

Linuxにおける動画画像処理の現状と応用

Video Image Processing with Linux and Its Applications

飯尾 淳*

iiojun@mri.co.jp

谷田部 智之*

tyatabe@mri.co.jp

比屋根 一雄*

hiya@mri.co.jp

概要 我々は動画画像認識処理のアルゴリズムを簡単に実装するフレームワークの提供を目的として汎用動画画像処理ソフトウェアライブラリ「MAlib」を開発し、オープンソースソフトウェアとして公開している。本稿では、主にオープンソースとして公開されている技術を中心としてLinuxプラットフォームにおける実時間画像処理技術の動向を検討する。さらにMAlibの現在の開発状況、MAlibの応用例、およびMAlib開発の今後の展望について述べる。

1 はじめに

DVDプレーヤやHDR(ハードディスクレコーダ)などの映像処理装置とPCの境界線は今日ではもはや曖昧であり、Linuxでもこれらの処理を行なうフリーのソフトウェアが現れ始めている。また家庭用の映像処理機器のための組込みOSとしてLinuxが採用されたもの(日本電算機製iBOX Server)が発表されるなど、これまでLinuxを載せたPCで画像処理研究を進めてきた我々の立場からすると逆転現象のように受け止められるニュースも報じられている。

動画画像の加工・認識処理に利用できるライブラリも、オープンソースのものがいくつか提供されるようになってきている。我々が開発しているライブラリ「MAlib」[1]も、2002年7月にリリースしたVer.0.5.0では初期の公開時と比較して倍の規模に成長した。MAlibは現在でも開発が進められているが、MAlibの開発に並行して、我々はMAlibを利用したアプリケーションの作成にも注力しはじめている[2, 3, 4]。

本論文ではまずMAlibと他のオープンソースで提供される動画画像認識処理用ライブラリとの比較を行ない、次にMAlibで利用できる入力装置の対応状況および開発の現状報告について述べる。さらに、我々が開発したMAlib応用アプリケーションのいくつかを紹介し、MAlibの使い方、MAlibを利用したことによる利点について報告する。また現時点でのMAlibの課題について述べ、MAlib開発の今後の展望について記す。

2 動画画像認識技術の現状

画像認識やコンピュータビジョンで利用できるライブラリの整備が進み、最近ではリンクスのFusionChecker [5]などLinuxを対象とした商用パッケージも発売されるようになってきている。オープンソースソフトウェアとして公開されているライブラリも、我々によるMAlibの開発と前後して整備が進み、いくつかの優れたものが提供されるようになった。現在は、動画画像認識アプリケーション作成にあたりその用途に応じてライブラリを選択できるまでに状況が改善されている。

2.1 オープンソース動画画像処理ライブラリ

Linuxで動作する動画画像認識処理に活用できるライブラリのうち、オープンソースソフトウェアとして公開されており、かつユニークな特長を持つものを紹介する。

OpenCV

Intelの研究所が提供しているコンピュータビジョン用のライブラリである。学会と協力してコンピュータビジョン用のアルゴリズムを幅広く集めており、様々な画像処理アルゴリズムを提供する。主なものはC++で実装されており、一部がC言語で実装されている。

OpenCVの開発版ライブラリは、SourceForgeのサイト¹からダウンロードすることができる。2002年7月時点でその最新版は、2001年11月末に公開されたbeta2.1バージョン(0.9.3)である。

*株式会社 三菱総合研究所 情報技術研究部

¹<http://sourceforge.net/projects/opencvlibrary/>

表 1: OpenCV、ARToolKit、MAlib の比較

	OpenCV	ARToolKit	MAlib
言語	C++, C	C, MatLab	C
プラットフォーム (太字は主対象)	Windows , Linux	Linux , SGI (IRIX), Windows, Mac (OS X)	Linux , FreeBSD, NetBSD
目的	ビジョン	拡張現実感	多目的
指向	研究指向	研究指向	実用指向
画像認識機能	豊富	AR 向けに特化	基礎的なもの中心
最適化レベル	高い (CPU 依存)	低い	低い
移植性	低い	高い	高い
規模	大 (20 万行超)	小 (2 万行弱)	小 (2 万行弱)
入力 (Linux 版)	V4L	V4L, IEEE1394, DV	V4L, IEEE1394, 他
ライセンス	BSD	GPL	LGPL(部分), GPL

