

Softwarekonzepte und Algorithmen für die 3D-Bildverarbeitung

Bernd Jähne¹ und Helmut Herrmann²

¹Interdisziplinäres Zentrum für Wissenschaftliches Rechnen, Universität Heidelberg
Im Neuenheimer Feld 368, D-69120 Heidelberg
Tel.: 06221/54 8827, Fax: 06221/54 8850
E-mail: Bernd.Jaehne@iwr.uni-heidelberg.de
²AEON Digital Image Processing
Fraunhoferstr. 51B, D-63454 Hanau
Tel.: 06181/2 35 25, Fax: 06181/257 954
E-mail: 100326.3156@compuserve.com

Zusammenfassung

Bildverarbeitungssoftware muß weitgehend plattformübergreifend, modular und offen aufgebaut sein, wenn sie nachhaltig zu einem erfolgreichen Einsatz in komplexen Aufgabenstellungen wie der 3D-Bildverarbeitung führen soll. Am Beispiel der Bildverarbeitungssoftware *heurisko* werden diese Prinzipien erläutert und Beispiele moderner Algorithmen gezeigt, die Bilddaten genauer, zuverlässiger und deutlich schneller auswerten können.

1 Einleitung

Die meisten am Markt erhältlichen Bildverarbeitungspakete und Entwicklungssysteme beschränken sich auf die 2D-Bildverarbeitung mit 8-Bit Bildern. Sie sind daher ungeeignet für die 3D-Bildverarbeitung. In diesem Übersichtsbeitrag werden die Prinzipien portabler und modularer Software für die Bildverarbeitung beleuchtet und dann am Beispiel von *heurisko* ein Softwarepaket vorgestellt, das nicht speziell für die 3D-Bildverarbeitung entwickelt wurde, sondern ein universelles Entwicklungsinstrument darstellt und neben 3D-Bilddaten auch Bildsequenzen und Mehrkanalbilder verarbeiten kann.

2 Prinzipien portabler und modularer Bildverarbeitungssoftware

2.1 Notwendigkeiten

Bedingt durch die rasanten Fortschritte in der Hardware werden die Aufgabenstellungen, die mit digitaler Bildverarbeitung gelöst werden können, immer komplexer. Eine Bildverarbeitungsaufgabe läßt sich nicht mehr durch das Zusammenstricken einer spezifischen Hardware lösen, sondern nur mit einem durchdachten Softwaresystem, das nicht nur auf eine bestimmte Aufgabenstellung zugeschnitten ist, sondern generische Lösungen anbietet, die sich flexibel auf neue Aufgabenstellungen anpassen

lassen. Es zeichnet sich deutlich ab, daß auch die Algorithmik in der Bildverarbeitung immer mehr Bedeutung gewinnen wird.

2.2 Möglichkeiten

Traditionsgemäß wurden Bildverarbeitungsaufgaben hardwarenah gelöst, um entsprechende Verarbeitungsgeschwindigkeiten zu erreichen. Man kann diese Tatsache auch so negativ gesehen auch so ausdrücken, daß die enormen Fortschritte in der Algorithmik für die Bildverarbeitung bisher weitgehend nicht eingesetzt werden konnten, weil die Softwareentwickler damit beschäftigt waren, relative einfache Algorithmen von einem System auf das nächste zu portieren.

Die Entwicklung der allgemeinverfügbaren Hardware hat heute in vielen wichtigen Punkten eine kritische Schwelle erreicht, so daß es für den überwiegenden Teil der Bildverarbeitungsaufgaben nicht mehr notwendig ist, auf Spezialhardware zurückzugreifen:

- Es bedarf keiner speziellen Videobussysteme mehr. Moderne Bussysteme zur Verbindung mit peripheren Ein-/Ausgabegeräten wie der PCI-Bus sind schell genug, daß damit Bilddaten in Videoechtzeit in den PC-Speicher übertragen werden können.
- Die Bilddarstellung bedarf keines zusätzlichen Aufwandes mehr. Sie ist Bestandteil der graphisch orientierten Benutzerschnittstellen.
- Die Rechenleistung der allgemein verfügbaren Rechner ist schneller gewachsen als die der Spezialhardware. Diese Entwicklung wird weiter beschleunigt durch die Integration von Multimediafähigkeiten in Standardhardware. Mit speziellen Instruktionssatzerweiterungen zur Pixelverarbeitung wie MMX (Multimedia Extension) von Intel und VIS (Visual Instruction Set) von Sun erfährt die Verarbeitung von Bilddaten einen enormen Leistungssprung (Jähne, 1996). Ein 200-MHz-Pentium mit MMX rückt in die Leistungsklasse von schnellen digitalen Signalprozessoren (DSP) wie der MVP von Texas Instruments und der Video-DSP Imagine von Arcobel.

