

# オープンソースソフトウェアの活用 －LinuxによるJava画像処理システム－

関東職業能力開発大学校 加部通明

An Application of Open Source Software  
－Java Image Processing System on Linux－

Michiaki KABE

**要約** 昨今、パソコンのクライアントOSの95%を占めるマイクロソフトのWindowsに對抗して、LinuxをクライアントOSに採用する動きがLinspireやHPなど一部の企業により徐々に実現されてきた。すでに、サーバOSではその流れが国や自治体およびヨーロッパにおいて経費節減や独自システムの構築という観点から、近々30%位の占有率を確保する見通しである。Linuxを初めとするソースコードが公開されている所謂オープンソースソフトウェア(Open Source Software 以下OSSと略記する)と呼ばれるソフトウェアは、自由に使用できて開発費や運用費をかなり削減でき、今後のシステム開発の活用において大いに注目されている。

従来、画像処理システムはパソコンOSの一極集中性とドライバの関係からWindows上でCやVCと言ったC言語系又はVBによって開発されることが多かった。そうした中で、世の中の流れを先取りすべくOSSを活用して画像処理システムを構築することは、応用課程の開発課題実習において企業から提示された課題を開発する場合等に、まさしく打ってつけな予算削減手段であり有意義であると思われる所以、本稿をOSS活用の一例として報告する。

## I はじめに

近年「情報格差」と言う言葉が差別用語の一種としてしばしばマスコミに登場している。その意味は国や地域における情報の利用を上手く享受できているか否かを表した言葉で、経済格差がそっくりそのまま情報にも当て嵌まり、地域間格差を新たに生む火種にもなっている。余り気が付かない簡単な例として、有償のソフトウェアについて取り上げてみよう。

マイクロソフトのパソコンOSやその上で稼働する様々なソフトは1万円前後する。それを貧しい国でも同じ価格で売れば、当然買う人は一部に限られその国の大部分の人達は現在のインターネットを初めとする情報技術の恩恵を受けることができない。ハードウェアはいろいろと中古があり、自動車と同じで部品を取り替えたりして改良すれば、ある程度の能力をもった

パソコンを安く手に入れることができるが、ソフトウェアはそうはいかない。ハード一つに対して必ず新しいソフトを先進国と同じ値段で購入しなければならないからである。このようなところから、貧しい国では未だに海賊版と言われる違法コピーが市場に出回っているのである。

こうした情報格差を解消する一つの手段として、OSSの活用が上げられる。無償だから全く価格の上からは問題ない。ではこのありがたいOSSが何故余り利用されていないのか、その理由を考えてみよう。それらを列挙すると次のようになる。

- ① どのようなOSSがあるのか分からぬ。
- ② OSSの利用方法が分からぬ。
- ③ 日本語処理対応がなされてないか、又は、不十分であって使いづらい。

今回、OSSの活用例を特にJNI(Java Native

Interface) に焦点を当てて説明し、更に、全て OSS を利用した画像処理システムの構築について報告する。

## II OSSについて

まず初めに、OSSの大まかな定義をここで確認しておこう。定義の詳細は参考文献(1)で公開されている。そのサイトではOSSに関する歴史や背景、更にOSSの見つけたなども説明している。

定義：OSSとは、次のような各項目についてある配布条件を満たすソフトウェアである<sup>(注1)</sup>。

- ① 自由な再配布
  - ② ソースコードの開示
  - ③ 派生ソースウェアの配布の許可
  - ④ 作者のソースコードの完全性
  - ⑤ 個人やグループに対する差別の禁止
  - ⑥ 利用する分野への制限の禁止
  - ⑦ ライセンスの分配
  - ⑧ 特定製品でのみ有効なライセンスの禁止
  - ⑨ 他ソフトウェアを制限するライセンスの禁止
  - ⑩ ライセンスは技術中立でなければならない