対象とするプラットフォームは Windows および Linux となっているが、上記 URL に記載されている説明によれば Windows がメインターゲットのようである。研究者が中心となって作成しているライブラリのため、画像認識用の機能は豊富に用意されているものの統一性にやや欠ける。また Intel の研究所が作成しているため、Intel 製の CPU に最適化されているとの特長がある。実際、ライブラリを構築する際には、Pentium の各アーキテクチャ毎に最適化したライブラリを生成する。

なお OpenCV は、BSD ライセンスに準拠した Intel のオープンソースライセンスのもとで公開されている。

ARToolKit

拡張現実感 (AR = Augmented Reality) のアプリケーションを容易に作成できるようにすることを目的として提供されているライブラリである。ワシントン大学の Human Interface Technology Laboratory が SIGGRAPH で展示した “Shared Space” プロジェクトをベースに整備されたもので、実時間でマーカの追跡とそのためのカメラキャリブレーションを簡単に実現することができる [6, 7]。

ARToolKit は移植性が高く、主プラットフォームである Linux の他、IRIX、Windows、Mac OS X など様々なプラットフォームでの動作が報告されている。C 言語で実装されているが、最近では MatLab バージョンの同ライブラリも公開されている。他にも ARToolKit の最新版では IEEE1394 カメラをサポートしているという利点もある。

ARToolKit のライセンスは、GPL となっている。ARToolKit は、同研究所の “Shared Space” を紹介しているサイト²から入手可能である。

MAlib

我々が開発を進めている動画像認識処理用のライブラリである。MAlib 開発開始時に既に OpenCV はアルファバージョンとして存在していた。しかし当時の OpenCV はまだ整備が不十分であり利用し難かったことが、MAlib のそもそもの開発の動機である。

OpenCV が完成度を高めた現在でも、実時間で連続的に画像処理を実行するフレームワークを提供する部分や入出力の抽象度が高い点などでのアドバンテージが存在すると考えている。また MAlib の持つ汎用フィルタを OpenCV と組み合わせることで、MAlib の枠組みのなかで OpenCV の機能を活用することも可能である。

MAlib の仕様や特長については文献 [1] を参照していただきたい。また新しく MAlib に追加された機能の詳細については次節で述べる。

また OpenCV、ARToolKit、および MAlib の比較を表 1 に示す。

MAlib 開発の意図として、動画像処理を活用した実用的なアプリケーションの迅速な開発を念頭に置いており、ひとつの研究開発に留まらないような統一的なフレームワークの提供が MAlib の特長となっている。

MAlib のライセンスは LGPL ないしは GPL である。その詳細については後述する。

²http://www.hitl.washington.edu/research/shared_space/

MALib は次の URL で公開しており、ソースコードその他をここから入手することが可能である。

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

その他

その他、Linux で利用できる有望なオープンソースの画像処理ライブラリとしては、ジョンホプキンス大学 Computational Interaction and Robotics Laboratory の XVision2³ や、名古屋工業大学岩田研の S-VIOS⁴ などがある。

前者は C++ で実装されたライブラリで、規模もかなり大きい。XVision2 も IEEE1394 カメラに対応しており、同研究グループでは今後は IEEE1394 カメラの入力に移行することを明示している。また後者は汎用 OS 上における画像処理のための擬似的なリアルタイム性を実現する、解像度を意識しない画像の取扱い、などを特徴としている。S-VIOS も C++ で実装されており、入力には Video for Linux 2 を利用する。