Die Vorteile dieser Entwicklung für die Bildverarbeitungssoftware sind enorm. Dem Softwareingenieur stehen ihm bekannte und wesentlich komfortablere Entwicklungswerkzeuge zur Verfügung als dies für Spezialhardware der Fall ist. Der Hauptteil der Entwicklungszeit kann nun tatsächlich in die Bildverarbeitungsalgorithmen selbst fließen und nicht in ihre ständige Anpassung an neue Hardware. Es ist offensichtlich, daß mit einem solchen Ansatz Bildverarbeitungssoftware schnell an eine neue Hardwareplattform angepaßt werden kann.

3 Fallstudie heurisko

Am Beispiel des Bildverarbeitungs-Entwicklungspakets heurisko soll in dieser Fallstudie erläutert werden, welche Vorteile mit einer portablen, modularen und offenen Softwareumgebung einhergehen. Gleichzeitig sollen eine Reihe moderner Algorithmen für die 3D-Bildverarbeitung vorgestellt werden.

3.1 Datenstrukturen

Ein konventionelles Bildverarbeitungspaket ist in der Regel nur für die Auswertung von 8 Bit tiefen Bilddaten ausgelegt. Moderne Sensoren liefern heute aber bis zu 16 Bit tiefe Bilddaten. Für die präzise 3D-Vermessung reichen 8 Bit ebenfalls nicht aus, sondern sind mindestens 16 Bit oder besser Gleitkommazahlen notwendig. Darüber hinaus verlangen viele moderne Algorithmen eine höhere Rechengenauigkeit, so daß sie am besten mit Gleitkommaarithmetik durchgeführt werden.

heurisko enthält eine Fülle von Datentypen für Bilddaten: Binärbilder, Festkommabilder mit 8, 16 und 32 Bit, Fließkommabilder mit einfacher und doppelter Genauigkeit und darüber hinaus auch komplexe Bilddatentypen (Tabelle 1). Die Bilddaten können in ihrer Größe völlig frei definiert werden. Unterstützt werden außerdem Mehrkanalbilder (wobei die einzelnen Kanäle unterschiedliche Datentypen

Tabelle 1: Datentypen in heurisko.

Typ	Beschreibung	Typ	Beschreibung
binary	Binär	float	Gleitkommazahl
byte	8-Bit-Integer	double	Gleitkommazahl doppeltgenau
short	16-Bit-Integer	fcomplex	Gleitkommazahl komplex
long	32-Bit-Integer	dcomplex	Gleitkommazahl doppeltgenau komplex
scomplex	16-Bit-Integer komplex	string	Zeichenkette
lcomplex	32-Bit-Integer komplex		

aufweisen können), Mehrgitterdatenstrukturen wie Pyramiden, Bildsequenzen und volumetrische (3D-) Bilder. Die Definition von Vektoren (z.B. für Histogramme oder Bildzeilen) und Skalaren ist ebenso möglich. Zur Beschreibung von Objektmerkmalen, lassen sich zudem lineare Listen aufbauen.

3.2 Bildverarbeitungsfunktionen als Operatoren

Alle Bildverarbeitungsfunktionen werden als Operatoren für die oben beschriebenen Objekte betrachtet. Damit sind eine Reihe von nachhaltigen Vereinfachungen möglich. Derselbe Operator kann auf Bilder mit unterschiedlichsten Datentypen angewendet werden. Ebenso berücksichtigt er automatisch die Dimension des Objektes. Wird eine zweidimensionale Faltungsoperation z.B. auf eine Bildsequenz angewendet, werden alle Bilder der Sequenz gefaltet, ohne daß eine Schleife programmiert werden muß. Die in heurisko eingebauten Operatoren muß man sich wie einen Werkzeugkasten mit einfachen Operatoren vorstellen. Mit Hilfe einer mächtigen Skriptsprache können die eingebauten Operatoren zu neuen zusammengesetzt werden. Diese lassen sich genauso wie bereits implementierte Operatoren verwenden. Da solche neu definierten Operatoren in einen schnellen Binärcode übersetzt werden, sind mit ihnen im Vergleich zu den eingebauten Operatoren kaum Geschwindigkeitseinbußen verbunden. Die Skriptsprache enthält vielfältige Steuerkonstrukte zur Programmierung von bedingten Anweisungen und verschiedene Arten von Schleifen.

Bildqualität und Bildverarbeitungsoperationen können durch maus- und operatorgesteuerte Bildinspektoren wie Zeilen- und Spaltenprofile, Linienplots, Lupenfunktion oder Histogramme von beliebigen Bildausschnitten gezielt und interaktiv beurteilt werden. Ist ein solcher Inspektor für ein Bildfenster aktiv gesetzt, berücksichtigt er automatisch die Änderung des Bildinhaltes. So kann beim Ablauf einer Bildsequenz z.B. die Änderung des Histogramms oder eines Profils mitbeobachtet werden.

