

CONFERENCE DIGEST

ロボティクス・メカトロニクス講演会2010
2010 JSME Conference on Robotics and Mechatronics

ROBOMECH2010 in ASAHIKAWA

ロボティクス・メカトロニクス・フロンティア・ビッグバン
Robotics・Mechatronics・Frontier・Big-Bang

June 13 Sun. - 16 Wed., 2010

Asahikawa TAISETSU Arena

主催 社団法人 日本機械学会 ロボティクス・メカトロニクス部門
The Japan Society of Mechanical Engineers, Robotics and Mechatronics Division



OpenCV 2.0 を用いた映像処理システムのための RT コンポーネント群

RT-Components for Image Processing System using the OpenCV 2.0

金 湘宰 (デジタルクラフト) ○李 賢徳 (産総研) 劉 惠想 (デジタルクラフト)
栗原 眞二 (産総研) ビグズ ジェフ (産総研) 正 安藤 慶昭 (産総研)

Sangjae KIM, Digital Craft Inc., sangjae.kim@aist.go.jp

Hyundeok LEE, National Institute of Advanced Industrial Science and Technology

Hyesang YU, Digital Craft Inc.

Shinji KURIHARA, National Institute of Advanced Industrial Science and Technology

Geoffrey BIGGS, National Institute of Advanced Industrial Science and Technology

Noriaki ANDO, National Institute of Advanced Industrial Science and Technology

In this paper, image processing RT-Components(RTCs) suite which is based on newly released OpenCV 2.0 beta is implemented on OpenRTM-aist-1.0-RELEASE. For encapsulation and modularity of image processing RTCs, a new data type for image data is introduced. Several types of image processing needs some information feedback such as clicked position, color and region by mouse events. In order to modularise image processing and image showing, the mouse event data-port is introduced and some RTCs are implemented by using is. RTC design guidelines are shown and discussed.

Key Words: 画像処理, RT コンポーネント, Reusability

1. はじめに

画像処理システムはフィルタ構造をとることが多く、比較的モジュール化しやすいシステムである。画像処理に関する様々な処理を RT コンポーネント化する試みは、[1] や [2] においても行われてきた。OpenRTM-aist においても、バージョン 0.4 からカメラ画像取得・表示コンポーネントが提供されている [3]。[2] や [3] においては、データポート間で伝送されるデータとして、汎用的なデータ型に画素情報のみを格納したものを採用していたため、RT コンポーネント間での画像サイズやその他の情報の伝達に問題があった。

本稿では、柔軟な画像データ伝送および処理を実現するための新たなデータ型 CameraImage 型を導入することにより、これらの問題を解決する。OpenCV 2.0 を利用した RT コンポーネント群、および OpenCV で扱えないカメラのためのコンポーネント群を、新たなデータ型を利用して開発したこれらのコンポーネント群の設計方針についても議論する。

2. 映像処理ライブラリ OpenCV

OpenCV (Open Computer Vision) はオープンソースのコンピュータビジョンライブラリである。OpenCV では、画像に関する様々な処理を行うために必要なデータ型や API 群が提供され、容易に利用できるようになっている。代表的な機能として画像処理 (Image Processing)、構造解析 (Structural Analysis)、モーション解析と物体追跡 (Motion Analysis and Object Tracking)、パターン認識 (Pattern Recognition)、カメラキャリブレーション (Camera Calibration) 等がある。現在、2.0beta が公開され、Windows と Linux 等の OS 上で使用することができる。

2.1 OpenCV1.x から 2.0 への変更点

OpenCV 2.0 では、新たに C++ 言語のためのインタフェースが導入され、バージョン 1.x と比較して、少ないコーディング量で効率よく処理を記述することができる。メモリ管理等も自動化されたため、メモリリーク等のバグが入り込みにくくなった。また、ヘッダ構成なども改善され、環境構築がより容易になった。

物体検出アルゴリズムでは、Corner Detection の処理速度の向上、顔認識・眼球認識などの機能が追加された。また、API が追加され、任意の台数のカメラからの複数の画像処理が可能となるなど、ステレオカメラのサポートが強化された。