残念ながら現在ではどちらもあまり積極的な開発がなされている様子が伺えない。しかしいずれもそのソースコードが GPL で公開されており自由に利用可能な状態であるので、選択肢のひとつとしては注目に値する。

3 MALib の開発状況

次節で MALib を応用したアプリケーション例を紹介する前に、まずは昨年度の報告時からの MALib 開発の進捗状況を述べる。

3.1 MALib 開発の現状

MALib は平成 12 年度の未踏ソフトウェア創造事業で開発がスタートした。情報処理振興事業協会 (IPA) による支援は終了したもののその後も開発は続き、MALib をベースとして利用している他プロジェクトからのフィードバックに基づき整備が続けられている。

平成 12 年度末の公開時から現在にかけて新たに実装された機能としては、フィルタ群の増強、MPEG やウェブカメラ、IEEE1394 入力などへの対応などがある。また他プラットフォームへの移植などの作業が行なわれ、Linux だけでなく FreeBSD および NetBSD で動作するように修正

³<http://www.cs.jhu.edu/CIRL/XVision2/>

⁴<http://mars.elcom.nitech.ac.jp/svios/>

が加えられている。また PC/AT 互換機以外にも、Linux Familiar をインストールした iPAQ での動作を確認し、iPAQ で動作する簡単な動画アプリケーションを作成した。

3.2 新たに追加された機能

MALib の最新版 (libmalib-0.5.0.tar.gz) に含まれるクラスのクラスツリーを図 1 に示す (斜体は抽象クラス)。

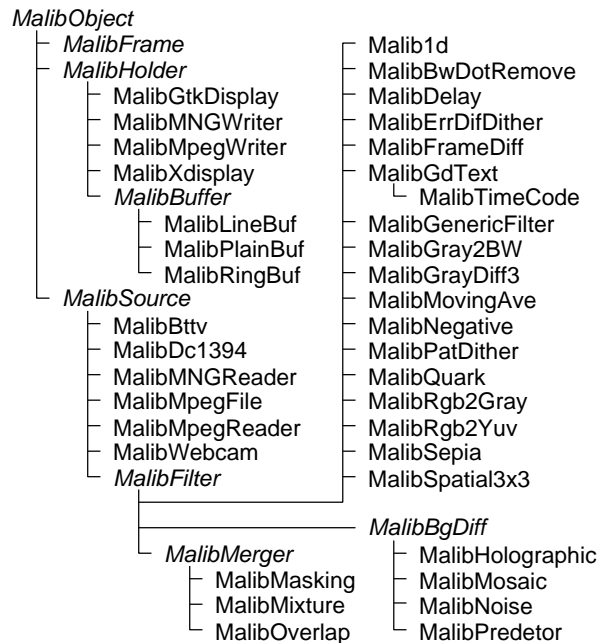


図 1: MALib のクラス構成

新たに追加されたクラスは、主に以下の機能を実現するものである。

各種フィルタ 様々な画像処理を行なうフィルタクラスが追加されている。MalibGdText および MalibTimeCode は文字 (時刻) を画像中に挿入する。MalibPatDither、MalibErrDifDither はディザリングにより二値化するフィルタである。また VJ (ビデオジョッキー) などに利用できるユニークな効果を持ついくつかのフィルタが、EffectTV⁵ の協力のもとに採用された。

入出力 Video for Linux (V4L) を利用した画像入力の他、IEEE1394 カメラ、ウェブカメラ (ネットワークカメラ)、各種ファイルからの入力アダプタとなるクラス (MalibDc1394、MalibWebcam など) が増強されている。また画面出力も GTK+ を利用したもの、ウィジェットを使わず X に直接描画するもの (MalibXdisplay)、ファイル

