

# 顔検出システム MACAO & FacePoint

MACAO & FacePoint, The Face Detection System

2003年10月30日

株式会社 三菱総合研究所  
飯尾 淳

## 概要

---

- 背景
  - ニーズ、技術シーズ、課題、問題点
- MACAO : 顔認識研究開発プラットフォーム
  - 目的と概要
  - 設計、プログラミング方法
- FacePoint : 頭部位置検出を利用したユーザインタフェース
  - 概要
  - アルゴリズム
- デモンストレーション & まとめ



# 背景: アプリケーションのニーズ

---

## ■ 人と機械のコミュニケーション

- 聴覚 (音声認識 / 音声合成)
- 視覚 (画像認識 / 画像生成)
- 触覚 (圧力センサ、他)

## ■ 視覚系

- コンピュータビジョン、ロボットビジョン
- VR (バーチャルリアリティ、仮想現実感)、MR / AR (拡張現実感)

## ■ ユーザインタフェースの見直し

- UIは保守的?
- いまだにキーボードとマウス...



# 技術シーズ

---

## ■ デバイスの普及と計算能力の向上

- 高性能小型カメラ
- 高速シリアルバス、IEEE1394、USB2.0
- GHz CPU、64bit化、etc...

## ■ 技術的背景

- MAiib (飯尾)
- 状況共有型ヒューマノイドロボット (伊達)



# 何ができていて、何ができていないか?

---

- 人物検出・顔認識技術研究は昔から活発
- 実用化製品もちらほら
  - オムロン、東芝、etc...

しかし!

- 「顔の同定のために重要となる特徴的な要素は何か」
  - この問題に関する答えはよくわかっていない
  - ソフトウェアも公開されている実装は少ない
    - ▶ 未踏ソフトでも顔認識のプロジェクト → オープンソースにしてもらえない



## 実時間顔検出・顔認識

---

- 既存技術
  - 知識ベース、特徴抽出、肌色検出、
  - テンプレートマッチ、EGM (Elastic Graph Matching)、
  - 固有顔、統計的手法 (ニューラルネット、SVM、HMM、など)
- 顔画像認識技術のレベル
  - 1. 顔位置の検出
  - 2. 顔の認識 (個人の認証)
  - 3. 個人属性の識別 (年齢や表情など)



- MACAO : 顔認識アルゴリズム開発テストベッド
  - カメラからの入力画像を解析して、顔を検出・認識
- FacePoint : 頭部位置情報を利用したユーザインタフェース
  - ステレオカメラ利用
  - 実は顔認識はしていない?
  - 単純なアルゴリズム故にロバスト



---

## MACAO : 顔認識アルゴリズム開発テストベッド

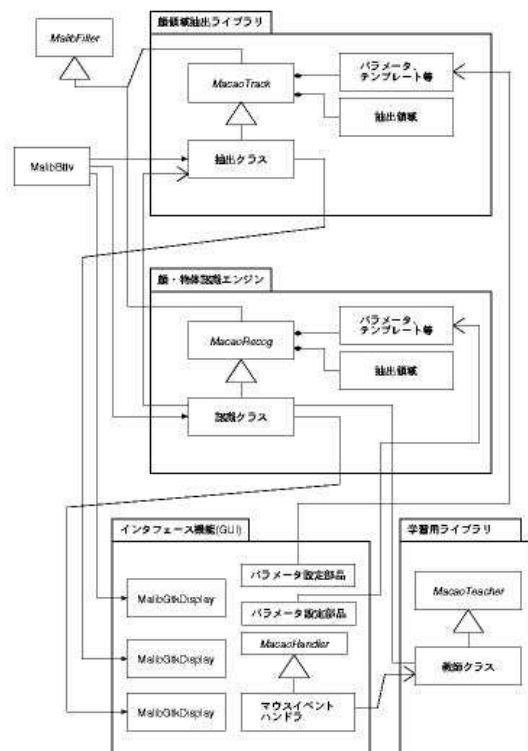
---

- 目的
  - 顔検出・顔認識アルゴリズム開発を簡単に
- 特徴
  - MAlibベース
    - ▶ (MAlib最新版(0.5.8)ではDV入力にも対応)
  - 検出・認識モジュールはライブラリで提供
  - 実時間動作で検証可能 (GUI付き)
  - 型の異なるパラメータを統一的に扱う仕組みを持つ



# MACAOの設計

- 顔検出・顔認識モジュールは、MAlibのフィルタの拡張 (サブクラス) として実装
- それぞれはアブストラクトクラス
  - MacaoTrack
  - MacaoRecog
- 各クラスを継承させて必要な認識クラスを作成



Linux Conference 2003

2002/10/30

Mitsubishi Research Institute, Inc.

