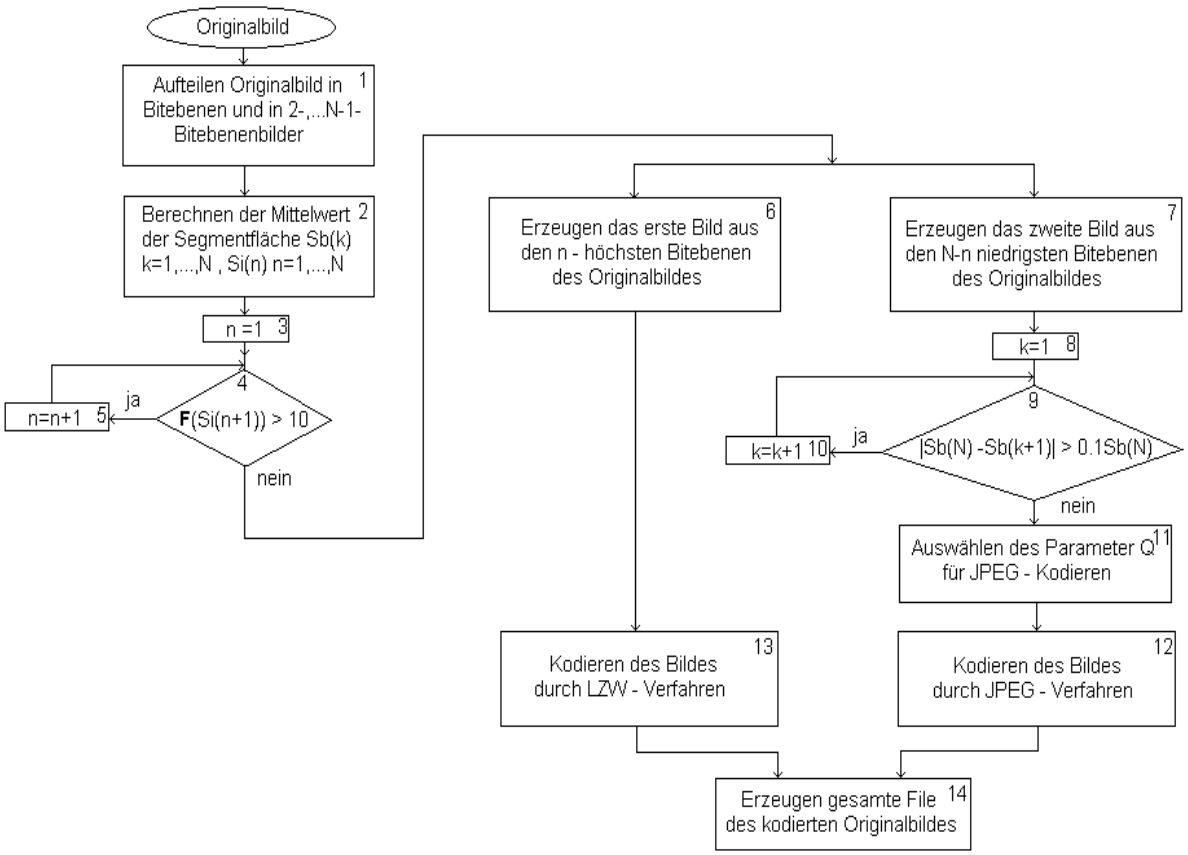


Abb. 4.13. Die Blockschema des LZW-JPEG Algorithmus

Tabelle 4.9: Kompressionsraten für LZW-JPEG, LZW, JPEG Algorithmen (PSNR > 40dB)

Modalität/ Algorithmus	CT	MR	US	NUK	CR	DS	Kompressions-rate (min)	Kompressions-rate (max)
LZW	1.3:1	1.5:1	1.8:1	1.5:1	1.4:1	1.4:1	1:1	2:1
JPEG	4:1	4:1	6:1	3:1	7:1	7:1	4:1	8:1
LZW-JPEG	7:1	7:1	15:1	6:1	12:1	12:1	5:1	16:1

Die Rechenzeit (Pentium200) für die Analyse eines DICOM-Bildes beträgt ca. 1-5 Sekunden. Für die Rekonstruktion wurde auf der Client-Seite unter den Internet-Browsern ein einfaches Java-Applet (600 Byte) verwendet. Die Zeit für die Rekonstruktion beträgt 1-10 Sekunden und ist abhängig von der Bildgröße.

Der beschriebene Algorithmus ist effizient für die Bildkompression bei Übertragungen über Netze, weil das LZW-Verfahren (GIF-Format) und JPEG in auf dem Markt

verfügbaren Internet-Browsern implementiert sind, weshalb man auf der Client-Seite das Rekonstruieren radiologischer Bilder nach der oben beschriebenen LZW-JPEG Kompression schnell mittels eines einfachen Java-Applets realisieren kann.

Der Vergleich der oben beschriebenen adaptiven Kompressionsalgorithmen liefert folgende Aussage:

- Durch den geringen visuell nicht wahrnehmbaren Informationsverlust ermöglicht der LZW-JPEG Kompressionsalgorithmus eine um ungefähr das 2- bis 3-fache größere Kompressionsrate als der adaptive Entropie-Kompressionsalgorithmus.
- Die Größe des Decodier-Applets für den LZW-JPEG-Kompressor ist um ungefähr das 10 – fache kleiner als für die adaptive Entropie-Kompression. Das ist sehr wichtig bei der starken Begrenzung der Übertragungsbandbreit im derzeitigen Netzwerk (z.B. ISDN).
- Da das LZW- und das JPEG–Verfahren standardisierte Verfahren in Internet-Browsern sind, ist der LZW-JPEG-Kompressor für Internet-Anwendung gut geeignet. Das heißt, er ist sicher und einfach in der Implementierung im Vergleich mit dem adaptiven Entropie-Kompressor.

4.9 Schlußfolgerungen

Trotz des schnellen Fortschritts in der Herstellung von Massenspeichern und in der Computernetzleistung, bleibt fortgesetzt die Forderung nach größerer Übertragungsgeschwindigkeit und mehr Speicherplatz im digitalen Radiologiebereich bestehen und treibt die Entwicklung neuer Lösungen für diese Probleme voran. Um diese zwei Hindernisse zu überwinden, zielt die Forschung in der radiologischen Bildkomprimierung darauf ab, eine low-bit-rate Kompression zu erzielen, wenn sie digitalisierte Bilder der verschiedenen Modalitäten darstellt; natürlich unter Beibehaltung einer, für den klinischen Gebrauch, annehmbaren Bildqualität.

Viele Arbeiten, die auf dem Gebiet der Kompression medizinischer Bilder durchgeführt wurden, deuten darauf hin, daß es denkbar ist, ein radiologisches Bild bis auf 10:1 zu komprimieren (oder sogar höher), ohne seine Diagnosequalität zu mindern. Wegen der zugelassenen und medizinischen Implikationen (Anwendungen), ist eine verlustfreie Komprimierung die momentan akzeptabelste Weise, medizinische Bilder zu komprimieren.

Allerdings ist durch die neue Annahme bestimmter Industriestandards, wie ACR/NEMA und DICOM und durch den expandierenden Markt für PACS und Teleradiologie eine merklich Zunahme derjenigen medizinischen Einheiten zu registrieren, die verlustbehaftete Kompressionsalgorithmen verwenden.

In diesem Kapitel wurden einige der wichtigsten Eigenschaften radiologischer Bilder, sowie die Schwerpunkte der medizinischen Bildkomprimierung und die Algorithmen für die radiologische Bildkomprimierung beschrieben. Im Rahmen dieses Kapitels wurde berichtet, wie zu Testzwecken eine größere Menge von DICOM-Bildern von unterschiedlichen Modalitäten mit verlustfreien Kompressionsmethoden (Huffman-Kodierung, RLE**,** LZW) sowie mit mehreren verlustbehafteten Kompressionsalgorithmen

(JPEG, Wavelet- Codierung, Vektorquantisierung) komprimiert wurden und die Testergebnisse miteinander verglichen wurden.

Eine interessante Entwicklung im Bereich der Bildkomprimierung ist die adaptive Kompression, die die Vorteile verschiedener verlustfreier und verlustbehafteter Verfahren kombiniert. Das adaptive Kompressionsverfahren kann man in vier Gruppen aufteilen:

- Die erste Gruppe beinhaltet örtliche adaptive Algorithmen. Zuerst wird das Bild auf Grund einiger Parameter in nicht-überlappende Bereiche segmentiert. Danach wird jeder Bereich durch einen geeigneten Kompressionsalgorithmus kodiert, der die höchste Kompressionsrate bei minimaler Verlust an Information (oder gar ohne Verlust an Information) für diese Bildbereiche liefert.
- Die zweite Gruppe besteht aus Algorithmen, die auf Grund lokaler Eigenschaften des Bildes ihre Parameter ändern. Z.B., Codebuch (Vektorquantisierung), Parameter der Quantisierungmatrix (JPEG), Huffman–Codierung etc..
- Die dritte Gruppe stellt die Algorithmen dar, die die globalen Eigenschaften des Bildes berechnen und danach auf der Basis dieser Parameter das effektivste Kompressions-verfahren für das ganze Bild verwenden.
- Die vierte Gruppe beinhaltet die Algorithmen, die die Bitebene des Originalbildes analysieren und danach das Originalbild in zwei oder drei Bilder teilen (z.B.:das erste Bild besteht aus den höchstwertigen Bitebenen des Originalbildes und das zweite aus dem Rest oder das erste Bild besteht aus den höchstwertigen Bitebenen des Originalbildes, das zweite aus den mittleren Bitebenen und das dritte aus dem Rest). Dann wird jedes Bild einzeln kodiert. Welches Kompressionsverfahren für welches Bild verwendet wird, ist abhängig vom Analyseergebnis.

In diesem Kapitel wurden zwei neue adaptive Algorithmen beschrieben. Der erste Algorithmus gehört zur dritten Gruppe, der zweite Algorithmus gehört zur vierten Gruppe. Für beide Algorithmen werden zwei Verfahren die Klassifizierung von Originalbildern und Bitebenen in unterschiedliche Klassen aufgrund einer vorläufigen Analyse des Bildinhaltes mit der nachfolgenden Kompression der Bilder mit einem geeignetem Kompressionsalgorithmus vorgeschlagen. Außerdem sind die Ergebnisse der statistischen Modellierung der beschriebenen adaptiven Verfahren für DICOM-Bilder angegeben, die die Effektivität der vorgeschlagenen Verfahren bestätigen. Hier werden die vorgeschlagenen adaptiven Kompressionsalgorithmen miteinander und mit bekannten verlustbehafteten Verfahren verglichen; das Ergebnis dieses Vergleichs zeigt die Vorzüge einer GIF-JPEG Kompression, die sie bekannten verlustbehafteten und verlustfreien Verfahren voraus hat (bei gegebener Begrenzung der Qualität des rekonstruierten Bildes (PSNR>40dB)). In dieser Hinsicht ist es sinnvoll, die vorgeschlagene adaptive GIF-JPEG Kompression für die Verwendung in internet-basierten PACS zu empfehlen.
Die vorgeschlagene adaptive GIF-JPEG-Kompression hat große Chancen auf Einführung in den medizinischen Alltag, da sie keine visuell wahrnehmbaren Veränderung zwischen Original und rekonstruiertem Bild erzeugen und bereits Web-orientiert sind. Die Hersteller von Web-basierten RIS für Krankenhäuser und Praxen brauchen also nur bereits Bekanntes in ihre Systeme zu integrieren, was zu verringerten Entwicklungskosten für die Hersteller führt.

Allerdings ist die Wahl eines speziellen Entwurfs für die optimale Komprimierung im Rahmen eines internet-basierten PACS ganz allgemein ein komplizierter Kompromiß zwischen dem Speicherplatzangebot des Archivierungssystems, der Übertragungsleistung des Netzwerks und den klinischen Anforderungen, der alle diese unterschiedlichen Faktoren berücksichtigen muß.

5. Implementierung von JAVA-basierten Übertragungs- und Visualisierungsverfahren für DICOM-Bilder

5.1 Einleitung

Intranet/Internet-basierte Anwendungen eröffnen weitreichende Möglichkeiten für Kommunikation und Datenvisualisierung [99-102]. Viele beispielhafte Anwendungen im World-Wide-Web (WWW) haben gezeigt, daß mit geringem Aufwand benutzerfreundliche und mächtige kommunikationsunterstützende Systeme erstellt werden können. Dateneingaben und Visualisierung bildhafter Daten lassen sich unter Nutzung bestehender Bibliotheksklassen einfach implementieren. Bereits bestehende Rechnerhardware (PCs, Macintosh-Rechner und Workstations) mit verschiedenen Betriebssystemen lassen sich unter Nutzung der Java-Technologie problemlos integrieren. Die einzige Voraussetzung hierfür ist das Vorhandensein der Java-Runtime Umgebung (JRE) für das jeweilige Betriebsystem und ein Browser mit integrierter Java-Unterstützung. Dieses Kapitel betrachtet in diesem Zusammenhang die Entwicklung JAVA-basierter Datenübertragungs- und Visualisierungssoftware für medizinische Bilder. Diese Anwendungen erlauben es dem Intranet/Internet-Benutzer, Bildverarbeitungsoperationen in Bezug auf die in der Datenbank oder lokal abgelegten medizinischen Bilder verschiedener DICOM-Modalitäten anzuwenden.

5.2 JAVA Client-Server Paradigma und JDBC-Technik

Eine wichtige Eigenschaft der Programmiersprache Java ist die enge Verbindung mit der Netzwerk-Technologie des Intranets/Internets. Die Java-Umgebung ist so aufgebaut, daß man leicht die Anwendung für den Datenaustausch und ihre verteilte Bearbeitung auf den verschiedenen Rechnern schreiben kann. Die Java-Konzeption des Datenaustausches unterstützt erstens das TCP/IP-Protokoll und basiert zweitens auf dem Client/Server-Kommunikationsmodell [103,104] und auf der JDBC-Technik [39,40] (Abbildund 5.1).

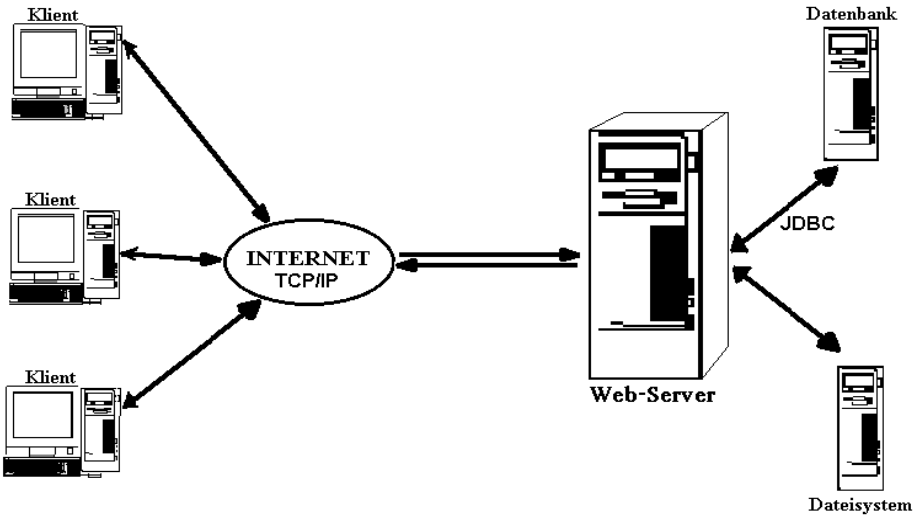


Abb.5.1: Das Client/Server-Kommunikationsmodell

Java hat zwei standardisierte Klassen, die es ermöglichen, Netzwerkanwendungen auf Socket-Basis zu erzeugen (java.net.Socket und java.net.ServerSocket). Das Socket ist eine Beschreibung der Netzwerkverbindung zu anderen Anwendungen (IP-Adresse und Portnummer des Client und ähnliche Daten über den Server). Auf der Basis dieser Klassen werden die Client/Server-Anwendungen erzeugt.