⁵<http://effectv.sourceforge.net/>

への出力 (MalibMpegWriter、MalibMNGWriter) などが追加された。

なおこの他にも、アルファチャンネルをサポートする RGBA 形式の画像を扱う枠組みが実装されはじめており、一部で利用されている。

3.3 入力ソースの対応状況

これまでの V4L のみの対応に加え、IEEE1394 で接続する非圧縮デジタルカメラやウェブカメラからの入力も可能になっており、様々な入力ソースを選択することができる。動画像解析アルゴリズムの実装側からはこれを MalibSource クラスとして抽象化することができ、入力ソースが交換可能とできることが MALib のひとつの特長となっている。しかし個々の入力クラスのインスタンス化、すなわち実際にハードウェアの設定をする箇所では個別の調整が必要である。

我々が実際に動作を確認した機器を表 2 に示す。

表 2: 入力機器の動作確認状況

ビデオキャプチャボード
GV-VCP/PCI, GV-VCP2M/PCI, GV-BCTV2/PCI, GV-BCTV4/PCI (I-O Data) KW607A (Kowell) VA1000 Lite (AOpen)
USB カメラ
WebCam Go Plus, WebCam 5 (Creative) Catch View, Catch Eye (Novac) QuickCam Pro (Logitech)
IEEE1394 カメラ
DFW-V500 (Sony) iBot (OrangeMicro) Fire-i400 (Unibrain)
ウェブカメラ
AX2100 (Axis Communications) mmEye2 (Brains)

V4L インタフェース

上記のうちビデオキャプチャボードからのアナログ入力と USB カメラは、V4L インタフェースを介して利用することが可能である。したがって MalibBttv クラスをそのまま用いて入力ソースとすることができる。実際には、とくに USB カメラの場合には、個々の機器のドライバの設定を

行なわないとうまく画像を取得できないので注意が必要である。またデバイスノードとチャンネルの設定が必要になる場合がある。

最新版の MALib では、簡単なプローブを行ない半自動で入力を設定する機能が盛り込まれているが、対応できない場合は再コンパイルが必要になる可能性が残っている。この点は設定ファイルで修正できるようにする予定であり、今後の課題として残されている。

IEEE1394 カメラ

1394 Trade Association のデジタルカメラサブワーキンググループが制定した産業用デジタルカメラ仕様 (IIDC Spec. = Instrumental and Industrial Digital Camera Specification)[8] に準拠した非圧縮画像転送を IEEE1394 上で行なうカメラからの入力が可能である。本機能は、同仕様に従った API を実装する libdc1394 プロジェクト⁶が提供するライブラリを利用して実現した。

ただし libdc1394 はまだ仕様が安定していないことに注意が必要である。現在公開されている libdc1394 のスナップショットでは MALib と正しくリンクできない場合がある。同プロジェクトが公開している CVS から最新版を取得して作成した libdc1394 とリンクしてほしい。

なお MALib に実装されている IEEE1394 カメラの入力クラスは現在のところまだプロトタイプ実装であり、一台のカメラからの入力しか対応していない。複数台のカメラから同時に画像を取得できるようにする修正作業を現在行なっている。

ウェブカメラ

ネットワーク経由でアクセスし、画像を取得するデバイスである。通常は HTTP でアクセスすると JPEG で圧縮された静止画を返す。インターネット上で定点観測サイトなどを実現する場合に利用されることが多い。

MALib にウェブカメラ用のアダプタとして用意されている MalibWebcam クラスは、汎用性を考慮して一回ごとに HTTP のセッションを確立し、静止画を逐次取得する設計となっている。したがって効率は非常に悪く、動画とは言い難い。カメラデバイスの実装によっては、MPEG を送信するものも存在する。これらの機器仕様に合わせてチューニングするには、MalibWebcam クラ

