

Effektive Implementierung von Algorithmen zum inhaltsbasierten Bildzugriff auf medizinische Bilddaten

Mark O. Güld, Daniel Schneider, Raoul Moritz, Alexander Craemer,
Christian Thies, Benedikt Fischer und Thomas M. Lehmann

Institut für Medizinische Informatik, RWTH Aachen, 52057 Aachen
Email: mgueld@mi.rwth-aachen.de

Zusammenfassung. Dieser Beitrag präsentiert Lösungsvorschläge zur effektiven Unterstützung bei der Implementierung und Bereitstellung von Verfahren zum inhaltsbasierten Zugriff auf Bilddatenbanken. Hierzu ist eine strikte Trennung zwischen Merkmalsextraktion, Merkmalsspeicherung, Merkmalsvergleich und den Benutzerschnittstellen sinnvoll. Dadurch wird es möglich, existierende Komponenten in verschiedenen Kontexten wiederzuverwenden, was Softwarequalität, Entwicklungszeit und einheitlichen Richtlinien bei den Benutzerschnittstellen zugute kommt.

1 Einleitung

Durch die zunehmende Verbreitung digitaler Röntgenarchive werden inhaltsbasierte Zugriffsverfahren immer interessanter. Diese Verfahren basieren auf der automatischen Extraktion und dem Vergleich von inhaltsbeschreibenden Merkmalen zu den in der Datenbank liegenden Bildern.

Momentan besteht eine Lücke zwischen zumeist monolithischen Implementierungen von Bilddatenbanksystemen auf der einen Seite und Werkzeugen zur Implementierung von Algorithmen der Bildverarbeitung auf der anderen. Bei existierenden Systemen zum inhaltsbasierten Bildzugriff sind der zugrundeliegende Algorithmus, die Merkmalsspeicherung und die Benutzerschnittstelle eng verflochten. Hierdurch sind insbesondere die in [1, 2] bzgl. variablem Anfragekontext und evolutionärem Inhaltsverständnis festgestellten Anforderungen an flexible Zugriffsverfahren für medizinische Bilder nur aufwendig umsetzbar, da neue Verfahren stets Änderungen an allen Systemkomponenten erfordern. Existierende Werkzeuge zur Implementierung von Bildverarbeitungsalgorithmen, wie sie z.B. bei der Merkmalsextraktion benötigt werden, bieten zwar eine große Anzahl vorgefertigter Algorithmen, ihnen fehlen aber Konzepte für die Organisation der Merkmalsspeicherung und einfache Bereitstellung der Verfahren für nicht-technische Endanwender. Zur effizienten Unterstützung sind eine plattformübergreifende Entwicklungsumgebung, die strikte Trennung von Algorithmen und Benutzeroberflächen, eine schnelle und unkomplizierte Verfügbarkeit neuer Verfahren sowie eine leistungsfähige Laufzeitumgebung notwendig.

2 Methoden

Soll ein neues Zugriffsverfahren integriert werden, sind durch den Entwickler Prozeduren zu Merkmalsextraktion und -vergleich bereitzustellen. Aufgabe des Systems ist es, ihn bei Entwicklung und Integration effektiv zu unterstützen.

Entwicklungsumgebung. Die Entwicklungsumgebung erlaubt eine Organisation von Verfahrensteilen in *Projekten*, verbirgt plattformabhängige Details des Übersetzungsprozesses und ermöglicht eine einfache Verwaltung von Abhängigkeiten. Verfügbare Drittsoftware wird architekturabhängig konfiguriert und unter symbolischen Namen verfügbar gemacht. Der Entwickler eines Projekts stellt die von ihm verwendeten internen Projekte sowie externe Bibliotheken über eine Konfigurationsdatei ein. Danach beschränkt sich sein Übersetzungsvorgang auf den Aufruf von GNU make, da das System Makefiles für Bibliotheksfunktionen, ausführbare Programme und Methoden zur Verfügung stellt, die die durch den Entwickler festgelegten Abhängigkeiten auswerten.

