

平成13年度

高度な技術ノウハウの蓄積に資する先進シーズ応用型研究開発

---

## 先進シーズ応用型研究開発

「動画像からの実時間身体姿勢認識を用いた  
3次元UI技術の研究開発」

操作手順書

# 目次

1	プログラム構成	1
1.1	プログラム、ライブラリとデータの一覧	1
1.1.1	プログラム一覧	1
1.1.2	データファイル一覧	1
2	インストール	2
2.1	動作環境	2
2.1.1	ハードウェア	2
2.1.2	ソフトウェア	2
2.2	インストール手順	4
2.2.1	環境に合わせたプログラムの修正	4
2.2.2	システムの構築 (コンパイル) とインストール	4
3	操作説明	5
3.1	システムの起動と停止	5
3.1.1	起動方法	5
3.1.2	終了方法	5
3.2	システムのキャリブレーション	6
3.2.1	カメラ位置の調整	6
3.2.2	カメラのキャリブレーション	7
3.2.3	背景画像データの取得	7
3.3	システムの操作方法	8
3.3.1	キーボードコマンドによる制御	8
3.3.2	システムの動作モード	8
	付録	9
A	カメラのキャリブレーション方法	9
A.1	Tsai の方法によるキャリブレーションソフトウェアの入手	9
A.2	キャリブレーション用マーカ	10
A.2.1	キャリブレーション用マーカの作成	10

A.2.2	世界座標への配置 . . . . .	11
A.2.3	各カメラでのキャリブレーション用マーカの撮影 . . . . .	11
A.2.4	画像座標の読みとり . . . . .	11
A.3	カメラパラメータ . . . . .	13
A.3.1	作成したファイルの移動 . . . . .	13
A.3.2	キャリブレーションソフトウェアのソースファイルの修正 . . . . .	13
A.3.3	カメラパラメータの生成 . . . . .	14
A.3.4	生成されたパラメータファイルの移動 . . . . .	15

# 1 プログラム構成

## 1.1 プログラム、ライブラリとデータの一覧

### 1.1.1 プログラム一覧

実行形式プログラムおよびライブラリの一覧を、表 1 に示す。

表 1: プログラムおよびライブラリ一覧

ファイル名	内容
main/rcv	メインプログラム
lib/libmario.so	3次元位置認識ライブラリ
lib/libmalib.so	汎用動画像処理ライブラリ
lib/libcam1394.so	IEEE1394 カメラドライバライブラリ
lib/libraw1394.so	IEEE1394 ドライバライブラリ

### 1.1.2 データファイル一覧

データファイル一覧を表 2 に示す。

表 2: データファイル一覧

ファイル名	内容
main/data/bg0.ppm	背景画像データ (0)
main/data/bg1.ppm	背景画像データ (1)
main/data/calib/calib100464.ppm	カメラキャリブレーションデータ (0)
main/data/calib/calib100465.ppm	カメラキャリブレーションデータ (1)
main/data/color/color_hand_l.ppm	左手に対応する参考色パラメータ
main/data/color/color_hand_r.ppm	右手に対応する参考色パラメータ
main/data/color/color_head.ppm	顔面に対応する参考色パラメータ
main/data/model/human.mdat	モデル定義データ
main/data/user_log.dat	行動記録データ

## 2 インストール

### 2.1 動作環境

動作環境として、ハードウェア、ソフトウェアの要件を記す。

#### 2.1.1 ハードウェア

カメラを含めたハードウェア環境を用意する。システムを実行する PC が必要とするハードウェア要件は以下の通りである。

CPU PentiumIII 相当 (Pentium-4 の利用が望ましい)。

記憶装置 主記憶 64MB 以上、HDD 5GB 以上。

ビデオカード OpenGL のハードウェアアクセラレーションを実装し、かつ X Window System でその機能をサポートしているものが望ましい。ビデオコントローラとして nVidia の GeForce を搭載したビデオカードで動作を確認済。

IEEE1394 ホストアダプタ Linux でサポートするもの。

IEEE1394 カメラ IEEE1394 経由で 2 台以上、接続する。

システムの設置方法については 3.2 を参照のこと。

#### 2.1.2 ソフトウェア

システムを実行する PC が必要とするソフトウェア要件は以下の通りである。

オペレーティングシステム GNU/Linux (カーネルバージョン 2.4.10)

ウィンドウシステム X Window System (XFree86 4.0.3, GLX 拡張モジュール対応)