⁶<http://sourceforge.net/projects/libdc1394/>

スを継承した新しいクラスを作成すればよい。

例として、Axis Communications の AX2100 用にチューニングを施されたクラスのサンプルコードが MALib ソースツリーの test/webcam/ ディレクトリに置かれている。AX2100 は HTTP のセッションを keep-alive で保持しつつ、撮影した JPEG データを順次送信する機能を持っている。ここで定義される MalibWebcamAx2100 クラスは、それに対応した画像の受信を実現することにより多少の性能向上を実現している。

動画像ファイル

以上に挙げた入力機器への対応の他、MNG ライブラリや FFmpeg⁷ が内部で利用するライブラリを共にリンクすることで、MNG や MPEG などのファイルを入力ソースとして活用することも可能となっている。

3.4 MALib のライセンス

MALib は製品に組み込むことも想定し、導入のしやすさを考慮して基本的には LGPL で提供したいと考えている。ただし MALib がリンクで利用するライブラリには MPEG デコーダやその他の GPL 準拠のライブラリが含まれる。例えば先に述べた FFmpeg の場合には、configure のオプションとして FFmpeg を構築したディレクトリを指定して FFmpeg のライブラリをリンクし、MPEG 関連の機能を利用できるようにしている。しかし FFmpeg は GPL で提供されているため、FFmpeg の一部をリンクした MALib もまた GPL とせざるを得ない。

そこで MALib のバイナリ構築時に GPL 依存部分を切り離して構築できるような設計とし、GPL で提供される機能を必要としない場合に限り LGPL でも利用できるようなライセンスを採用した。先の例では、MPEG 入出力機能が必要でなければ FFmpeg のコードとのリンクを行なう部分を MALib から削除することができる。具体的には、FFmpeg の機能を利用する MalibMpegReader、MalibMpegWriter は FFmpeg と共にコンパイルするという configure オプションを与えない限りは構築されない。その他の GPL 依存部分に関しても同様である。したがって、これらの機能が不要な場合は、MALib を LGPL のまま提供することが可能となっている。

⁷<http://ffmpeg.sourceforge.net/>

4 MALib アプリケーション例

MALib が持つ機能を活用したアプリケーションの数々を紹介する。個々の実装で MALib が効果的に利用されている点に着目していただきたい。

4.1 ハンドヘルド画像処理装置

もともとが組み込み用 Linux 向けの画像処理基盤ソフトウェアとの題目のもとに開発が進められた [2] との経緯もあり、まず、組み込み向け Linux (Linux Familiar) を利用して、iPAQ で動作する画像処理装置のサンプルを作成した (図 2)。



図 2: ハンドヘルド画像処理装置

概要

図 2 に示す iPAQ は、Windows CE を削除して Linux Familiar ver 0.5 に入れ換えられたものである。また初期化スクリプトに細工をしてブートすると画像処理アプリケーションが起動するようになっており、完全なハンドヘルド画像処理アプリケーションとして機能する。

この iPAQ で動作している画像処理アプリケーションは、ウェブカメラからの画像を受信して表示するウェブカメラビューアである。その骨子はほぼ MalibWebcam と MalibGtkDisplay を直結したものと考えてよい。

本アプリケーションでは、206MHz の StrongARM を搭載した iPAQ H3660 において毎秒 5 フレーム程度の表示性能を得た。実験に用いたウェブカメラの AX2100 は毎秒 10 フレーム程度の画像送信能力を持つので、およそ半分のデータの欠損が認められる。

開発手法

開発環境には式神 [9] が提供する SDK を利用した。実際には開発時に iPAQ 用の V4L で駆動

できる画像入力デバイスが無かったため、MAlibのソース一式から MalibBttv に関連するコードを削除してライブラリを構築した。

式神 SDK が提供する ARM 用ライブラリがほとんど問題なく整備されていたことにより、移植に際してはソースコードの内部に手を入れる必要がほとんど無く、労力をかけずに移植作業を遂行できた。したがってアプリケーションの開発とデバッグを x86 ワークステーションで行ない、完成したものを式神 SDK と共にリコンパイルして iPAQ にインストールする、という効率のよい開発スタイルをとることができた。循環的な開発サイクルにおいて、そのメリットは非常に大きい。

