

Eine Sprache zur Implementation von Methoden der Bildverarbeitung

M. Woste, S.J. Pöpl
Gesellschaft für Strahlen- und Umweltforschung
Ingolstädter Landstr. 1
8042 Neuherberg

Zusammenfassung:

Die Bildverarbeitungssprache ELEPHAND (Extensible Language for Easy Picture Handling) bemüht sich, zwischen sauberer formaler Spezifikation von Bildverarbeitungsmethoden und deren effizienter Implementation einen Kompromiß zu schließen. Um maximale Effizienz zu erreichen, können vorhandene Algorithmen einfach gegen schnellere ausgetauscht werden; außerdem besteht die Möglichkeit, auf schnelle Bildverarbeitungs-Spezialrechner zuzugreifen. Die Sprache deckt den Bereich zwischen Bildverbesserung und Segmentierung ab, in ihr geschriebene Programme stellen Daten zur Verfügung, die anhand üblicher Mustererkennungsmethoden analysiert und interpretiert werden können.

1. Problemstellung:

Bedingt durch die Verwendung verschiedenster Aufnahmegeräte mit unterschiedlichen Bildfehlern sowie die Aufnahme verschiedenartiger Objekte gibt es kein einheitliches Bildmaterial, so daß jede Anwendung andere Algorithmen zur Bearbeitung der Bilder benötigt. Ob ein Verfahren anwendbar ist, kann nur durch 'trial and error' - Methoden durch visuelle Überprüfung des Ergebnisses festgestellt werden. Daher ist die Interaktivität eine wichtige Anforderung an jedes Bildverarbeitungssystem.

Der spezielle Charakter der Bildverarbeitung, ein völlig unstrukturiertes, umfangreiches Rohdatenmaterial bearbeiten zu müssen, das mit vielen nicht beeinflussbaren Störgrößen behaftet ist, stand einer Methodenstandardisierung bisher im Wege, und die Verschiedenheit der Anwendungen und der für diese Anwendungen eingesetzten Spezial-Bildverarbeitungsrechner machten eine Geräteunabhängigkeit nahezu unmöglich.

Daher gestaltet sich eine Klassifikation existierender Verfahren schwierig und wird in der Standardliteratur nach den verschiedensten Gesichtspunkten gehandhabt, wie, um zwei Extreme zu nennen, nach den zur Anwendung kommenden mathematischen Grundlagen oder nach Performance-Kriterien wie z.B. Grad der Parallelisierbarkeit < 1-4 > .

Die Beseitigung schwerwiegender Hardwarebeschränkungen, z.B. durch Vergrößerung des Speicherplatzes, Verfügbarkeit hochauflösender Displays, Erweiterung des Leistungsspektrums von Bildverarbeitungsrechnern) läßt es sinnvoll erscheinen, Werkzeuge zu schaffen, die eine problemorientierte, leicht zu erlernende und zu handhabende Notation auf der einen Seite mit einer effektiven Codierung von Algorithmen auf der anderen Seite verbinden.

2. Realisierungsansatz

Interaktive Bildverarbeitungssysteme sind in der Praxis häufig als hierarchisch menügetriebene Systeme aufgebaut. Die Anzahl der bekannten Methoden der Bildverarbeitung macht es jedoch wahrscheinlich, daß einige davon schwer auffindbar sind. Außerdem erfolgt die Kombination von existierenden Methoden unter vollständiger Steuerung des Benutzers, was bei zeitaufwendigen Methoden zu langen Wartezeiten vor dem Bildschirm führt. Schließlich ist noch das Problem der mangelnden Mittelbarkeit der Algorithmen an andere gegeben.

In der hier vorliegenden Arbeit wurde aus diesen Gründen der Realisierung als Bildverarbeitungssprache der Vorzug gegeben. Die elementaren Datenstrukturen und Operatoren sind auf die Belange der Bildverarbeitung abgestimmt, hierbei wurden Konstrukte und Operatoren, die schon aus verbreiteten Hochsprachen wie PASCAL, C oder PL/I bekannt sind < 5-12 > , verwendet und sinngemäß zur Bearbeitung von Bilddatenstrukturen erweitert. Zwischen den Operatoren sind die üblichen Prioritäten definiert, so daß der Gebrauch von Klammern minimiert werden kann.