一方で、OpenCV 2.0 は現在ベータバージョンであり、今後、さらにバグの改修等が行われる可能性がある。Linux OS

上で、低価格の Logitech や Microsoft の JPEG Camera から画像を取得する場合、画像データの一部分が正常に取得できないなどの不具合も報告されている。

2.2 IplImage 構造体

OpenCV ライブラリでは画像のデータを処理するため、イメージ情報を格納出来る IplImage 構造体 (表 1) を提供している。この構造体を用いて、チャンネル数、画像サイズ、画素データなど、画像に関する汎用的なデータを扱うことができる。

Table 1 IplImage data structure

IplImage data structure	
int nSize	IplImage のデータサイズ
int nChannels	チャンネル数
int depth	ピクセルごとのデプス
int dataOrder	カラーチャンネル
int origin	原点位置
int width	画像の幅
int height	画像の高さ
int imageSize	画像データのバイト数
char *imageData	画像データへのポインタ
int widthStep	画像データの横 1 行ごとの幅
char *imageDataOrigin	画像データへのポインタ

3. RT コンポーネント設計方針

本節では、OpenCV 2.0 の各種画像処理アルゴリズムを OpenRTM-aist1.0-RELEASE 上で RT コンポーネント化する際の設計方針について議論する。

3.1 TimedOctetSeq データ型の問題点

OpenRTM-aist-0.4.2-RELEASE のサンプル RT コンポーネントとして公開された、OpenCV 1.0 を利用した USB-CameraAcquire・USBCameraMonitor コンポーネントでは、画像データ入出力のため、TimedOctetSeq データ型を利用した。

TimedOctetSeq データ型は、タイムスタンプ情報と、octet (CORBA の無変換バイト型) のシーケンス型 (CORBA の動的配列型) で構成されている。これは、バイト列をシーケンスに格納するだけの単純なデータ型であり、画像データのためのデータ型ではない。上記の TimedOctetSeq データ型を画像データの処理のために利用した場合は、データには画素データしか格納することができず、データのフォーマットや画像のサイズのような付属情報を伝送することができない。し

たがって、画像データを送受信するコンポーネント間で、予め画像サイズを合わせておく必要等があった。

3.2 CameraImage データ型の導入

上記の問題を解決するため、新たに CameraImage データ型を導入した。このデータ型は、OpenRTM-aist-1.0-RELEASE に付属する InterfaceDataType.idl において定義されている。

```
interface CameraImage
{
    Time tm;
    unsigned short width;
    unsigned short height;
    unsigned short bpp;
    string format;
    double fDiv;
    sequence<octet> pixels;
};
```

CameraImage データ型はサイズ情報、フォーマット情報、1画素あたりのビット数、ピクセルデータ情報などで構成されている。このデータ型を利用することにより、画像データの送信側から受信側に、必要な情報を伝達することができる。

3.3 画像サイズの動的変更への対応

TimedOctetSeq 型を用いて実装した前述の RT コンポーネントでは、画像サイズをコンフィギュレーションパラメータとして定義し、処理に必要なメモリ領域を確保していた。一方で、画像サイズの動的変更に対応可能な RT コンポーネントを実現するためには、入力された画像データから取得した画像サイズを利用して IplImage メモリ領域を確保する必要がある。

```
RTC::ReturnCode_t Flip::onExecute(RTC::UniqueId ec_id)
{
    // Get image from InPort
    ....
    // Check size of image and secure a memory
    if( image_previous.width != cameraImage.width ||
        image_previous.height != cameraImage.height)
    {
        // Get Infomation of image size
        image_previous.width = cameraImage.width;
        image_previous.height = cameraImage.height;

        // clear memory
        if(image != NULL)
        {
            cvReleaseImage(&image);
        }

        // secure a new memory
        image = cvCreateImage(
            cvSize(cameraImage.width, cameraImage.height),
            IPL_DEPTH_8U, nChannels);
    }
    // handle of data process
    ....
}
```

以前のコンポーネントでは、onActivate 関数でメモリの確保、onDeactivate 関数でメモリの解放を行っていたが、動的な画像サイズの変更に対応するためメモリの確保と解放を画像サイズが変化したときのみ onExecute 関数で行うものとする。