## 処理の流れ

1. MacaoTrackで顔領域のおおまかな位置を検出
  2. MacaoRecogではその情報を参照しつつ、顔認識処理を行なう
- 必要に応じて学習も可能 (MacaoTeacherを使用)



Linux Conference 2003

2002/10/30

Mitsubishi Research Institute, Inc.

# MACAOプログラミング例

## ■ 使用クラス

- 顔検出クラス: MacaoSkin
  - ▶ 肌色画素判定に基づく顔領域検出アルゴリズムを実装
- 視線の検出: MacaoLook
  - ▶ テンプレートマッチを用いた注視点検索
- 学習作業はMACAOテストベッドを利用可能

```
MalibSource * src;
MalibSource * filter0, * filter1;
MalibBuffer * buf0;
MalibGtkDisplay* disp[3];

/* initialization */
malib_init (&argc, &argv);

/* start capturing */
src = (MalibSource*)malib_bttv_new_with_size(320, 240);
malib_bttv_start_capture ((MalibBttv*) src);

/*
 * MalibBttv
 *   -> MalibPlainBuf
 *         -> MacaoSkin -> MalibGtkDisplay
 *         -> MacaoLook -> MalibGtkDisplay
 */
buf0 = (MalibBuffer*)malib_plainbuf_new_with_source(src);
filter0 = (MalibSource*)macao_skin_new_with_buf(buf0);
filter1 = (MalibSource*)macao_look_new_with_buf(buf0,
                                                (MacaoTrack*) filter0);

/* setup displays */
disp[0] = malib_gtkdisplay_new_with_source (filter0);
disp[1] = malib_gtkdisplay_new_with_source (filter1);
disp[2] = NULL;

malib_gtkdisplay_auto_play2 (disp);
```



## 公開情報

- テストベッド、ライブラリをまとめたパッケージ
- ライセンスはLGPL
- コンパイルと実行には MALib が必要

## ■ 公開URL

[http://www.malib.net/applications/index\\_j.html](http://www.malib.net/applications/index_j.html)



# FacePoint の概要

- ステレオカメラの画像を利用した顔位置検出システム
- 新しいユーザインタフェースの提案 (参考 → LC2002資料)

## ■ 開発動機

- いつまでマウスを使いつづけるの?
- キーボードから手を離したくない
- 3次元データの扱いにくさ
- 下から覗くと……?



## 改良点 (昨年から進化した点)

### ■ アルゴリズムを一新

- 認識の安定性が向上
  - ▶ 肌色情報に基づく検出は不安定
  - ▶ 環境の変化に対して敏感
- 実用化までもう少し?

### ■ システム小型化

- 小型の専用ステレオカメラ導入
  - ▶ PointGrey社製、Bumblebee
- 出張デモセット
  - ▶ 鞆ひとつで世界中どこへでも

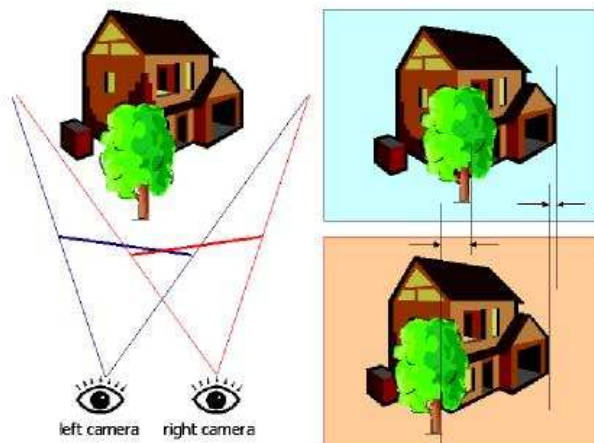


# 動作原理

1. 視差マップを利用したユーザ画像の切り出し
2. トラッキング領域の逐次更新による頭部位置追跡

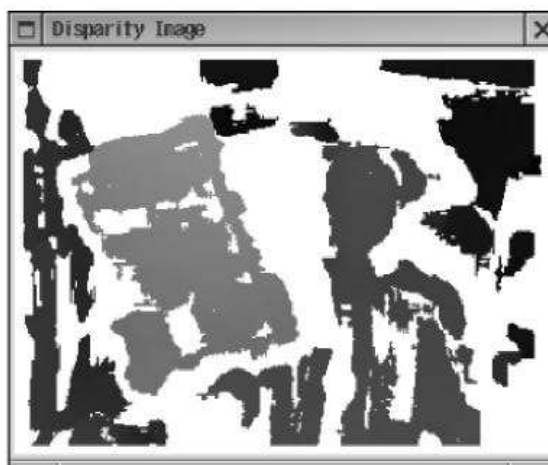
## ■ 視差マップとは?