その他、表 3 に示すライブラリがインストールされていること。

表 3: システムが依存するライブラリの一覧

libstdc++-libc6.2-2.so.3	標準 C++ ライブラリ
libglut.so.3, libGLU.so.1, libGL.so.1, libGLcore.so.1	OpenGL 関係
libpthread.so.0	pthread
libjpeg.so.62	jpeg
libgdk_imlib.so.1	IMlib
libgtk-1.2.so.0	GTK+
libgdk-1.2.so.0	GDK
libgmodule-1.2.so.0, libglib-1.2.so.0	GLib 関係
libdl.so.2	動的リンク
libXi.so.6, libXext.so.6, libX11.so.6, libXmu.so.6, libXt.so.6, libSM.so.6, libICE.so.6	X Window System 関係
libm.so.6	数学ライブラリ
libc.so.6	標準 C ライブラリ
/lib/ld-linux.so.2	GNU リンカ

## 2.2 インストール手順

インストールは、管理者権限で行なう。また、インストールに際しては、環境に合わせてシステムの再構築を行なう必要がある。

### 2.2.1 環境に合わせたプログラムの修正

IEEE1394 カメラは各々の機器で固有の ID を持つ。事前に IEEE1394 カメラの ID を調べておき、メインプログラムのソースファイル `mario/src/rcv.c` において以下の箇所 で定義されるカメラ ID を個々の環境で利用するカメラ ID に修正し、再コンパイルする。

```
#define LEFT_1394CAM_ID      100464
#define RIGHT_1394CAM_ID    100465
```

上記の例は、左側に設置されたカメラの ID が 100464、右側に設置されたカメラの ID が 100465 のものを利用することを示す。

また、視点数を 3 個以上に増やす場合は、同ソースファイルにおいて `CAMERA_NUM` として定義されるマクロを修正し、また対応するデータファイルの定義を追加してシステムの再構築を行なう。

### 2.2.2 システムの構築 (コンパイル) とインストール

システムの構築 (コンパイル) は次の手順で行なう。

1. ソースツリーを作業領域にコピーする。以下の操作はソースツリーのトップディレクトリに移動して行なう。
2. `mario/src/rcv.c` をシステム環境に合わせて修正する。
3. ブートストラップスクリプト `./bootstrap` を起動し、設定ファイルを生成する。
4. `./configure; make` により、システムを構築する。
5. ルートになり、`make install` によりシステムのインストールを行なう。
6. また必要に応じて、データディレクトリ `mario/src/data` 以下をシステムを実行するディレクトリにコピーする。

## 3 操作説明

以下では、システムを実行するディレクトリを\$HOME/rcvとし、データディレクトリを\$HOME/rcv/dataに配置するようにインストールしたものと仮定して説明を行なう。

### 3.1 システムの起動と停止

#### 3.1.1 起動方法

ktermなどのターミナルエミュレータから、以下のコマンドを入力する。

```
$ cd $HOME/rcv/  
$ ./rcv
```

#### 3.1.2 終了方法

起動したターミナルエミュレータ上で「Ctrl」+「C」(コントロール・キーとCキーを同時に押す)により終了させるか、またはアバター表示ウィンドウ上で「Esc」キー(エスケープ・キー)を入力する。

## 3.2 システムのキャリブレーション

### 3.2.1 カメラ位置の調整

まず2台のカメラ位置の調整を行なう(図1)。ただしキャリブレーションを後で行なうので厳密に設置する必要はない。なおカメラの設置場所に関しては、窓際等の日光の影響を強く受ける恐れのある場所は避けた方がよい。

また2台のカメラで追跡を行なうので、被写体の座る位置(机の端から200mmと想定)は2台のカメラで同時にとらえる必要があることと、被写体の顔および両手をロックするために、とくに顔に影が被らないような光源の位置が確保できることの3点に注意する必要がある。また3次元位置推定の精度を高めるためには、斜め上から見下ろす位置に設置するとよい。

