Optimierung eines konnektionistischen Graphmatchers zum inhaltsbasierten Retrieval medizinischer Bilder

Christian Lappe[1], Benedikt Fischer[1], Christian Thies[1],
Mark-Oliver Güld[1], Michael Kohnen[2] und Thomas M. Lehmann[1]

[1]Institut für Medizinische Informatik
Rheinisch-Westfälische Technische Hochschule (RWTH), 52057 Aachen
[2]Klinik für Radiologische Diagnostik, Universitätsklinikum Aachen, 52057 Aachen
Email: clappe@mi.rwth-aachen.de

Zusammenfassung. Die Bestimmung von Ähnlichkeiten zwischen medizinischen Bildern erfordert in manchen Kontexten die Einbeziehung struktureller Bildinformation, wie Nachbarschaftsbeziehungen oder Hierarchierelationen zwischen Teilregionen der Bilder. Werden attributierte Graphen zur Modellierung der strukturellen Information als Bildrepräsentation eingesetzt, so führt die Bestimmung von Ähnlichkeiten auf ein NP-vollständiges Graphmatchingproblem. Dieser Beitrag zeigt, wie durch Parallelisierungstechniken die Leistung eines auf Neuronalen Netzen basierenden Matchingalgorithmus so optimiert werden kann, dass er auch für große Datenvolumina, wie sie medizinisches Bildmaterial erfordern, eingesetzt werden kann. Eine Partitionierung des Neuronalen Netzes sowie eine Synchronisierung der gewonnenen Cluster bilden den Kern der Optimierung. Das Konzept eignet sich sowohl für parallele Berechnungen in verteilten Systemumgebungen als auch für den Einsatz auf isolierten Simulationsrechnern.

1 Einleitung

Die Bestimmung von Ähnlichkeiten zwischen medizinischen Bildern ist grundlegend für viele Anwendungen im Bereich der medizinischen Bildverarbeitung. So greifen beispielsweise Systeme zum inhaltsbasierten Retrieval von Bildern (CBIR) auf automatisch extrahierte globale wie lokale, also regional beschränkte, Merkmale von Bildern zu, um durch paarweise Vergleiche dieser Merkmale möglichst „ähnliche“ Bilder für den betreffenden Kontext in umfangreichen Referenzdatenbanken zu detektieren. Erlaubt der Anwendungskontext eine Beschränkung auf niedrig-dimensionale Merkmalsräume, wie beispielsweise Farbmerkmale, so stehen effiziente Datenstrukturen und Algorithmen zur Verfügung, die über eine Indizierung dieser Merkmale Retrievalergebnisse in sublinearer Laufzeit (bezüglich der Größe der Referenzdatenbank) generieren können. Im Projekt „Image Retrieval for Medical Applications“ (IRMA) wird ein Rahmenwerk zum inhaltsbasierten Retrieval medizinischer Bilder definiert, das mit möglichst wenigen Einschränkungen an Anfragekontexte oder Bildgenerierungsmodalitäten

auskommen soll. Für diese Anwendung ist wesentlich, möglichst viele Merkmalstypen flexibel in die Datenstrukturen zu integrieren und auf Basis dieser Merkmale Ähnlichkeitsvergleiche führen zu können. Eine multiskalare und somit von Anfragekontexten weitgehend unabhängige Bildrepräsentation ist mit hierarchisch organisierten, attributierten Graphen gegeben [1]. Deren Knoten repräsentieren Regionen im Bild, die mit Vektoren lokaler Merkmale verknüpft werden können. Die Kanten der Graphen modellieren Adjazenz- und Inklusionsrelationen zwischen den Regionen – die Inklusionsrelation bildet dabei eine hierarchische Struktur aus. Die Frage der Bildähnlichkeit führt somit auf ein Matchingproblem ihrer Graphrepräsentationen, das als inexaktes Subgraph-Isomorphismus-Problem klassifiziert werden kann. Da diese Problemklasse als NP-vollständig bekannt ist, d.h. ihre Probleme im Allgemeinen nicht effizient lösbar sind, und die Datenvolumina zur multiskalaren Repräsentation eines medizinischen Bildes erheblich sind, muss auf approximative Lösungsverfahren zurückgegriffen werden, die zudem eine besondere Flexibilität bezüglich der verwendeten Merkmale aufweisen müssen. Das von Schädler und Wysotzki vorgestellte Matchingverfahren auf Basis Neuronaler Netze [2] erfüllt diese Anforderungen. Es wurde für eine Evaluierung im Rahmen des IRMA-Projektes ausgewählt [3], jedoch stand bisher keine ausreichend performante Implementierung für sehr große Netze zur Verfügung. Mit dem hier vorgestellten Parallelisierungskonzept konnte die Leistung so weit optimiert werden, dass der Matcher nun für die Forschung an Fragen des IRMA-Projektes eingesetzt werden kann.

