

UNIX 環境下でのコンパクトな画像処理支援ツール IV-Box

7K-1

加藤二郎* 坂上勝彦† 横矢直和† 山本和彦† 松原 仁† 山崎祐一†

*三菱化成 †電子技術総合研究所 ‡久保田鉄工

1. はじめに

最近、UNIX¹の持つ利点とその周辺をサポートするソフトウェアの充実により、画像処理実験環境における UNIX ワークステーションの比重が増してきている。そこで今回それらの既存ソフトウェアとの併用が可能なコンパクトな画像処理ソフトウェアツール (IV-Box: Image and Vision toolBox) を開発したので、ここに報告する。なおこのツールは、画像処理アルゴリズムの開発者及び画像処理モジュールの利用者を対象として、新規のアルゴリズム開発を支援するためのデータ管理・表示と履歴管理の基本機能を提供する。

2. 設計思想

UNIX 環境下の画像処理ソフトウェアツールはいくつか^{[1][2]}あるが、それに対し我々のツールの特徴は以下の通りである。

- ・アルゴリズム開発者に対し、アルゴリズム以外の部分 (画像ファイルのロード・セーブ、画像の表示、履歴管理等) をカバーする最小限の関数を提供する。この関数は、画像のデータ型によらず同じ関数が使えて、また詳細な指定をしなくても適切に機能する仕様とする。

- ・他のツールとの併用が可能な「開かれたソフトウェア環境」の実現を目指す。このために、ヘッダーファイルと画像データファイルを独立にした画像ファイル形式とする。

- ・特殊な装置・ソフトウェアなしで、つまりワークステーションの標準構成 (OS: UNIX, C 言語, XWindow²) だけで使用できるものとする。

3. 実現方法

上記の設計思想を満足させるために、画像ファイルの形式、画像データ型、IV-Box 関数、メモリーの動的管理、オートヒストリー機能を開発した。以下項目別に述べる。

3.1 画像ファイルの形式

IV-Box の画像ファイルは、表-1 に示すようにヘッダーファイルとデータファイルとヒストリーファイル (後述) の3つ別々のファイルから成る。ヘッダーファイルに

は画像のデータ型・サイズ等の画像属性が、データファイルには生のデータが入る。ヘッダーはSIDBA仕様^[3]のサブセットになっており、内容は表-2に示す。1画像 (フレーム) に対して、1ヘッダー、1データ (カラー画像と距離画像に対しては、それぞれRGB型とXYZ型を許す) を必ず対応させ、サブフレームという概念は使わない (SIDBAフォーマットに基づくSFDIファイル形式^[4]におけるフレーム=サブフレームの場合に相当する)。このヘッダーとデータの分離によってIV-Boxと他のツール (例えば、XVision^[2]等) との併用が容易になる。ヘッダーファイルは、文字型とする。明示的に書かない項目はIV-Boxの関数側でデフォルト処理をし、また通常のエディタでの作成、修正が容易であり、人間とのインターフェイスがよい。

(file-name)	内容
ヘッダーファイル (xxx.ivhdr)	画像のデータ型、サイズ等の属性
データファイル (xxx.ivimg)	画像データ
ヒストリーファイル (xxx.ivhst)	原画像からの処理履歴

表-1 IV-Box ファイル

ITEM	SYMBOL	DEFAULT
data name	dname	default
image data type	dtype	gray8
machine dependencies	mdepend	IEEE
frame size	fx,fy	512,512
gray level information		
assigned bits	nbit	8
effective bits	ebit	=nbit
left or right		
justification of ebit	lr	left
gray scale type	scale	intensity
statistics		
maximum value	max	
minimum value	mini	
mean	mean	
standard deviation	sd	
color map model	cmmapmodel	GRAY
date	date	
location	location	
comment	comment	

表-2 ヘッダー内容

¹UNIX は米国 AT&T 社の登録商標です。²XWindow System は MIT の商標です。

A compact image processing tool IV-Box in the UNIX environment

Jiro KATO* Katsuhiko SAKAUE† Naokazu YOKOYA† Kazuhiko YAMAMOTO† Hitoshi MATSUBARA† Yuichi YAMAZAKI†

*Mitsubishi Kasei Co. †Electrotechnical Laboratory ‡Kubota Ltd.