もし既設の光源に対して適当な位置を確保できない場合には、被写体の前面からの補助光投影もできるようにする。

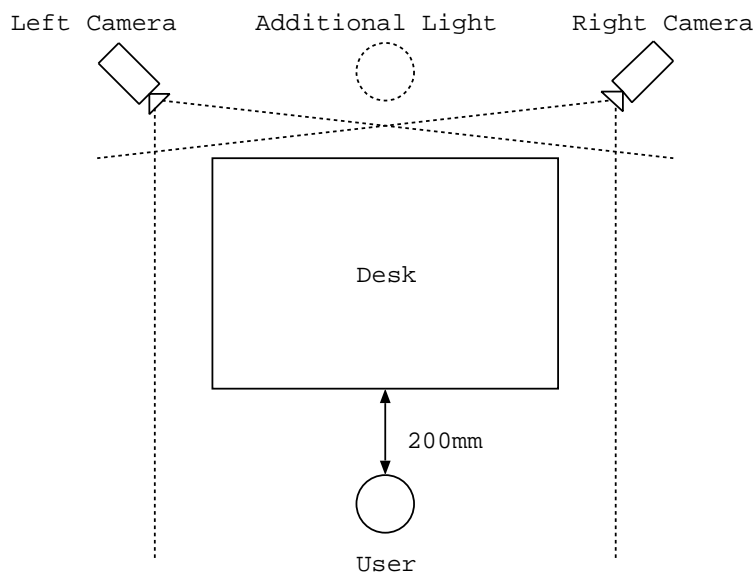


図 1: システム配置の一例

カメラを設置したら、カメラの絞りや配置が適当かを確認するためにシステムを起動させる。

```
$ cd $HOME/rcv/  
$ ./rcv
```

認識画面の出力を見ながら、起動したままで画角や絞りを調節する。

### 3.2.2 カメラのキャリブレーション

カメラシステムのキャリブレーションや、それに伴う設定ファイルの変更を行なう。その詳細については Appendix A を参照のこと。

### 3.2.3 背景画像データの取得

次に背景画像を得るために、カメラシステムを起動させる。背景画像の取得はシステム実行の前に必須の準備である。なお背景画像取得の際には、ユーザ自身がカメラの画角に含まれないように注意する必要がある。

```
$ cd $HOME/rcv/  
$ ./rcv -b
```

撮影された画像は\$HOME/rcv/data ディレクトリに bg0.ppm および bg1.ppm という名で格納される。

### 3.3 システムの操作方法

#### 3.3.1 キーボードコマンドによる制御

システムはキーボード入力を受け、表 4 に示す操作を実行することができる。なおキーボード入力はアバター表示ウィンドウ上にマウスポインタが置かれている状態で行なう必要があることに注意する。

表 4: キー入力によるコマンドの一覧

1、2	アバターの位置を左右に移動させる
a、z	アバターの位置を前進・後退させる
A、Z	アバターの位置を上下に移動させる
c	視点の位置を切り替える
d	アバターの表示方式を切り替える
b、B	アバターの胴体の向きを左右に回転する
ESC	プログラムを終了する

#### 3.3.2 システムの動作モード

システムの起動時に、表 5 のコマンドラインオプションを付けることでシステムの動作モードを選択することができる。

表 5: コマンドラインオプション

短い定義	長い定義	オプション引数	動作
-b	--bg	—	背景画像取得モードで起動
-p	--play	行動記録ファイル名	行動再現モードで起動
-r	--record	行動記録ファイル名	行動記録モードで起動
-h または -?	--help	—	利用法を表示して終了

## 付録 A カメラのキャリブレーション方法

### A.1 Tsai の方法によるキャリブレーションソフトウェアの入手

本システムが利用するカメラのキャリブレーションは、Tsai の方法による。Tsai の方法によるキャリブレーションソフトウェアは以下の URL からダウンロードすることで入手可能である。

```
http://www-2.cs.cmu.edu/afs/cs.cmu.edu/user/rgw/www/TsaiCode.html
```

キャリブレーションを行なうソフトウェアは上記から入手できる次のソフトウェアを用いる。

```
Tsai-method-v3.0b3.tar.Z
```

以降の説明では、このファイルをホームディレクトリで展開し以下に示す作業用ディレクトリが生成されているとして、キャリブレーションの手順を記す。

```
$HOME/Tsai-method-v3.0b3/
```

## A.2 キャリブレーション用マーカ

カメラは任意の場所に設定されているために、キャリブレーションを行なう必要がある。本節では Tsai の方法を用いてカメラのキャリブレーションを行なう方法を説明する。具体的な方法として、3次元上にマーカを配置してキャリブレーションを行なう方法と2次元平面にマーカを配置して行なう方法があるが、ここでは後者の方法を用いる。

### A.2.1 キャリブレーション用マーカの作成

B2 サイズのスチレンボードを利用し、その上に間隔が 150mm の線を縦横に引く。できた格子点上にシール等で目印を作成する (図 2 参照)。撮影された画像から座標を読みとる関係上、10mm 程度のものがよい。