ここで、あるWebサイト<sup>(2)</sup>が把握しているOSS情報からその現状と数量的推移を抜粋してみよう。2003年12月中旬と2004年11月中旬の11ヶ月間におけるOSS推移状況を表1に示す。これよりOSSが漸次増加傾向にあることと、計画中のものや試作品に類するものが数多く存在することが分かる。

表 1 OSS の開発状況とその増減

	08/12 中旬	04/11 中旬	増減
計画中	13,783	14,462	+679
プレα	9,784	10,976	+1,192
α	9,239	11,042	+1,803
β	11,277	14,073	+2,796
製品安定	8,995	11,676	+2,681
成熟	852	1,052	+200
休止	376	922	+546

### III 画像処理システム

今日、画像処理はパソコンの演算能力の飛躍的な進歩によりいろいろな分野で応用されている。例えば、工場の生産現場では、従来人間が行っていた目視によ

る製品検査や欠陥検査等をコンピュータで自動的に行うようになって来ている。これらの検査は、従来ある単位で部品を抜き取りそれを検査してその単位での部品の良否を判定していたが、近頃では、小さな部品のビスでも全数検査という大変厳しい要求になっている。人間の目視では、検査する人の作業能力や時間の経過による検査精度の低下という不確実な要素を内包しているので、こうした検査をより確実に継続的に行うような自動検査システムの開発要求が従来からあった。

特にものづくりの支援を標榜する能力開発施設では、パソコンの周辺機器として機械制御や画像入力等を利用する場合が多い。以前はそれらをCプログラムで制御してきたが、今後はWebや通信、ビジネス関係等広範囲に対応できる汎用性のあるJavaプログラムを使用する機会が益々増えることが予想される。そうした状況の中で、画像処理システムをJavaで構築することは時代の流れに沿ったものと考えられる。

1 機能

表2に示したOSSを活用して表3に示すような画像処理機能を持ったシステムを構築した。

表2 利用するOSS.bmp

O S S	機能
Red Hat Linux 9	パソコン O S
Video for Linux two	ビデオキャプチャ用
bttv.o	ビデオキャプチャボード用ドライバ
C	A P I 使用言語
Java	画像処理言語

表 3 处理機能

メニュー	処理項目	別記事項
主：会員登録	ファイル	選択したファイルを記述を入力する
	セイタク（属性用）	Jetリスト表示する
	セイタク（一覧表示）	セイタクの基礎情報をより詳細な表示する
	学習データ（学習用）	日本で園庭土にある学習データをクリップで入力する
	種類データ（学習用）	日本で園庭土にある種類データをクリップで入力する
	属性カスタム	入力したファイルを属性画面表示する
植物標識 (フィルタ)	名前	園庭地の適性ヒストограмを参考にして標識を入力する
	輪郭推定	輪郭推定 Lagine
	露水推定	
	葉平均	
	葉標準	
	平滑化	
被写体 (フィルタ)	粗粒化	光强度分布図上の高さを標準表示する
	被写体密度	
	先粗粒化	
	輪郭推定 Sobel	
	輪郭推定 Edge	
	種類（2種類用）	セイタクの適性並びにデータから七種選択する
被写体分析	被写体	セイタクを通過園庭土を基準に標識ヒストограмを作成する
	被写体変換	標引変換により被写体形状を正確に検出する
	目視判別	
被写体判別	セイタクレジ	三種標識の標識をセイタクレジする
	コントラスト閾値	濃度フィルターを実行した標識を表示する
	輪郭は輪郭高	セーリングされた輪郭の輪郭高を算出する
	輪郭（輪郭）	輪郭からその種類の輪郭を算出する
パターン 認識	平滑	平滑データを強調した結果表示する
	輪郭	輪郭データを平滑に取り付ける

## 2 ハードウェア構成

職業能力開発大学校応用課程の授業で使用しているCCDカメラと実勢価格約1万円という安価なビデオキャプチャを用いて<sup>(注2)</sup>、図1に示すような画像処理のハードウェアを構成した。このシステムでは、カメラから撮像した映像は直接パソコン上に表示されるので、特別なモニタは必要ない。

## 3 ソフトウェア構成

ソフトウェア構成を図2に示す。今回使用したソフトウェアは全てOSSを利用しているのでソフトウェアに関しては経費ゼロである。今後のシステム導入では、ハードウェア経費10に対してソフトウェア経費3と言う現状の高い割合をOSSの利活用で究極的には0に近い比率まで持っていくことが理想である。厳しい予算を余儀されることが予想される中で、各人が具体的な行動や対策として取ることができるのがOSSの利活用と考えられる。

## 4 画面設計

表3に示したようにメニューには、入力、前処理(フィルタ)、基本処理、パターン認識をそれぞれ設けた。このパターン認識はニューラルネットワークを利用し、対象物が特定されないあらゆる画像に対応できる意味で、究極の特徴抽出と考えられる濃淡画像の全画面を抽出した。これより、抽出画素数とメモリとの制約から150×150画素程度の画像を3種類しか分類することができない<sup>(注3)</sup>。

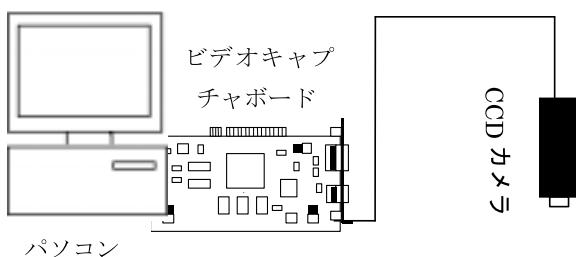


図1 ハードウェア構成

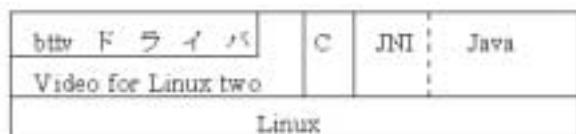


図2 ソフトウェア構成

## IV OSSの活用方法

ここでは、III章で説明した画像処理システムを取り上げて、ビデオキャプチャ<sup>(注4)</sup>選定から画像処理プログラムの作成と実行までのOSS活用方法を説明する。

### 1 Video for Linux two

Video for Linux two<sup>(3)</sup>(以下v4l2で表記する)は、ビデオキャプチャ用のドライバとカーネルとの情報の橋渡しをするモジュールvideodev.oと画像取得用プログラムを集めたAPIから構成されている。そして、それはLinux上でビデオキャプチャがカメラより画像を取得するために、そのAPIの関数をCプログラムに組み込んで利用される。

具体的には、v4l2の開発サイトからダウンロードして得られた圧縮ファイルを解凍し、その後makeしてモジュールvideodev.oを作成しそれを利用する。モジュール以外の全般的な利用方法を図3に示す。

なお、Linuxには既にカーネル2.2.x乃至2.4.xにおいて標準で最初からvideodev.oが組み込まれているので、それを利用してもよい。



図3 v4l2の利用方法

### 2 ビデオキャプチャの選定

画像処理システムを構築する上で一番厄介なのがビデオキャプチャの選定である。現在、国内ではv4l2用に開発されたドライバで稼働できるビデオキャプチャはほとんどが外国製であって、信頼性に幾分不安がある。以前販売されていたv4l2対応の画像処理チップBt848<sup>(注5)</sup>を使用していたアイ・オー・データ製のビデオキャプチャボードGV-VCP2/PCIが2002年に製造中止になり、その後継製品がv4l2に対応していないのでそれに代わる廉価な製品を探すのが難しい。

### 3 言語

C言語をv4l2を呼び出す制御用プログラムで使用し、Java言語をJNIを利用してその制御用プログラムを呼び出したり、GUIに優れていることから画像処理を行うプログラムで使用する。

## 4 画像処理プログラム実行方法

全体の流れを図4に示し、以下細部を説明する。

### 4.1 JNIを活用したv4l2による画像取得

JNI<sup>(4)</sup>はJavaと他の言語とを結び付ける道具である。Javaの短所にはドライバを直接操作することができない点がある。一般的に言うと、最近ではドライバのソースプログラムはC言語で書かれていることが多い、そのドライバを利用してプログラムを作成する場合にもC言語が適切である。今回利用したビデオキャプチャボードのドライバもC言語で使用する。そこで、JavaからCプログラムを呼び出すのがJNIの役割である。今までJNIを取り上げた例が少ないので、その具体的な利用方法を、今回の画像処理システムのカメラ撮像を例にして、ここで順番を追って説明する。

(1) 先ず、画像処理を行うメインのプログラムImageProv4l.javaを作成する。その中で、カメラ撮像を行う手続GetImagev4l()<sup>(注6)</sup>をオブジェクトimagev4lを用いて呼び出す文章を記述する。

```
imagev4l.GetImagev4l();
```

(2) (1)で作成したプログラムImageProv4l.javaをコンパイルする。

```
$javac ImageProv4l.java
```

(3) コンパイル後のクラスファイルImageProv4l.classを利用して、Cプログラムgetcamerav4l.cが使用するJNI用ヘッダファイルImageProv4l.hを作成する。

```
$javah ImageProv4l
```

(4) v4l2のAPIであるプログラムvlib.cとilib.cをmakeコマンドによりコンパイルして、目的プログラムvlib.oとilib.oをそれぞれ作成する。

(5) 画像取得用プログラムgetcamerav4l.cを作成して、makeコマンドによりコンパイルし目的プログラムgetcamerav4l.oを作成する。

(6) 3つの目的プログラムgetcamerav4l.o、vlib.o、ilib.oとをmakeコマンドで結合して、共有ライブラリlibgetcamerav4l.soを作成する。

(7) 以上全てエラーなく終了した場合、videodev.oとドライバbtv.oをモジュールに組み込み、その後、画像表示プログラムxawtvを実行しビデオキャプチャボードおよびドライバが正常に動作しているかを確認する。

```
#modprobe btv
```

```
#xawtv
```

エラーがある場合、途中にエラーメッセージが出力されていると思うので、それを参照し再度ドライバ等を組み込む。

(8) ビデオキャプチャボードおよびドライバの正常動作を確認した後、xawtvを終了させて画像処理プログラムImageProv4lを実行する。

```
$java ImageProv4l
```

以上の説明で使用したMakefileの内容を図5に示す。

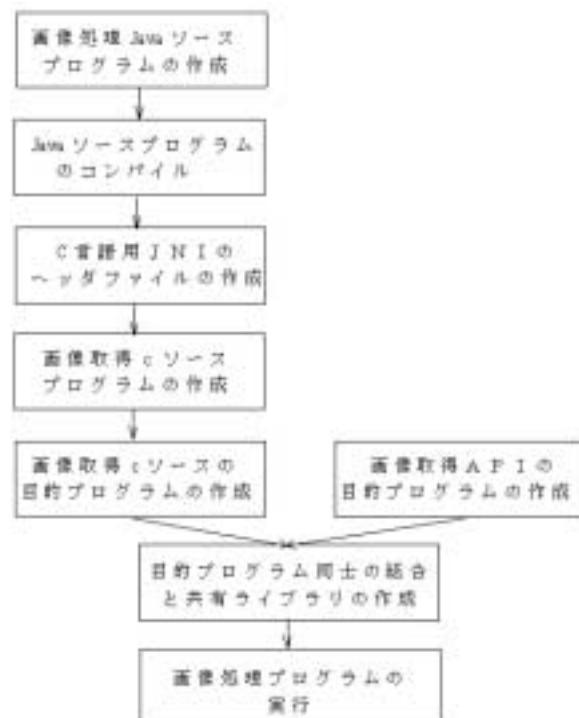


図4 プログラム作成と実行

```

#video4linux
CFLAGS=-Wall -O2 `imlib-config --cflags`
LFLAOS=`imlib-config --libs`
OBJECTS=libc.o libv.o
C2 = -shared
J1 = -I/usr/local/j2sdcl4/include
J2 = -I/usr/local/j2sdcl4/include/linux
#make shared library by linking object files
$(B)
        gcc -c $(B) $(C2) $(OBJECTS) $(E) $(LFLAOS)
#make object file
C2:
        gcc -c $(J1) $(J2) $(CFLAGS) $(E)
clean:
        rm -f $(OBJECTS) $(B).
  
```

図5 v4l2関連のMakefile

### 4.2 v4l2について

カメラ撮像をパソコン画面に出力するv4l2の関数にipdisp()があるが、これを2回以上使用すると、図6に示すエラーメッセージが出てアボードしてしまうので、1回のみの使用に止める。もし、カメラ撮像を連

続的にパソコン画面に出力させたい場合には、ipdisp()を使用せずに、v4l2の関数v4lread()を使用して画像情報を取得し、それを画素単位に配列したオブジェクト渡しを利用して、Javaの画像処理プログラムの方に渡す。その後、Javaの画像処理プログラムでComponentクラスのrepaint()命令を使用して取得した画像を画面に表示すればよい。v4l2を利用して画像取得するCプログラムを図7に示す。

また、カメラ撮像した画像をv4l2及びJavaで表示した様子をそれぞれ図8、図9に示す。

An unexpected exception has been detected in native code outside the VM  
Unexpected Signal : 11 occurred at PC=0b4CAC4E2  
Function=0xF9FreeGC+0xD1  
Library=/usr/X11R6/lib/libX11.so.6

図6 エラーメッセージ

図7 v4l2APIとJNIを活用したCプログラム

## V J N I 活用の他事例

## 1 應用課程

職業能力開発大学校生産情報システム技術科の標準課題実習の1つに「計測制御システム構築課題実習」がある。その中で画像処理を行う部分があり、そこでは日立製の画像処理ボードIP5005とそのボード専用に開発された画像処理ライブラリIP5000/CDを備えたC言語が使用されている<sup>(5)</sup>。IP5000/CDには、画像処理に関する数多くの処理が関数の形で含まれおり、その個数は凡そ160種類ある。IP5005は当初Windows NT用に開発されたものであり、それを LASER5



図8 v4l2による画像表示



図9 Javaによる画像表示と濃度ヒストグラム

Linux 上でも使用できるようにあとから改良された。各種配布版Linuxは毎年いろいろとバージョンアップされているが、画像処理ボードのLinux版は当初開発されたままであるから、その時のLinux版でしか動かない不便さがある。更に、画像処理ボード自体が高価であるから、思わぬ故障による修理代も高価になりそうであり、耐久性を考えると、もっと廉価な画像処理ボード乃至はビデオキャプチャの使用が今後望まれる。因みに廉価なものはIP5005の1/40程度の価格である。

こうした状況において、今回紹介した画像処理システムでは、ビデオキャプチャ以外経費がかかるないので、廉価なシステム構築に打って付けである。

そこで、既存の日立製画像処理ポートを利用したJava画像処理システムの構築方法について説明する。

## 1.1 現状と変更

標準課題実習や画像計測実習で使用されている画像処理システム部の構成は図10に示す通りである。

## [システム仕様]

OS : LASER5 Linux 6.2

画像処理ボード：日立製IP5005

## 画像処理ソフトウェアライブラリ：IP5000/CD

パソコン、画像表示専用モニタ、CCDカメラ

次にIP5000/CDを用いた現状のC画像処理プログラムで作成した処理の流れを説明する。

[現状]

- (1) CCDカメラから画像を取得し、それをパソコンとは別のモニタに表示する<sup>(注7)</sup>。
  - (2) 取得した画像に対して、IP5000/CDの画像処理ライブラリを用いてCプログラムにより画像処理を行う。

上の操作と同じ流れで画像処理をJavaプログラムで変更した場合の処理の流れを説明しよう。

[变更]

- (1) C C D カメラから画像を取得するCプログラムをIP5000/CDライブラリとJ N Iを利用して作成し、取得した画像をモニタとパソコン画面両方に表示する。
  - (2) Javaプログラムで表示された画像に対してJavaプログラムで画像処理を行う。