Der Datenaustausch zwischen den Komponenten des Informationssystems für die Übertragung und Archivierung von medizinischen Bildern innerhalb des Krankenhauses kann man gemäß zweier Schemata verwirklichen.

Das erste Schema: Client-Anwendung - Server - (Dateisystem oder Datenbank) – Server – Client-Anwendung. Zur Implementation dieses Schemas ist es nötig, auf jedem Arbeitsplatzrechner der Ärzte die Client-Anwendungssoftware zu installieren. Jeder Arbeitsplatzrechner ist über das lokale Netz mit dem Server der DICOM-Datenbank und dem Archiv verbunden. Nach diesem Schema fragt der Client den Server und der Server liest die notwendigen Informationen entweder aus dem Dateisystem oder aus der Datenbank. Danach schickt der Server die Informationen zurück an den Client. Für die Realisierung solcher Dialoge nach diesem Schema beinhaltet das Ablaufprotokoll sechs grundlegende Schritte:

1. Der Client verbindet sich mit dem Server;
2. Der Server schickt die Bestätigung, daß die Verbindung erfolgreich ist;
3. Der Client fragt die gewünschten Informationen ab;
4. Der Server sucht und ließt die Daten aus dem Dateisystem und schickt danach die benötigten Daten zurück zum Client;
5. Schritte 3 bis 4 werden bis zum Ende des Dialogs wiederholt;
6. Die Verbindung wird beendet.

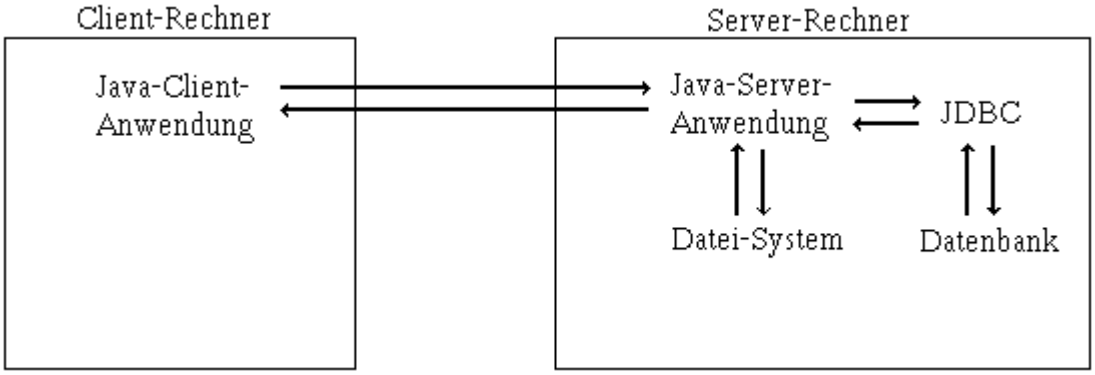


Abb. 5.2: Ablauf der Datenübertragung nach dem ersten Schema

Der Ablauf des oben beschriebenen Datenaustausches wird oben (Abbildung 5.2) dargestellt. Die Java-Klasse „Client“ und die Klasse „Server“, die dieses Protokoll realisieren, werden im Anhang D dargestellt.

Wenn die Quellinformation als Datenbank dient, verwendet der Server die JDBC-Software für die Anfragen an die Datenbank. JDBC ist eine Java-Schnittstelle für die Verbindung mit den Datenbanken (ACCESS, Sybase, Oracle, ...) oder anderen Informationsquellen.

JDBC liefert eine Möglichkeit, alle Methoden zur Datenmanipulation in Datenbanken durch Java-Anwendungen oder Applets zu implementieren. JDBC erstellt die Verbindung zwischen den Tabellen der Datenbank, überträgt über diese Verbindung die SQL-Befehlen für das DBMS und erhält auf demselben Wege die Ergebnisse und Dienstinformationen zurück.

Das zweite Schema: Client-Applet – Server - Datenbank – Server – Client-Applet. Zur Imple-mentation dieses Schema ist auf dem Arbeitsplatzrechner des Client / Arztes nur ein Web-Browser, jedoch keine spezielle Softwareinstallation nötig. Durch die HTML-Seite und das Applet, das auf dem Web-Server installiert ist, greift der Arzt via Intranet/Internet direkt auf die Daten aus der Datenbank zu. Das Java-Applet wird automatisch in seinem Browser gestartet. Medizinische Bilddaten werden in originärer Qualität via ISDN oder dem Internet an den beteiligten Client übertragen. Der Datenaustausch gemäß diesem Schema beinhaltet fünf Schritte und sieht folgendermaßen aus:

1. Der Client ruft mit der URL eine HTML-Seite auf, die das Applet zur Verfügung stellt;
2. Der Client stellt durch das Applet eine konkrete Anfrage an die Datenbank, die auf demselben Server installiert ist;
3. Durch JDBC organisiert das Applet die Verbindung mit der Datenbank;
4. Durch JDBC sucht und liest das Applet Daten aus der Datenbank;
5. Das Applet schickt die gefundenen Daten zurück zum Client.

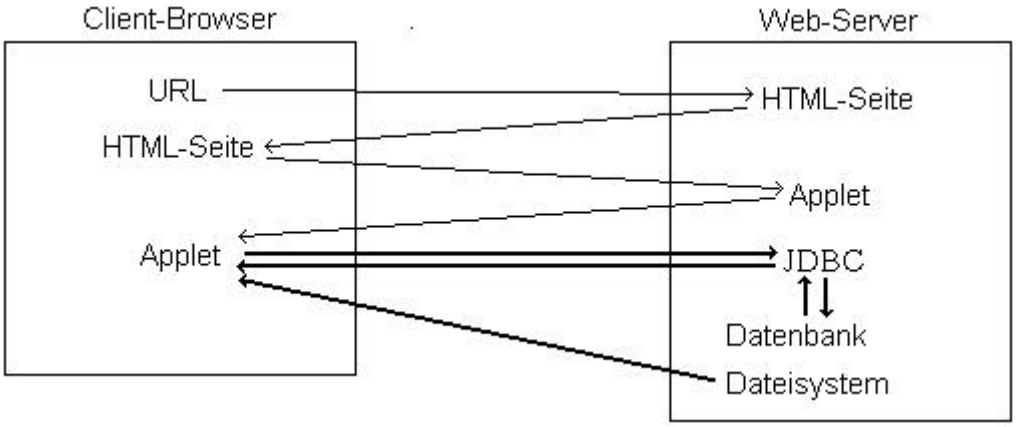


Abb. 5.3: Ablauf der Datenübertragung nach dem zweiten Schema

Der Ablauf des oben beschriebenen Datenaustausches wird unter (Abbildung 5.3) dargestellt.

5.3 Servlet Technik

Eine alternative Lösung dieser Aufgabe ermöglicht die Servlet-Technik der Firma Sun [44-46,104]. Sun Microsystems hat mit den Servlets eine in Java realisierte Server API entwickelt. Die Einbettung von Servlets in eine HTML-Seite erfolgt mit dem Auszeichnungselement <SERVLET> oder mit der URL des Programms für die Bearbeitung der durch das Auszeichnungselement <FORM> angegebenen Daten. Der Server parst ein HTML-Dokument. Wird über eine HTML-Seite eine Datenbank-Anfrage initiiert, verbindet sich das entsprechende Servlet mit Datenbank und schickt das Resultat

der Anfrage zum Client zurück. Da Java-Servlet-Klassen dynamisch geladen werden, funktionieren Servlets ähnlich wie dynamisch ladbare Bibliotheken. Sie werden beim ersten Aufruf geladen und verbleiben dann im Speicher. Der Unterschied zwischen Standart–HTML und Servlets besteht darin, daß nach dem Öffnen eines Socket die Verbindung zwischen Servlet und Client als Dialog funktionieren kann.

Das Servlet stellt alle Möglichkeiten einer Java-Anwendung zur Verfügung. Einer der Vorteile des Servlets ist, daß das Servlet nicht jedes Mal eine erneute Installation verlangt. Nach der ersten Installation im Speicher kann das Servlet auf jede Anfrage an diese Seite antworten. Servlet und Anwendungen haben keine Beschränkungen hinsichtlich der Netz-Verbindung.

Der Datenaustausch durch Servlet-Technik beinhaltet sechs Schritte und sieht folgendermaßen aus:

1. Der Client ruft mittels der URL eine HTML-Seite auf, die vom Servlet zur Verfügung gestellt wird;
2. Der Client stellt eine konkrete Anfrage;
3. Der Server nimmt die Anfrage entgegen und versucht das Servlet zu identifizieren;
4. Das indentifizierte Servlet liest die Daten aus dem Dateisystem oder durch JDBC aus der Datenbank. Danach erzeugt das Servlet eine neue HTML-Seite und eine temporäre Datei mit den benötigten Daten;
5. Der Server schickt die erzeugte HTML-Seite zurück zum Client;
6. Der Client ruft mittels der erzeugten HTML-Seite ein Applet auf, der die Datei mit den benötigten Daten auf dem Server liest und zurück zum Client überträgt;

Der Ablauf des oben beschriebenen Datenaustausches wird unter (Abbildung 5.4) dargestellt.

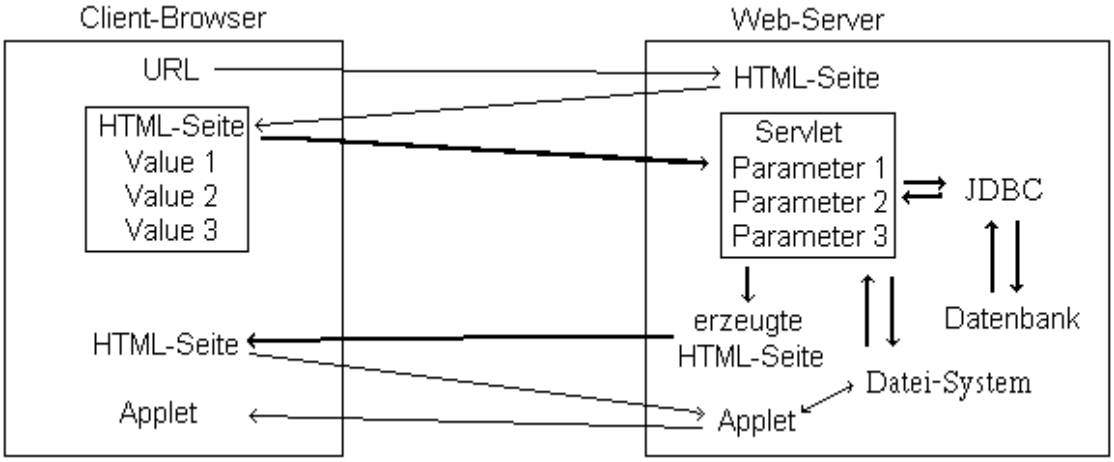


Abb. 5.4: Ablauf der Datenübertragung entsprechend der Servlet-Technologie

Das Erstellen von Servlets ist relativ einfach, da das Servlet API bereits viele Basisklassen enthält, die das gleichzeitige Ausführen mehrerer Anfragen unterstützen. Ein Vorteil von Servlets gegenüber Server API-Anwendungen ist der, daß der Code von Servlets auch von

einem anderem Rechner geladen werden kann. Servlets werden mittlerweile von einigen Web-Servern unterstützt.

5.4 DICOM- Viewer- Anwendung

Als Befundungsstationen dienen die Arbeitsplatzrechner der Ärzte, im allgemeinen also PCs mit hochauflösender Graphikkarte und entsprechenden Monitoren. Auf diesen Rechnern ist eine DICOM-Viewer-Anwendung (DV-Anwendung) installiert. Die DV-Anwendung ist ein Visualisierungsprogramm für DICOM-Bilder [106-108]. Das Programm ist in der Programmiersprache Java geschrieben und somit plattformunabhängig.

Die DV-Anwendung unterstützt das TCP/IP-Protokoll zur Datenübertragung und verwendet das JDBC-Interface. In dieser Hinsicht paßt sich die DV-Anwendung für die Verwendung in beiden, im Abschnitt 5.1 beschriebenen, Schemata des Datenaustausches zwischen den Komponenten des Informationssystem zur Übertragung und Archivierung von medizinischen Bildern an.

Jeder Arbeitsplatzrechner ist über das lokale Netz indirekt mit der DICOM-Datenbank verbunden. Innerhalb der DV-Anwendung wählt der Arzt die gewünschten medizinischen DICOM-Dokumente in der Datenbank oder im Dateisystem aus und aktiviert damit die Übertragung der ausgewählten Bilddaten vom Server zum Client-Rechner. Die DV-Anwendung besteht aus zwei Teilen: DICOM-Patientenauflistung-Viewer (PAV) und DICOM-Bilder-Viewer (DBV).

Die DV-Anwendung kann so konfigurieren werden, daß sie automatisch beim Einschalten des Arbeitsrechners des Arztes startet. Nach erfolgreicher Authentifikation des Benutzers erscheint das Fenster des Patientenauflistungs-Viewer (PAV) (s. Abbildung 5.5). Das Fenster beinhaltet zwei Tabellen: Die erste Tabelle beinhaltet die Patienteliste, die zweite die Auflistung der Daten für einen bestimmten Patienten. In Tabelle 1 befinden sich die Informationen über Patientendaten, die in der abgefragten DICOM-Datenbank archiviert sind. Im einzelnen beinhaltet Tabelle 1 den Patientennamen, das Geburtsdatum, die ID-Nummer des Patienten, das Geschlecht, das letzte Untersuchungsdatum, die gesamte Zahl von Dokumenten zu jedem Patienten, die ID-Nummer des überweisenden Arztes, sowie das Eingangsdatum auf dem Arzt-Arbeitplatzrechner. Die Informationen über die DICOM-Bilder aus einer ausgewählten Quelle befinden sich ebenfalls in Tabelle 1. Außerdem gibt es die folgenden Möglichkeiten die Informationen in Tabelle 1 zu bearbeiten:

- Ordnen der gesamten Patientenliste nach: Geburtsdatum des Patienten, Untersuchungsdatum und Patientenname;
- Information suchen mit anschließendem Ordnen nach dem oben genannten Parameter in der gesamte Patientenliste:

  Alphabetisch;
  Nach einem bestimmten Patientennamen;
  Nach einem Geburtsdatum des Patienten;
  Nach der ID-Nummer des Patienten (medizinische Einrichtung);
  Nach der ID-Nummer des überweisenden Arztes;

Diese Funktion wird mittels Choice "Suchparameter ", TextField "Wert " und Button "Suche" ausgeführt. Sie wird mit dem Button ”Suche” gestartet. Mittels Doppelklicken auf den Namen eines ausgewählten Patienten in der Tabelle 1 erhält man die Informationen über diesen Patienten in der Tabelle 2 - Auflistung für einen bestimmten Patienten. In Tabelle 2 befinden sich das Untersuchungsdatum, die Untersuchungs-UID, die Bildnummer in Bildserien, die Modalität des Bildes, sowie Bemerkungen zur Untersuchung. Die Informationen in Tabelle 2 kann man mit den Auswahlparametern analog zur Tabelle 1 durchsuchen. Diese Funktionen werden mittels Button ”Suche” für Tabelle 2 gestartet. Nach jeder Suche können Sie zurück zur ersten Patientenliste kommen. Das wird über den Menüpunkt ”Patient” und der Auswahl ”Zur ersten Auflistung zurück” ermöglicht.