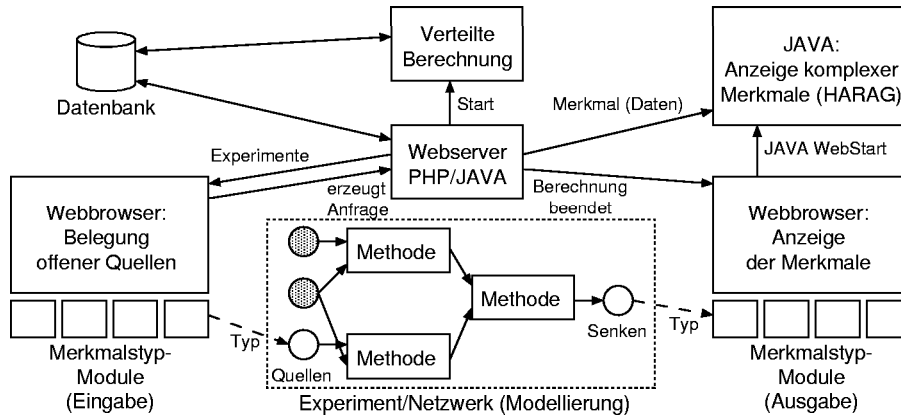
Verfahrensentitäten und deren Schnittstellen. Der Entwickler kann das System um verschiedene Entitäten erweitern: *Merkmalstypen*, *Methoden*, *Netzwerke*, d.h. Algorithmen, und *Experimente*, d.h. teilweise parametrisierte Netzwerke, aus denen durch das System *Anfragen* generiert werden können. Sämtliche Entitäten werden über die eXtensible Markup Language (XML) definiert und durch ein jeweils entsprechendes Dienstprogramm in das System eingespielt. Es werden typisierte Merkmale verwendet, d.h. jedes Merkmal trägt die Information, welche Semantik die in ihm enthaltenen Werte besitzen. Dementsprechend wird für ein neues bildinhaltsbeschreibendes Merkmal ein entsprechender Typ definiert, z.B. für ein Texturmaß:

```
<featuretype name="texture_tamura">
  A 3D histogram of texture features proposed by TAMURA:
  Directionality, contrast and coarseness.
  <isa> vector </isa>
</featuretype>
```

Das System erlaubt symbolische Vererbung, um Kompatibilität bzgl. Methoden auszudrücken, z.B. ist ein Texturmerkmal als Tupel von Fließkommawerten möglicherweise wie ein normaler Vektor behandelbar, wie er von diversen Distanzmaßen erwartet wird. Methoden kapseln implementierte Transformationen von Merkmalstupeln, auf die die Methode über ihre Ein- und Ausgänge zugreift. Der Entwickler definiert neben der Methodenimplementierung deren Merkmalschnittstelle, die festlegt, an welchem Ein- bzw. Ausgang welcher Merkmalstyp erwartet wird:

```
<method name="extract_tamura" type="1:1" project="texture_tamura">
  <input featuretype="image" keeps_ref="1"> Input image. </input>
  <output featuretype="texture_tamura"> Texture feature. </output>
  Extracts texture features proposed by TAMURA from an image.
</method>
```

Abb. 1. Interaktion der Benutzerschnittstelle (Frontend) mit dem Restsystem (Backend). In PHP erstellte Benutzerschnittstellen sind von den Verfahren entkoppelt und über Attribute der Verfahrensentitäten teilweise automatisch generierbar.



Danach sind Methoden in beliebigem Kontext wiederverwendbar. Einzelne Methoden werden innerhalb von Netzwerken zu Algorithmen verbunden, wobei die Typinformation der Merkmalschnittstelle für Validitätsprüfungen benutzt wird. Netzwerke drücken den Datenfluß zwischen Methoden und deren Abhängigkeiten aus. Wie eine Methode besitzt ein Netzwerk eine Merkmalschnittstelle, die durch die Quell- und Senkeknoten des Netzwerks gegeben ist. Das Netzwerk stellt ferner eine konsistente Parametrierung der Methoden sicher, z.B. gleiche Parameter für eine Extraktionsmethode, die sowohl auf Referenzen als auch das Testdatum angewendet wird. Experimente erlauben, Netzwerke teilweise zu parametrieren. Dies geschieht durch die Definition von Belegungen von Quellknoten mit geeigneten Merkmalen, z.B. einer zuvor empirisch ermittelten Parametrierung eines Extraktionsverfahrens.

