

## カラー画像の表示と解析に関する検討

百田正広<sup>\*1</sup> 山田健仁<sup>\*1</sup>

### Picture Display Using GTK+ and Tracking Object Using Color Information

Masahiro MOMOTA<sup>\*1</sup> and Takehito YAMADA<sup>\*1</sup>

#### Abstract

This paper proposes to display an image(16 division images) and a sequential images on personal computer (PC). This PC has the GTK+ system for graphical user interface(GUI) on Linux(Red Hat Linux) and a IEEE1394 image capture board (CCM-DS250 : SONY INC.). It was not high-speed software tool, it turns out that there is sufficient performance to display images. Moreover, based on the color information on the taken-in measurement object, since an object tracking was tried. This paper describes how the system can display color images and tracking object images on a GUI.

**Key Words :** GTK+, Sequential image, Color information, Tracking object, Linux

#### 1. まえがき

近年, CCD カメラ付きの携帯電話が発売されるなど, 画像情報が積極的に利用されるようになってきた. また, コンピュータ(以下 PC)の低価格化, 処理能力の向上を背景に, デジタル画像処理や 3 次元シーンの情報を復元することを目的としたコンピュータビジョンの研究が活発に行われ, ロボットビジョンなど多くの試みがなされている<sup>1)~3)</sup>. 我々も, 画像処理応用として, 固定磁気ディスク装置への非圧縮画像を大容量かつ高速に蓄積するシステムの可能性やインターネットカメラ制御などを検討してきた<sup>4)~6)</sup>.

一方, PC を利用する際, MacOS や Windows, UNIX 上での X といった GUI(Graphical User Interface)を利用することが多い. 最近では, GUI の構築も容易になり, なかでも GTK+(Gimp Tool Kit Plus)はウィンドウひとつだけの簡単なプログラムから, 大規模なアプリケーションプログラムまで容易に組むことが可能なフリー

なツールである. また, LINUX では多くのディストリビューションに標準で含まれており, OS インストール時に容易に導入可能である. これらのことから, 多くの参考書<sup>7)</sup> が出版されている.

一方, 本論文でも一部使用する HALCON<sup>8)</sup>をはじめとする, GUI を搭載した多くの画像処理ソフトが市販されている. しかし, 連続取り込みや追跡が可能なソフトは高価なものも多く, 特殊な用途で利用されている. このような背景のもと, 本論文では, デジタルカメラから取り込んだカラー画像を, フリーソフトである GTK+を用いて表示するアプリケーションソフトの開発を試みた. また取り込んだ測定対象物のカラー情報をもとに, 対象物の追跡を試みたので報告する.

#### 2. 画像取り込みと表示

画像の取り込みには, IEEE1394 デジタルカメラ (SONY:型番 COM-DS250)を使用した. IEEE1394 は高速の

<sup>\*1</sup> 情報電子工学科

シリアルバスで、転送速度 400Mbit/s で動作する。この IEEE1394 は、接続ポートを持つデバイス同士で接続可能であり、同一バス上に複数の PC を接続することも可能である。このカメラで取り込まれた画像は、効率の良い伝送や圧縮を実現している YUV フォーマット形式である。画像データの取り扱いを容易にするため、YUV フォーマットから RGB フォーマットに変換した。

### 3. GUI の開発

本システムでは GTK+ という GUI を用いたが、アプリケーション(API)との橋渡しには GDK や GLIB が必要であり、OS インストール時に同時に導入されている。今回作成したプログラムでは、ウインドウウィジェット、垂直パッキングボックス、水平パッキングボックス、ボタンウィジェットなどを組み合わせて GUI を構成した。次に、GTK+ のウィジェットとして、画像を表示可能にするために Imlib を用いた。この Imlib を利用すると、様々なフォーマットの画像ファイルを読み込むことが可能となる。また、画像サイズの変更や回転といった操作も可能となる。画像の表示には、画像が変更不可の方法と、変更可能な方法の異なる 2通りの方法を用いた。前者の方法は、16 分割した静止画像などに向いており、表示は容易にできる。後者の方

法は、連続画像を表示する場合に向いており、手順がやや複雑となる。

#### 3.1 分割表示

16 枚の連続画像を表示する手順について述べる。

1. ウィンドウを作成する。
2. 4×4の行列を持つパッキングテーブルを作成し配置する。
3. Imlib を用いて画像ファイルを読み込む。
4. GTK+ のウィジェットのピクスマップに変換し、パッキングテーブルに配置する。
5. gtk\_widget\_show 関数を用いて、ピクスマップとパッキングテーブルを表示可能にする。
6. ウィンドウを表示する。

図1に 16 分割画像を示す。

#### 3.2 連続画像再生

画像再生手順について簡単に述べる。

1. 画像データを読み込む。
2. GdkImlibImage 型の画像データに変換する。
3. ドローイングエリアサイズに、画像データをリサイズする。
4. GdkPixmap 型の画像データに変換する。
5. ドローイングエリアに貼り付ける。



図1 16 分割画像表示例

タイマー関数(`gtk_timeout_add`)を利用すると、一定時間毎に画像の更新が可能となる。また、画像のフレーム番号を減少させることで、逆再生も可能となる。

#### 4. 色情報の抽出

画像中のピンポン玉(オレンジ色)の、カラー情報を抽出することを試みた。色抽出のみ処理を容易にするため、HALCON(リンクスコーポレーション製)画像処理用ソフトウェアを用いた。ここでは、3.2 で示した方法により取り込んだ画像を対象とした。HALCONでは、カラー情報をRGB空間、HSI空間やHSV空間で処理が可能である。ここでは取り込んだカラー画像がRGB画像であることや後の処理を容易にするため、HALCONで得られたR、G、Bそれぞれの閾値からオレンジ色の抽出を試みた。レッド(R)の画素値は、 $130 < R < 255$ の範囲にあること。グリーン(G)の画素値は、 $0 < G < 220$ の範囲にあること。ブルー(B)の画素値は、 $0 < B < 95$ の範囲にあること。この3つを満足する範囲を、オレンジ色とし、求めるピンポン玉の範囲とした。図2にピンポン玉の検出例を示す。以下、連続画像中のオレンジ色の範囲は変化しないとの仮定のもと、これらの閾値をもとに解析を行った。

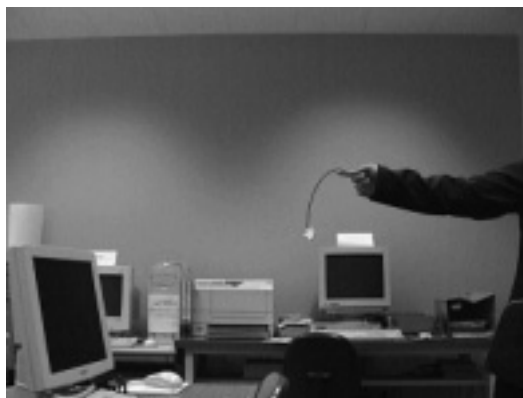


図2 ピンポン玉の検出結果

#### 5. 対象物体の中心座標の検出

カメラから取り込まれた画像は、PPM形式のファイルとして保存されている。PPM形式は、画像データの先頭にヘッダが付いている非圧縮のフォーマットである。このためデータ処理を行うには、ヘッダ部分を読み飛ばすことで可能となる。今回、RGBカラー画像処理を行うため、カラー画像データがどのように格納されているか調べておく必要がある。画像の高さをH画素、画像の幅をW画素としたとき、画像データは配列の先頭から順に、(0, 0)の点のR要素、(0, 0)の点のG要素、(0, 0)の点のB要素、(0, 1)の点のR要素、…、以下同様に(0, w)の点のR要素、…、最後のデータが(H, w)の点のB要素という順に格納されている。前節で求めたオレンジ色に関する情報をもとに、画像中のR、G、Bの値と比較することでピンポン玉の位置を検出することを試みた。検出された範囲において、上端と下端の座標、および左端と右端の座標の平均をとることで、ピンポン玉の中心座標とした。座標解析にはC言語<sup>9)</sup>を用いた。

図3に3枚の連続画像から、ピンポン玉の中心座標を求めた結果を示す。カラー画像を、白黒印刷しているため、見難くなっている。画像上で座標の原点(X, Y) = (0, 0)は、左上隅とする。

得られた座標位置を記憶させ、この中心座標を、直線で結ぶことで、物体軌跡を示した。結果を、図4に示す。白線の描画には、`gtk_draw_lines`関数を使用した。図に示すように、オレンジ色のピンポン玉の中心座標を追跡していることが判る。