- 左右のカメラの画像から生成
- 部分画像の相関を探索して計算
- 視差の大きさは距離に反比例



# 視差マップの例

## ■ 入力画像(左)と視差マップ(右)



## ■ 視差が計算できない例

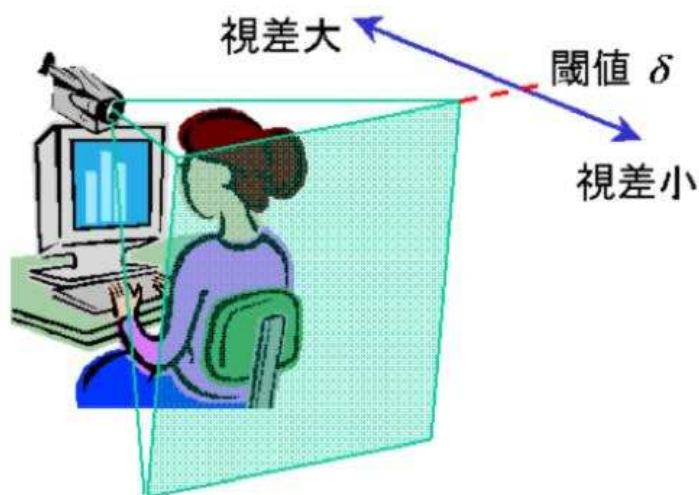
- オクルージョン
- 平板画像
- 規則的なパターン





# 視差による背景分離

## ■ その基本的なアイデア



## ■ 背景分離の例

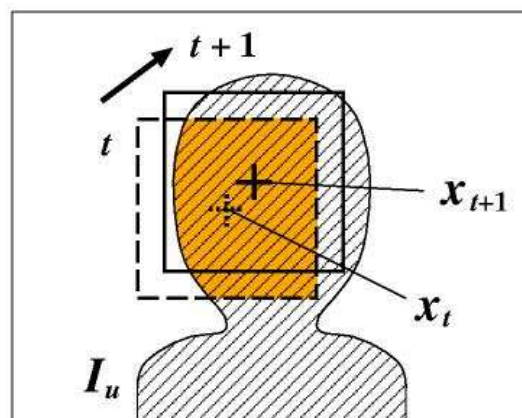


# トラッキングアルゴリズム

1. ユーザの顔中心を囲む領域をトラッキング領域とする

2. 切り出されたユーザ画像のうち、トラッキング領域に含まれる部分の重み付き平均を計算する

3. 求められた重心を中心とするよう、トラッキング領域を更新する



# 3つの仮定

---

- ユーザはシステムの前 (カメラの前) に存在する
- 人間の頭の大きさはほぼ一定、かつ、画像中の顔領域の大きさは距離に反比例する
- 人間の頭はユーザ像の最上部に位置する



# 3次元位置の推定

---

- XY平面での点( $X'$ ,  $Y'$ )に対応する実空間上での位置( $x', y', z'$ ) ※ は、次の要素から計算される
  - 視差  $D(X', Y')$
  - レンズ間のベースライン長  $b$
  - レンズの焦点距離  $f$
- トラッキング領域の中心を、視点として近似 → UIに適用

※ ただし ( $x', y', z'$ ) はカメラ座標系であることに注意



# ユーザインタフェースへの応用

## ■ プロトタイプシステム

- バーチャル地球儀 (改)
- VRMLビューアへの組込み
- Open Inventor のサンプルプログラム

## ■ 性能

- 10~12 fps
  - ▶ Pentium 4 (1.8GHz)
  - ▶ メモリ768MB
  - ▶ GForce3 ビデオカード



# 性能測定

## ■ 実験

- 顔モデルを10cm刻みで移動させて測定
  - ▶ カメラ座標に変換して誤差を計測
  - ▶ ワールド座標 → カメラ座標への変換行列は、全ての測定値についての最小二乗誤差を得る変換行列を推定した
- 環境を変えて測定
  - ▶ オフィスユース (会議室)
  - ▶ ホームユース (リビングルーム)



