

```
#####  
# 画像キャプチャコンポーネントの使い方      #  
# 2011/02/16                                #  
#####
```

## 1.概要

本プログラムは OpenCV を利用した画像キャプチャ RT コンポーネントです. OpenCV により画像キャプチャが可能な USB/IEEE1394 カメラで利用が可能です. 画像描画 RT コンポーネントも同封されています. データポートのデータ型に `Img::TimedCameraImage` 型を, キャプチャ指令用に `Img::CameraCaptureService` インタフェースを利用しています.

## 2.依存ライブラリ

OpenCV 2.0.0 以上

## 3.コンパイル

### (1)依存ライブラリのインストール

```
$ sudo aptitude update
```

```
$ sudo aptitude install libcv4 libhighgui4 libcvaux4 libcv-dev libhighgui-dev  
libcvaux-dev opencv-doc
```

### (2)GUI の有無の指定

Makefile 内の `WITH_GUI` を `yes` にするとウィンドウを表示し, 出力画像を描画する.

### (3)コンパイル

```
cd src/
```

```
make -f Makefile.CaptureTimedCameraImage
```

## 4.実行

### (1)コンポーネントの実行

```
./CaptureTimedCameraImageComp
```

### (2)コンポーネント接続例

```
- example_img/example1.png
```

キャプチャ RTC を能動的かつ連続的に実行させる場合

```
- example_img/example2.png
```

キャプチャ RTC を受動的に実行させ, キャプチャ指令を `CameraCaptureService`

の Consumer 側から制御させる場合

## 5.設定

コンフィギュレーションセットを設定する

### - camera\_id

キャプチャするカメラの ID。複数カメラが接続されている場合に利用する。

### - output\_color\_format

出力ポートへ書き出す画像のカラーフォーマット。キャプチャのモードと一致しない場合は変換される。

グレースケール (GRAY) またはカラー (RGB) から選択する

### - camera\_param\_filename

カメラパラメータデータのファイル名。

OpenCV サンプル (テスト時 OpenCV2.2.0 の cpp サンプル) の calibration で出力された yml ファイルを指定する。

### - undistortion\_flag

レンズひずみ補正の実施フラグ。

0 を指定するとひずみ補正を行わず、そのままの画像を出力する。

1 を指定するとカメラパラメータに基づき、レンズのひずみ補正を行う。

レンズひずみ補正の際、出力データに記述されるひずみパラメータはゼロにリセットされる。

### - cap\_continuous\_flag

実行時に連続キャプチャを有効化するオプション

0 を指定すると、アクティブ時はサービスポート CameraCaptureService より、

take\_one\_frame(), take\_multi\_frames(), start\_continuous()のいずれかが

指定されるまで何もせず待機状態となる。内部的にも画像キャプチャ処理が実行されない。

1 を指定すると、連続キャプチャおよびデータの送信を実行する。

## 6.入出力仕様

### 6.1 出力ポート

- image: 画像データ。

-- TimedCameraImage 詳細のデータ構造は Img.idl を参照。

```

--- RTC::Time tm      # RTM 形式の時間(キャプチャ時間と同じ)
--- CameraImage data # カメラ画像データ
---- RTC::Time captured_time      # キャプチャ時間
---- ImageData image      # 画像データ
---- CameraIntrinsicParameter intrinsic # カメラの内部パラメータ
----- double matrix_element[5]      # カメラの内部行列
      a[0] a[1] a[3]
      0   a[2] a[4]
      0    0   1
----- sequence<double> distortion_coefficient # レンズ歪パラメータ
      OpenCV のものと同じ, k1, k2, p1, p2 [, k3]
---- Mat44 extrinsic # カメラの外部パラメータ
--- long error_code # エラーコード (使用していない)

```

## 6.2 サービスポート (Provider)

```

- インタフェース名:CameraCaptureService
  インタフェース型:Img::CameraCaptureService
-- void take_one_frame();      # 画像 1 枚のみキャプチャ
-- void take_multi_frames(in long num); # 画像を指定枚数キャプチャ
-- void start_continuous();    # 画像の連続取得開始
-- void stop_continuous();     # 画像の連続取得終了

```

## 7.RTC::Logger ログ

```

- ERROR
-- Unable to open video device. [<camera_id>]
    カメラデバイスのオープンに失敗した際に出力される. (onActivated)
    このエラー発生時にはエラー状態に移行する.
-- Unable to open camera parameter file. [<camera_param_filename>]
    カメラパラメータファイルのオープンに失敗した際に出力される. (onActivated)
    このエラー発生時にはエラー状態に移行する.
- WARN
-- Image data is empty.
    キャプチャ画像が空データの時に出力される (onExecute)
    この警告発生時には画像データの出力ポートへの書き込みは行わず onExecute を抜
    ける.
- DEBUG

```

-- Waiting capture command from continuous\_flag or Service Port.

このメッセージは画像キャプチャの指令が与えられていない際に出力される.