2 Methode

Das grundlegende Verfahren aus [2] sieht vor, aus zwei gegebenen Eingabegraphen einen Kompatibilitätsgraphen, der Mappinghypothesen zwischen den Eingabegraphen der Form „bilde Knoten X auf Knoten Y ab“ codiert, und hieraus ein Neuronales Netz der Hopfield-Art zu konstruieren. Dessen Neuronen tauschen in einer Simulationsphase über gewichtete Verbindungen excitatorische, d.h. stärkende, oder inhibitorische (hemmende) Erregungen iterativ aus, bis sich ein garantierter, global konvergenter Zustand einstellt. Die Simulation entspricht dabei einem Gradientenabstiegsverfahren im Raum der Neuronenpotentiale, also einer Heuristik, die ein Optimierungsproblem approximativ löst. Stabile Zustände des Systems bilden (lokale) Minima einer Energiefunktion, die an ein globales Distanzmaß zwischen Bildern, das sich aus deren Merkmalsvektoren und strukturellen Relationen berechnet, gekoppelt wird. Durch den Gradientenabstieg wird somit versucht, eine minimale (globale) Distanz zwischen den Bildern, aus denen die Eingabegraphen mittels eines Regiongrowing-Partitionierungalgorithmus konstruiert wurden [1], zu berechnen. „Fehlschläge“ in Form suboptimaler Lösungen, d.h. lokaler aber nicht globaler Minima, müssen unter Berücksichtigung der Komplexitätsklasse akzeptiert werden.

2.1 Netzpartitionierung

Ansatzpunkt für die Leistungsoptimierung bietet die an sich synchrone Aktualisierungsvorschrift der Neuronen des Netzes. Prinzipiell werden alle Neuronen einmal pro Iteration gleichzeitig aktualisiert, d.h. ihre Potentiale unter Berücksichtigung aller eingehenden Erregungen neu berechnet. Diese Gleichzeitigkeit ermöglicht eine parallele Verarbeitung der Neuronen, auch auf verschiedenen Rechnern, die für die Simulationsphase zusammengeschlossen werden. Bedingung ist, dass die Neuronen nicht voneinander abhängig sind, dass also keine Verbindung zwischen ihnen im Netz besteht. Das Netz wird also in einem ersten Schritt partitioniert, in Cluster von Neuronen aufgeteilt, wobei die Abhängigkeiten zwischen verschiedenen Clustern zu minimieren sind. Vollkommen unabhängige Cluster werden in der Praxis nur selten zu realisieren sein, da die verwendeten Netze relativ dicht sind, d.h. einen hohen Verbindungsgrad aufweisen. Eine Minimierung der Abhängigkeiten kann angestrebt werden, allerdings ist das Problem der exakten Minimierung wiederum NP-vollständig. Es wurde ein Algorithmus für diese Partitionierung entwickelt, der schon während der Konstruktionsphase des Netzes durch systematisches Verfolgen von Kanten eine Einteilung in Cluster vornimmt, und in ersten Tests positive Ergebnisse zeigte. Als eigenständige Komponente implementiert, kann er jedoch auch ausgetauscht oder durch spezialisierte Partitionierungsalgorithmen ergänzt werden.