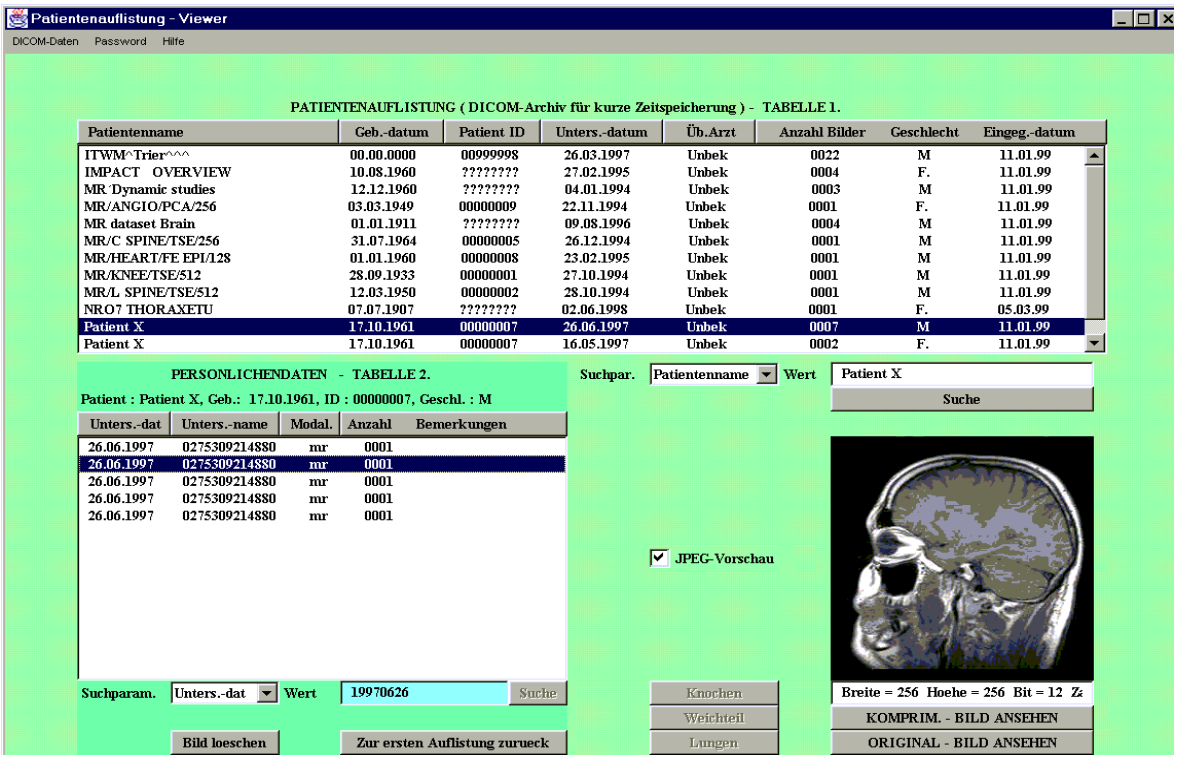


Abb. 5.5: Das Fenster der PAV-Anwendung

Außerdem ermöglicht der PAV mittels der Checkbox "Bild-Vorschau" eine Voransicht des in Tabelle 2 ausgewählten DICOM-Bildes in skalierter Version. Eine Kontrolle der Bildparameter - Bildhöhe, Bildbreite, Bit pro Pixel, Nummer des Bildes in Bildfolgen - wird ermöglicht. Bei CT-Bildern kann man zwischen der Ansicht der Weichteile, Knochen und der Lungen wählen. Dies wird durch die drei Buttons ”Weichteil”, ”Knochen”, ”Lungen” realisiert.

Der PAV bietet grundsätzlich zwei verschiedene Möglichkeiten, DICOM-Daten einzulesen. Zum einen kann er DICOM-Daten aus dem Verzeichnis ”dicombilder” von der eigenen Festplatte lesen und daraufhin eine Liste aller Patienten generieren, deren DICOM-Daten in diesem Verzeichnis gespeichert sind. Zum anderen kann über den Menübefehl *”Patientenliste aus der DICOM-Datenbank”* die gesamte oder nach bestimmten Parametern geordnete Patientenliste aus der Datenbank gelesen werden.

Danach lassen sich über diese Liste dort vorhandene DICOM-Bilder unmittelbar auswählen und ansehen. Außer dem Lesen der Information vom Server bietet der PLV über die der Funktion ”*Bild verschicken zur DICOM-Datenbank*” die Möglichkeit, DICOM-Bilder von der Festplatte der Befundungsstation zur Datenbank zu versenden. Der Menübefehl ”*Bild verschicken von DICOM-Datenbank zum DICOM-Archiv*” ermöglicht es dem Arzt, ausgewählte Bilder aus dem Kurzzeitspeicher im DICOM-Archiv dauerhaft zu archivieren. Mit dem Button ”*Bild löschen*” kann das ausgewählte Bild von der Festplatte der Befundungsstation oder aus der Datenbank zur Kurzzeitspeicherung gelöscht werden.

Für die Kommunikation zwischen Befundungsstation, Datenbank und DICOM-Archiv wird das TCP/IP-Protokoll verwendet. Für die Verbindung werden die IP-Adresse des Zielrechners, die Portnummer des Servers und der Name der Datenbank verwendet. Jedes in der Patientenfliste ausgewählte DICOM-Bild kann durch Betätigung des Button ”*Komprimiertes Bild ansehen*” oder ”*Original Bild ansehen*” visualisiert werden. Dabei startet der DBV (s. Abbildung 5.6). Der Button ”*Komprimiertes Bild ansehen*” erlaubt es, die komprimierte Version des DICOM-Bildes zu betrachten. Die komprimierte Version kann 5 - 20 mal schneller als das Original geladen werden. Der Button ”*Originalbild ansehen*” zeigt das Original-Bild.

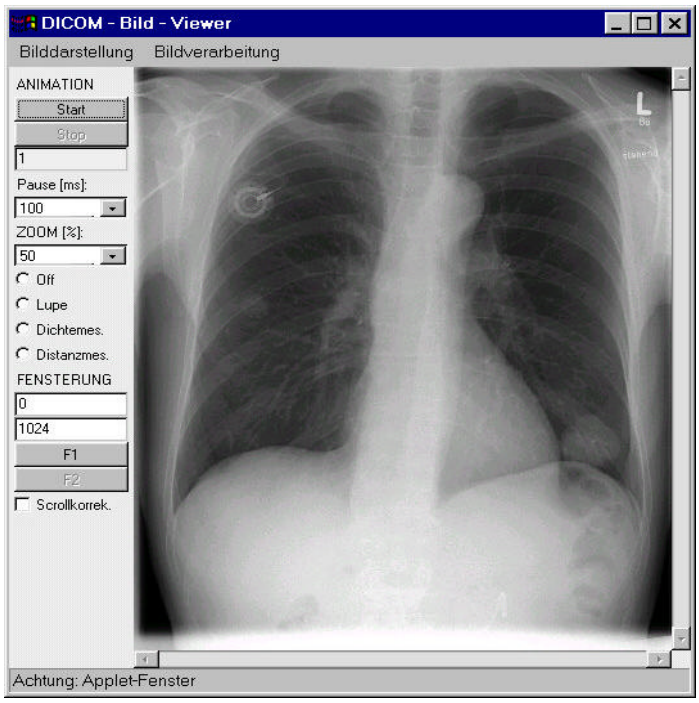


Abb. 5.6: Das Fenster des DBV

Über das folgende Protokoll organisieren Client (Befundungsstation) und Server den Bildaustausch:

- Der Client schickt dem Server die Anforderung „BILD LESEN“;
- Der Server analysiert die empfangenen SQL-Befehle und quittiert diese gegenüber dem Client mit ”OK”;

- Sobald der Client der Schnittstelle dieses ”OK” erhält, schickt er den Namen des DICOM–Bildes, z.B. ”MR10673490”, zum Server;
- Sobald der Server den Namen erhält, antwortet er durch Angabe der Länge des ausgewählten DICOM-Bildes;
- Zum Schluß verschickt der Server die Daten des DICOM-Bildes im Format ”*BYTE*”.

Im DBV werden verschiedene Methoden der Bilddarstellung und der interaktiven Bildverarbeitung bereitgestellt. Der DICOM-Bilder-Viewer stellt die folgenden Funktionen zur Verfügung:

- Layout Aufteilung;
- Scrollen;
- Manuelle Änderung der Kontrastfensterwerte;
- Movie / Cine;
- Bild spiegeln & drehen;
- Dichtemessung (für CT - Bilder);
- Bild in Zwischenablage laden;
- Lupe;
- Zoomfunktionen;
- Automatische Fensterung;
- Bild invertieren;
- Distanzmessung;
- Bildinformation ein- / ausschalten;
- Bild drucken:
- Darstellung von Einzelbildern und Bildgruppen sowie Darstellung von Bildsequenzen (z.B. CT- / MR- Applikationen) mit interaktiv einstellbaren Ablauffolgen.

Die Anwendung unterstützt sowohl Bilder mit einer Digitalisierungstiefe von 8 Bit als auch Bilder mit maximal 16 Bit pro Pixel sowie DICOM-Farbbilder mit 24 Bit pro Pixel. Die Benutzeroberfläche des Programms wird automatisch angepaßt und optimiert, insbesondere in Bezug auf unterschiedliche Bildschirmauflösungen und benutzerabhängige Browser-Einstellungen. Außerdem gewährleistet der DBV das sichere Lesen der DICOM-Bilder von verschiedenen Herstellern radiologischer Geräte.

Durch die vollständige Unterstützung des DICOM-Format und durch die geringe Größe der Programmdatei (etwa 220 Kbyte) kann die Java DV-Anwendung sehr flexibel in einem Netzwerk mit medizinischen Bilddaten eingesetzt werden

5.5 DICOM-Viewer-Applet

Das DICOM-Viewer-Applet (DV-Applet) wird auf einem WWW-Server installiert und dient der Übertragung und der Visualisierung von DICOM-Bildern auf Internet/Intranet-Client-Rechnern. Das DV-Applet ist vollständig in Java programmiert und kann auf einem beliebigen Computer mit Internetzugang und unter einem frei wählbaren Betriebssystem gestartet werden [27,47]. Als Server kann ein beliebiger Webserver (z.B. Apache) eingesetzt werden und als Client-Laufzeitumgebung dient ein Java-fähiger Internetbrowser (Netscape Communicator, MS Internet Explorer etc.). Auf diese Weise muß der

Internet/Intranet-Client keine spezielle Installation der DICOM-Viewer-Software auf seinem Rechner vornehmen. Das DV-Applet startet automatisch im Internetbrowser des Client. Medizinische Bilddaten werden in originärer Qualität via ISDN und dem Internet an die beteiligten Clients übertragen. Der DV-Applet Viewer interpretiert beliebige Bilddaten nach dem DICOM 3.0-Standard.

Ähnlich der DV-Anwendung besteht auch das DV-Applet aus zwei Teilen: der DICOM-Patientenliste-Viewer und DICOM-Bilder-Viewer. Nach erfolgreicher Authentifikation startet ein autorisierter Arzt in einem Internetbrowser das DV-Applet (aus einer HTML-Seite heraus). Nach dem Start erscheint das Fenster des PAV(Abbildung 5.7).

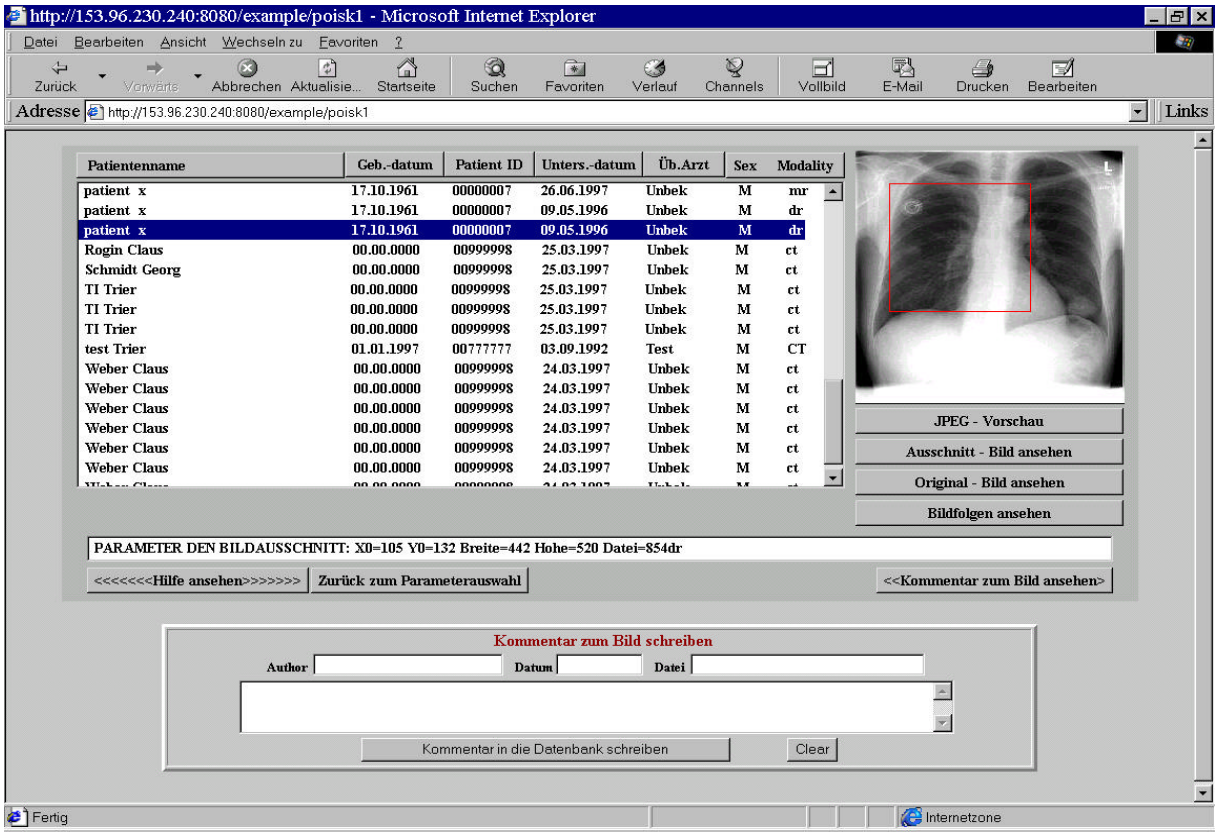


Abb. 5.7: Das Fenster des PAV-Applet

Das Fenster beinhaltet die Tabelle mit der Patientenliste. Darin befinden sich die Informationen über die Patienten und die entsprechenden DICOM-Bilder, die in der abgefragten DICOM-Datenbank nach bestimmten Kriterien ausgewählt wurden. Die Tabelle beinhaltet im einzelnen den Patientennamen, das Geburtsdatum, die ID-Nummer des Patienten, das Geschlecht, das Untersuchungsdatum, die ID-Nummer des überweisenden Arztes, die bildgebende Modalität. Für jedes in der Tabelle ausgewählte Bild kann man im Textfeld „Bildparameter“ die Bildhöhe, Bildbreite, Bit pro Pixel, Anzahl der Bilder in Bildserie betrachten. Die Informationen in der Tabelle kann man nach Geburtsdatum des Patienten, Untersuchungsdatum und Patientennamen ordnen.