Trennung zwischen Benutzerschnittstelle und Verfahren. Experimente bilden die Schnittstelle definierter Verfahren zum Endanwender, da sie Teile der Parametrierung verbergen und durch die noch zu belegenden Quellen die Parameter definieren, welche durch den Endanwender festzulegen sind (Abb. 1). Durch die Merkmalstypinformation der noch offenen Quellknoten und der Senkeknoten ist eine grafische Benutzerschnittstelle zu Algorithmen generierbar, ohne deren Details zu kennen. Hier genügen Module für die Ein- und Ausgabe verwendeter Merkmalstypen, die entsprechend der Experimentdefinition eingebunden werden. Durch die Auswahl bzw. Eingabe von Merkmalen erzeugt der Endanwender eine vollständige Belegung aller Quellknoten des Experiments. Diese wird als Anfrage bezeichnet und durch die grafische Benutzeroberfläche an das Backend zur Bearbeitung übertragen. Nach der Berechnung kann die Benutzerschnittstelle an den Senken auf resultierende Merkmale zugreifen und diese gemäß ihres Typs anzeigen.

Plattformunabhängige Benutzerschnittstellen. Sämtliche Schnittstellen des Systems zum Endanwender basieren auf JAVA-Applikationen bzw. über PHP generierten HTML-Dialogen, die über einen Webserver zur Verfügung gestellt werden. Somit ist ein Zugriff auf das System über das Internet möglich, der Anwender benötigt lediglich einen Webbrowser und eine JAVA Virtual Machine (VM). Die Ausführung von Anfragen kann durch ein PHP-Skript angestoßen werden und erfolgt für den Benutzer transparent über das Backend. Für die Anzeige und Verarbeitung von hierarchischen Regionenbeschreibungen (engl. Hierarchical attributed region-adjacency graph, HARAG), die aus Partitionierungsverfahren resultieren, sind in JAVA implementierte Oberflächenmodule verfügbar, die über JAVA WebStart in Webseiten integriert werden. Der Zugriff auf die als Merkmale abgelegten HARAGs erfolgt über das Netzwerkprotokoll HTTP. Zur schnellen Entwicklung von Benutzerschnittstellen in PHP wurde eine umfangreiche Klassenbibliothek implementiert, die Templates zur Trennung von Daten und deren Darstellung einsetzt, sowohl bei der Eingabe als auch bei der Ausgabe.

Laufzeitumgebung. Bei der Berechnung einer Anfrage werden die entsprechenden Methodenaufrufe automatisch durch das System generiert. Das System implementiert Merkmale als einen Container über elementaren Datentypen, dessen Inhalt transparent bei einem Aufruf zur Verfügung gestellt wird. Ebenso wird das resultierende Merkmalstupel eines Aufrufs automatisch durch das System auf einem Dateiserver bzw. in der Datenbank abgelegt. Dadurch sind Methodenaufrufe auf einzelne Rechner in einem Netzwerkverbund verteilbar [3]. Hierbei wurde zusätzlich großer Wert auf die Stabilität des Systems gelegt, sodaß fehlerhafte Methoden die Plattform nicht beeinträchtigen können. Die Datenbankorganisation erlaubt es, bereits berechnete Merkmale wiederzuerkennen und entsprechende Berechnungsschritte ggf. zu überspringen. Nach der Berechnung sind die resultierenden Merkmale einer Anfrage über die Senkeknoten verfügbar.