4.2 3D ユーザインタフェースエージェント

次に実時間姿勢認識によるユーザインタフェースエージェント [3] を作成した (図 3)。本アプリケーションは、複数台接続した IEEE1394 カメラのステレオ画像からユーザの顔と手先の位置を計測する。その結果に逆運動学解法⁸を適用し、利用者の姿勢を実時間でアバターに反映させるものである。

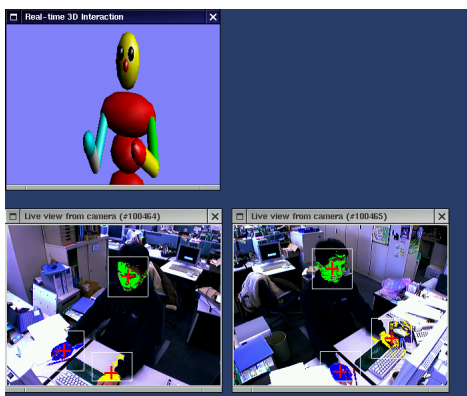


図 3: 3D ユーザインタフェースエージェント

概要

このシステムでは、システムを操作するユーザの正面上部にユーザの挙動を実時間観測するカメラが複数台設置されている。図 3 に示された 3 個のウィンドウのうち、下の 2 個にはそれぞれ左斜め上および右斜め上からユーザを観測する画像が表示されている。

このアプリケーションでは、まずユーザの顔と手先の空間的な位置を認識する。それぞれのカメラ

⁸肩や肘の回転角と腕の長さから手の位置を求めることを運動学といい、その逆に手先位置から関節位置を推定することを逆運動学という。

ラでは肌色と思われる画素の塊 (ブロップ) を認識し、それらの画像上での位置を決定する。それぞれのカメラの 3 次元位置は事前にキャリブレーションを行なっておく [10]。したがってそれぞれのカメラの画像上での位置が決定すれば三角測量の原理で各ブロップの空間上の 3 次元位置を求めることが可能となる。

このようにして求めた空間上の手先位置と顔の位置から、逆運動学の解法を適用して肩肘の位置を推定する。その結果をリアルタイムに反映した結果が図 3 上部に示されたウィンドウに表示されている人形 (アバター) である。

MAlib の利用

本システムにおいて、各カメラで得られる画像中の顔位置、手先位置を求めるアルゴリズムの実装に問しては、まさに MAlib の能力を發揮できる部分である。これらは各カメラ毎に処理が行なわれ、最終的に統合されて 3 次元位置が決定される。個々のカメラで行なわれる処理の手順は以下の順序に分割することができ、まさにそれぞれが MAlib のフィルタクラスとして実装され得るものである。

1. 各カメラから入力された画像と、あらかじめ用意しておく背景画像との差分をとる。その結果、一定の閾値を超えたものだけを処理の対象とする。
2. 背景から分離された画素を対象として、肌色に近い画素を抽出する。なお肌色の分別に関しては、明度の影響を受けることが少ないとされている手法 [11] を用いた。
3. 各カメラ毎に抽出された肌色画素のブロップをトラッキングし、最後にそれらの 2 次元位置情報を統合して 3 次元位置を決定する。

図 4 は以上のプロセスを実現するクラスの関連とデータの流れを示すブロック図である。

各カメラ毎に上記の 1. および 2. の手順がそれぞれ MarioBgDiff および MarioBlob2D フィルタとして実装され、各カメラのデータに対して個別に動作する。各カメラからの出力をまとめるクラス MarioBlob3D が、全ての 2 次元情報から 3 次元位置の計算を行なうクラスである。