4. RT コンポーネント群

本節では、OpenCV 2.0 対応 RT コンポーネントの機能を紹介する。

4.1 コールバック関数対応モニタ RTC

物体追跡処理等でマウスやキー操作等の入力を伴う RT コンポーネントにおいては、マウスイベントやキーイベントを取り扱う必要がある。OpenCV では、こうしたイベントをコールバック関数により処理する枠組みが提供されている。コールバック関数を利用するためには、OpenCV 用の Viewer が活性化されている必要がある。しかし、Viewer のみ表示する

既存の RT コンポーネントは、コールバック関数による命令を処理することはできなかった。

本稿で新たに作成した RTC 群のひとつ USBCameraMonitor コンポーネントでは、図 1 に示すように、KeyEvent, x 座標, y 座標, および mouseEvent 等モニタ画面でのマウスイベントやキーイベントに関連するデータを OutPort から出力するようになっている。これ等のデータを必要とする ObjectTracking などのコンポーネントに対して情報をフィードバックすることが可能となり、処理部分と表示部分の分割とモジュール化を実現した。

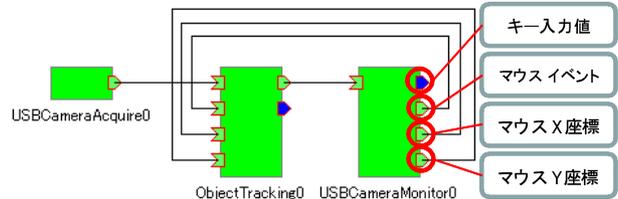


Fig.1 Callback DataPorts for USBCameraMonitor Component

4.2 PGR カメラコンポーネント

PointGray 社から発売されている IEEE1394 や USB 接続の各種カメラは、ロボット研究においても広く利用されているものの、そのままでは OpenCV 2.0 から利用することはできない。画像はベイヤーパターンとして取得されるため、これを RGB フォーマットに変換し、OutPort から出力する必要がある。OpenCV 2.0 対応 RT コンポーネント群の一つである PGRCameraCapture コンポーネントは、上記の機能を実装した RT コンポーネントである。

4.3 DirectShow 対応カメラキャプチャコンポーネント

Windows 環境の OpenCV ライブラリでは、IEEE1394 デバイスから映像を取得する機能を提供していない。DirectShow (マイクロソフト社が提供するマルチメディアフレームワーク API) を利用することで、IEEE1394 接続のカメラから画像を取得することができる。本 RT コンポーネントでは、IEEE1394 デバイスからカメラデバイスを検索し、Direct Show の Capture Filter の仕組みを利用することにより、画像取得を実現している。

4.4 他の画像処理コンポーネント

本節では、実装した各 RT コンポーネントについて説明する。上述の設計方針に従って実装した RT コンポーネント群の略称と機能を表 2 に示す。

画像の入出力に関するコンポーネントとしては、図 2 に示す USBCameraAcquire, USBCameraMonitor (コールバック関数機能付き), および PGRCameraCapture, IEEE1394Camera コンポーネントがある。



Fig.2 USBCameraAcquire and USBCameraMonitor Components

画像データの位置や座標の制御に関する RT コンポーネント群としては、Perspective (図 3), Rotate (図 3), Affine,

Table 2 Image Processing RT-Components

Image Processing RT-Components	
Affine	入力映像を指定した行列で変換
Binarization	入力映像を 2 値化白黒に変換
Chromakey	特定色領域を除去, ある映像にすり替える
DilationErosion	バイナリイメージの膨張/収縮
Edge	エッジの抽出
Findcontour	輪郭の検出
Filp	映像の上下左右回転
Histogram	色相情報の検出
Houghline	直線抽出
ImageCalibration	外部/内部パラメータ取得
ImageSubtraction	画像比較
ObjectTracking	物体追跡
Perspective	遠近投影
PGRCameraCapture	PGR カメラキャプチャ
RockPaperScissors	物体認識
Rotate	映像角度回転
Sepia	暗褐色変換
Template	パターン認識
Translate	映像の平行移動
USBCameraAcquire	USB カメラキャプチャ
USBCameraMonitor	映像の出力
IEEE1394Camera	IEEE1394 カメラキャプチャ



