Verteilte Bearbeitung inhaltsbasierter Suchanfragen auf ein medizinisches Bildarchiv

Mark O. Güld[1], Alexander Craemer[1], Bartosz Plodowski[1], Christian Thies[1], Benedikt Fischer[1], Michael Kohnen[2] und Thomas M. Lehmann[1]

[1]Institut für Medizinische Informatik, Pauwelsstraße 30, 52057 Aachen
[2]Klinik für Radiologische Diagnostik, Pauwelsstraße 30, 52057 Aachen
Email: mgueld@mi.rwth-aachen.de

Zusammenfassung. Es wird ein verteiltes System vorgestellt, das die graphische Programmierung von Methoden für Bildverarbeitung, Merkmalsextraktion und Merkmalsvergleich mit Automatismen für transparente Merkmalsspeicherung, Ablaufkontrolle (Scheduling) und Wartung in einem heterogenen Netzwerkverbund kombiniert. Die Modellierung wird hierbei um Konzepte zur Verarbeitung von Merkmalsmengen erweitert. Das System unterstützt sowohl die Entwicklung wie auch die spätere Anwendung von beliebigen Algorithmen zum inhaltsbasierten Bildzugriff, z.B. im Rahmen von Image Retrieval in Medical Applications (IRMA*).

1 Einleitung

Für eine präzise Antwort auf Anfragen an medizinische Bildarchive zur Diagnoseunterstützung ist ein inhaltsbasierter Ansatz, engl. content-based image retrieval (CBIR), notwendig. Dabei wird ein Bild durch automatisch extrahierte Merkmale beschrieben, die zur Identifikation und zum Vergleich der Bilder benutzt werden. Insbesondere medizinische Bilder tragen für den Anfragekontext relevante Informationen in lokalen Details und stellen somit hohe Anforderungen an die benötigte Inhaltsrepräsentation. Ferner muss die stetige Evolution medizinischen Wissens flexibel abbildbar sein [1]. Die Merkmalsextraktion, -speicherung sowie die Bearbeitung von Anfragen auf dieser Datenstruktur erfordern dabei in großem Umfang Speicherplatz und Rechenzeit, die durch eine geeignete Infrastruktur verfügbar gemacht werden müssen.

2 Stand der Technik

Existierende CBIR-Systeme greifen meist auf ein einziges Verfahren zur Bildinhaltsbeschreibung zurück, wobei allenfalls die Vergleichsoperation durch den

* Das IRMA-Projekt wird gefördert von der Deutschen Forschungsgemeinschaft DFG (Le 1108/4)

Anwender zur Anfragezeit parametriert werden kann. Nichtmedizinische CBIR-Systeme konzentrieren sich ferner auf große Mengen von Farbbildern, deren Inhalt durch sehr wenige globale Merkmale, z.B. Histogramminformationen bzgl. Farbe, Textur und Form, beschrieben werden. Die Merkmalsextraktion findet für die vorhandenen Bilder offline statt, was eine sehr schnelle Anfragebearbeitung ermöglicht.

Die Implementierung der Merkmalsextraktion eines CBIR-Systems kann durch Werkzeuge aus dem Bildverarbeitungsbereich unterstützt werden. Bei diesen Werkzeugen sind graphische Modellierungen von Algorithmen als Datenfluss-Graphen stark verbreitet, was neben interdisziplinärer Kommunizierbarkeit Vorteile bei Implementierung und Abwicklung derartig formulierter Algorithmen birgt, z.B. hinsichtlich Trennung von Benutzerschnittstelle und Funktionalität, Modularisierung, Datenabhängigkeiten und Parallelisierbarkeit [2]. Das weit verbreitete, kommerziell verfügbare Werkzeug Khoros [3] arbeitet mit halbautomatischer Verteilung der anfallenden Berechnungen. Des weiteren können vorgeschlagene Datenformate (z.B. KDF oder Vista) Anwendungen der Bildverarbeitung flexibel und gleichzeitig uniform in der Schnittstelle für den Programmierer unterstützen. Diese Entwicklungswerkzeuge unterstützen andererseits nicht die Verwaltung extrahierter Merkmale.

3 Methoden

3.1 Zentrale Datenbank

Alle Entitäten des Systems (Modellierung, Merkmale, Infrastruktur) werden über eine zentrale relationale Datenbank verwaltet. Für größere Datenelemente, z.B. lokale (d.h. pro-Pixel-)Merkmale oder die Bilder selbst, sind Transportfunktionen implementiert, die verteilte Datenspeicherung mit ortstransparentem Zugriff anbieten.

3.2 Modellierung von Algorithmen

Zur Modellierung von Algorithmen werden gerichtete Graphen verwendet, deren Kanten den Datenfluss von Merkmalen zwischen Knoten darstellen. Die Knoten beschreiben benutzerimplementierte Transformationen (sog. Methoden) bzw. Steuerelemente für bedingte Ausführung, Schleifen und Mengenmanipulation. Ein Merkmal ist ein Container, in dem Tupel über elementaren Datentypen Ganzzahl, Fließkommazahl, Text, ID, Bild und hierarchische Bildinhaltsbeschreibung zusammen mit einer benutzerdefinierten Typinformation abgelegt werden. Quellen und Senken eines Netzwerks stellen die Schnittstelle des Algorithmus zum Anwender dar. Graphisch werden diese mit Hilfe eines modularen Werkzeugs in PHP implementiert, was webbasiertes Arbeiten erlaubt. Im Gegensatz zu anderen visuellen Datenfluss-Programmierwerkzeugen unterstützt das System die Verarbeitung von Merkmalsmengen, was gerade in den Bereichen Kategorisierung, Prototypbildung, statistische Datenanalyse und im finalen

Vergleichsschritt des Anfragemerkmals mit allen Referenzmerkmalen notwendig ist. Dementsprechend existieren drei elementare Methodentypen: Neben der für ein Prozessnetzwerk typischen 1:1-Transformation eines Merkmalstupels in ein Merkmalstupel existieren T:1-Methoden, die nach der Verarbeitung einer Menge von Merkmalstupeln ein neues Tupel generieren, wobei T die zeitlich sequentielle Verarbeitung der Eingabetupel symbolisiert. Das entsprechende Analogon bilden Methoden vom Typ 1:T.

3.3 Ausführung von Algorithmen

Ein im Zuge der Bearbeitung eines Netzwerks notwendiger Aufruf einer Methode mit Eingabemerkmalen wird als Job bezeichnet. Die Bearbeitung eines Jobs erfolgt durch die Schnittstelle der Methode zur Plattform orts- und nebenläufigkeitstransparent, wobei die Abwicklung durch einen zentralen Scheduler übernommen wird. Dieser kommuniziert mittels eines über http abgewickelten Protokolls mit Job-Daemons. Ein solcher Job-Daemon kontrolliert hierbei alle Berechnungen des Rechners, auf dem er läuft. Der Scheduler macht sich aus dem Graphen ableitbare Abhängigkeiten von Merkmalen und damit Möglichkeiten zur Parallelisierung zunutze und verteilt anfallende Jobs auf den bzgl. Hardwareausstattung und Auslastung bestpassenden Rechner im Verbund. Der empfangende Job-Daemon kann vor dem Start der zum Job gehörenden Methode deren Aktualität überprüfen und installiert ggf. über das concurrent versioning system (CVS) aus dem Quelltext eine neue Version. Die Bereitstellung benötigter Eingabemerkmale sowie die Speicherung der Ausgabemerkmale erfolgt transparent durch das System. Zu erzeugten Merkmalen wird ferner die Information über deren Generierung (Methode, Eingabemerkmale) gespeichert, so dass bei späterem Auftreten dieses Jobs auch in anderem Kontext auf das existierende Merkmal zurückgegriffen werden kann.