(onExecute)

このメッセージが出力される場合は, 画像キャプチャそのものが実行されず onExecute を抜ける.

画像キャプチャを行うには, サービスポート CameraCaptureService からキャプチャ指令を与えるか, または, コンフィギュレーションパラメータ cap\_continuous\_flag の初期設定に true を指定する. ただし, cap\_continuous\_flag が true に指定された場合でも, サービスポート CameraCaptureService より stop\_continuous()が呼び出された場合は, キャプチャ停止状態となる.

- TRACE

-- Capture image size <width> x <height>

キャプチャされた画像の画像サイズ (幅 x 高さ) が出力される. (onActivated)

-- Depth <depth>, Channels <channels>

キャプチャされた画像の深さ, チャンネル数が出力される. (onActivated)

## 8.開発環境

- ubuntu 10.04
- OpenRTM-aist 1.0.0
- omniORB 4.1.2-1
- RTCBuilder 1.0.0
- G++コンパイラ 4.4.3
- OpenCV 2.2.0 / 2.0.0

## 9.ディレクトリ構成

- src : RTCBuilder で生成したひな形に実処理を実装したソースコード
- test : テスト用プログラム
- example\_img : RTC 接続例の画像

## 10.テスト用プログラム

### 10.1 画像ビューアコンポーネント (timed\_camera\_image\_viewer)

#### 10.1.1 概要

カメラ画像データを取得し、描画を行うコンポーネント

#### 10.1.2 依存ライブラリ

OpenCV 2.0 以上

#### 10.1.3 コンパイル

```
cd test/timed_camera_image_viewer
make -f Makefile.TimedCameraImageViewer
```

#### 10.1.4 実行

```
./TimedCameraImageViewerComp
```

#### 10.1.5 設定

コンフィギュレーションセットを設定する

- frames\_num

キャプチャを要求する画像枚数を指定.

0 では連続取得, 1 以上では指定枚数を要求する.

ただし, あくまで要求する枚数であり, 画像は取得されれば指定枚数に限らず描画される.

#### 10.1.6 入出力仕様

##### 10.1.6.1 入力ポート

- image: 画像データ

##### 10.1.6.2 サービスポート (Consumer)

- インタフェース名: CameraCaptureService

インタフェース型: Img::CameraCaptureService

onActivated, onDeactivated 時に必要に応じて利用される

-- void take\_one\_frame(); # 画像 1 枚のみキャプチャ

-- void take\_multi\_frames(in long num); # 画像を指定枚数キャプチャ

-- void start\_continuous(); # 画像の連続取得開始

-- void stop\_continuous(); # 画像の連続取得終了

### 10.1.7 RTC::Logger ログ

#### - ERROR

- Configuration Param <frames\_num> should be over 0. [<frames\_num>]  
コンフィギュレーションパラメータ frames\_num に負の値を指定した場合に出力される。frames\_num にはゼロ以上の値を指定する。このエラー発生時にはエラー状態に移行する。(onActivated)

#### - INFO

- Send command of "start continuous" via CameraCaptureService.  
サービスポート CameraCaptureService に「start continuous」コマンドを発行した際に出力される。(onActivated)
- Send command of "take one frame" via CameraCaptureService.  
サービスポート CameraCaptureService に「take one frame」コマンドを発行した際に出力される。(onActivated)
- Send command of "take multi frames" via CameraCaptureService.  
サービスポート CameraCaptureService に「take multi frames」コマンドを発行した際に出力される。(onActivated)
- Send command of "stop continuous" via CameraCaptureService.  
サービスポート CameraCaptureService に「stop continuous」コマンドを発行した際に出力される。(onDeactivated)

#### - TRACE

- Capture image size <width> x <height>  
取得された画像の画像サイズ（幅 x 高さ）が出力される。(onExecute)
- Channels <channels>  
取得された画像のチャンネル数が出力される。(onExecute)
- Intrinsic matrix element[<num>] <value>  
取得された画像のカメラパラメータ内部行列が出力される。(onExecute)
- Distortion parameter[<num>] <value>  
取得された画像のレンズ歪パラメータが出力される。(onExecute)

## 11.ChangeLog

- 2011.02.16

各種バグフィックス.

バージョン 0.2.1 としてリリース

- 2011.02.07

サービスポート側の実装を追加.

RTC::Logger を使ったログメッセージの追加.

各種バグフィックス.

バージョン 0.2.0 としてリリース

- 2011.01.04

OpenCV 利用カメラキャプチャコンポーネントを試作

OpenCV 利用カメラキャプチャコンポーネントのテスト用ビューアコンポーネントを試作

## 12.特記事項

本モジュールのライセンスについては同封の LICENCE.txt、LICENSE-BSD.TXT をご参照下さい。

### 【連絡先】

奈良先端科学技術大学院大学 情報科学研究科 ロボティクス研究室

〒630-0192 奈良県 生駒市 高山町 8916-5

TEL: 0743-72-5376 FAX: 0743-72-5379 Email: robotics-staff@is.naist.jp