Neu definierte Operatoren und auch die definierten Bilddaten, die für eine bestimmte Bildverarbeitungsaufgabe benötigt werden, können in sogenannten Workspaces abgespeichert werden. Durch einfaches Laden eines solchen Workspaces ist die Software sofort an eine bestimmte Bildverarbeitungsaufgabe angepaßt. Neu definierte Operatoren werden automatisch in die dynamisch aufgebauten Menüs eingefügt. Mit modernen Framegrabbern auf Basis des PCI-Bus und einem einfachen Satz von Operatoren zur Kontrolle der Echtzeit-Digitalisierung per DMA-Transfer können in dieser Skriptsprache auch Echtzeitbildverarbeitungsroutinen auf einfachste Weise programmiert und mit der Bildaufnahme synchronisiert werden.

3.3 Offener und modularer Aufbau

heurisko ist modular aufgebaut und besteht aus drei Teilen, der Benutzerschnittstelle, einem Interpreter/Compiler und dem Programmkern mit den eigentlichen Bildverarbeitungsfunktionen, die über standardisierte Schnittstellen miteinander kommunizieren (Abb. 1). Der Interpreter kann mehrere Kerne steuern. Somit ist in dem Ansatz paralleles und verteiltes Rechnen enthalten. Der Benutzer kann sowohl die vorhandene Benutzerschnittstelle ersetzen als auch den Programmkern um zusätzliche

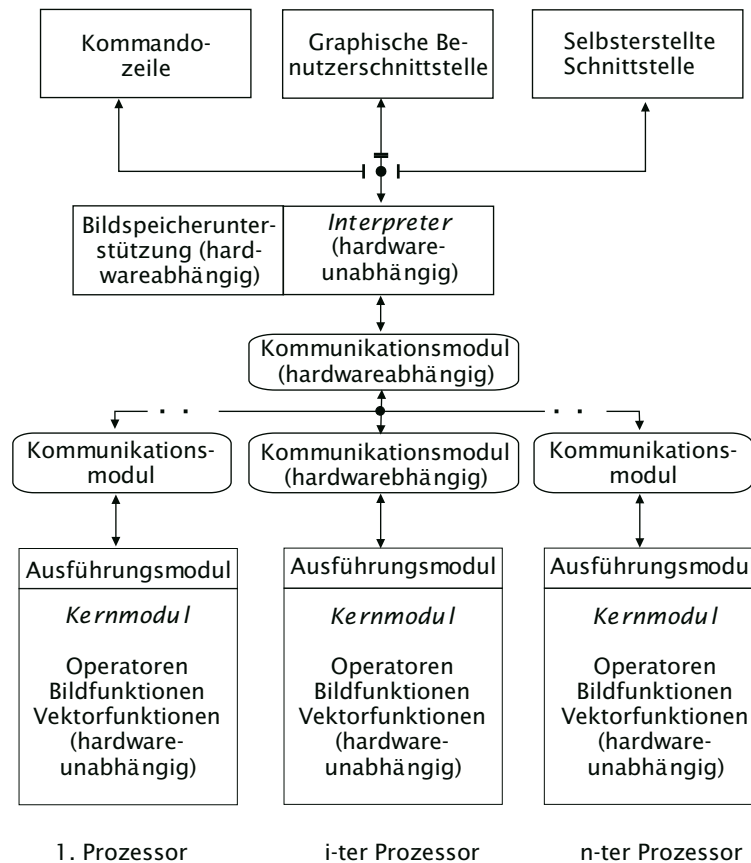


Abb. 1: Blockdiagramm des modularen und hierarchischen Aufbaus des Softwarepakets heurisko

Funktionen erweitern. Der heurisko-Bildverarbeitungskern ist streng hierarchisch in drei Ebenen aufgebaut. Auf der äußeren Ebene befinden sich die sogenannten Operatoren, die auf die Objekte wirken. Diese Funktionen entsprechen direkt den im Befehlszeileninterpreter oder in der graphischen Benutzerschnittstelle verfügbaren Befehlen.

Ein spezieller, in der Benutzerschnittstelle nicht sichtbarer Typ von Operatoren sind die sogenannten generischen Operatoren. Diese nutzen konsequent die Tatsache, daß es nur eine kleine Anzahl verschiedener Klassen von Bildverarbeitungsalgorithmen gibt, wie z. B. Punktoperationen, Nachbarschaftsoperationen, Mehrgitteralgorithmen und globale Transformationen. Damit ist in den generischen Operatoren ein Großteil der Steuerstrukturen eines neuen Operators bereits vorhanden, und der Benutzer muß nur eine neue Bild- oder Vektorfunktion schreiben, die vom generischen Operator als Parameter aufgerufen wird. Auf diese Weise sind die meisten in heurisko eingebauten Operatoren implementiert. Eine Ebene darunter befinden sich die Bildfunktionen, die auf 2D- bis 4D-Bilder angewandt werden. Auch auf dieser Ebene gibt es wiederum generische Funktionen, welche die Programmierung erheblich erleichtern. Auf der untersten Ebene finden sich schließlich die Vektorfunktionen, die nur auf Vektoren wirken. Fast alle Bildfunktionen und Operatoren rufen schließlich diese Funktionen auf.