4 Ergebnisse

Das System modelliert alle im Rahmen eines Retrievals mit hoher Bildinhaltsabstraktion anfallenden Schritte [4] uniform. Entwickler nutzen die Plattform als wachsenden Werkzeugkasten mit wiederverwendbaren Komponenten zur Bildverarbeitung, Merkmalsextraktion und zum Mustervergleich, wobei die einheitliche Schnittstelle zur transparenten Merkmalsübergabe und das Typkonzept für Merkmale die Entwicklung in Gruppen effektiv unterstützt. Das System ist mit einer webbasierten Schnittstelle für das Retrieval gekoppelt. Deren modulares Rahmenwerk [5] erlaubt die merkmalstypbezogene Wiederverwendung von Komponenten für auftretende Eingabemerkmale (z.B. Bilder, numerische Parameter) und Ausgabemerkmale (z.B. das Suchergebnis als bewertete Bildliste). Auf der Anwenderseite ist die Rechenleistung und der Massenspeicher des Netzwerkverbundes transparent verfügbar, wobei der damit verbundene Wartungsaufwand durch automatische Aktualisierung benutzerimplementierter Komponenten adressiert wird. Die Plattform benutzt einen heterogenen Verbund aus

Workstations unter Intel/Linux und Sparc/Solaris. Die Bearbeitung von Anfragen (Scheduling des Netzwerks) findet transparent statt und generierte Merkmale bleiben zum etwaigen späteren Rückgriff verfügbar. Insbesondere kann zur Anfragezeit auf bereits berechnete, da vom Anfragebild unabhängige, Merkmale für die Referenzen zurückgegriffen werden. Das System ist im laufenden Betrieb durch das Einspielen neuer Merkmale, Methoden und Netzwerke, sowie das Hinzufügen von Hardware erweiterbar.

5 Diskussion

Der durch die Merkmalsspeicherung und die Kosten für Verteilung entstehende Overhead muss detailliert dem Gewinn durch die Ausnutzung von Parallelitäten im Ablauf und den Rückgriff auf bereits berechnete Merkmale gegenübergestellt werden. Bislang beschränken sich die Interaktionen in implementierten Anwendungen auf Anfragestart und Anfrageende, was der Belegung von Netzwerkquellen und dem Auslesen und Anzeigen von Netzwerksenken entspricht. Komplexere Anfragen, die z.B. die Markierung einer Bildregion (engl. region of interest, ROI) auf einem vorverarbeiteten Anfragebild umfassen, erfordern ein Konzept für GUI-Knoten, die als anwendergesteuerte, nichtdeterministische Merkmalstransformation aufgefasst werden können. Ein weiteres Zeil ist es, das Konfigurationsmanagement des Netzwerkverbunds zu verfeinern, insbesondere da nicht alle benötigten externen Bibliotheken und damit alle Details des automatischen Übersetzungs- und Installationsvorgangs für Methoden initial bekannt sind. Dementsprechend ist die Modellierung der Clusterarchitektur zu erweitern und die Entwicklungsumgebung anzupassen.

Literaturverzeichnis

1. Tagare HD, Jaffe CC, Dungan J: Medical Image Databases: A Content-Based Retrieval Approach. Journal of the American Medical Informatics Association 4(3): 184–198, 1997.
2. Koelma D, Smeulders A: A visual programming interface for an image processing environment. Pattern Recognition Letters 15(11): 1099–1109, 1994.
3. Argiro D, Farrarr K, Kubica S: Cantata: The Visual Programming Environment for the Khoros System. Visualization, Imaging, and Image Processing. Proceedings of the IASTED International Conference 697–702, 2001.
4. Lehmann TM, Wein BB, Dahmen J, Bredno J, Vogelsang F, Kohnen M: Content-Based Image Retrieval in Medical Applications – A Novel Multi-Step Approach. Procs SPIE 3972: 312–320, 2000.
5. Plodowski B, Güld MO, Schubert H, Keysers D, Lehmann TM: Modulares Design von webbasierten Benutzerschnittstellen für inhaltsbasierte Zugriffe auf medizinische Bilddaten. Procs BVM 03: 383–387, 2003.