2.2 Synchronisierung

Können nicht alle Abhängigkeiten eliminiert werden, so wird eine Synchronisierung der Cluster erforderlich, bei der einmal pro Iteration Informationen zwischen den Clustern ausgetauscht werden. Zu diesem Zweck bildet jeder Cluster sog. „Travelunits“, mobile Einheiten eines Neurons, die an andere Cluster übermittelt werden können, um gezielt Potentialwerte abzufragen, gleichzeitig aber genügend Informationen enthalten, um das abgefragte Zielneuron gleichzeitig zu aktualisieren. Eine definierte Ordnung auf den Neuronen aller Cluster sorgt dafür, dass für jede Verbindung des Netzes jeweils nur eine Travelunit erzeugt wird (Uplink-Strategie). Travelunits werden in Exportlisten gesammelt, um sie effizient und systematisch nach einem festgelegten Versandschema zwischen den Clustern übertragen zu können. Als Effekt der Uplink-Strategie kann dieses Versandschema an unterschiedliche Leistungsstärken der an der Simulation beteiligten Rechner angepasst werden. Je kürzer die Exportlisten, desto wirkungsvoller war die Partitionierung, und desto weniger zeitlicher Overhead entsteht als Folge der Parallelisierung. Ist ein Cluster einem Rechner zur Bearbeitung zugeordnet, so braucht nur dieser Rechner die Simulationsdaten des Clusters – aktuelle Potentiale, Akkumulatoren etc. – lokal vorrätig zu halten. Die Größe der einzelnen Cluster kann dabei so dimensioniert werden, dass die Daten vollständig im physischen RAM aller Rechner gehalten werden können und somit immer schnell zur Verfügung stehen - der Partitionierungsalgorithmus verwendet hierfür eine benutzerdefinierbare Partitionstabelle.

In der Simulationsphase werden für jeden Cluster iterativ die folgenden vier Schritte durchgeführt, sie bilden den internen Clusterzyklus:

– Integration der eigenen Exporte. Zurückgekehrte, bearbeitete Travelunits werden reintegriert.
– Iterationswechsel. Skalierungen, Normierungen.
– Cluster-interne Aktualisierungen. Alle hierfür benötigten Potentialwerte sind sofort verfügbar.
– Bearbeitung externer Exportlisten.

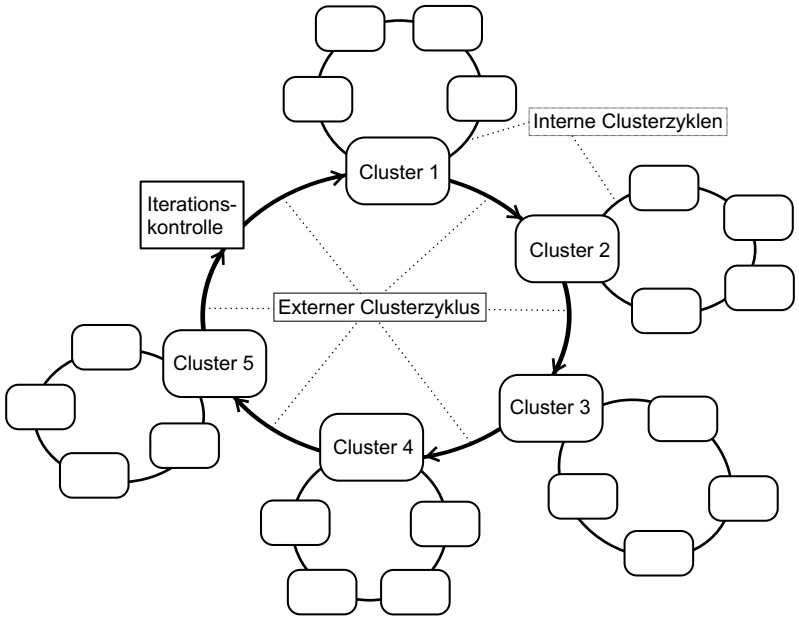
Die vorgestellte Parallelisierung wurde ursprünglich für die verteilte Entwicklungsplattform des IRMA-Projektes [4] konzipiert, ihre Mechanismen können aber auch auf einem isolierten Simulationsrechner gewinnbringend eingesetzt werden. Zum leistungsdrosselnden Flaschenhals wird hier der beschränkte physische Speicherausbau, der die für IRMA benötigten großen Netze nicht vollständig verfügbar machen kann. In einer naiven Implementierung müssen Auslagerungsmechanismen des Betriebssystems diesen Mangel kompensieren, die jedoch uninformiert über den Simulationsverlauf keine gute Leistung zeigen. Die Parallelisierung ermöglicht auch hier, das Netz in ausreichend kleine Cluster zu zerlegen, und diese untereinander zu synchronisieren. Dabei werden die Daten eines Clusters in zusammenhängenden Speicherbereichen abgelegt, die durch eine kontrollierte Auslagerung nach Bearbeitung eines Clusters sehr effizient ausgetauscht werden können. Diese gezielten Clusterwechsel bilden den externen Clusterzyklus, den Abb. 1 schematisch darstellt.