Die Portabilität und von heurisko wird belegt durch Implementierungen unter DOS, Windows 3.1, Windows'95, Windows NT, verschiedenen UNIX-Plattformen (einschließlich Linux) und OS9 (ELTEC Hardware). Dem hierarchisch aufgebauten Bildverarbeitungskern ist es zu verdanken, daß ein Großteil der Funktionen in heurisko in kürzester Zeit in Assembler auf MMX optimiert werden konnte.

3.4 Effiziente und genaue Algorithmen

heurisko zeichnet sich durch mathematisch fundierte Algorithmen aus. Dabei wurde sowohl auf die Geschwindigkeit als auch auf die Genauigkeit Wert gelegt. Es finden sich eine Fülle von Algorithmen, die nicht in den üblichen Bildverarbeitungspaketen enthalten sind. Für die 3D-Bildverarbeitung sind von besonderer Bedeutung:

3D- und 4D-Nachbarschaftsoperationen : Alle Nachbarschaftsoperationen (Faltungen, Binär- und Grauwertmorphologie) lassen sich nicht nur auch 2D- sondern auch 3D- und 4D-Bilddaten anwenden.

Hilbert-Filter : Grundbaustein für die schnelle und subpixelgenaue Bestimmung der Lage von Signalen, z.B. der Streifen bei der Streifenprojektionsmethode

Pyramiden : Grundbaustein für die schnelle Implementierung vieler Algorithmen

Normalisierte Faltung : berücksichtigt den Fehler einzelner Bildpunkte und ist daher unmittelbar geeignet, unvollständige Daten zu extrapolieren

4 Schlußbemerkungen

Bei aller Bedeutung der Hardware wird die Software und Algorithmik eine immer wichtigere Rolle spielen. Sie verlangt allerdings fundiertes und breitangelegtes mathematisches und physikalisches Verständnis. Deshalb kommt einer engen Zusammenarbeit zwischen Forschungsgruppen an den Universitäten und der weitgehend mittelständig orientierten Bildverarbeitungsindustrie eine wichtige Rolle zu.

In diesem Sinne haben die regelmäßig stattfindenden ABW Workshops eine wesentliche Bedeutung. Um den Austausch zwischen Industrie, Fachhochschulen und Hochschulen im südwestdeutschen Raum in einem breiteren Themenbereich zu fördern und eine zeitgemäße Weiterbildung anzubieten, wurde 1995 das Heidelberger Bildverarbeitungsforum gegründet. In drei regelmäßigen Nachmittagsveranstaltungen — jeweils am ersten Dienstag in den Monaten November, März und Juli — werden aktuelle Entwicklungen der Bildverarbeitung von namhaften Experten zusammenfassend dargestellt. Das ermöglicht den Anwendern aus der Praxis einen schnellen Einstieg in neue Techniken. Der Beirat für das Forum setzt sich aus führenden Vertretern von Hochschulen, Fachhochschulen und der Industrie zusammen.

Das Heidelberger Bildverarbeitungsforum wird von der Akademie für Weiterbildung an den Universitäten Heidelberg und Mannheim e.V. organisiert. Einen Überblick über die bisher durchgeführten und kommenden Veranstaltungen gibt die folgende Tabelle:

Nr.	Datum	Thema	Veranstaltungsort
1	7. Nov. 1995	Segmentierung	Universität Heidelberg
2	5. März 1996	Bildgebende Sensorik	Universität Heidelberg
3	2. Juli 1996	Texturanalyse	ABB Forschungszentrum, Heidelberg
4	5. Nov. 1996	Schnelle Hardware und Algorithmen	Deutsches Krebsforschungszentrum, Heidelberg
5	4. März 1997	Farb- und Mehrkanalbildverarbeitung	Forschungs- und Entwicklungszentrum Heidelberg Druckmaschinen, Heidelberg
6	1. Juli 1997	Bildkommunikation und Bilddatenbanken	TecMath GmbH, Kaiserslautern

Literatur

Jähne B, (1996) SIMD-Bildverarbeitungsalgorithmen auf PCs, demonstriert am Multimedia Extension Instruktionssatz (MMX) von Intel, 4. Heidelberger Bildverarbeitungsforum, 5. November 1996, DKFZ, Heidelberg, Akademie für Weiterbildung an den Universitäten Heidelberg und Mannheim e.V.