Der PAV ermöglicht es, eine ausgewählte, skalierte Version eines DICOM-Bildes durch einen Druck auf den Button "JPEG-Vorschau“ anzusehen. Sollte die skalierte Version des DICOM-Bildes nicht ausreichend für die visuelle Analyse sein, kann durch Anklicken des

Buttons ”*Original Bild ansehen*” das entsprechende Bild aufzurufen werden. Dabei startet der DBV, der das Original oder das rekonstruierte DICOM-Bild visualisiert. Wenn der Arzt nur einen Ausschnitt des DICOM-Bildes ansehen möchte, markiert er mit der Maus auf der skalierten Version des DICOM-Bildes die benötigte rechteckige Bildregion und ruft sie durch Anklicken des Button ”*Ausschnitt-Bild ansehen*” auf. Auch dabei startet der DBV, der diese Bildregion dann vergrößert visualisiert. Dadurch wird die Zeit stark reduziert, die für die Übertragung der benötigten Bilddaten über das Netz notwendig ist.

Das Fenster des DBV sieht ähnlich wie das Fenster des DBV für die DV-Anwendung aus. Der DBV ermöglicht alle Funktionen des DBV für die DV-Anwendung außer „*Bild drucken*“. Innerhalb des DBV-Applets können ganze Schichtbildserien mit ca. 40 bis 120 Bildern durchgesehen werden.

Die Dateigröße des Java DV-Applet hat etwa 20 KByte (PAV) und etwa 30 Kbyte (DBV). Deshalb kann es sehr schnell über ein Netz medizinische Bilddaten übertragen.

5.6 Datenaustausch zwischen Internet/Intranet-Browser und Klinikarchiv

Im letzten Abschnitt dieses Kapitels wird das neuartige internet/intranet-orientierte radiologische Informationssystem (I^2RIS) zur Übertragung, Visualisierung und Bearbeitung der DICOM-Bilder betrachtet. Die Abbildung 5.8 zeigt die System-Darstellung und den Informationsfluss im System [109-111]. Mit Hilfe von I^2RIS kann der Arzt, der einen Zugang zum Internet hat, alle notwendigen Daten über einen Patienten, inklusive aller radiologischen Aufnahmen, von den DICOM-Modalitäten oder aus dem DICOM-Archiv eines Krankenhauses bekommen. Das System beinhaltet drei grundlegende Komponenten:

- Die bildgebenden DICOM-Modalität oder das DICOM-Archiv;
- Der Transmissions-Kontroll-Rechner (Rechner 1);
- Der Client-Internet-Browser (Rechner 2);

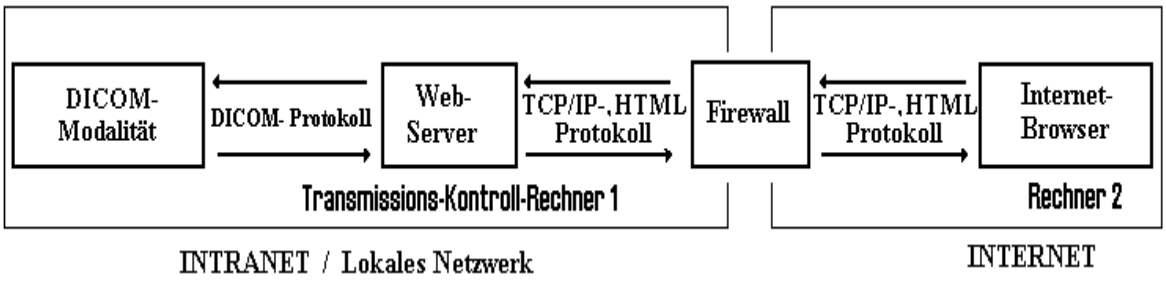


Abb. 5.8: Die System-Darstellung und den Informationsfluss im System

Die DICOM-Daten befinden sich auf dem DICOM-Modalität oder auf dem DICOM-Archiv. Auf dem Transmissions-Kontroll-Rechner wird ein Web-Server, eine Benutzerdatenbank mit den Benutzerdaten sowie die Servlet- und Applet-Klassen installiert, die die Datenverwaltung auf den DICOM-Modalitäten und auf dem DICOM-Archivierungssystem des PACS sowie die Kommunikation mit dem autorisierten Internet-

Client organisieren. Auf dem Client-Rechner wird ein normaler Internet-Browser installiert.

Die Implementationsbasis des I²RIS gründet sich auf der DICOM-Standardisierung, der modernen, plattformunabhängi-gen JAVA-Servlet-Technologie, der JAVA-Secutity-Technologie, der Intranet/Internet Technologie und der neuartigen adaptiven Bildkomprimierungstechnologie.

Das Ablaufschema für den Datenaustausch zwischen den Komponenten des Systems basiert auf der Servlet-Technologie und ist auf der Abbildung 5.9 dargestellt.

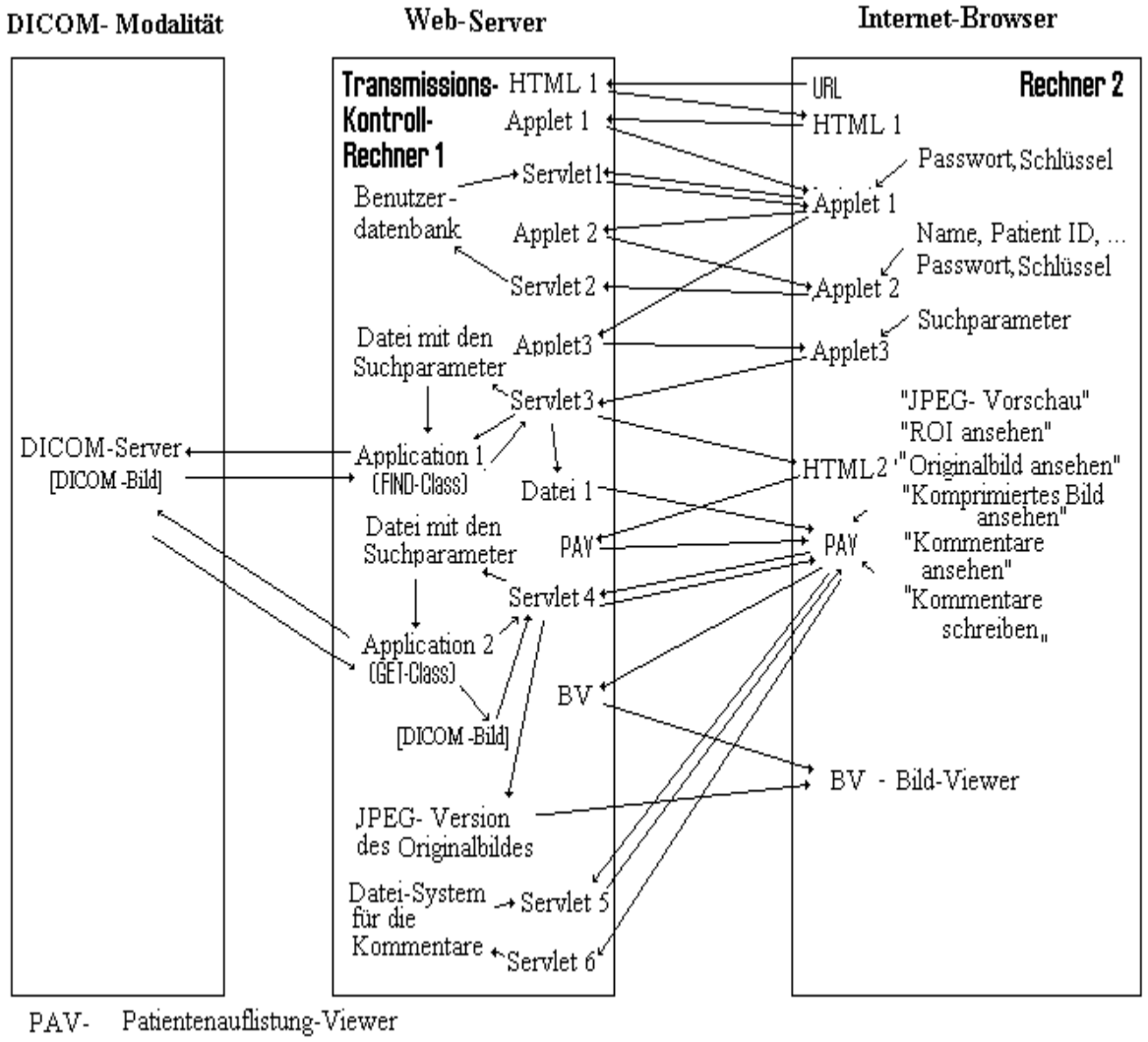


Abb. 5.9: Ablaufschema für den Datenaustausch zwischen den Komponenten des I²RIS

Der Client gibt die Internetadresse (URL) des I²RIS der medizinischen Einrichtung in seinem Internet-Browser an und ruft damit die erste HTML-Seite, die das Applet 1 beinhaltet, auf. Der Anwender wird nun aufgefordert, sich durch sein Passwort als autorisierter Benutzer zu identifizieren und die Schlüssel für die Verschlüsselung der DICOM-Daten einzugeben. Applet 1 verschlüsselt das Passwort und schickt es zum Web-Server (Rechner 1). Das Servlet 1 (Rechner 1) entschlüsselt und prüft das Passwort. Wenn

die Authentifikation nicht erfolgreich war, wird durch ein neues Applet (Applet 2) zur Angabe und Verschlüsselung persönlicher Daten des Anwenders aufgerufen. Nach der Überprüfung dieser Daten wählt der Anwender ein eigenes Passwort und die Schlüssel für die Verschlüsselung der Patienten-Daten. Passwort und die Schlüssel, sowie die persönlichen Daten des Anwender, speichert das Servlet 2 in der Benutzerdatenbank (Rechner1). Nach erfolgreicher Authentifikation ruft das Applet 1 das Applet 3 zur Angabe und Verschlüsselung der Suchparameter der benötigten DICOM-Bilder auf, die in der DICOM-Modalität gesucht werden sollen. Applet 3 startet das Servlet 3 und überträgt die Suchparameter an das Servlet 3. Servlet 3 entschlüsselt die Suchparameter, generiert die Datei mit Suchparametern und startet die Client-Java-Anwendung 1, die die DICOM-FIND-Service Class implementiert. Anwendung 1 überträgt durch das DICOM-Protokoll die Datei mit den Suchparametern an die DICOM-Modalität und empfängt die Datei mit den gefundenen Daten. Dann verschlüsselt das Servlet 3 diese Daten, speichert sie als Datei 1 und erzeugt die HTML-Seite, die das Patientenauflistung –Viewer-Applet (PAV) aufruft. Der PAV liest die Datei 1, entschlüsselt gespeicherte Daten und visualisiert sie im Internet-Browser des Client (Rechner 2).

Bei jeder Darstellungsart, egal ob nun das Originalbild, Bildausschnitte oder nur eine Preview-Version vom Client angefordert wurde, werden die Suchparameter des ausgewählten Bildes vom PAV verschlüsselt zum Web-Server übertragen und dadurch wird das Servlet 4 gestartet. Das Servlet 4 entschlüsselt die Suchparameter, generiert die Datei mit den Suchparametern und startet die Client-Java-Anwendung 2, die DICOM-OET-Service Class implementiert. Die Anwendung 2 überträgt durch das DICOM-Protokoll die Datei mit den Suchparametern an die DICOM-Modalität und empfängt das gefundene DICOM-Bild. Danach erzeugt das Servlet 4 eine skalierte Version des DICOM-Bilders, (bzw. einen Ausschnitt des Originalbildes oder das komprimierte Originalbild. Zum Schluß wird durch das Servlet 4 aus dem PAV heraus das Bild-Viewer-Applet gestartet. Das DBV-Applet liest das Originalbild (alternativ das komprimierte Bild oder den Bildausschnitt), lädt und visualisiert es in den Internet-Browsern des Benutzer (Rechner 2). Zum ausgewählten Bild kann der Benutzer (Rechner 2) die Kommentare ansehen. Dafür startet er das Servlet 5, welches auf dem Web-Rechner 1 die Datei mit den Kommentaren für das ausgewählte Bild liest, verschlüsselt und sie zum PAV auf dem Rechner 2 schickt. Der PAV entschlüsselt diese Kommentare und zeigt sie im Kommentar-Fenster an. Zum Erzeugen neuer Kommentare bietet der PAV ein Formular, in dem der Benutzer die relevanten Daten für die Zuordnung des Kommentars einträgt. Danach verschlüsselt der PAV dieses Kommentar und startet auf dem Webrechner das Servlet 6. Das Servlet 6 entschlüsselt die Kommentardaten und speichert sie in der Datei zusammen mit den Kommentaren zu dem ausgewählten Bild.

Die allgemeine Sicherheitskonzeption des Systems basiert auf der Authentifizierung, der Verschlüsselung der DICOM-Daten, sowie der Applet-Signierung. Implementationsbasis hierfür ist die Java – Kryptographie.

5.7 Schlußfolgerungen

Mit der oben beschriebenen Software können verschiedenste Anwendungsszenarien erfüllt werden. Ein erstes Beispiel für eine Internet/Intranet-Anwendung wurde in Abschnitt 5.5 beschrieben. Ein zweites einfaches Beispiel für die Anwendung in Rahmen eines lokalen Netzwerkes könnte eine kostengünstige Bildbetrachtungsworkstation sein, die mit einen

Java-Server, einer relationalen Datenbanksoftware und einer Java DV-Anwendung ausgerüstet wäre, die die DICOM-Bilder sowie patientenbezogene Informationen über das Netzwerk empfängt, die Bilder und DICOM-Header-Informationen in ihren lokalen Datenbanken speichert und danach eine patientenbezogene Bildmappenliste für den Arzt zur Verfügung stellt, um die Bilder auszuwählen und zu betrachten. Dabei können Java-Server mit relationaler Datenbanksoftware und Java DV-Anwendung getrennt auf den verschiedenen Rechnern installiert sein.

Die ersten Internet/Intranet-Anwendungsszenarien wurde in einer Laborumgebung mit den Browsern Netscape Communicator 4.01 und MS Explorer 4.0 sorgfältig getestet und insbesondere hinsichtlich der Funktionalität, der Möglichkeit des Lesens des Bildmaterial von unterschiedlichsten Herstellern und der Ausführungsgeschwindigkeit ausgewertet. Die Zeit für die Entschlüsselung und Verschlüsselung von DICOM-Daten wurde dabei nicht berücksichtigt. Die Anwendung erwies sich unter beiden Browsern im Betrieb als absolut stabil. Das vorcompilierte Java-Programm benötigt im Netscape-Browser zum Einlesen eines 512*512 Pixel großen 16-Bit Bildes ca. 1 s (Pentium-PC 200 MHz mit 100 MB/s Netzanbindung unter Win-95,Win-NT 4.0). Eine beliebige 16- auf 8-Bit Transformation des Bildes erfolgt in 1.0 s. Der Lesetest an zahlreichem Bildmaterial unterschiedlichster Hersteller hat ein problemloses Arbeiten des DBV für verschiedene Bildmodalitäten von allen bekannten Bildgeberherstellern ergeben. Tests unter dem Browser Netscape Communicator 4.01 hat die Speicherbegrenzung bei der Visualisierung von Bildern mit der Größe von mehr als 1Kb x1Kb Pixel gezeigt, die den Start des DBV-Applet unmöglich macht.