3 Ergebnisse und Diskussion

Erste Testläufe bestätigen das Konzept und zeigen eine erhebliche Leistungssteigerung gegenüber der originalen Implementierung aus [2]. Werden dort Simulationsläufe mit mehr als 300.000 Verbindungen im Netz aus Effizienzgründen zurückgewiesen, so werden nun selbst Netze mit 68.000 Neuronen und 15 Millionen Verbindungen in zwölf Minuten auf Standardhardware vollständig berechnet. Die Parallelisierung erzeugte einen zeitlichen Overhead von zwei Minuten während der Netzkonstruktion, führte jedoch zu einer Beschleunigung der Simulationsphase um 30 Prozent pro Iteration. Der eingesetzte Partitionierungsalgorithmus zeigte eine Reduzierung der Abhängigkeiten zwischen Clustern um 60 Prozent gegenüber einer zufälligen Verteilung von Neuronen über alle Cluster.

Die Optimierung des Graphmatchers ermöglicht seinen Einsatz zur Untersuchung von Ähnlichkeiten medizinischer Bilder im Rahmen des IRMA-Projektes. Besonders vielversprechend ist die Skalierbarkeit des Verfahrens, da es an verschiedenen Stellen an den physischen Speicherausbau und die Geschwindigkeit beteiligter Simulationsrechner adaptiert werden kann und Auslagerungsbereiche auf Sekundärmedien, die nahezu unbegrenzt zur Verfügung stehen, nun effizienter genutzt werden können. Eine detailliertere Analyse der Leistungssteigerung in einzelnen Komponenten empfiehlt sich, da hier noch Potential für weitere Verbesserungen gesehen wird.

Abb. 1. Externe und interne Clusterzyklen steuern den Simulationslauf der parallelen Implementierung auf einem isolierten Rechner. Das Beispiel zeigt eine Partitionierung des Netzes in fünf Cluster, die in Kontextwechseln sukzessive in den physischen Speicher des Rechners eingelesen werden (externer Zyklus). Für jeden der Cluster wird einmal der interne Zyklus durchlaufen, in dem alle Aktualisierungen und nötigen Synchronisierungen vorgenommen werden. Eine Iteration ist abgeschlossen, wenn der externe Zyklus einmal durchlaufen wurde.



Literaturverzeichnis

1. Thies C, Metzler V, Aach T: Content-Based Image Analysis: Object Extraction by Data-Mining on Hierarchichally Decomposed Medical Images. Procs SPIE. 5032: 579–589, 2003.
2. Schädler K, Wysotzki F: Comparing Structures using a Hopfield-style Neural Network. Applied Intelligence. 11:15–30, 1999.
3. Fischer B, Thies C, Güld MO, Lehmann TM: Matching von Multiskalengraphen für den inhaltsbasierten Zugriff auf medizinische Bilder. Procs BVM 2003. 353–357, 2003.
4. Güld MO, Fischer B, Thies C et al.: A platform for distributed image processing and image retrieval. Procs SPIE. 5150: 1109–1120, 2003.