3 Ergebnisse

Das System wurde ausschließlich unter Verwendung offener Standards und freier Software implementiert und wird momentan unter Intel/Linux und Sparc/Solaris eingesetzt. Die Entwicklungszeiten neuer Zugriffsverfahren sind durch die Entwicklungsumgebung, die umfangreiche Softwarebibliothek und die einfache Anbindung der Algorithmen an den Bilddatenbestand deutlich verkürzt. Einfache Vorverarbeitungsschritte wie z.B. Bildfilter können innerhalb weniger Minuten implementiert und zur Integration in beliebige Verfahren verfügbar gemacht werden. Derzeit sind sechs Verfahren zum automatischen Vergleich bzw. Kategorisierung von Bildern anhand eines medizinischen Schlüssels implementiert. Die Entwicklungsumgebung leistet insbesondere einen konstruktiven Beitrag zur Pflege diverser Projekte unter hoher Personalfuktuation, z.B. im Rahmen von Studienarbeiten.

Die Verwendung von Merkmalstypen und Schnittstellendefinitionen von Methoden und Netzwerken ermöglichen die einheitliche Dokumentation, effiziente

Wiederverwendung von Verfahren und die Entkopplung der Implementierung von Verfahren (Backend) und deren Benutzerschnittstellen (Frontend).

Alle Schnittstellen des Endbenutzers sind PHP- bzw. JAVA-basiert, was einen transparenten Zugriff auf das Backend über das Internet ermöglicht. Hierdurch wird vermieden, daß ein nicht-technischer Anwender mit der Wartung und Konfiguration von Software konfrontiert ist. Da sämtliche Zugriffsverfahren momentan nach dem Prinzip der k nächsten Nachbarn bei Anfrage mit einem Beispielbild arbeiten, ist die webbasierte Benutzeroberfläche in der Schnittstelle zum System für alle Verfahren identisch und kann wiederverwendet werden. Das Konzept und die Umgebung für die Implementierung von Algorithmen im Backend sowie die Klassenbibliothek unter PHP sind weit fortgeschritten, während sich das Analogon auf der JAVA-Seite noch im Anfangsstadium befindet.

Die verteilte Bearbeitung von Algorithmen beschleunigt insbesondere die Merkmalsextraktion. Diese wird offline, d.h. nicht zur Anfragezeit sondern vorab, durchgeführt und tritt bei der Integration eines neuen Zugriffsverfahrens oder dem Einspielen neuer Bilder in das System auf. Die meisten Konzepte zur Beschreibung des Bildinhalts arbeiten isoliert auf jedem Bild, wodurch diese sehr gut verteilbar sind. In ersten Tests zeigte sich die Laufzeitumgebung hier lediglich durch die Bandbreite des zur Merkmalspeicherung verwendeten Dateiservers begrenzt und erzielte für 10 Stationen eine achtfache Beschleunigung.

4 Diskussion

Der verwendete allgemeine Ansatz zur Modellierung von Verfahren zum Bildzugriff erzeugt einen Overhead, sodaß die Effizienz eng gekoppelter monolithischer Systeme nicht ganz erreicht werden kann. Dies wird jedoch durch die Vorteile bzgl. Erweiterung, Wartung und Wiederverwendung ausgeglichen. Zukünftig ist eine Integration des Systems in bestehende Lösungen für die Radiologie geplant, was eine direkte Schnittstelle zu digitalen DICOM-Röntgenarchiven (Picture Archiving and Communications System, PACS) und deren Benutzerschnittstellen (z.B. das Clinical Applications Interface, CAI, der Firma Sectra) erfordert.

Danksagung. Diese Arbeit ist Teil von Image Retrieval in Medical Applications (IRMA) und wird von der Deutschen Forschungsgemeinschaft gefördert (DFG, Le 1108/4).

Literaturverzeichnis

1. Tagare HD, Jaffe CC, Dungan J. Medical Image Databases: A Content-Based Retrieval Approach. *JAMIA* 1997;4(3):184-98.
2. Müller H, Michoux N, Bandon D, Geissbuhler A. A Review of Content-Based Image Retrieval Systems in Medical Applications. *Clinical Benefits and Future Directions. Int Journal of Medical Informatics* 2004;73:1-23.
3. Güld MO, Craemer A, Plodowski B, Thies C, Fischer B, Lehmann TM. Verteilte Bearbeitung inhaltsbasierter Suchanfragen auf ein medizinisches Bildarchiv. In: *Procs. BVM; 2004. p. 347-50.*