Fig.6 ObjectTracking Component



Fig.7 ImageSubtraction Component

Flip, Translate がある。

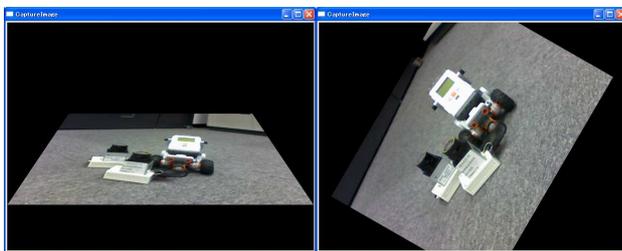


Fig.3 Perspective(Left) and Rotate(Right) Components

画像データの画素情報の処理に関する RT コンポーネント群としては, Edge (図 4), Binarization (図 4), Chromakey, DilationErosion, Findcontour, Histogram, Houghline, Sepia を実装した。

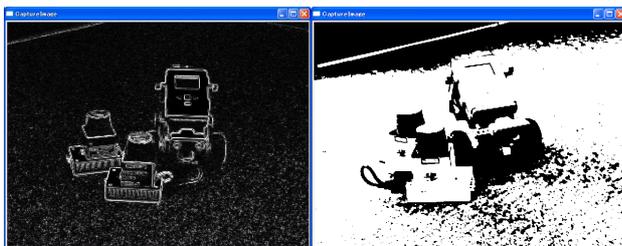


Fig.4 Edge(Left) and Binarization(Right) Components

動的画像処理に関する RT コンポーネント群としては, RockPaperScissors(図 5), ObjectTracking(図 6), ImageSubtraction(図 7) 等がある。

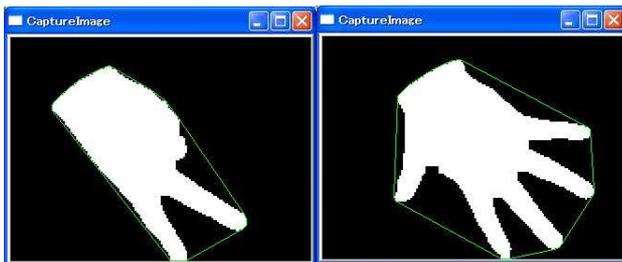


Fig.5 RockPaperScissors Component

パターン認識及びカメラのパラメータ情報を取得できる RT コンポーネント群としては, Template(図 8), ImageCalibration がある。

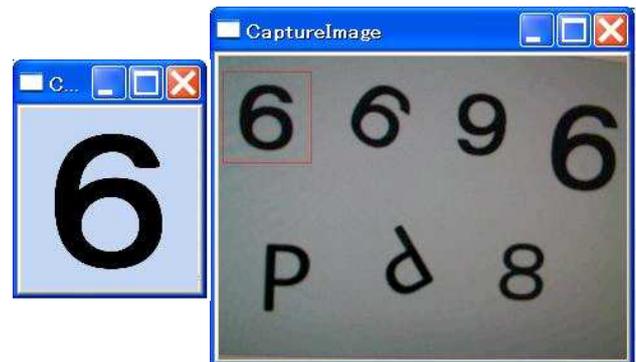


Fig.8 Template Component

これらの RTC を組み合わせることで, 様々な画像処理システムを容易に構築することができる。また, CameraImage データ型を使用しているため, 画像の詳細情報をデータポートから取得可能であり, 柔軟な処理を実現することが可能となった。

5. おわりに

本稿では, 新たにリリースされた OpenCV 2.0 を利用した画像処理コンポーネント群を実現した。様々な画像処理アルゴリズムをモジュール化するに当たり, モジュールの独立性を高めるために新たな画像データ型を導入した。また, マウス入力等を必要とする処理モジュールと表示モジュールを分割する方法として, マウスイベント情報用データポートを導入した。本稿で実装した RT コンポーネント群はオープンソースとして OpenRTM-aist の Web 上で公開する予定である [3]。

文献

- [1] G. Sziebig et al., "Robot Vision for RT-Middleware Framework", 2007 IEEE Instrumentation and Measurement Technology Conference, 2007
- [2] 田窪 他, "画像処理学習用 RT コンポーネントライブラリ", SI2008, pp.1L4-4, 2008
- [3] OpenRTM-aist Official Website, <http://www.openrtm.org/>