# 性能測定 (2)

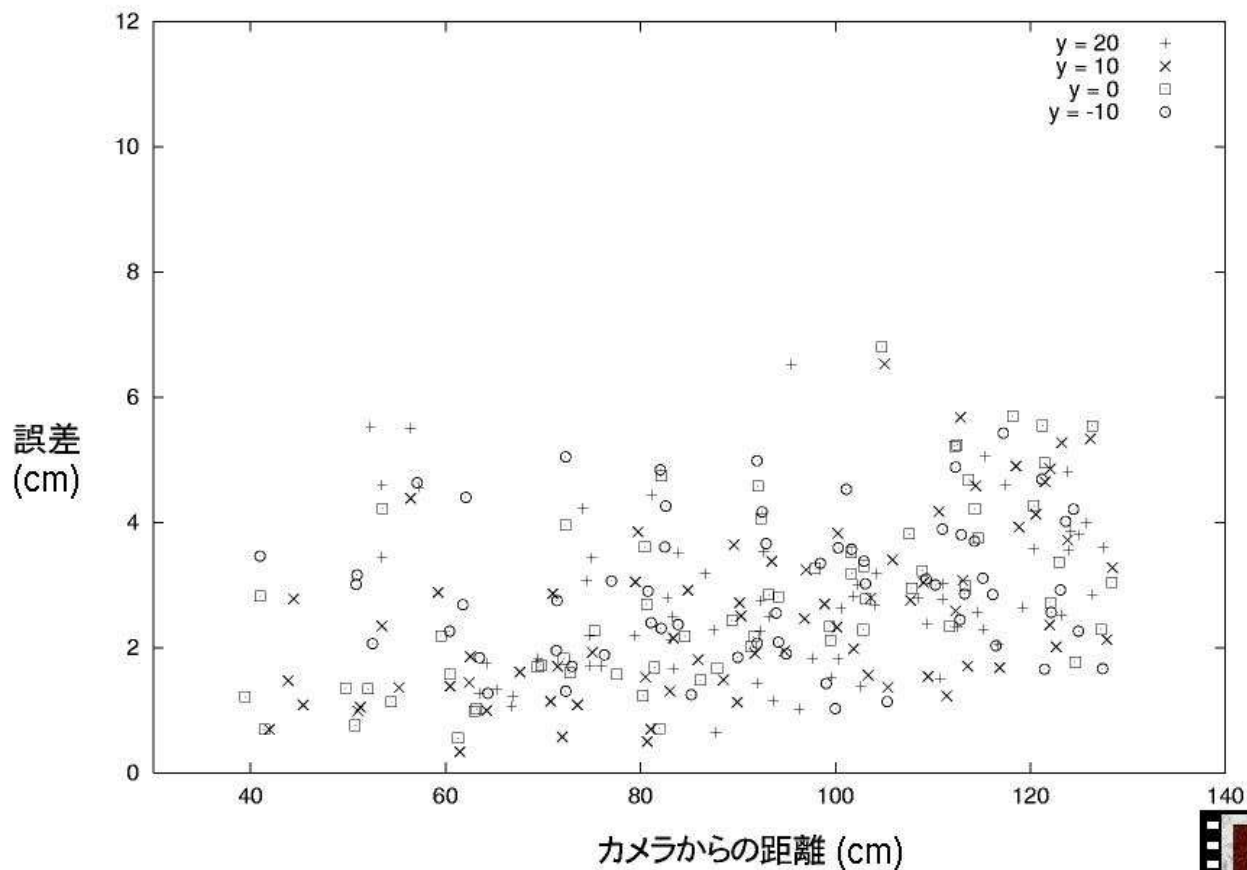
## ■ オフィス環境



## ■ 家庭環境



## 計測結果





## FacePointの抱えている課題

---

### ■ 問題点

- 現状はまだ作り込みが足りない
- ステレオカメラが特殊
- 視差マップ作成ライブラリがプロプラエタリ

### ■ オープンソース化に至るロードマップ

- 普遍的デバイスへの対応
- USBカメラを用いたステレオ視ライブラリ



- MACAO システムを紹介しました
  - 顔認識アルゴリズムのテストが簡単にできます
  - ぜひモジュール開発にご参加下さい
  
- FacePoint ユーザインタフェースを紹介しました
  - 昨年度提案したプロトタイプと比べて、「使える」度アップ
  - システム構築提案中

おしまい