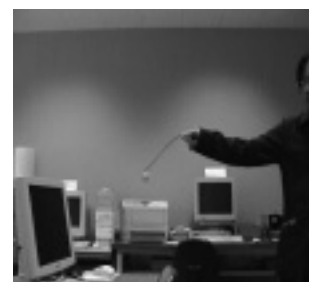
なお、`gtk_draw_lines`関数は、drawableウィジェットに線を描画するため、ウィンドウやドローイングエリアのサイズを変更すると、表示ずれが生じる。これを、防止するにはメモリ上の値を変化させて対応することが可能である。この点については、今後の課題とする。



(213,160)



(186,166)



(138,147)

図3 ピンポン玉の中心座標検出結果



図4 ピンポン玉の追跡結果表示

## 6. ユーザーインターフェース

Gtk+を使用してグラフィカルユーザインターフェース(GUI)を作成した(図5)。各ボタンに対してコールバック関数により動作を定義する。ウィンドウが閉じられたとき、アプリケーションは終了する。ウィンドウが閉じられるときに発生するシグナルは“destroy”である。このシグナルを受け取ったときに終了処理を行うようにコールバック関数を定義する。

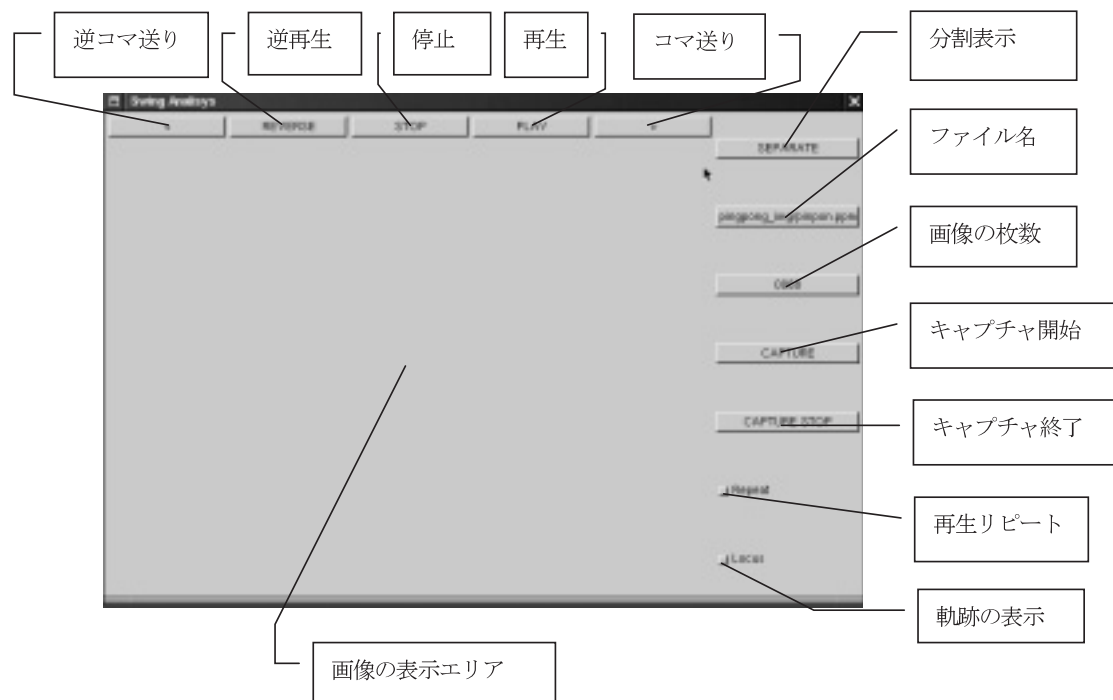


図5 作成したGUI

このGUIには水平パッキングボックスと垂直パッキングボックスを2つずつ使用している。まず水平パッキングボックスをウィンドウに配置し、そこに垂直パッキングボックスを2つパッキングする。そして、左側に水平パッキングボックスとドロワーリングエリアを配置する。この水平パッキングボックスに再生ボタンや停止ボタンをパッキングする。ドロワーリングエリアは640×480のサイズとして画像を表示する。右側には縦方向に分割表示やキャプチャを開始するボタンや設定を変更するボタンをパッキングする。

ファイル名と画像枚数の変更ではダイアログウィンドウを新たに作成し、そのテキストエントリに入力するようにする。最初はエントリウィジェットを直接パッキングしたが、1文字変更するごとにシグナルを受け取ってしまい、存在しないファイル名をしばしば参照するので、ダイアログウィンドウにしてボタンが押されるまで設定を変更しないようにした(図6)。