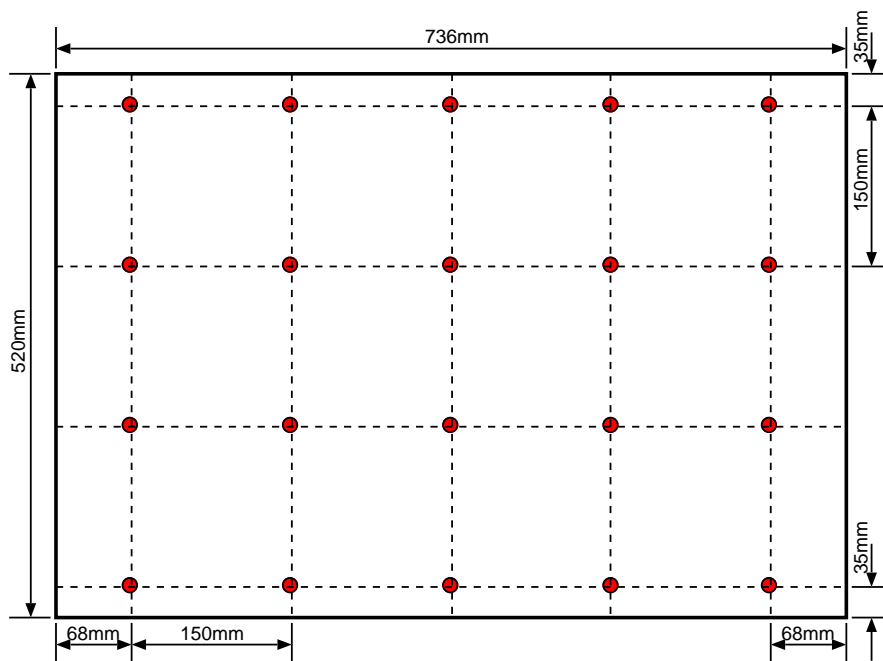


図 2: キャリブレーション用マーカの配置例

## A.2.2 世界座標への配置

作成されたキャリブレーション用マーカを世界座標上へ配置する。本システムは机の縁に原点を持ち、被写体側を  $+x$  方向 (被写体は  $-x$  方向を向いて座っている) とし、被写体の左手方向が  $+y$  方向であるような世界座標を規定している。

キャリブレーションは世界座標と画像座標の座標変換を正しく行なうための作業であるために、各マーカの世界座標を知っておかねばならない。そこで先ほど規定した原点から各マーカまでの  $x$  方向、 $y$  方向それぞれの距離を計測し、その距離をそのまま世界座標上での座標とする。ただし距離の単位は mm とする。

## A.2.3 各カメラでのキャリブレーション用マーカの撮影

キャリブレーション用紙の配置と世界座標の読みとりが終わったら、用紙の撮影を行なう。手順は以下の通りである。

```
$ cd $HOME/rcv
$ ./rcv -b
```

撮影された画像は `$HOME/rcv/data` ディレクトリに `bg0.ppm` および `bg1.ppm` という名で格納される。

## A.2.4 画像座標の読みとり

撮影された画像データから各マーカの画像座標を読みとる。ここでは GIMP を用いて `bg0.ppm` ファイルを読む。

```
$ cd $HOME/rcv/data
$ gimp bg0.ppm
```

画像上でマウスを移動させると、GIMP の画面左下に  $x,y$  座標が表示される。この座標が画像座標であるために、キャリブレーション用のマークをポイントして座標を読みとる。読みとる際に可能であれば、主観的に小数点以下まで読みとることで若干の精度の向上が期待できる。

またキャリブレーション用紙が画角から外れたり、隠れてしまっているような場合は、再度調節を行なった後に、「各カメラでのキャリブレーション用マーカの撮影」を行なう。

読みとった座標は各マーカの世界座標と画像座標を組にして、カレントディレクトリにファイルを作成して書き込む（ここではファイル名を `l_data.dat` とする）。書式は、「世界座標の  $x$  座標、世界座標の  $y$  座標、世界座標の  $z$  座標、画像座標の  $x$  座標、画像座標の  $y$  座標」であり、マーカを 20 点用意した本説明での場合では、ファイルに書き込まれる内容は 20 行 5 列になる。