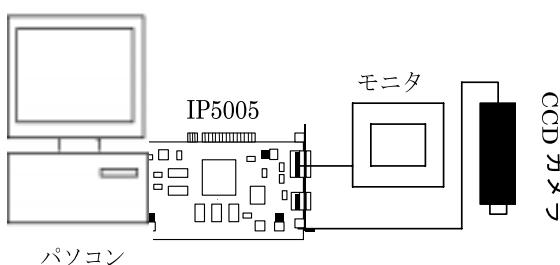


図10 ハードウェアシステム構成

## 1.2 JN1を活用したIP5000/CDによる画像取得

カメラで撮像した画像をJavaプログラムで処理するために、v4l2と同様にIP5000/CDによる画像取得データを画素単位で配列にしたオブジェクト渡しを行う。その処理を行うCプログラムとそのコンパイル方法を記述したMakefileの例をそれぞれ図11、図12に示し、更に取得した画像をJavaプログラムで画面に表示した様子を図13に示す。

2 在職者訓練

工場の生産工程で製品や部品を自動的に検査する場合、一つの方法として、画像処理の利用が考えられる。

図11 IP5000/CD とJ NIを活用したCプログラム

図12 IP5000/CD関連のMakefile



図13 IP5005によるJavaプログラムの画像表示

そこで、現在容易に手に入る比較的廉価なインターフェース製PCI-5520<sup>(6)</sup>なる画像キャプチャボードを使用した画像処理システムの構築を考えよう。

## 2.1 提案するシステム概要

v4l2で説明したものと同様なシステムで、次のような仕様となっている。

[システム仕様]

OS : LASER5 Linux 6.2

## 画像キャプチャボード：インターフェース製PCI-5520

ドライバ: dp0100.o、cp5220.o<sup>(注8)</sup>

画像取得ライブラリ:gpg5520

## パソコン、CCDカメラ

PCI-5520で使用するソフトウェアや取扱説明書は製造会社の説明によると、環境保全、ペーパレス推進のため媒体での提供を行わずに、インターフェース社のWebサイト(6)からID登録した後ダウンロードして取得する。次に、それを解凍して展開されたディレクトリやファイルに納められた説明書を良く読んで、ドライバのインストールやデバイスの初期設定、画像取得用プログラムの作成方法等を習得する。PCI-5520が備えるライブラリには、画像取得のみに関する関数が32種類用意されている。画像処理そのものについては、グラフィック処理機能を持つOSSのgtkを使用しCプログラムにより行うことを想定している。また、そのサンプルプログラムが用意されている。

## 2.2 JNIを活用したgpg5520による画像取得

PCI-5520の付属ライブラリlibgpg5520.soにある専用関数を用いて画像取得するCプログラムをJNIを活用して作成する。更に、Java画像処理プログラムで画像取得用Cプログラムの制御と画像処理を行う。

J N I を活用した画像取得用のCプログラムとその Makefile をそれぞれ図14、図15で示し、併せて画像取得した画像表示の様子を図16に示す<sup>(注9)</sup>。

VI 終わりに

OSSの活用例として、画像入力ボードのドライバを制御するCプログラムをJava プログラムから呼び出す方法であるJNIを中心にして説明した。

また、OSSの活用を促進する意味から標準課題実習で既に構築されている画像処理システムを、一部有償ソフトウェアを利用しLinuxとJavaを使用して再構築を試みた。更に、現在は手に入りにくいv4l2を利用するビデオキャプチャに代わるものとして、インターフェース製画像キャプチャボードを使用してOSSを活用するよう提案した。