ELEPHAND ist, um Interaktivität zu gewährleisten, als Kommandointerpreter konzipiert. Aus der Kommandoumgebung können bereits implementierte Standardoperationen als Funktionen oder Operatoren aufgerufen werden und werden dann sofort ausgeführt.

Wenn eine Methode sich im Test als anwendbar herausgestellt hat, ist es wünschenswert, sie in das Gesamtsystem einzubetten. Die Ausführungsgeschwindigkeit der Methode sollte maximierbar sein. Daher stellt ELEPHAND außer einer Kommando-umgebung noch einen Prozedurinterpret bereit, dessen Sprache der Kommandosprache entspricht. Um Nebenwirkungsfreiheit zu erreichen, handelt es sich um eine funktionale Sprache mit einem 'call by value/result' - Parameterübergabemechanismus, in der es keine globalen Variablen gibt. Objekte dieser Sprache sind Funktionen und Atome (= Bildvariablen, Look Up Tables, Konstanten etc.). Zur Vereinfachung der Bedienung wurde der funktionale Ansatz innerhalb der Funktionen nicht konsequent realisiert: Dort können allgemein übliche Programmsteuerungsanweisungen wie z.B. Iterationsstatements (Schleifen) verwendet werden. Zusammengesetzte Objekte (wie z.B. Listen) existieren nicht < 13 > .

Da eine interpretative Abarbeitung von Kommandos stets zeitaufwendig ist, können vom Benutzer in der Wirtssprache geschriebene Funktionen auch in Maschinencode übersetzt und in das System eingebettet werden. Diese sog. 'Primitive Functions' werden während der Laufzeit des Interpreters auf Anfrage geladen und zum Modul hinzugebunden.

Um technischen Innovationen Rechnung zu tragen, ist maximale Portabilität, d.h. Betriebssystem- und Hardwareunabhängigkeit, eine wichtige Anforderung; andererseits müssen spezielle Bildverarbeitungsrechner maximal effizient verwaltet werden können. Daher kann das System auf einen angeschlossenen Spezialrechner mit spezieller Pipeline- oder Arrayarchitektur zugreifen, ohne andererseits jederzeit von der tatsächlichen Präsenz dieser Zusatzhardware abhängig zu sein:

Lokale Speicher werden logisch mittels spezieller 'Hardware'- Datentypen erfaßt. Alle elementaren Bildverarbeitungsoperationen sind generisch implementiert, d.h. dasselbe Kommando kann sowohl für host-basierte als auch für Hardware-Variablen eingegeben werden. So ist keiner Operation anzusehen, wo und wie sie ausgeführt wird, dies ergibt sich einzig und allein aus den Typen der verwendeten Variablen.

Laden und Entladen der Daten aus dem Bildverarbeitungsrechner wird in Form von Typkonversionen angegeben. Lediglich die Datentypen und der Aufruf dieser Konversionsfunktionen weisen darauf hin, daß ein Spezialrechner verwendet wird. Ansonsten sind ELEPHAND-Prozeduren völlig unabhängig davon, in welchem Datentyp eine Operation durchgeführt wird.

3. Datenstrukturen

Von Kamera eingezogene Bilder stehen als Matrix von Grauwerten (typisch: 8 bits/pixel) zur Verfügung. Bildverarbeitungssprachen benötigen daher einen (in anderen Programmiersprachen nicht üblichen) Datentyp

CARD - 8 bit vorzeichenlose ganze Zahl
(Wertebereich 0..255)

Andere Datentypen sind

BOOLEAN - 1 bit logischer Wert
INT - 16 bit ganze Zahl mit Vorzeichen
REAL - 32 bit Gleitkommazahl
COMPLEX - Realteil, Imaginärteil jeweils 32 bit

Alle weiteren Datentypen sind - zumindest was ihre Implementation betrifft - stark vom verwendeten Bildverarbeitungsprozessor abhängig. zur interaktiven Verarbeitung sind auf jeden Fall die Typen

DISPLAY - Bildausgabeveriable und
CLUT - Colour Look Up Table

notwendig. Die Verwendung des Systems IBM 7350 in der Pilotinstallation (Pipelinearchitektur mit lokalem Speicher und in Hardware existierenden Look Up Tables) führte zur Definition der Datentypen

MASK - BOOLEAN in lokalem Speicher
BUFCARD - CARD
BUFINT - INT
ILUT - Look Up Table mit 256 Einträgen
OLUT - Look Up Table mit 65536 Einträgen