Die Software für die zweiten Anwendungsszenerien wurde in einer Laborumgebung mit der Java-Runtime-Umgebung JRE 1.15 und dem JavaPerformancePack unter dem Betriebssystem Windows 95/NT 4.0 getestet. Die Anforderungen an die Software sind ähnlich den oben genannten Anforderungen gewesen. Die Anwendung erwies sich unter beiden Betriebssystemen auch als absolut stabil. Dabei gibt es keine Begrenzung bei der Bildgröße. Die Prüfung der Qualität der Datenübertragung durch die Java-Client-Server-Technologie auf der Basis des TCP/IP-Protokolls hat eine hohe Sicherheit und Geschwindigkeit für diese Art Netzverbindung gezeigt. Der Vergleich der Ausführungsgeschwindigkeit des DV-Applet mit der DV-Anwendung bestätigt bessere Parameterwerte im Bereich Ausführungs-geschwindigkeit für die DV-Anwendung. Damit ist die Java DV-Anwendung nur wenig langsamer als eine entsprechende compilierte Programmversion in einer herkömmlichen Hochsprache (C++, Pascal).

6. Schlußbetrachtungen

Die Teleradiologie hat durch die schnelle Entwicklung digitaler Netzwerktechnologien wie ISDN und dem Internet in den vergangenen Jahren im klinischen Alltag an Bedeutung gewonnen. Diese Entwicklung ermöglicht neue Wege der Kommunikation und des Datenaustausches für Krankenhäuser sowie für niedergelassene Ärzte. Hierbei kommt multimedialen Systemen, die den Austausch und die Besprechung medizinischer Bilder (Röntgenbilder, CT, MR-Aufnahmen etc.) in Kombination mit Befunden, Patientenstammdaten etc. ermöglichen, besondere Bedeutung zu.

Die vorliegende Arbeit hat die heutige technische Ausrüstungssituation der Radiologie in den einzelnen Krankenhäusern kritisch betrachtet, und darauf aufbauend die grundlegenden Anforderungen an ein PACS und dessen Struktur bestimmt. Darüberhinaus bot sie mögliche adäquate Modelle von DICOM-Datenbanken und entwickelte Algorithmen zur adaptiven verlustfreien DICOM-Bildkompression. Dies alles wurde in das hier vorgestellte, neue, auf der Programmiersprache Java basierende Intranet/Internet Radiologiesystem für die Archivierung und Übertragung medizinischer Daten im DICOM-Format integriert.

Im ersten Kapitel werden bekannte telemedizinische Systeme und Software von unterschiedlichen Herstellern radiologischer Geräte betrachtet und analysiert. Aufgrund der Ausgangssituation an den Kliniken und in den radiologischen Praxen sind die Aufgaben dieser Doktorarbeit formuliert, deren Lösungen in den folgenden Kapitel dargelegt werden.

Das zweite Kapitel präsentiert die gesamte Aufbaukonzeption für ein preiswertes, intranet-/internet-basiertes, modulares, radiologisches Informationssystem für Kliniken und niedergelassene Ärzte. Das System ermöglicht den internen/externen Zugriff auf DICOM-Daten, die von den verschiedenen bildgebenden Modalitäten auch unterschiedlicher Hersteller stammen können. Dies wurde durch die Verwendung der DICOM-Standardisierung, der modernen plattformunabhängigen Programmiersprache Java und der Intranet/Internet-Technologie erreicht.

Kern des Systems sind die digitalen bildgebenden Modalitäten (Bildgebergeräte), das Archivierungssystem, der Security Rechner, die Befundungsstationen, die Praxenrechner und das Netzwerk. Dabei ist die Integration aller digitalen Modalitäten über den DICOM-Standard eine essentielle Voraussetzung des Systemskonzepts.

Das Archivierungssystem besteht aus einer zentralen DICOM-Datenbank und einem verteilten Datenbanksystem. Die DICOM-Datenbank speichert kurzfristig alle digitalen Aufnahmen vor ihrer Analyse durch den Arzt. Das verteilte Datenbanksystem dient der langfristigen Speicherung aller DICOM-Bilder, die nach der Analyse für die langfristige Speicherung akzeptiert worden sind. Das verteilte objektorientierte Datenbanksystem besteht aus einer relationalen Datenbank und einer objektorientierten DICOM-Datenbank als selbständige Objekte. Eine solche Architektur gibt dem Datenbanksystem Offenheit, Sicherheit und eine hohe Geschwindigkeit bei der Datenmanipulation. Alle DICOM-Daten, archiviert und bearbeitet durch die Komponenten des Systems, sind über das lokale

(Intranet) und globale (Internet) Netzwerk durch den Einsatz moderner Java- und Web-Technologie auf den PC- und Stationsarbeitsplätzen der Ärzte und Praxen verfügbar.

Das zweite Kapitel betrachtet ferner die praktischen Grundlagen des Informationsmodells für DICOM-Daten und Datenbankmodelle, gewährt eine kurze Übersicht über heute vorhandene DICOM-Datenbanken und schlägt eine neue Struktur der relationalen DICOM-Datenbank für die Speicherung von Header-Attributen und einem Dateisystem für die Speicherung von Rohdaten als Original-, komprimierte und skalierte Bilder vor.

Danach beschreibt dieses Kapitel die Struktur einer objektorientierten DICOM-Datenbank, indem ein neues geeignetes Modell von Klassenhierarchien entwickelt wird. Die Konzeption der so ausgearbeiteten OODDB basiert auf zwei Objekten. Das erste Objekt ist das „Server DBMS“-Objekt, das die DBMS für OODDB implementiert und das DICOM-Protokoll für den Datenaustausch unterstützt. Das zweite Objekt ist das Objekt „Image“, das das reale DICOM-Bild und die Sammlung der Methoden für die Manipulation der Attribute und Rohdaten darstellt. Die vorgeschlagene Struktur der objektorientierten DICOM-Datenbank eignet sich sehr gut für verteilte Netzwerkimplementierung. Die Verwendung der JAVA-Technologie verleiht der Software der OODDB Unabhängigkeit vom Betriebsystem und große Flexibilität bei der Einführung im Klinikum und in der Praxis.

Das dritte Kapitel gibt einen Überblick über die bekannten Bildformate und verlustfreien und verlustbehafteten Bildkompressionsverfahren, sowie die Beschreibung der Bildeigenschaften, die für die Auswahl des effektivsten Kompressionsverfahren entscheidend sind. Wenn die Bildeigenschaften den Eigenschaften des Bildmodells entsprechen, das im Algorithmus verwendet wird, dann liefert dieser Algorithmus größere Kompressionsraten als andere Algorithmen, bei gleichen Bedingungen für die Qualität des rekonstruierte Bildes. Außerdem werden in diesem Kapitel die wichtigsten Anforderungen an Kompressionsalgorithmen für medizinische Bilder betrachtet.

Dieses Kapitel bietet zwei neue, verlustfreie Komplexitätsalgorithmen für die (adaptive) DICOM-Bildkompression, die aus verschiedenen verlustfreien und verlustbehafteten Algorithmen besteht. Die Kombination dieser Algorithmen ist von den Bildeigenschaften des zu komprimierenden Bildes abhängig. Im Rahmen der Entwicklung dieser adaptiven Kompressionsalgorithmen werden zwei neue Verfahren zur Analyse der Ortstruktur der DICOM-Bilder und ein neues Kriterium zum Vergleichen zweier Bilder vorgestellt.

Für den ersten Algorithmus wird ein Verfahren zur Klassifizierung von Originalbildern in unterschiedliche Klassen aufgrund einer vorläufigen Analyse des Waveletkoeffizienten, mit der nachfolgenden Kompression der Bildern mit einem geeignetem verlustfreien Kompressionsalgorithmus vorgeschlagen.

Für den zweiten Algorithmus wird ein Verfahren zur Klassifizierung der Bitebenen in unterschiedliche Klassen (aufgrund einer vorläufigen Analyse der Segmentparameter der Bitebenen) für die effektivste Aufteilung des Originalbild in zwei Bilder mit anschließender Kodierung jedes Bildes entsprechend GIF- und JPEG-Verfahren geboten.

Zum Schluß des Kapitel werden die Ergebnisse der statistischen Modellierung der beschriebenen adaptiven Verfahren für DICOM-Bilder dargestellt, die die Effektivität der

vorgeschlagenen adaptiven Verfahren bestätigen. Es handelt sich dabei um einen Vergleich der vorgeschlagenen adaptiven Kompressionsalgorithmen miteinander und mit bekannten verlustbehafteten Verfahren, der den Vorzug der GIF-JPEG-Kompression vor bekannten verlustbehafteten und verlustfreien Verfahren (bei der Begrenzung der Qualität des rekonstruierten Bildes (PSNR>40dB)) belegt. Dabei wurden beide Algorithmen mit der plattformunabhängigen Programmiersprache JAVA realisiert. Die Größe des Decodier-Applets, insbesondere für GIF-JPEG –Dekoder, beträgt ungefähr 600 Byte, was sehr wichtig ist, bedenkt man die starke Begrenzung der Übertragungsgeschwingigkeit derzeitiger Netzwerke (z.B. ISDN). Da LZW und das JPEG–Verfahren standardisierte Verfahren in Internet-Browsern sind, ist der LZW-JPEG-Kompressor für Internet-Anwendung besonders gut geeignet.

Das fünfte Kapitel widmet sich der Implementation der Komponenten des Radiologie Informationssystem in der Programmiersprache Java.

Im Rahmen dieses Kapitel werden folgende Softwarekomponenten ausgearbeitet und vorgestellt:

- Zwei Schemata und die entsprechenden Java-Klassen für den Datenaustausch zwischen den Komponenten des RIS mittels des TCP/IP-Protokolls im lokalen Netz des Krankenhauses auf der Basis der Java-Anwendung, des Java-Applets und der JDBC-Software

- Eine alternative Lösung der oben genannten Aufgabe mittels der Servlet-Technologie für den DICOM-Datenaustausch über Intranet/Internet.

- Software zur Visualisierung und Bearbeitung von DICOM-Bildern in Form einer Java DICOM-Viewer-Anwendung und eines Java-DICOM-Viewer-Applet. Dabei unterstützt die DICOM-Viewer-Anwendung die Datenübertragung mittels TCP/IP-Protokoll und wird lokal auf jeder Befundungsstation installiert. Das DICOM-Viewer-Applet wird auf dem Web-Server-Rechner installiert und startet automatisch in einem Internetbrowser des Clients, ohne zusätzliche Installation auf dem Client-Rechner. Außerdem werden wichtige Funktionen für die Bearbeitung der patientenbezogenen Daten und Bilddaten beschrieben.

- Die Schemata der Anwendungsszenarien des Datenaustausches zwischen Internet/Intranet-Anwender und Klinikarchiv basieren auf dem Web-Server, dem Java-Server, der relationalen Datenbank, dem DICOM-Viewer-Applet, der Web- und Servlet-Technologie zur Datenübertragung und -bearbeitung basiert. Die Testergebnisse, die sich aus der Implementation dieser Internet/Intranet-Anwendungsszenarien in einer Laborumgebung mit den Browsern Netscape Communicator 4.01 und MS Explorer 4.0 ergeben, belegen die gute Funktionalität und die hohe Ausführungsgeschwindigkeit, wobei Bildmaterial von unterschiedlichsten Herstellern problemlos gelesen wurde. Die Software erwies sich dabei als stabil und im Datenübertragungstest konnte dank des Client/Server-Kommunikationsmodells und der Java-Servlet- und Java-Applet-Technik sowohl eine hohe Sicherheit als auch eine hohe Geschwindigkeit der Datenübertragung sichergestellt werden.

Somit entwirft diese Doktorarbeit ein neues, auf der Programmiersprache Java basierendes Web-orientiertes Teleradiologiesystem für die Archivierung und Übertragung medizinischer Daten, dessen Konzept im Folgenden vorgestellt wird.

Mit dem Teleradiologiesystem können medizinische Bilddaten unabhängig von vorhandenen Betriebssystemen oder Hardwareplattformen archiviert, in eine eigene, verteiltete Datenbank übertragen und während einer Diagnosestellung an verschiedenen Orten visualisiert und bearbeitet werden.

Das Teleradiologiesystem beinhaltet eine verteilte Bilddatenbank und einen WWW-Server. Die verteilte DICOM-Datenbanksystem kann auch aus stark heterogener Hardware (PCs, Apple Macintosh Rechner, Workstations) bestehen, auf der verschiedene gängige Betriebsysteme (UNIX, Machintosh, OS/2, DOS, Windows NT) und relationale Datenbanken (Orakle, Sybase, Access usw.) laufen können.

Der WWW-Server basiert auf einem Client/Server-Modell und dient der Kommunikation des Internet-Clients mit der verteilten DICOM-Datenbanksystem. Als Server kann der frei verfügbare Apache Webserver und als Client ein beliebiger Web-Browser z.B. Netscape Navigator eingesetzt werden. Die Verarbeitung und die Auswertung kann auf dem Java-Server parallel durchgeführt werden.

Die Software des Teleradiologiesystems wurde in Java programmiert. Daher kann das Programm auf einem beliebigen Computer mit Internetzugang und unter einem frei wählbaren Betriebssystem gestartet werden. Als Laufzeitumgebung dient ein Java-fähiger Internetbrowser. Für die Übertragung der Bilddaten unterstützt das System das Kommunikationsprotokoll TCP/IP. Die auf dem Client hergestellten Bilder, die aus der lokalen Datenbank des verteilten Bilddatenbank ausgewählt wurden, werden durch eine Java-Anwendung und ein Java-Servlet auf den WWW-Server geladen. Durch das Java-Applet DICOM-Viewer werden sie zum Client zurückgeleitet und dort vom Web-Browser am Bildschirm angezeigt. Dieser Java-DICOM-Viewer interpretiert beliebige Bilddaten nach dem DICOM 3.0-Standard.

Um die Vertraulichkeit und Authentizität der übertragenen Datenpakete zu gewährleisten, werden die übertragenen DICOM-Daten und DICOM-Bilder durch die Java-API-Security mit eigenen Kodierungsprogrammen verschlüsselt. Diese Kodierungsprogramme sind unsere eigene Entwicklung und bestehen aus verlustfreien und verlustbehafteten Kompressionalgorithmen. Die Kodierung ermöglicht eine verbesserte Datensicherheit und gleichzeitig eine 3 bis 15mal höhere Geschwindigkeit der Datenübertragung ohne den Verlust von Bildinhalten. Das entsprechende Decodierungs-Applet ist im Java-DICOM-Viewer implementiert.

Am Institut für Telematik werden Experimente durchgeführt, in denen die automatische Auswertung der DICOM-Bilder durch einen WWW-Server verwirklicht wurde. Der Anwender (Arzt) kann die Bilder mit Hilfe eines Netscape Navigators aus der verteilten Bilddatenbank auswählen (siehe Demo: http://www.ti.fhg.de/dicomv), dann auf den WWW-Server laden und sie dann auf dem Bildschirm des eigenen Arbeitsrechners visualisieren.

Ausblicke zur Arbeit

Die Sicherheit von Datenübertragungen über das Internet ist heute von enormer Bedeutung. Im Bereich der Teleradiologie ist dieses Thema von besonderer Brisanz, da es sich bei den zu übertragenden Daten um Informationen handelt, die, außer den normalen datenschutzrechtlichen Bestimmungen, auch dem ärztlichen Schweigegebot unterliegen. Der Schutz dieser Informationen vor unberechtigtem Zugriff muß also unbedingt gegeben sein. In dieser Arbeit konnte dieses Thema nicht erschöpfend behandelt werden, denn das hätte in jedem Fall den Rahmen einer Doktorarbeit gesprengt. Dieses Feld ist das Thema für eigenständige Doktorarbeiten.