また、`bg0.ppm` に関して作業が終了した後に、同様の作業を `bg1.ppm` に関して行ない座標対のファイルを作成する（ここで `bg1.ppm` に関して作成されたファイルの名称は `r_data.dat` とする）。

## A.3 カメラパラメータ

### A.3.1 作成したファイルの移動

両カメラに関して世界座標と画像座標の組を作成することができたならば、その作成したファイルを Tsai のプログラムで実行することで、それぞれのカメラに関するパラメータを得ることができる。そこで作成した座標対の記述されたファイルを、Tsai の方法のプログラムが置いてあるディレクトリにコピーする。

```
$ cp $HOME/data/l_data.dat $HOME/Tsai-method-v3.0b3/  
$ cp $HOME/data/r_data.dat $HOME/Tsai-method-v3.0b3/
```

### A.3.2 キャリブレーションソフトウェアのソースファイルの修正

キャリブレーションデータから未知のカメラパラメータを算出する。あらかじめテクニカルマニュアル等でわかるパラメータについては、その値を利用して計算を行なう。

まず使用しているカメラデータに合わせて、キャリブレーションソフトウェアのソースファイルを修正する。

変更する設定ファイルは\$HOME/Tsai-method-v3.0b3/内の `cal_main.c` ファイルの、92 行目から 209 行目にかけての関数 `initialize_photometrics_parms()` である。変更箇所と変更内容は次の通り。

- 194 行目の”...”内を今回使用しているカメラ名に変更
- 196 行目の `cp.Ncx` の値を使用しているカメラの x 方向の CCD の素子数に変更
- 197 行目の `cp.Nfx` の値を 1 走査線のサンプル数に変更
- 198 行目の `cp.dx` の値を x 方向の CCD 素子の間隔に変更
- 199 行目の `cp.dy` の値を y 方向の CCD 素子の間隔に変更
- 202 行目の `cp.Cx` の値を  $(x \text{ 方向の CCD の素子数})/2$  の記述に変更
- 203 行目の `cp.Cy` の値を  $(y \text{ 方向の CCD の素子数})/2$  の記述に変更
- 207 行目の `cp.Cx` の値を  $(x \text{ 方向の CCD の素子数})/2$  の計算値に変更
- 208 行目の `cp.Cy` の値を  $(y \text{ 方向の CCD の素子数})/2$  の計算値に変更

具体的には次の通り (変更後の内容を記載)。

```

192 void      initialize_photometrics_parms ()
193 {
194     strcpy (camera_type, "SONY DFW-V500");
195
196     cp.Ncx = 659;          /* [sel]      */
197     cp.Nfx = 320;        /* [pix]      */
198     cp.dx = 0.0074;      /* [mm/sel]   */
199     cp.dy = 0.0074;      /* [mm/sel]   */
200     cp.dpx = cp.dx * cp.Ncx / cp.Nfx; /* [mm/pix]   */
201     cp.dpy = cp.dy;      /* [mm/pix]   */
202     cp.Cx = 320 / 2;     /* [pix]      */
203     cp.Cy = 240 / 2;     /* [pix]      */
204     cp.sx = 1.0;        /* []         */
205
206     /* something a bit more realistic */
207     cp.Cx = 160.0;
208     cp.Cy = 120.0;
209 }

```

次に変更を反映させるためにコンパイルを行なう。

```
$ make -f makefile.unx all
```

### A.3.3 カメラパラメータの生成

コンパイルの結果生成された実行ファイルを用いて座標対の記述されているファイルを読み込み、カメラパラメータを生成させる。実行方法は次の通り。実行結果は l\_cal.dat に記述される。

```
$ ./ccal l_data.dat > l_cal.dat
```

またもう一台のカメラについても同様に行なう。

```
$ ./ccal r_data.dat > r_cal.dat
```

#### A.3.4 生成されたパラメータファイルの移動

Tsai の方法によって生成されたデータを、カメラシステムで用いるために移動させる。なおカメラパラメータ定義ファイルのファイル名は、メインプログラムの `rcv.c` 内で定義するファイル名と整合性をとる必要がある。

```
$ cp $HOME/Tsai-method-v3.0b3/l_cal.dat $HOME/rcv/data/calib100464.dat  
$ cp $HOME/Tsai-method-v3.0b3/r_cal.dat $HOME/rcv/data/calib100465.dat
```

以上でカメラパラメータの準備は完了する。

Copyright (C) 2002 情報処理振興事業協会 (IPA)

Copyright (C) 2002 Information-technology Promotion Agency, Japan