## 7. まとめ

フリーソフトであるGTK+を用いた、画像表示ソフトの開発とカラー情報を用いた対象物体追跡を試みた。今回用いたGTK+などは、高速なライブラリではないが、

動画像を表示させるには十分な可能性があることが判った。画像表示用 PC の OS としては Windows が主流であるが、Linux では多くのディストリビューションに標準で含まれており、OS インストール時に容易に導入可能なことを考慮すると、今後、広く使われる可能性をもったソフトウェアと考えられる。

画像取り込みにおいては、今回 IEEE1394 カメラを使用した。一般にこのカメラは、Windows 対応のドライバはメーカーが用意しているため、動作は安定している。今回、Linux 上での動作を試みたが、ドライバが十分に対応していないため、不安定な動作をすることがあった。今後、Linux 用ドライバの開発を行う必要があると思われる。今後の課題である。IEEE1394 カメラの設定については、付録に記載している。



図6 ファイル名入力ウィンドウ

### 謝辞

本研究に協力いただいた、平成13年度卒業生の田島英朗君に感謝の意を表します。

### 文献

- 1) 松本和雄ほか：画像概論 I, 丸善, 1994.
- 2) 松本隆司ほか：コンピュータビジョン, 新技術コミュニケーションズ, 1998.
- 3) 山田健仁ほか：動画像情報によるアーム型倒立振子の安定化制御, 徳山高専紀要, 第24号, pp. 7-14, 2000.
- 4) 百田正広ほか：固定磁気ディスク装置への画像取り込みに関する一考察, 徳山高専紀要, 第22号, pp. 9-13, 1998.
- 5) 百田正広ほか：WWWブラウザによるライブカメラ制御システムの試作, 高専教育, 第22号, pp. 107-113, 1999.
- 6) 長谷晃裕ほか：Linuxを用いた画像取り込み・表示に関する一検討, 徳山高専紀要, 第25号,

pp. 37-41, 2001.

- 7) 竹田英二：GTK+ではじめる X プログラミング, 技術評論社, 1999.
- 8) FEST Project 編集委員会：実践画像処理, シュプリンガー・フェアラーク東京, 2000
- 9) David A Curry：UNIX C プログラミング, アスキー出版局, 1993

### 付録

#### Linux の設定

Red Hat Linux のネットワークインストールには ftp サーバを利用して行った。今回は Windows95 用 FTP サーバ「Tiny FTP Daemon Ver5.2b」を使用した。

Red Hat Linux のカーネル再構築

```
cd /usr/src/linux
make xconfig
IEEE1394 Support をモジュールとして選択
OHCI Support
OHCI Video Support
Raw IEEE1394 I/O Support をモジュールとして選択.
make dep
make bzImage
make modules
make install
```

libraw1394 のインストール

```
$tar xzf libraw1394-0.9.0.tar.gz
$cd libraw1394-0.9.0
$./configure
$make
$make install
$make dev
$chmod a+rw /dev/raw1394
/etc/ld.so.conf に/usr/local/libを追加
$/sbin/ldconfig -v
```

libcam1394 のインストール (cam1394 というプログラムが /usr/local/bin/ にインストールされる)

```
$ tar xvfz libcam1394-0.1.1.tar.gz
$ cd libcam1394
$ ./configure
$ make
$ make install
$ /sbin/ldconfig -v
```

モジュールをロード  
\$ cd ieee1394-20010922  
\$ /sbin/insmod ieee1394.o  
\$ /sbin/insmod ohci1394.o  
\$ /sbin/insmod raw1394.o  
\$ /sbin/insmod ohci1394\_core.o  
\$ /sbin/insmod ohci1394\_isoTx.o  
\$ /sbin/insmod ohci1394\_isoRx.o  
\$ /sbin/insmod ohci1394\_fb.o  
\$ /sbin/insmod ohci1394\_low.o  
これで cam1394 を使用できるようになる。

カメラ ID の確認. int 型にキャストしていないので表示がおかしい.

\$cam1394 - I

カメラから画像送信を開始.

\$cam1394 - S

カメラからの映像を X Window 上に表示

\$cam1394 - D

カメラからの送信を終了

\$cam1394 - P

このドライバを使用して取り込みを行ったところ、約 25fps で取り込むことができた。また画面に表示しながらでも約 17fps のスピードで取り込みを行えた。ドライバやライブラリは

<http://limu.is.kyushu-u.ac.jp/~yosimoto/ieee1394/driver.html>

で公開されている。

(2002. 9. 5 受理)