Nur sei soviel gesagt, daß es wichtiger ist, die DICOM-Header-Informationen zu schützen, als die Bildrohdaten, da die personenbezogenen Daten dort abgelegt werden. Der enorme Umfang der Standard-DICOM-Header ist aber für Codierungszwecke unpassend, da ein nicht unbeträchtlicher Zeitaufwand betrieben werden muß, um diese Datenmenge zu verschlüsseln, die codierten Daten zu übertragen und danach die Daten wieder zu entschlüsseln. Eine Möglichkeit, diesen Zeitaufwand so klein wie möglich zu halten, ist die in dieser Arbeit schon vorgestellte Minimierung der Attributmenge auf medizinisch relevante Attribute.

Die wohl optimale Lösung zur Gewährleistung der sicheren Übertragung von Daten über offene Netzwerke bieten heute Trust Center, wie das des Institutes für Telematik in Trier. In absehbarer Zeit werden hier die notwendigen Voraussetzungen geschaffen worden sein, um Systeme wie das hier beschriebene RIS durch Trust Center-Zertifizierung nach neuesten Erkenntnissen und Methoden sicher für die Nutzung im Internet zu machen.

7. Anhang

7.1 A.

Parameter der Systeme für die Archivierung, Übertragung, Verarbeitung und Visualisierung von medizinischen Bildern :

System-name	Program-mier-sprache	Protokoll Datenaus-tauschen	Daten-Format	Betriebs-system	Daten-bank-modell	Daten-archiv	Schnitt-stelle mit anderen KIS, RIS	Netzver-bindung über	Soft-ware-modu-larität
Advan-tage Win.	C++	HTML, TCP/IP, DICOM	DICOM	UNIX	Rela-tional	+	HL7 DICOM	WEB-FTP-server	-
TIM	Delphi-3	TCP/IP, DICOM	DICOM TIFF JPEG	Windows	Rela-tional	-	DICOM	DICOM-FTP-server	-
Medora-PACS	C++	TCP/IP, DICOM	DICOM	Windows	Rela-tional	+	HL7, SQL, DICOM	DICOM-FTP-server	+
Kame-din	C++	TCP/IP, DICOM	DICOM	UNIX Windows	Rela-tional	+	DICOM, SQL	DICOM-FTP-server	-
SIENET	C++	TCP/IP, DICOM	DICOM	UNIX	Rela-Tional	+	HL7 DICOM	DICOM-FTP-server	-
Medicus	C++	TCP/IP, DICOM	DICOM	UNIX	Rela-tional	+	HL7 DICOM	DICOM-FTP-server	+
CHILI	C++	HTTP TCP/IP, DICOM, E-mail	DICOM File-Format	UNIX	Rela-tional	+	HL7 DICOM	DICOM-FTP-E-mail-server	+
DOIA	HTML PERL Vis.Basic C++	HTML TCP/IP	JPEG HTML	Web-basiert	Ver-teiltes System, Rela-tional	-	-	WEB-server	+
ISIS	C++	TCP/IP	JPEG DICOM	UNIX	Rela-tional	+	HL7, SQL DICOM	DICOM-FTP-server	-

7.2 B.

Der DICOM-Header eines MR-Bildes

Object type:	ELEMENT LIST		
Object size:	137134		
Group: 0008,	Length:	398	
0008 0008	26 //	ID Image Type//	ORIGINAL\PRIMARY\UNDEFINED
0008 0016	26 //	ID SOP Class UID//	1.2.840.....5.1.4.1.1.4
0008 0018	48 //	ID SOP Instance UID//	1.3.12.2.....199707162028330686308
0008 0020	8 //	ID Study Date//	19970626
0008 0022	8 //	ID Acquisition Date//	19970626
0008 0023	8 //	ID Image Date//	19970626
0008 0030	14 //	ID Study Time//	090415.543000
0008 0032	14 //	ID Acquisition Time//	090416.451000
0008 0033	14 //	ID Image Time//	090714.000000
0008 0040	2 //	ID Data Set Type (RET)//	Unimplemented
0008 0041	8 //	ID Data Set Subtype (RET)//	Unimplemented
0008 0050	0 //	ID Accession Number//	
0008 0060	2 //	ID Modality//	MR
0008 0070	8 //	ID Manufacturer//	SIEMENS
0008 0080	26 //	ID Institution Name//	KRANKENHAUS TRIER
0008 0090	0 //	ID Referring Physician's Name//	
0008 1010	16 //	ID Station Name//	mrc
0008 1090	26 //	ID Manufacturer Model Name//	MAGNETOM IMPACT
Group: 0010,	Length:	106	
0010 0010	26 //	PAT Patient Name//	DUCK, DONALD
0010 0020	12 //	PAT Patient ID//	123/97
0010 0030	8 //	PAT Patient Birthdate//	19560911
0010 0040	2 //	PAT Patient Sex//	M
0010 1010	4 //	PAT Patient Age//	040Y
0010 1030	6 //	PAT Patient Weight//	000120
Group: 0018,	Length:	498	
0018 0010	8 //	ACQ Contrast/Bolus Agent//	NONE
0018 0020	2 //	ACQ Scanning Sequence//	RM
0018 0021	10 //	ACQ Sequence Variant//	NONE\NONE
0018 0022	0 //	ACQ Scan Options//	
0018 0023	0 //	ACQ MR Acquisition Type //	
0018 0024	8 //	ACQ Sequence Name//	se1
0018 0025	2 //	ACQ Angio Flag//	N
0018 0050	14 //	ACQ Slice Thickness//	004.000000E+00
0018 0080	14 //	ACQ Repetition Time//	005.400000E+02
0018 0081	14 //	ACQ Echo Time//	001.500000E+01
0018 0083	6 //	ACQ Number of Averages//	000002
0018 0084	14 //	ACQ Imaging Frequency//	004.046476E+01
0018 0085	8 //	ACQ Imaged Nucleus//	1H
0018 0086	6 //	ACQ Echo Number//	000001
0018 0087	8 //	ACQ Magnetic Field Strength//	0.95007
0018 0088	4 //	ACQ Spacing Between Slices//	1.2
0018 0090	6 //	ACQ Data Collection Diameter//	000240
0018 0091	2 //	ACQ Echo Train Length//	1
0018 0093	4 //	ACQ Percent Sampling//	100
0018 0094	4 //	ACQ Percent Sampling Field of View//	100
0018 1000	26 //	ACQ Device Serial Number//	5603
0018 1020	8 //	ACQ Software Version//	VB31B
0018 1200	8 //	ACQ Date of Last Calibration//	19961022
0018 1201	14 //	ACQ Time of Last Calibration//	212037.000000
0018 1250	16 //	ACQ Receiving Coil//	CP Head
0018 1314	2 //	ACQ Flip Angle//	90
0018 1316	2 //	ACQ SAR//	0
0018 4000	52 //	ACQ Comments//	Unimplemented

```
0018 5100        4 //    ACQ Patient Position//                    HFS

Group: 0020,     Length: 684
0020 000d        48 //   REL Study Instance UID//                  1.3.12.2.11.......19970716202753084918 1
0020 000e        48 //   REL Series Instance UID//                 1.3.12.2.11.......19970716202753092148 8
0020 0010        6 //    REL Study ID//                            000002
0020 0011        6 //    REL Series Number//                       000001
0020 0012        6 //    REL Acquisition Number//                  000001
0020 0013        6 //    REL Image Number//                        000004
0020 0030        42 //   REL Image Position (RET)//                Unimplemented
0020 0032        26 //   REL Image Position Patient//              -47.6289\-122.055\118.151
0020 0035        84 //   REL Image Orientation//                   Unimplemented
0020 0037        30 //   REL Image Orientation (Patient)//         0\1\-0\-0.0174524\-0\-0.999848
0020 0050        6 //    REL Location (RET)//                      Unimplemented
0020 0052        48 //   REL Frame of Reference UID//          1.3.12.2.1107.....19970716202753092172 1
0020 0070        8 //    REL Image Geometry Type (RET)//           Unimplemented
0020 0080        0 //    REL Masking Image (RET)//                 Unimplemented
0020 1001        6 //    REL Acquisitions in Series//              Unimplemented
0020 1020        30 //   REL Reference (RET)//                     Unimplemented
0020 1040        0 //    REL Position Reference Indicator//
0020 1041        6 //    REL Slice Location//                      -49.7
0020 3100        0 //    REL Source Image IDs (RET)//              Unimplemented
0020 3401        0 //    REL Modifying Device ID (RET)//           Unimplemented
0020 3402        0 //    REL Modified Image ID (RET)//             Unimplemented
0020 3403        0 //    REL Modified Image Date (RET)//           Unimplemented
0020 3404        0 //    REL Modifying Device Mfr (RET)//          Unimplemented
0020 3405        0 //    REL Modified Image Time//                 Unimplemented
0020 3406        0 //    REL Modified Image Description (RET)//    Unimplemented
0020 5000        12 //   REL Original Image ID (RET)//             Unimplemented
0020 5002        50 //   REL Orig Image ID Nomenclature (RET)//    Unimplemented

Group: 0028,     Length:  246
0028 0002        2 //    IMG Samples Per Pixel//                   1          1
0028 0004        12 //   IMG Photometric Interpretation//          MONOCHROME2
0028 0005        2 //    IMG Image Dimensions (RET)//              Unimplemented
0028 0010        2 //    IMG Rows//                                100        256
0028 0011        2 //    IMG Columns//                             100        256
0028 0030        28 //   IMG Pixel Spacing//                       009.375000E-01\09.375000E-01
0028 0040        4 //    IMG Image Format (RET)//                  Unimplemented
0028 0050        0 //    IMG Manipulated Image (RET)//             Unimplemented
0028 0060        4 //    IMG Compression Code//                    Unimplemented
0028 0100        2 //    IMG Bits Allocated//                      10         16
0028 0101        2 //    IMG Bits Stored//                         c          12
0028 0102        2 //    IMG High Bit//                            b          11
0028 0103        2 //    IMG Pixel Representation//                0          0
0028 0200        2 //    IMG Image Location//                      Unimplemented
0028 1050        12 //   IMG Window Center//                       000758\00758
0028 1051        12 //   IMG Window Width//                        001572\01572
0028 1052        6 //    IMG Rescale Intercept//                   000000
0028 1053        6 //    IMG Rescale Slope//                       000001

7fe0 0010        131072// PXL Pixel Data//                        Data on disk
DCM Dump Elements Complete
```

7.3 C.

Die momentan gebräuchlichen Bitmap-Formate:

Das GIF-Format stammt vom amerikanischen Online-Dienst CompuServe [59,60]. Es wurde entwickelt, um die Übertragungszeiten von Bildinformationen gering zu halten. Es ist neben JPEG das meistverbreiteste Grafikformat im Internet.

Ein GIF-Bild besitzt eine Farbpalette, die bis zu 256 Farben oder Grauwerten (8 Bit pro Pixel) enthalten kann. Je kleiner die Tabelle, desto geringer ist der Platzbedarf der Datei. Gute Grafik- und Bildbearbeitungsprogramme erlauben eine Umrechnung der Farbtiefe in 1-Bit-Schritten, wodurch der Platzbedarf der Dateien individuell angepaßt werden kann. Die Bildabmessungen sind auf 16000x16000 Pixel begrenzt. Weiterhin werden die Bilddaten durch die verlustfreie Lauflängenkodierung (LZW) komprimiert.

Das GIF-Format eignet sich besonders für Grafiken, die wenige optische Informationen enthalten, also beispielsweise Bilder mit Schrift. Im Idealfall wenn ein Bild nur aus einer Farbe besteht wird eine Komressionsrate von 30:1 erreicht. Im ungünstigsten Fall - wenn sich jeder Pixel von seinem Nachbarn unterscheidet - erreicht man mit der LZW noch eine Kompression von 65 Prozent.

Im JPEG-Format (Joint Photographic Experts Group) lassen sich Bilder ab 8 Bit Farbtiefe, bzw. Graustufenbilder, speichern [59,61,62]. Die Speicherung eines, in Graustufen umgewandelten, S/W-Bildes im JPEG-Format führt auch bei höchster Kompressionsrate zu einer höheren Datenmenge als wenn es das Bild im GIF-Format gespeichert wird. Folglich ist JPEG für S/W-Bilder ungeeignet.

In JPEG-Dateien werden ähnliche Farbflächen zusammengefaßt. Dabei kann der Grad der Kompression variiert werden, wodurch die Bildqualität mehr oder weniger verschlechtert wird. Je höher man den Kompressionsgrad wählt, desto schlechter wird die Bildqualität.

Das JPEG-Kompressionsverfahren ist ein verlustbehaftetes Verfahren. Es ist sehr gut geeignet für Graustufen- und Farbbilder aller Farbtiefen, was ein Vorteil der JPEG-komprimierten Grafiken gegenüber GIF ist. Bilder mit fein abgestuften Verläufen, die wesentlich mehr als 256 verschiedene Farben enthalten, sollten als JPEG-Datei gespeichert werden.

TIFF (Tag Image File-Format) ist eine Entwicklung der Firma Aldus Corporation [59]. Dieses Format hat sich in den letzten Jahren zu einem der wichtigsten Formate für Rasterdateien entwickelt. Es wurde von Anfang an sehr umfangreich konzipiert. Somit bietet es eine Vielzahl von Speichermöglichkeiten und neben den eigentlichen Grafikdaten auch Angaben wie der Name der benutzten Grafiksoftware oder der Scannertyp.

Dieses Format ist universell für alle Anwendungen (wenn das Format beherrscht wird). Es ist jedoch auch leicht möglich Fehler einzubauen, die zur Fehlinterpretation führen. Selbst namhafte Hersteller von Grafikprogrammen sind nicht in der Lage dieses Format in der ganzen Vielfalt zu überblicken.

TIFF ist in der Lage, Schwarz/Weiß-, Grauwert- und Farbbilder bis zu 24 Bit RGB und bis zu 32 Bit CMYK zu speichern. Die Bildgröße ist auf ca. 4 Milliarden Bildzeilen "begrentzt". Es ist möglich in einer TIFF-Datei mehrere Bilder zu speichern.

BMP ist das unter dem Betriebsystem MS-Windows 3.0 entstandene und durch die weiteren Versionen dieser Betriebssystemfamilie häufig verwendete Bildformat [59,63]. Die Daten können optional mit Komprimierung abgelegt werden. Leider wird in den meisten Fällen auf eine Komprimierung verzichtet. Das Format ist unabhängig vom verwendeten Ausgabegerät definiert (device-independent), es kann sowohl auf verschiedenen Bildschirmen oder Druckern ausgegeben werden und wird mittlerweile auch von Programmen anderer Betriebsysteme unterstützt. Dem BMP-Format liegt das RGB-Farbmodell zugrunde. Durch den geeigneten Aufbau der Farbtabellen ist auch die Kodierung von monochromen und von Graustufenbildern möglich.

Das DICOM-Format [29,30] ist TAG-basiert, das heißt, daß Datenfelder durch Kennungen in Form von standardisierten Einzelfeld- und Gruppennummern gekennzeichnet sind. Dadurch ist ein Datenzugriff unabhängig von der Reihenfolge der Datenfelder möglich. Außerdem ergeben sich nach diesem Prinzip optionale, nicht obligatorisch vorhandene Datenfelder abhängig vom gewünschten Dateiinhalt. Gleichzeitig wird zu einzelnen Datenfeldern die Länge des Datenfeldes gespeichert.

Für die Interpretation der Bilddaten bedeutet dies, daß aus einer entsprechenden Bilddatei die Informationen zu Bildgröße, Länge des Bildrohdatenblocks und bitweisem Aufbau der Pixeldaten ausgelesen werden.

Die durch DICOM definierten Übertragungsprotokolle werden für die Darstellung im WWW nicht genutzt.