今後は、画像処理システムを実用的なシステム、例えば開発課題実習での欠陥品や製品等の検査システム

図14 gpg5520とJNIを活用したCプログラム

図15 gpg関連のMakefile

に組み込み、昨今の厳しい経費削減に対応できるよう  
に更なる活用を考えたい。



図16 PCI-5520によるJavaプログラムの画像表示

経費削減の具体的な対策の一つとして、本稿では、ビデオキャプチャの使用を前提として画像処理システムの構築を考えた。しかし、今後はそれを必要としないUSBカメラによる画像取得を行えば、それこそ数千円のカメラ代のみの経費でシステム構築が出来てしまう。USBカメラによる撮像の精度と動画の取り込みが出来るかと言う課題が解決されれば、後はv4l2等のOSSにその製品が対応しているか否かが問題となるだけである。

### [注]

- (注1) ここで述べる定義はあくまでもOSSの推進を唱える団体によるものであって、ライセンスに係わる項目等は今後の社会情勢で変わる可能性がある。
- (注2) IV章1で説明するが、当初想定していたビデオキャプチャが製造中止になったので、現在では、中古店又はオークション等でしか手に入らない。
- (注3) 3層型バックプロパゲーションのニューラルネットワークを利用して、本稿のような手法でパターン認識を行う場合、プログラムの大部分を占める入力層と中間層間の結合係数のメモリ量が問題になる。今、記号M、N、T、wを以下のような意味とする。  
M：パソコンのメモリリソースで約50メガバイト。  
N：検査画像の一辺の画素数。(N×Nが画像の大きさであり、また入力層のユニット数ともなる。)  
T：中間層のユニット数。  
w：入力層と中間層間の結合係数で、具体的にはw[T][N][N]なるfloat型配列。  
Javaではfloatは4バイトだから、配列wのメモリ量 $4 \times T \times N \times N$ はM未満となる。これを目安にパソコンで実験調査すると、およそN=150となった。
- (注4) 本稿では、ビデオキャプチャ製品名を製造元が使用している呼称を用いて表記している。

GV-VCP 2 /PCI：ビデオキャプチャボード

IP5005：画像処理ポート

PCI-5520：画像キャプチャボード

(注5) Bt848のドライバがbtv.oであり、それはvideodev.oを通してカーネルと情報のやり取りを行っている。従って、IV章4.1(7)で説明しているように、modprobe btvを行うことによって、videodev.oとbtv.oの両方がモジュール組み込みされる。

(注6) Cプログラムのファイル名はgetcamerav4l.cであるが、その共有ライブラリのファイル名はlibgetcamerav4l.soとなる。Javaプログラムでその共有ライブラリを定義するときには

```
public static native byte [][] GetImagev4l();
static{
```

```
    System.loadLibrary("getcamerav4l");
```

```
}
```

として共有ライブラリ名をgetcamerav4lで表記する。実際にそのプログラムを呼び出す場合には、getcamerav4l.cの名前と関係ない名前が使えGetImagev4l()とした。

(注7) IP5005では、カメラ撮像した画像やそれを処理した画像をパソコン画面に表示することができないので、別のモニタに表示する。もし、パソコン画面に画像を表示したい場合には、今回提案したようなシステム構築をしなければならない。

(注8) dpg0100.oは各カーネルバージョンにおけるAPI仕様や内部情報の違いを吸収する為のモジュールであり、cp5520.oがドライバ本体である。それらは、ちょうどvideodev.oとbtv.oとの関係と同じである。

(注9) IP5005及びPCI-5520の画像取得解像度はそれぞれ $512 \times 440$ 、 $640 \times 480$ である。

### [参考文献]

- (1) <http://www.opensource.org/docs/definition.php>
- (2) [http://sourceforge.net/softwaremap/trove\\_list.php?form\\_cat=6](http://sourceforge.net/softwaremap/trove_list.php?form_cat=6)
- (3) <http://www.thedirks.org/v4l2/>
- (4) <http://java.sun.com/j2se/1.3/ja/docs/ja/guide/jni/>
- (5) [http://www.hitachi.co.jp/Div/omika/prdcts/h-ip/ip5000\\_0.htm](http://www.hitachi.co.jp/Div/omika/prdcts/h-ip/ip5000_0.htm)
- (6) <http://www.interface.co.jp/catalog/prdc.asp?name=pci-5520>