3.2 画像データ型

画像を画像データ型で区分する。画像データ型は、意味画像データ型と物理画像データ型(1pixelのデータ長)の組合せで定義する。例えば、濃淡画像で1byte/pixelのものを、gray8 という画像データ型とし、ラベル画像で2bytes/pixelのものを、label16とする。物理データ型としては、C言語で扱えるデータタイプはすべて許す。表-3にサポートする画像データ型を示す。画像はメモリー上ではC言語のストラクチャー(以下画像ストラクチャー)に展開される(表-4)が、この画像ストラクチャーの種類は画像データ型の種類だけある。これは、画像の操作を画像データ型単位で考えているからである。

意味	物理的サイズと表現形式	キーワード
1 濃淡	unsigned 8bits integer	*fx*fy gray8
2 濃淡	signed 16bits integer	*fx*fy gray16
3 濃淡	signed 32bits integer	*fx*fy gray32
4 濃淡	32bits real	*fx*fy grayf
5 濃淡	64bits real	*fx*fy grayd
6 カラー	unsigned 8bits integer	*fx*fy*3 color8
7 2値	unsigned 1bit integer	*fx*fy bin1
8 2値	unsigned 8bits integer	*fx*fy bin8
9 ラベル	signed 16bits integer	*fx*fy label16
10 ラベル	signed 32bits integer	*fx*fy label32
11 距離1	32bits real	*fx*fy range1
12 距離2	32bits real	*fx*fy*3 range2

fx: 画像の X 方向のサイズ
fy: 画像の Y 方向のサイズ

表-3 サポートする画像データ型

操作関数へのポインター配列
ヘッダー
カラーマップのポインター
画像データのポインター (画像データの先頭アドレス)
画像データの配列

表-4 画像ストラクチャーの構造

関数名	機能
IVload	IV-Box ファイルからメモリー上の画像ストラクチャーへ展開
IVdalloc	画像ストラクチャーのメモリー領域確保
IVstore	画像ストラクチャーから IV-Box ファイルに保存
IVdisplay	XWindow 環境での画像の表示
IVtyconv	画像データ型の変換
IVhst	ユーザーヒストリーの作成
free	メモリーの開放 (C 言語の標準関数)

表-5 IV-Box 関数一覧表

3.3 IV-Box 関数

ユーザーに対して画像データの操作とメモリー管理機能を提供する IV-Box 関数の一覧表を表-5に示す。IV-Box の関数は、引数に画像データのポインターを与えるだけで、その画像の属性に応じた動作をする。例えば、

表示関数 IVdisplay の場合、画像の x 方向と y 方向のサイズや1データの物理的大きさや機種によるデータ順を考慮して、適切な表示を行う。別な言い方をすれば、使用者がデータ型とか、サイズとかを意識しなくていいように、データ型によらず同じ関数、同じコーリングシーケンスになっている。この動作は、画像ストラクチャーの中にヘッダー情報とローデータと別に IV-Box の操作関数も持たせることと、その操作関数へのポインターを利用することにより実現している。

3.4 メモリーの動的管理

IV-Box 関数 IVdalloc において、C 言語の malloc 関数を用い、画像データに対して画像データ型とサイズに応じた領域の動的な獲得と開放を行う。これによって、画像処理プログラム実行時にメモリーの有効利用が計れる。

3.5 オートヒストリー機能

履歴管理は、画像に対するヒストリーファイルとユーザーに対するヒストリーファイルにより自動的に行なわれる。

画像に対するヒストリーファイルは、IV-Box 関数 IVstore により画像ファイルのセーブと同時に自動的に作成される。ここで言うヒストリーは、原画像からの処理手順(=コマンド行と日時)を順次積み上げて残していったもので、原画像からの処理履歴である。したがって、原画像さえあれば、最後のヒストリーファイルの実行によりすべての画像(途中の画像も)を再び作り出すことができる。

また、ユーザーに対するヒストリーも1つのファイルとして残るので、使用者の試行錯誤の過程も容易に確認することができる。

4. あとがき

画像処理支援ツール IV-Box の概要について述べた。XWindow 環境を利用した IV-Box 関数 IVdisplay の詳細については別途報告したい。今後は、既存ツールの接続を実現するとともに、従来の手法での実験や新手法の前処理、後処理がより効率よくできるように、画像処理の基本手法を IV-Box 関数に組み入れて充実させていく予定である。

謝辞 画像データ型は、東北大学工学部松山隆司助教授の画像データ型を参考にさせていただきました。

参考文献

- [1] 田村、佐藤：“画像ワークステーションのためのソフトウェア環境” 情報第 36 回全国大会,4V-6,1988
- [2] Dr.John Rasure(University of New Mexico),et al.：“XVision File Format” Aug 5,1987
- [3] 尾上他：“イメージプロセッシングの振興と標準化” 情報処理、Vol.21,No.6,pp.645-659,june1980
- [4] 田村、坂上、横矢：“画像処理実験支援のためのデータファイル管理ソフトウェア” 情報処理, CV24-5, 1983