7.4 D.

JAVA-Klasse „Server“

```
import java.io.*;
import java.net.*;
import java.util.*;

public class server {

private static int SERVER_PORT=1704;
private static final int Max_Clients =50;
static String path="c:\\dicombilder\\";
private static final File Data_File=new File(path);
private ServerSocket listSock=null;
private String[] stockInfo;
private Date stockInfoTime;
private long stockFileMod;
private boolean keepRunning=true;
static utilit ut=new utilit();
String data[][];

public static void main(String[] args)
{
  if (args.length>=1)
  { SERVER_PORT=ut.StrInRad((args[0].trim()),10);}
  if (args.length>1)
  { path=args[1];}

        String spis[][]=new String[2][23];
        for(int i=0;i<2;i++)
        for(int j=0;j<23;j++)
        {spis[i][j]="k";}
        spis[0][0]=""+1;
        spis[0][1]=path+"dicom_bild\\";
        ut.wristr(spis,path+"data.txt",2,23);
          }

dicomserver server=new dicomserver();
  server.serverQuotes();
}

public dicomserver()
 {
  load_data();
  try{ listSock=new ServerSocket(SERVER_PORT,Max_Clients);}
  catch(IOException excpt) {
      System.err.println("Keine Möglichkeit den Port hören"+SERVER_PORT+": "+excpt);
      System.exit(1); }
  }

 protected boolean load_data()
 {
  data=new String[2][23];
  data=ut.redstr(path+"data.txt",23);
  int k=ut.StrInRad(data[0][0],10);
  System.out.println("k="+k);
  stockInfoTime=new Date();
  return true;
 }

 public void serverQuotes()
 {
   Socket clientSocket=null;
```

```
  try {  while(keepRunning)
          {
            clientSocket=listSock.accept();
            if(stockFileMod != Data_File.lastModified()) {load_data();}
            stockquotehandler newHandler = new stockquotehandler(clientSocket,data,
            stockInfoTime,path);
            Thread newHandlerThread=new Thread(newHandler);
            newHandlerThread.start();
          }
         listSock.close();
      }
  catch(IOException excpt)
      {
       System.err.println("1_Failed I/O: "+excpt);
      }
 }

protected void stop()
 {
   if(keepRunning)
   {keepRunning=false;}
  }
}

class stockquotehandler implements Runnable {

   utilit ut1=new utilit();
   head_ser he=new head_ser();
   private Socket mySocket=null;
   private PrintStream clientSend = null;
   private DataInputStream clientReceive=null;
   private String[][] stockInfo;
   private Date stockInfoTime;
   private String path="";
   byte b[];
   Date data1;
   int dat_jahr=0,dat_mon=0,dat_tag=0,dat_hour=0,dat_min=0,dat_sec=0;
   Runtime ry=Runtime.getRuntime();
   Process p=null;
   Process p1=null;
   byte buf_bmp[];
   int numer=0;

   public stockquotehandler (Socket newSocket,String[][] info, Date time,String path1)
   {
      mySocket=newSocket;
      stockInfo=info;
      stockInfoTime=time;
      path=path1;
   }

   public void run()
   {
     String nextLine;
     int k=ut1.StrInRad(stockInfo[0][0],10);
     System.out.println("k1="+k);

     String quoteResponse;
     try{
        clientSend=new PrintStream(mySocket.getOutputStream());
        clientReceive=new DataInputStream(mySocket.getInputStream());
        clientSend.println("+Hello "+stockInfoTime);
        clientSend.flush();
        while((nextLine=clientReceive.readLine())!=null)
         {
```

```
        nextLine=nextLine.trim();
        System.out.println("nextLine="+nextLine);
//-----------------------------------------------------------------------
        if(nextLine.equals("lesendata"))
//-----------------------------------------------------------------------
         {
          quoteResponse=getQuote(0,0);
          clientSend.println(quoteResponse);
          clientSend.flush();
          for(int i=0; i<k;i++)
          for(int j=0;j<23;j++)
           {
            quoteResponse=getQuote(i,j);
            clientSend.println(quoteResponse);
            clientSend.flush();
           }
         }
//-----------------------------------------------------------------------
        if(nextLine.equals("schreibendata"))
//-----------------------------------------------------------------------
         {
          clientSend.println("ok");
          clientSend.flush();
          nextLine=clientReceive.readLine();
          k=ut1.StrInRad((nextLine.trim()),10);
          System.out.println("k2="+k);
          stockInfo=new String[k][23];
          for(int i=0; i<k;i++)
          for(int j=0;j<23;j++)
          {
           nextLine=clientReceive.readLine();
           stockInfo[i][j]=nextLine;
          }
         ut1.wristr(stockInfo,path+"data.txt",k,23);
         }
//-----------------------------------------------------------------------
       if(nextLine.equals("holenfile"))
//-----------------------------------------------------------------------
        {
         clientSend.println("ok");
         clientSend.flush();
         nextLine=clientReceive.readLine();
         System.out.println("nextLine="+nextLine);
         File f1=new File(path+nextLine);
         long faillang=f1.length();
         int lang=(int)faillang;
         b=new byte[lang];
         try{
               FileInputStream f2=new FileInputStream(path+nextLine);
               f2.read(b,0,lang);
               f2.close();
             }
         catch(IOException e){;}
         String lang_s=""+lang;
         clientSend.println(lang_s);
         clientSend.flush();
         clientSend.write(b,0,lang);
         clientSend.flush();
        }
//-----------------------------------------------------------------------
      if(nextLine.equals("schickenfile"))
//-----------------------------------------------------------------------
        {
          clientSend.println("ok");
          clientSend.flush();
```

```
            nextLine=clientReceive.readLine();
            String filename=nextLine;
            nextLine=clientReceive.readLine();
            int lang=ut1.StrInRad(nextLine.trim(),10);
            b=new byte[lang];
            clientReceive.readFully(b,0,lang);
            try{
                FileOutputStream f1=new FileOutputStream(path+filename);
                f1.write(b,0,lang);
                f1.close();
              }
            catch(Exception e){;}
          }
    //-----------------------------------------------------------------------
       if(nextLine.equals("loschen"))
    //-----------------------------------------------------------------------
          {
            clientSend.println("ok");
            clientSend.flush();
            nextLine=(clientReceive.readLine()).trim();
            System.out.println("1nextLine="+nextLine);
            File fil=new File(path+nextLine);
            fil.delete();
          }
    //-----------------------------------------------------------------------
    //-----------------------------------------------------------------------
        }
         clientSend.println("+BYE");
         clientSend.flush();
      }
     catch(IOException excpt)
        { System.err.println("2_Failed I/O: "+excpt); }

     finally
        {
         try
           {if(clientSend!=null) clientSend.close();
            if(clientReceive!=null) clientReceive.close();
            if(mySocket!=null) mySocket.close();}
         catch(IOException excpt)
          {  System.err.println("3_Failed I/O: "+excpt);}
        }
      }
//-----------------------------------------------------------------------
  protected String getQuote(int i,int j)
   {
    return stockInfo[i][j];
   }
//-----------------------------------------------------------------------
}
```

JAVA-Klasse „Client“

```
import java.awt.*;
import java.io.*;
import java.net.*;

public class dicomclient
{
//-----------------------------------------------------------------------
    private static int SERVER_PORT=1704;
    private static String serverName;

    private Socket quoteSocket=null;
```

```
   private DataInputStream quoteReceive=null;
   private PrintStream quoteSend=null;
   private String[] stockIDs;
   private String[] stockInfo;
   byte b[];
   Private String currentAsOf=null;
   utilit ut=new utilit();
   int k=1;
   String data[][];
   int kcon=1;
//-----------------------------------------------------------------------
 public static void main(String[] args)
  {
   serverName=args[0];
   System.out.println("serverName="+serverName);
   dicomclient client=new dicomclient(serverName,1024,"schickennachdata","c:\\dicombilder\\mr0003.d","mr0003.d");
   client.printQuotes(System.out);
   try{System.in.read();}
   catch(Exception e){;}
   System.exit(0);
  }
//-----------------------------------------------------------------------
 public dicomclient(String serverName1,int port,String befehl,String pathname,String filename)
  {
   if(port!=1024) SERVER_PORT=port;
   String serverInfo;
   serverName=serverName1;
   serverInfo=contactServer();
   if(serverInfo!=null)
     {
      currentAsOf=serverInfo.substring(serverInfo.indexOf(" ")+1);
      System.out.println("serverInfo="+serverInfo);
     }
   getQuotes(befehl,pathname,filename);
   quitServer();
  }
//-----------------------------------------------------------------------

 protected String contactServer()
  {
   String serverWelcome=null;
   try {
       quoteSocket=new Socket(serverName,SERVER_PORT);
        quoteReceive=new DataInputStream(quoteSocket.getInputStream());
        quoteSend=new PrintStream(quoteSocket.getOutputStream());
        serverWelcome=quoteReceive.readLine();
       }
   catch (UnknownHostException excpt)
       {
        System.err.println("Unknown host "+serverName+" : "+excpt);
        kcon=0;
        }
   catch (IOException excpt)
       {
        System.err.println("1Failed I/O to "+serverName+" : "+excpt);
        }
   return serverWelcome;
  }
//-----------------------------------------------------------------------

 protected void getQuotes(String num,String pathname,String filename)
  {
   String response;
   if(connectOK())
    {
```

```
try
  {
 //-------------------------------------------------------------
    if(num.equals("lesendata"))
  //-------------------------------------------------------------
    {
      quoteSend.println(num);
      quoteSend.flush();

      response=quoteReceive.readLine();
      k=ut.StrInRad((response.trim()),10);
      System.out.println("k2="+k);
      data=new String[k][23];
      for(int i=0; i<k;i++)
      for(int j=0;j<23;j++)
      {
        response=quoteReceive.readLine();
        data[i][j]=response;
      }
    }

  //-------------------------------------------------------------
    if(num.equals("schreibendata"))
  //-------------------------------------------------------------
    {
     data=new String[2][23];
     data=ut.redstr(pathname+filename,23);
     k=ut.StrInRad(data[0][0].trim(),10);
     quoteSend.println(num);
     quoteSend.flush();

     response=quoteReceive.readLine();
     if(response.equals("ok"))
      {
        quoteSend.println(data[0][0]);
        for(int i=0;i<k;i++)
        for(int j=0;j<23;j++)
        {quoteSend.println(data[i][j]);}
        quoteSend.flush();
      }
    }
   //-------------------------------------------------------------
   if(num.equals("holenfile"))
  //-------------------------------------------------------------
    {
      quoteSend.println(num);
      quoteSend.flush();
      response=quoteReceive.readLine();

      if(response.equals("ok"))
      {
        quoteSend.println(filename);
        quoteSend.flush();
        String lang_s=quoteReceive.readLine();
        int lang=ut.StrInRad(lang_s.trim(),10);
        System.out.println("lang="+lang);
        b=new byte[lang+1];
        quoteReceive.readFully(b,0,lang);
        quoteSend.flush();
        try{
            FileOutputStream f1=new FileOutputStream(pathname);
            f1.write(b,0,lang);
            f1.close();
           }
        catch(Exception e){;}
```

```
            }
          }
        //-------------------------------------------------------------
          if(num.equals("schickenfile"))
        //-------------------------------------------------------------
          {
            quoteSend.println(num);
            quoteSend.flush();
            response=quoteReceive.readLine();
            if(response.equals("ok"))
            {
              quoteSend.println(filename);
              quoteSend.flush();
              File f1=new File(pathname);
              long faillang=f1.length();
              int lang=(int)faillang;
              b=new byte[lang];
              try{
                  FileInputStream f2=new FileInputStream(pathname);
                  f2.read(b,0,lang);
                  f2.close();
                 }
              catch(IOException e){;}
              String lang_s=""+lang;
              quoteSend.println(lang_s);
              quoteSend.flush();
              quoteSend.write(b,0,lang);
              quoteSend.flush();
            }
          }
        //-------------------------------------------------------------
         if(num.equals("loschen"))
         //-------------------------------------------------------------
          {
            quoteSend.println(num);
            quoteSend.flush();

            response=quoteReceive.readLine();
            if(response.equals("ok"))
            {
              quoteSend.println(filename);
              quoteSend.flush();
            }
          }
         //-------------------------------------------------------------
         //-------------------------------------------------------------
        }
       catch (IOException excpt)
        {
         System.err.println("Failed I/O to "+serverName+" : "+excpt);
        }
    }
  }
//-----------------------------------------------------------------------------

 protected String quitServer()
  {
    String serverBye=null;
    try
     {
      if(quoteSend!=null) quoteSend.close();
      if(quoteReceive!=null) quoteReceive.close();
      if(quoteSocket!=null) quoteSocket.close();
     }
    catch (IOException excpt)
```

```
      {
           System.err.println("Failed I/O to "+serverName+" : "+excpt);
      }
      return serverBye;
   }
//-----------------------------------------------------------------------------

 public void printQuotes(PrintStream sendOutput)
   {
     if(currentAsOf!=null)
       {
         sendOutput.print("INFORMATION ON ....."+"\n\tCurrent As Of: "+currentAsOf+"\n\n");
       }
   }
//-----------------------------------------------------------------------------

 protected boolean connectOK()
   {
     return(quoteSend!=null&& quoteReceive!=null&&quoteSocket!=null);
   }
//-----------------------------------------------------------------------------
}
```

8. Literatur

[1] http://www.ge.com/medical/xray/msxadvwk.htm

[2] http://www.ge.com/medical/iis/msiisdaw.htm

[3] http://camrd4.mc.duke.edu/project%20pages/AWT%20project

[4] http://www.ivs-solutions.de/

[5] http://www.innomed.de/produktkatalog/produktbeschr_ris.htm#Produktbeschreibung MEDORA-PACS

[6] http://www.medinf.mu-luebeck.de/kamedin.html

[7] Handels, H., Busch, C., Encarnacao, J., Hahn, C., Kühn, V., Miehe, J.; Pöppl, S.J., Rinast, E., Roßmanith, C., Seibert, F., Will, A. (1997) KAMEDIN: A Telemedicine System for Computer Supported Cooperative Work and Remote Image Analysis in Radiology Computer Methods and Programs in Biomedicine; 52, 3, 175-183.

[8] Rinast, E., Grande-Nagel, I., Handels, H., Hahn, C., Roßmanith, C., Pöppl, S.J., Weiss, H.-D. (1997) Interaktive Teleradiologie auf ISDN - Das KAMEDIN-System FOCUS MUL,14, 2, 100-103.

[9] http://www.siemens.at/med/index.htm

[10] http://www.expasy.ch/www/UIN/html1/projects/osiris/osiris.html

[11] Ligier Y., Ratib O., Logean M., Girard C.: OSIRIS : A MEDICAL IMAGE MANIPULATION SYSTEM, M.D. Computing Journal, July-August 94, Vol 11 No 4, pp 212-218.

[12] Ligier Y., Ratib O., Girard C., Logean M.: EVOLUTION OF THE OSIRIS IMAGING SOFTWARE, Proceedings of EuroPACS 94, Geneva, 22-24 Sept 94.

[13] http://www.roe.med.tu-muenchen.de/pacs/katalog3.html

[14] http://mbi.dkfz-heidelberg.de/mbi/medicus/doxx/

[15] Engelmann U, Schröter A, Baur U, Schroeder A, Werner O, Wolsiffer K, Baur HJ, Göransson B, Borälv E, HP: TELERADIOLOGY SYSTEM MEDICUS. In: Lemke (Ed). CAR `96: Computer Assisted Radiology, Computer Assisted Radiology, 10th International Symposium and Exhibition, Paris, 26-29 June 1996.

