

Konzept einer universellen Programmiersprache für Bildverarbeitungsanwendungen

W. Eckstein, S.J. Pöpl

Institut für Medizinische Informatik und Systemforschung der GSF,
Ingolstädter Landstraße 1, D-8042 Neuherberg

Zusammenfassung

Beschrieben wird die Programmiersprache PSIWAG (Picture Segmentation and Interpretation With Attributed Graphs), die als Werkzeug für verschiedene Ansätze der Bildanalyse entwickelt wurde. Gekennzeichnet ist sie durch einen regelorientierten Programmaufbau, wobei einzelne Regeln Prozeduren repräsentieren, über die die Datenbasis manipuliert wird. Konzipiert ist das System als Interpreter, der erweitert und automatisch konfiguriert werden kann, zusammen mit einem Kommandosystem für Programmentwicklung und Anwendung.

Einführung

Eine Großzahl von Untersuchungen in der Bildverarbeitung beschäftigen sich mit Techniken zur Wissensrepräsentation[1,2]. Dies umfaßt sowohl Methoden der Programmierung als auch der Darstellung und Modellierung von Daten. Dabei wurden Vorgehensweisen aus anderen Fachrichtungen übernommen und auf die Bedürfnisse der Bildverarbeitung angepaßt (z.B. Syntax-Analyse, Logikprogrammierung, Datenbanken, Techniken der KI). Vorteil hierbei ist, daß größtenteils fundierte Theorien zur Verfügung stehen. Als Problem erwies sich jedoch unter anderem, daß manche Konzepte schlecht miteinander verträglich sind und außerdem der Datenumfang der Bilddaten eine effiziente Umsetzung nicht erlaubte. Dies führte zu einer Reihe unterschiedlicher Systeme, die nur Anwendungen für bestimmte Aufgabengebiete zulassen.

PSIWAG wurde entwickelt, um verschiedene Ansätze der Bildanalyse gleichzeitig realisieren zu können. Dies ermöglicht es, in einem Programm sowohl klassische Ansätze der Bildverarbeitung (z.B. Filterung, Segmentation), als auch weitergehender Techniken (z.B. abstrakten Bildbeschreibung, Syntaxanalyse, semantische Netze usw) in einem einheitlichen Konzept realisieren zu können.

Daten und Prozeduren

Für ein einheitliches Programmkonzept ist ein universeller Datentyp (Bildobjekt genannt), der die Repräsentation von Informationen auf verschiedenen Abstraktionsebenen (Graubild, Objektbereich, Attribut usw) zuläßt, sinnvoll:

Def.: Ein Bildobjekt o ist ein Tupel: $o = (T, G, A, I)$ mit

$$T \subset (1 \dots \text{Format})^2$$

$$G : ((1 \dots \text{Format})^2 \rightarrow 1 \dots 255) + \perp$$

$$A : (\mathbb{N}^+ \times \text{ID}) \rightarrow (\text{VAL} \cup \{\perp\})$$

$$I \in \mathbb{N}^+$$

wobei:

Format: maximale Anzahl von Zeilen, bzw. Spalten eines Bildes

VAL: Menge von Werten

ID: Menge von Bezeichner

Dabei liegt den 4 Komponenten folgende Interpretation zugrunde:

T ist die "Ausdehnung" eines Objektes, die im Fall eines Bildes alle Pixel umfaßt, ansonsten eine beliebige Teilmenge davon sein kann.

G ist die Grauwertzuordnung, die jedem Bildpunkt aus T einen Grauwert zuweist. G kann auch die undefinierte Funktion sein, d.h. daß den Bildpunkten keine Grauwerte zugeordnet sind. Ein Objekt besitzt dann nur eine Ausdehnung, aber keine Grauwerte (vergleichbar mit einem Binärbild).

A ist eine Attributierungsfunktion, mit der dem Objekt beliebige Werte (z.B. ganze Zahlen, Listen, Felder usw) über einen Bezeichner zugeordnet werden können.

I dient zur Identifizierung von Objekten.

Def.: Sei O die Menge aller Bildobjekte o . Ein Variablenbereich D ist definiert als die Potenzmenge von O :

$$D := P(O).$$

D ist der Variablentyp in PSIWAG. Die Elemente von D sind Mengen von Bildobjekten.

Def.: Eine PSIWAG - Prozedur p ist gekennzeichnet durch:

$$p : D^+ \rightarrow D^+$$

Aus der Definition für Prozeduren ergibt sich die einfachste Form einer Anweisung, die sich als eine Regel darstellt. Sei E_1, \dots, E_n die Menge der Eingabevariablen und A_1, \dots, A_m die Menge der Ausgabevariablen und P ein Prozedurname, so ergibt sich folgende Form:

$$A_1 | \dots | A_m :- E_1 * \dots * E_n > P.$$

Standardoperationen

Zum Aufbau einfacher Ausdrücke stehen drei Standardoperationen zur Verfügung. Diese beziehen sich auf die Ausdehnung der Objekte (im mengentheoretischen Sinn). Seien A, A_1, A_2 Ausdrücke: