

顔検出システム MACAO & FacePoint

MACAO & FacePoint : The Face Detection System

飯尾 淳[†]

iiojun@mri.co.jp

伊達 章[‡]

date@crl.go.jp

概要 様々な動画認識アプリケーションをできるだけ簡単に構築できるようにするために、C 言語ソフトウェアライブラリ MAlib の開発を行なっている。この MAlib をベースとして、入力映像に含まれる顔を実時間検出するためのライブラリ MACAO と、ステレオカメラからの映像を用いて顔の 3 次元位置を検出し、ポインティングデバイスとして利用するユーザインタフェースシステム FacePoint を開発した。本論文では MACAO および FacePoint の設計と実装について報告する。

1 はじめに

人とロボットのコミュニケーションは、ロボット研究者にとっての永遠の研究課題である。ロボットとのコミュニケーションにおいては、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) など、様々なチャンネルを介した情報の交換が試みられている。

一方で人とコンピュータとのインタフェースは、いまだにキーボードとマウスが主流である。音声入力のためのマイクや、最近では USB で接続する安価なカメラが備えられた PC も増えつつあるとはいえ、キラーアプリケーションの不在によりそれらのデバイスが普遍化するところまでには至っていない。カメラを意識させない映像入力手段など、新しいインタフェースの登場が望まれている段階である。

知能をもったロボット・コンピュータを活躍させるためには、まず、外界の様子を、画像を通じて認識することが必要になる。特に、人物を同定する必要性は様々な場面で生じることから、人物検出・顔認識技術については以前から活発に研究がおこなわれている。ここ数年は、カメラが小型化し価格が安くなったことで発展が著しく、携帯電話、監視カメラなどに組み込

まれた製品としても身近になりつつある。しかし、人の顔を認識することは日常の行為であるにもかかわらず、顔の同定をおこなう際に重要な特徴要素は何かということはよくわかっておらず、顔画像認識技術は未だ揺籃期にある。さらにオープンソースで公開されている実装は多くはない [1]。

本報告で紹介する顔検出システムは、各アルゴリズム別にライブラリとしてまとめることができるように設計されている。そのため各アルゴリズムを検証するテストベッドとしても利用できる。本システムの一部は既に LGPL のもとで公開しており、自由に利用できるようになっている。

2 開発の背景

三菱総合研究所の応用数理技術チームでは、PC におけるユーザインタフェースの新たな局面を実現すべく、画像認識を活用した PC への情報入力方式の研究開発を進めている [2, 3, 4]。またその応用範囲としてロボットインタフェースへの適用も想定して、汎用的に使用できる研究環境の構築を実施してきた。

本報告のベースとなっている MAlib は、これらの研究を進めるための重要なツールとして整備されている動画認識用 C 言語ソフトウェアライブラリ [5, 6] である。

MAlib の動作プラットフォームは Linux に限ったものではないが、開発は主として Linux

[†] 株式会社三菱総合研究所 情報技術研究部

[‡] 通信総合研究所 けいはんな情報通信融合研究センター

上で行なわれている。MAlib はオブジェクト指向設計による拡張の容易さを実現している他、対応する入力デバイスも Video4Linux だけでなく IEEE1394 カメラや DV カメラ*1、ネットワークカメラ、各種の動画ファイルなど多岐にわたっていることを特長としている。

通信総合研究所社会的インタラクシヨングループでは、Linux で動作する「状況共有型ヒューマノイドロボット」を用い、自然なコミュニケーションに必要な不可欠な要素は何かについて研究を進めている。具体的には、人間どうしのやりとりを観察することによりモデルを構築し、そのモデルをロボットに実装することで、モデルの検証・精密化を行っている。

ロボットと人間が状況を共有するためには、相手の顔の表情などを読むことは重要である。第4節で報告する MACAO は、このロボットに実装すべく開発が進められたものである。第5節で述べる FacePoint は、MACAO とは独立に開発したものであるが、ロボットのステレオ視を利用することにより、統合が可能である。

3 実時間顔画像検出

画像中に含まれる人間の顔を検出する技術は、コンピュータビジョンの中でも重要な分野である。顔検出技術は、知識ベース、特徴抽出、肌色検出、テンプレートマッチ、グラフマッチ、固有顔、統計的手法(ニューラルネットワーク、SVM、HMM)など、様々な手法が試みられている [7, 8]。

人間と機械との間における画像を用いたコミュニケーションにおいては、顔画像の検出と認識は重要な要素である。「目は口ほどにものを言う」という格言もあるとおり、言語以外に

*1 ここで IEEE1394 カメラとは IIDC (Instrument and Industrial Digital Camera) 仕様に準じた非圧縮画像データ伝送を可能とする IEEE1394 (FireWire) 接続の産業用デジタルカメラを指し、DV カメラとは家庭用デジタルビデオカメラのことを指す。後者も IEEE1394 で PC と接続されるが、画像データは DV コーデックで圧縮されて伝送される点で異なる。DV カメラによる画像入力機能は、2003 年 8 月末にリリースした MAlib の第 0.5.8 版でサポートされた。

よる情報入力としてユーザの顔情報を効果的に取得し認識できれば、より使いやすいユーザインタフェースを実現する可能性は非常に高まることが期待できる。

なお顔画像検出や顔画像認識技術には、処理の複雑さを基準として次のレベルが存在する。

1. 顔位置の検出
2. 顔の認識 (個人の認証)
3. 表情や属性 (年齢など) の識別

1. と 2. の間のレベルとしては、顔の向きの認識技術がある。さらに 3. 以降の認識としては顔画像を介したメンタルモデルの認識に至り、人工知能や認知心理学の領域となる。

我々の研究開発では、ユーザインタフェースに利用するという最終的な目標のもとで実時間で処理できることを優先し、1. および 2. のレベル程度の認識アルゴリズムの実現を対象と考えている。ただし将来 CPU の計算能力が更に拡大し表情の識別やそれ以上のレベルでの認識が実時間でできるようになれば、それらの実装も対象に含むこととなる。

4 顔検知ライブラリ: MACAO

4.1 概要

MACAO は、MAlib 上に構築される顔検出・顔認識処理 C 言語ライブラリである。顔検出機能や顔認識を行ないたいユーザは、アプリケーションへの MACAO を組み込みによって、逐次入力される画像を解析して顔領域の検出や顔認識処理を実時間で簡単に実現できるようにすることを目的としている。

次に説明するように顔検出や顔認識処理はフィルタとして実装されている。現在はまだ実装例が少ないが、様々なアルゴリズムをフィルタとして実装し、ライブラリ化している。ライブラリユーザは適宜、必要なフィルタを選択してアプリケーション中で使用できる。

また MACAO のパッケージには、フィルタを集積したライブラリの他、顔検出モジュールと顔認識モジュールの動作をテストしたり、各モジュールが持つパラメータの調整を行なうためのテストベッドが含まれている。

4.2 MACAO ライブラリ

MACAO ライブラリは、顔検出モジュールとさらにそれを拡張した顔認識モジュール、および各々に付帯するパラメータモジュール、教師モジュールなどから構成される。

MACAO テストベッド稼働時のソフトウェアモジュール構成は図 1 の状況となる。なおこれらは MALib が提供するオブジェクト指向設計に準じているので、それぞれがクラスとして実装されている。

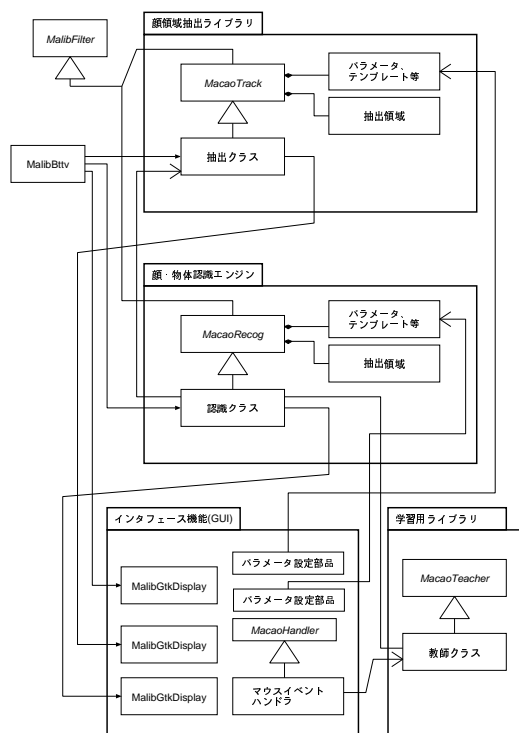


図 1: MACAO を構成するソフトウェアモジュール

顔検出処理，顔認識処理を行なうクラス (MacaoTrack, MacaoRecog) は，MALib におけるフィルタ処理を行なう MalibFilter のサブクラスとして実現される。さらに，認識処理は検出処理の特化したものと考えられ，認識クラスは検出クラスのサブクラスとなる。なお検出クラス，認識クラスは抽象クラスとして定義され，具体的なクラスはそれらを継承するクラスとして実装される。

また MacaoRecog は MacaoTrack と対になって利用されることを想定している。一般的な

処理の流れは次の手順となる。

1. MacaoTrack クラス (のサブクラス) によって，顔領域の大まかな位置が検出される。
2. MacaoRecog クラス (のサブクラス) は，その情報を参照しつつ，入力画像から顔認識処理を行なう。

認識クラスは，検出クラスの持つ性質に加えて各種のパラメータを学習可能とするために教師クラス (MacaoTeacher) へのポインタを持つ。

MacaoTeacher もやはり抽象クラスであり，実体として各 MacaoRecog クラスの実装に対応した具象クラスが用意される。

4.3 GUI によるテストベッド

MACAO ライブラリに含まれる顔領域のトラッキングあるいは認識機能は，MACAO テストベッドを利用することにより対話的に調整・確認することができる。MACAO テストベッドは GUI を備え，マウスとキーボードの操作により各機能の調整を行なうための汎用的なインタフェースを提供する (図 2)。



図 2: MACAO テストベッド

MACAO テストベッドの左側の上下に表示されている画像は，顔検出モジュールが顔領域を抽出した結果と顔認識モジュールが顔特徴・向きなどを認識した結果を表示するものである。これらの画像は各モジュールの出力によって異なる。

右上には入力画像そのものと、操作パネルが並ぶ。操作パネルでは入力の一時的停止と再開の制御を行なうことができる。

右側中央から右下に現れているパネルには、各モジュールが持つパラメータを調整するためのフィールドが並ぶ。これらは各モジュールのパラメータ毎に設定され、テストしたいモジュールのパラメータ定義を参照することで自動的にパラメータ調整用のウィジェットが配置される。

4.4 利用例

図 3 は、顔検出クラスとして MacaoSkin を使い、顔検出後の認識処理として MacaoLook を適用するプログラミング例である。この例では、入力として MAlib の Video4Linux インタフェースである MalibBttv を使い、上記ふたつのモジュールから出力される認識結果を MalibGtkDisplay クラスによりリアルタイムに表示させている。

```
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);
```

図 3: MACAO プログラム例

なお、MacaoSkin はカラー画像を対象とした肌色画素判定に基づく顔領域検出アルゴリズムを実装したクラスである。

また MacaoLook クラスではテンプレートマッチによる注視点認識を行なう。MacaoLook は事前に学習したテンプレート画像を用いて、その画像に対応した注視点を検索する処理を行

なう。テンプレートと注視点の対応関係に関する学習作業は、MACAO テストベッドを利用して実施できるように設計されている。

4.5 クラスの追加

MACAO では MAlib における画像処理フィルタを拡張するといった形式で顔検出クラスや顔認識クラスを実装する。したがって、入力画像に対する顔検出処理、顔認識処理といったアルゴリズムの本質的な部分については、MAlib のフィルタを新たに用意するプログラム手法に準じる。

ただし MACAO ではパラメータインタフェースの共通化の枠組みが導入された点が新しく、顔検出・認識処理に特化している部分である。すなわち MACAO で用意されるフィルタの雛形である MacaoTrack では様々なパラメータを統一的に扱うためのパラメータクラス MacaoParam が用意され、また先に述べたように MacaoRecog にはそのパラメータを学習させる処理を記述する教師クラスが用意されている。テストベッドが用意する GUI のパラメータ調整パネルの自動生成は、これらのパラメータアクセスの標準化によって実現されている例である。

新しい顔認識アルゴリズムを実装し、ライブラリに追加するには、そのアルゴリズムの実装の他にパラメータのオブジェクト化を行なう必要がある。ただしひとたびオブジェクトとしてパラメータを用意すれば、MACAO テストベッドを利用し、アルゴリズムの検証やパラメータの調整が簡単になるというメリットを享受できる。

5 3D 顔トラッカ : FacePoint

5.1 概要

FacePoint システムは、ステレオカメラからの映像を利用してユーザの顔の 3 次元位置を検出し、ユーザインタフェースとして利用するためのシステム [10] である。

FacePoint では入力機器として市販のステレオカメラを利用した。実際に使用しているカメラは PointGrey 社の BumbleBee [9] である (図 4)。

本カメラはIEEE1394でPCと接続し、WindowsおよびLinuxで利用できる。Linux向けには、libraw1394を介して画像データを取得し、さらに視差の計算までを行なうことができるライブラリが用意されている。



図 4: LCD 上に設置されたステレオカメラ

FacePoint では視差マップを入力とし、そこからユーザの顔領域を抽出することによって顔の検出、トラッキングを実現している。さらにその結果と視差の値から顔の3次元位置を計算し、3次元ポインティングデバイス [11] としての利用を検討した。

5.2 視差マップによる背景除去

FacePoint では、端末の前に一人のユーザが座っているという利用状況を想定している。つまり入力画像 I から背景 I_b を分離したものがユーザ画像 I_u である、と定義する。

入力画像に対する背景の分離は、視差マップを利用して行なわれる。

視差マップとは、2枚のステレオ画像の各画素の対応関係から生成した画像であり、キャリブレーション済のステレオカメラにおいては距離画像に相当する (図 5)。

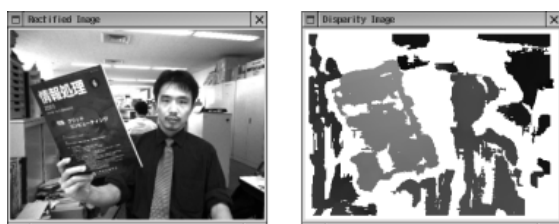


図 5: 視差マップ

視差マップは、各画素を中心として左右の画像におけるそれらの周囲の部分画像の相関を計算することによって作成される。比較中の部分画像に写る対象物が近くに存在すればするほど、左右の画像における部分画像の位置は離れたものとなり、視差は大きくなる。一方、対象物が遠くであれば左右の画像でほぼ同じ位置に写像されるため、視差は 0 に近づく。

図 5 は、カメラの前に物体を突き出した状態 (左) で視差マップ (右) を作成したものである。右側の視差マップは、濃度が薄くなるほど被写体が近いことを示している。ただし白抜きの部分は、視差が求められなかった箇所である。

一般に視差マップの計算においては、カメラの視差によるオクルージョンの差異や、平板ないしは繰り返しテクスチャによる一意性の欠如などの理由により、画像中の全ての画素について視差が求められるわけではないことに注意が必要である。

ところで先に挙げた仮定からユーザはカメラから一定の距離の範囲内に存在すると考えてよい。したがって式 (1) のとおり、各画素 X についてその視差が閾値 δ_D を越えるかどうか判定し、条件を満たす画素をユーザ画像 I_u として採用することで、背景画像を分離することが可能である。

$$X \in \begin{cases} I_u, & \text{if } D(X, Y) > \delta_D \\ I_b, & \text{if } D(X, Y) \leq \delta_D \end{cases} \quad (1)$$

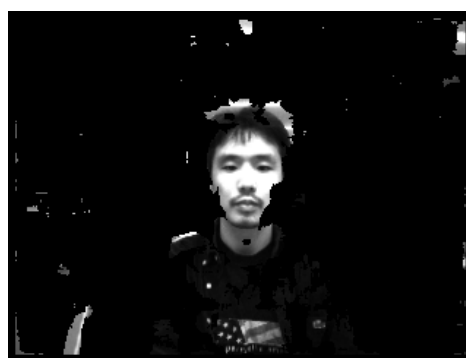


図 6: 距離を利用した背景除去

図 6 にユーザ画像が抽出されている様子を示す。周囲の黒く塗り潰されている部分が、背景画像として削除された部分である。

5.3 トラッキングアルゴリズム

ユーザ顔領域のトラッキングは、次の手順で行なわれる。

1. ユーザの顔領域中心 \bar{X}_t を囲む矩形領域をトラッキング領域とする。
2. ユーザ画像 I_u 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. 計算された重み付き平均で、ユーザの顔領域中心 \bar{X}_{t+1} を更新する。

重み付き平均の計算においては、重み $w(\cdot)$ として y 方向の上向きに比例した偏差を与える。その結果、トラッキング領域は上方向に移動することになるが、ユーザの頭の上には通常は何も存在しないため、ちょうど顔の位置で安定する。その結果として、顔領域が自動的に追跡されることとなる。

上記手順は次式で表される。

$$\bar{X}_{t+1} = \frac{\sum_{p \in \mathcal{D}(\bar{X}_t)} w(p) p}{\sum_{p \in \mathcal{D}(\bar{X}_t)} w(p)} \quad (2)$$

ここで $\mathcal{D}(\bar{X}_t)$ は、 \bar{X}_t を中心とする矩形領域と I_u の積集合として構成される領域である (図 7)。

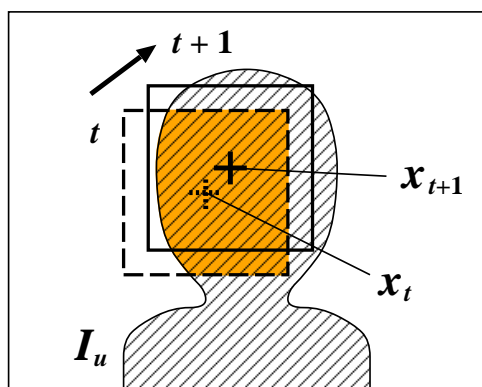


図 7: 顔領域のトラッキング

なお顔領域中心位置の初期値 \bar{X}_0 は、ユーザの顔が画像の中心付近に存在するとの仮定から撮影画像の中央に置く。さらにユーザ不在の状態や、異常状態を検知したときなどは初期値に戻るように設定した。

またトラッキングに用いる矩形領域は、 \bar{X}_t における視差を参照してユーザが遠くに位置するときには小さく、また近くに位置するときには大きく設定するようにしており、ほぼ顔の大きさに等しい範囲を覆うよう自動的に調整される (図 8)。

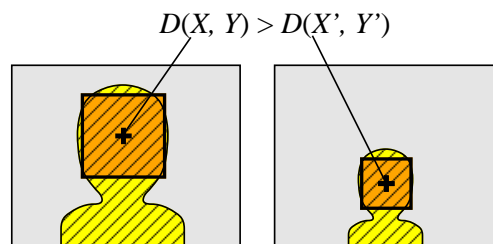


図 8: トラッキング範囲の自動調整

5.4 3次元位置の推定

撮影された画像平面 $\{X, Y\}$ と撮影対象が存在する実空間 $\{x, y, z\}$ の関係は、式 (3), (4) から計算できる。

XY 平面での $X' = (X', Y')$ に対応する実空間上での位置 $x' = (x', y', z')$ は、 X' おける視差、レンズ間のベースライン長 b 、およびレンズの焦点距離 f を用いて、以下の式で表される。

$$z' = bf/D(X', Y') \quad (3)$$

$$x' = X'z'/f, \quad y' = Y'z'/f \quad (4)$$

トラッキングの結果得られた領域中心点を仮想的に視点位置として用い、上式を適用することによりユーザの頭部位置の3次元位置が実時間で求められる。

5.5 プロトタイプシステム

FacePoint のプロトタイプシステムは、[6] で報告したバーチャル地球儀システムの顔認識部を入れ換える形で開発が進められた。現在では、SGI 社がオープンソースソフトウェアとして GPL で公開している Open Inventor [12] を利用し、同ソフトウェアがサンプルプログラムとして提供している VRML ビューアに顔位置認識部を組み込む形で、プロトタイプ開発と実験、アルゴリズムの検証を行なっている。

VRML ビューアと組合せたプロトタイプシス

テムでは、汎用的な 3D モデルビューアを実現できており、モデルデータを VRML 形式で与えることにより任意のモデルを覗き込むように観測することができる (図 9)。



図 9: FacePoint プロトタイプシステム

現段階でのデモ用プロトタイプシステムでは、CPU が 1.8 GHz の Pentium 4、搭載メモリ 768MB、GForce 3 によるハードウェアアクセラレーションを有効にしたレンダリング、という実行環境で毎秒 10~12 フレームの性能を確保している。ユーザインタフェース用の実時間性能としてはほぼ問題ない速度を実現できているレベルである。

6 まとめ

Linux で利用できるオープンソース顔検出認識システムとして開発した MACAO および FacePoint の設計と実装について報告した。

MACAO ライブラリおよびテストベッドをまとめたソースコード一式および関連ドキュメントを、現在、下記 URL で公開しているので、興味を持たれた方にはぜひとも評価していただきたいところである。なお MACAO ライブラリの利用許諾条件は LGPL としている。

http://www.malib.net/applications/index_j.html

まだフレームワークが用意された段階であり、実装されている認識アルゴリズムの拡充は今後の課題である。本フレームワークを評価していただいた開発者の方々から、積極的な実装

の提案をいただければ幸である。

また 3 次元顔位置検出システム FacePoint は、現時点ではまだアルファ版の品質を伴わないプロトタイプ実装であることと、入力機器として特殊なステレオカメラを用いていることからオープンソース開発のやり方にそぐわないなどの理由から、まだ公開には至っていない。ただし希望があればソースコードを公開してオープンソース型開発を進めることに問題はない。

例えば市販の USB カメラ 2 個でステレオ視を実現する研究もあり [13]、これらの成果を取り入れるなどによりハードウェア的な開発の敷居が低くなった時点で、こちらもオープンソースソフトウェアとして公開していきたいと考えている。

参考文献

- [1] R. W. Frischholz, "Face Detection Home Page - Software," <http://home.t-online.de/home/Robert.Frischholz/facedetection/software.htm>
- [2] 飯尾, 谷田部, 比屋根, "顔の向きと頭部位置を利用したユーザインタフェースの試作," 第 8 回画像センシングシンポジウム, pp.573-576, Jul. 2002.
- [3] 飯尾, 谷田部, 比屋根, 米元, 谷口, "動画画像処理ライブラリ MALib を利用した 3 次元ユーザインタフェースの実装," オブジェクト指向 2002 シンポジウム, pp.117-120, Aug. 2002.
- [4] 飯尾, 谷田部, 比屋根, "観測位置計測を備えた 3D モデル表示システムとその応用," 3D 映像, 17 (2), pp.43-46, 2003.
- [5] 飯尾, 谷田部, 比屋根, "汎用動画画像処理ソフトウェアライブラリ- MALib -, " Linux Conference 2001, <http://lc.linux.or.jp/lc2001/papers/malib-paper.pdf>, Sep. 2001.
- [6] 飯尾, 谷田部, 比屋根, "Linux における動画画像処理の現状と応用," Linux Conference 2002, <http://lc.linux.or.jp/lc2002/papers/iio0920h.pdf>, Sep. 2002.
- [7] E. Hjelmås, "Face Detection: A Survey," *Computer Vision and Image Understanding*, 83, pp.236-274, 2001.
- [8] M. Yang, N. Ahuja, and D. Kriegman, "Detecting Faces in Images: A Survey," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24 (1), pp.34-58, 2002.
- [9] Point Grey Research Inc., "Products and Services :: Bumblebee," <http://www.ptgrey.com/products/bumblebee/index.html>

- [10] 飯尾, 谷田部, 比屋根, “Disparity Map を用いたユーザ位置の検出と 3 次元インタフェースへの適用,” 第 2 回情報科学技術フォーラム (FIT2003), 情報技術レターズ, Vol.2, pp.271–272, Sep. 2003.
- [11] J. Iio, T. Yatabe, and K. Hiyane, “3D Model Viewer with Real-time Viewpoint Tracking System,” The 9th IFIP TC13 International Conference on Human-Computer Interaction (INTERACT2003), pp.1107–1108, Zürich, Sep. 2003.
- [12] Silicon Graphics Inc., “SGI – Developer Central Open Source | Open Inventor,” <http://oss.sgi.com/projects/inventor/>
- [13] D. O. Gorodnichy and G. Roth, “Affordable yet Robust and Precise Face Tracking using USB Cameras with Application to Designing Hands-free User Interfaces,” The 15th User Interface Software & Technology (UIST’02), Paris, Oct. 2002.