[16] Baur HJ, Saurbier F, Engelmann U, Schröter A, Baur U, Meinzer HP. Aspects of Data Security and Privacy in Teleradiology. In: Lemke HU (Ed): CAR `96:

Computer Assisted Radiology, 10th International Symposium and Exhibition, Paris, 26-29 June 1996

[17] http://mbi.dkfz-heidelberg.de/mbi/chili/

[18] U.Engelmann, A.Schröter, U.Bauer, H.Müller, O.Werner, M.Schwab, H.P.Meinzer, Design of the next Generation of Teleradiology, EuroPACS 97 Proceedings, Pisa: Tipografia Editrice Pisana (1997) 121-124.

[19] Engelmann U, Schröter A, Evers H, Gundel S, Schwab M, Meinzer HP. CHILI: Eine Integrationsplattform für Bildverarbeitungsmethoden. In: Evers H, Glombitza G, Lehmann T, Meinzer HP (Eds). Informatik Aktuell - Bildverarbeitung für die Medizin 1999 - Algorithmen, Systeme, Anwendungen. Berlin, Heidelberg, New York: Springer (1999), 248-252.

[20] http://www.imse.med.tu-muenchen.de/rtb/3_10_5/

[21] Findling A, Horsch A, Zink A: An Open Distributed Medical Image and Signal Data Server Network with World Wide Web Front-End. IEEE Infocom '97 (The Conference on Computer Communications), Kobe, 1997

[22] http://www.derma.med.uni-erlangen.de/bilddb/Index_d.htm

[23] http://www.hipax.de/home_d.htm

[24] http://expasy.hcuge.ch/UIN/html1/projects/pacs/pacs.html

[25] Ratib O., Ligier Y., Girard C., Logean M., and al.: DESIGN AND IMPLEMENTATION OF THE GENEVA PACS : A PROGRESS REPORT, Proceedings of EuroPACS 94, Geneva, 22-24 Sept 94.

[26] S. Hludov, Chr. Meinel.: PACS FOR RADIOLOGY: The international conference „INNOVATIONS-98“ , Proceedings, Moscow, 1998. pp 50-53

[27] A.Heuer, S.Hludov, Ch.Meinel.: JAVA-based PACS. Computer Assisted Radiology and Surgery.13th International Congress on CAR, Paris, June, 1999, S. 516-521.

[28] Sergej Hludov, Lutz Köhler, Christoph Meinel, Thomas Engel.: PACS für Radiologie Preprint 98-08, Institut für Telematik

[29] http://www.xray.hmc.psu.edu/dicom/dicom_home.html

[30] NEMA Standart Publication PS3.X(1994): Digital Image and Communications in Medicine, Parts 1-10

[31] A. Heuer: Objektorientierte Datenbanken. Addison-Wesley, Bonn, 1992.

[32] Kudlich: Verteilte Datenbanken, Siemens-Nixdorf, 1993

[33] Meier: Relationale Datenbanken, Springer, 1994

[34] E. Schicker: Datenbanken und SQL: eine praxisorientierte Einführung. – Stuttgart: Teubner, 1996

[35] H.-C. Klaiber, H. Handels, S.J. Pöppl.: Bilddatenaustausch und radiologische Telekooperation auf der Basis von Java. 3. Workshop „Bildverarbeitung für die Medizin 1999“, Springer-Verlag Berlin Heidelberg, 1999, S. 226-230

[36] JAVA: http://www.java.sun.com/

[37] Jerry R. Jackson / Alan L. McClellan: JAVA an Beispiel: Sprach- und Applet-Struktur.- Hannover: Heise, 1996, S. 335

[38] S. Middendorf, R. Singer, S. Strobel: Java: Programmierhandbuch und Referenz. – 1. Aufl.- Heidelberg: dpunkt, Verl. Für digitale Technologie, 1996, S. 811

[39] Sun, JDBC Homepage.- http://www.java.sun.com/products/jdbc/

[40] G. Reese: Database Programming with JDBC and JAVA. O’Reilly & Associates, 1997, p. 234

[41] V. Turau: Techniken zur Realisierung Web-basierter Anwendungen. Informatik-Spektrum 22: Springer-Verlag 1999, S. 3-12

[42] I.D. Agranat, Engineering Web Technologies for Embedded Aplikations, IEEE Internet Computing, Vol.2, Nr. 3, 1998, S. 40-45

[43] D. Raggett, A.Le Hors und I. Jacobs, HTML 4.0 Specification, W3C Empfehlung, 1997

[44] Sun, Servlet Homepage, http://www.java.sun.com/products/servlet/

[45] W. Kaiser, J. King: Fernsteuerung durch Java-Servlets. Java-Spektrum 6: Springer-Verlag 1997, S. 16-20

[46] Servlet API, Dokumentation und Whitepaper: http://www.javasoft.com/ products/servlet/

[47] S.Hludov, Ch.Meinel, F.Warda, G.Noelle.: PACS for Teleradiology. 12 th IEEE Symposium on Computer-Based Medical System. Stamford, Connecticut. June, 1999, S.6-11.

[48] http://www.medos.de/

[49] F.Unglauben, R.Indefrey, W.Hillen: Multimediale Darstellung und Verarbeitung medizinischer Bilddaten in Rechnernetzen. 2. Workshop „Bildverarbeitung in der

Medizin“ Springer-Verlag Berlin Heidelberg, 1998, (http://www.juelich.fh-aachen.de/fachbereiche/ physik/medinfo/Java.html)

[50] http://www.aplusg.de/pegasos.htm

[51] Bahner ML, Engelmann U. Meinzer HP, van Kaick G. Anforderungen an ein Teleradiologiesystem - Erfahrungen aus dem MEDICUS-2 Feldtest. Radiologe 37, 269-277, 1997.

[52] R. Schaarschmidt, K. Bühnert, A. Herbst, K. Küspert, R. Schindler, Konzepte und Implementierungsaspekte anwendungsorientierten Archivierens in Datenbanksystemen, Informatik Forsch. Entw., 13: 79-89, Springer Verlag, 1998.

[53] J. Bernarding, A. Thiel, J. Braun, Ch. Koennecke, T. Schaaf, J. Hohmann, G. Bellaire, K.-J. Wolf, T. Tolxdorff.: Augbau einer JAVA/DICOM-basierten Hirninfarkt-Bilddatenbank mit integriertem Datenschutz. 3. Workshop „Bildverarbeitung für die Medizin 1999“, Springer-Verlag Berlin Heidelberg, 1999, S. 342-346

[54] J.Gosling, B. Joy, und G. Steele, The Java Language specification, Addison-Wesley, Reading Mass., 1996.

[55] J.-R. Ohm: Digitale Bildcodierung: Repräsentation, Kompression und Übertagung von Bildsignalen. – Springer-Verlag, 1995, S. 488.

[56] Wong S., Zaremba L., Gooden D., Huang. H.K.: Radiologic Image Compression. Proceedings of the IEEE, Vol. 83, NO. 2, February 1995.

[57] http://www.intercap.com/cgminfo/Papers/vector.htm

[58] http://www.intercap.com/cgminfo/cgm.htm

[59] http://www.why-not.com/articles/formats.htm

[60] http://documents.cfar.umd.edu/imageproc/lzw.and.gif.doc

[61] Gregory K. Wallace.: The JPEG Still Picture Compression Standart.: IEEE Transactions on Consumer Electronics, 1991

[62] W.B. Pennebaker and J. L. Mitchell, JPEG Still Image Data Compression Standard. New York: Van Nostrand Reinhold, 1993.

[63] http://www.students.cs.uu.nl/people/eheukels/BMP.html

[64] S. Hludov, Chr Meinel.: Adaptive LZW-JPEG Kompression radiologischer Bilder. 3. Workshop „Bildverarbeitung für die Medizin 1999“, Springer-Verlag Berlin Heidelberg, 1999, S. 297-301

[65] S. Hludov, C. Schröter, Chr. Meinel.:Adaptive Compression of Image Data. SYBEN'98 Broadband Networks, Zürich,1998 SPIE Proceedings Vol. 3408, pp 520-527

[66] S. Hludov, T. Engel, Chr. Meinel.: Adaptive Compression of DICOM-Image Data. International Symposium on Electronic Image Capture and Publishing, Zürich,1998, SPIE Proceedings Vol. 3409, pp 260-266.

[67] H. Baierl, A. Klappenecker: Strukturadaptierte Praediktion in der verlustfreien Kompression medizinischer Bilddaten. 3. Workshop „Bildverarbeitung für die Medizin 1999“, Springer-Verlag Berlin Heidelberg, 1999, S. 211-216

[68] http://www.brunel.ac.uk/~dc95anm/home.html

[69] http://www.rasip.fer.hr/research/compress/algorithms/adv/fakscomp/index4.html

[70] http://www.cs.cmu.edu/~guyb/real-world/compress/lzexplained.html

[71] http://www.amara.com/current/wavelet.html

[72] Meyer Y.: Wavelets and Applications. Springer Verlag-Berlin. 1992.

[73] L.P. Panych. Theoretical comparison of Fourier and Wavelet encoding in Magnetic Resonance Imaging. IEEE Trans. On Medical Imaging, 15(2): 141-153, 1997

[74] I. Urriza, L.A. Barragan, J.I. Artigas, J.I. Garcia and D. Navarro. Choice of word length in the design of a specialized hardware for lossless wavelet compression of medical images. Optical Engineering, 36(11):3033-3042, 1997

[75] Antonine M., Barland M.: Digital image compression using vector qantization and the wavelet transform. in Meyer Y.: Wavelets and Applications. s. 160-174. 1992.

[76] J. Wang, H.K. Huang. Medical image compression by using three-dimensional wavelet transformation. IEEE Transactions on Medical Imaging, 15(4), 1996

[77] S. Hludov.: Wavelet transformation and image compression. Avtometriya, Novosibirsk, 2, 1997, p. 98-106.

[78] C.F.Barnes and R.L. Frost. Residual Vector Quantizers with Jointly Optimized Code Books. In Advances in Electronics and Electron Physics,pages 1-59. Amsterdam: Elsevier, 1992.

[79] http://www.cis.ohio-state.edu/hypertext/faq/usenet/compression-faq/part2/faq-doc-7.html

[80] C.F.Barnes and R.L. Frost. Vector Quantizers with Direct Sum Codebooks. IEEE Transactions on Information Theory, 39:565-580,March 1993.

[81] S. Mitra and S. Yang. High Fidelity Adaptive Vector Quantization at Very Low Bit Rates for Progressive Transmission of Radiographic Images, Journal of Electronic Imaging, Vol.8, No. 1, 1999, pp. 23-35.

[82] X. Ginesta and S.P. Kim. Vector Quantization of Contextual Information for Lossless Image Compression. In Proceedings Data Compression Conference, pages 390-399. Los Alamitos, CA: IEEE Computer Society Press, 1994.

[83] S. Panchanathan and M. Goldberg. Adaptive Algorithm for Image Coding Using Vector Quantization. Signal Processing: Image Communication, 4:81-92, 1991.

[84] A. Klappenecker, F.U. May, T. Beth: Filterbankstrukturen zur verlustfreien Kompression medizinischer Bilddaten. 2. Workshop „Bildverarbeitung in der Medizin“ Springer-Verlag Berlin Heidelberg, 1998 (http://www.klinikum.rwth-aachen.de/webpages/mib/MBV/ bvm98/ abstracts/071.html)

[85] R.B. Arps and T.K. Truong. Comparison of International Standards for Lossless Still Image Compression. Proceedings of the IEEE, 82:889-899,June 1994.

[86] Y. Fisher, editor. Fractal Image Compression: Theory and Application. New York: Springer Verlag, 1995.

[87] http://www.cis.ohio-state.edu/hypertext/faq/usenet/compression-faq/part2/faq-doc-8.html

[88] Michael Barnsley, Lyman Hurd. Fractal Image Compression. AK Peters Ltd, 1993.

[89] US. Patent # 5347600, 1994.

[90] US. Patent # 4941193, 1990.

[91] US. Patent # 5065447, 1991.

[92] K. Sayood. Introduction to Data Compression. M. Kaufmann Publishers, Inc. 340 Pine Street, Sixth Floor San Francisco, CA. USA. 1996.

[93] Lohscheller,H A subjectively adapted image communication system. IEEE Trans. Comman. Com-32 (Dec. 1984), pp.1316-1322.

[94] S. Hludov, Ch. Meinel, Th. Engel.: Flächen- und Volumenmessung lokaler Objekte in DICOM-Bildern und –Bildfolgen. 3. Workshop „Bildverarbeitung für die Medizin 1999“, Springer-Verlag Berlin Heidelberg, 1999, S. 292-296

[95] S.Hludov, Ch.Meinel.: DICOM-Image Compression. 12 th IEEE Symposium on Computer-Based Medical System. Stamford, Connecticut. June, 1999,S 282-287.

[96] E. Ridley: Universities collaborate on de facto standard for lossy compression. Telemedicine and Telehealth Networks, Vol. 3, NO. 6, 1997.

[97] A. Bruckmann, A. Uhl: Ein Vergleich von Wavelet und JPEG basierten selektiven Methoden im Bereich der medizinischen Bildkompression. 3. Workshop „Bildverarbeitung für die Medizin 1999“, Springer-Verlag Berlin Heidelberg, 1999, S. 216-220

[98] A. Bruckmann, A. Uhl: Selective medical image compression using wavelet techniques. Journal of Computing and Information Technology, 2(6): 203-213, 1998

[99] A. Becker, L. Grünwoldt, Ch. Meinel. TeleMedizin - Eine Übersicht, Institut für Telematik, 1996

[100] Ch. Meinel. Wie das Internet funktioniert. Preprint 97-01, Institut für Telematik, 1997

[101] Ch. Meinel, S. Müller. Trierer Telemedizin Symposium. Proceedings, Preprint 98-11, Institut für Telematik, 1998

[102] A. Abrardo, A.L. Casini: Embedded Java in a web-based teleradiology system. IEEE Internet Computing, 1998: p. 60-68

[103] R. Orfali, D. Harkey, Client/server Programming with JAVA and CORBA, Wiley, 1998

[104] P. Sridharan. Advanced Java networking. Prentice HALL Upper Saddle River, ISBN 0-13-749136-0, 1997, p 368

[105] T. Hübenthal. Bestellsystem als Java-Applet mit JDBC-Zugriff.- JavaSprktrum, Juli/August 1998.

[106] S. Hludov, Ch. Meinel.: JAVA-DICOM-Viewer. Preprint 98-05, Institut für Telematik

[107] L. Grünwoldt, S. Hludov, Ch. Meinel, F. Thiesse.: DICOM-Datenbank und -Viewer. Innovatives Bildmanagement in der Medizin. Preprint 97-05, Institut für Telematik

[108] S. Hludov, T. Engel, Chr. Meinel.: JAVA-basierte DICOM-Viewer. Telemedizinführer Deutschland - Ausgabe 2000 (ISBN 3-0000-4589-9) , 254-258 S. 1999.

[109] S. Hludov, L. Vorwerk, Ch. Meinel.: Intranet/Internet-basierte PACS. Telemedizin-führer Deutschland - Ausgabe 2000 (ISBN 3-0000-4589-9) , 250-253 S. 1999.

[110] L. Vorwerk, S. Khludov, Ch. Meinel.: Concept for Increased Security for Internet/Intranet-Based Administration of Patient Data. 8 th Iternational Conference on Computer Craphics, Visualization and Interactive Digital Media' 2000 (WSCC' 2000). Plzen, Czech Republic, February 2000, pp 124-129.

[111] S. Khludov, L. Vorwerk, Ch. Meinel.: Internet-orientated medical information system for DICOM-Data tranfer, visualization and revision. 13 th IEEE Symposium on Computer-Based Medical System. CBMS 2000, Houston, Texas, June 2000