なおこれらの画像処理は各カメラ毎に順次呼び出されることで実現されているが、アバターを

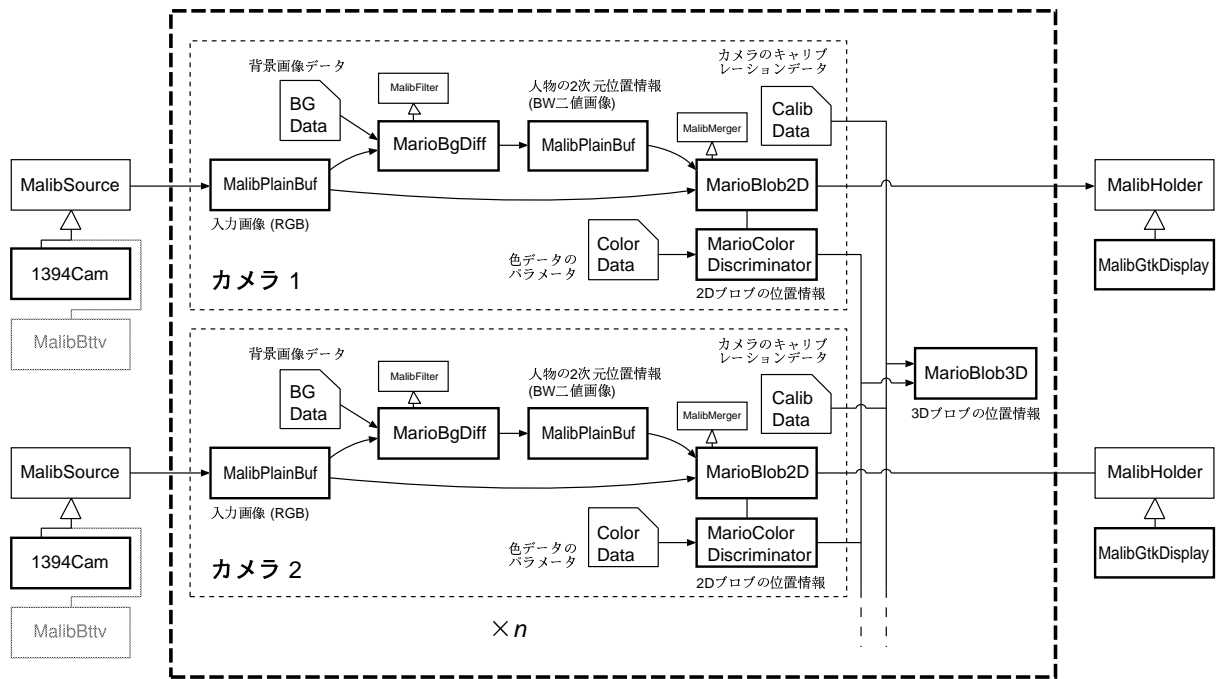


図 4: 画像認識処理のブロック図

表示する手続きとは別スレッドで動作する設計としている。

MAlib 利用によるメリット

MAlib を利用することにより、このような連続的なフィルタの適用という形式を用いて、アルゴリズムを素直に実装することが可能になっている。

もうひとつ、MAlib を利用した重要な利点を述べておく。今回の開発では実績と効率を重視して入力に IEEE1394 カメラを用い、またそのドライバとして九州大学で作成されたものを利用している。IEEE1394 上の非圧縮転送をサポートするカメラは性能はよいが、一般への普及は遅れているうえ USB カメラと比較して高価であり入手も容易ではない。

本アプリケーションは、MAlib の利用によりカメラデバイスの交換が非常に簡単になっている。実際、カメラの焦点距離など 3 次元キャリブレーション用のパラメータさえ分かれば、USB カメラあるいはビデオキャプチャボードを利用したアナログカメラからの入力に変更することは多少の修正を施すだけで可能である。

4.3 デムパ

撮影された画像からユーザの顔領域を実時間で切り取り、あらかじめ用意した背景に貼り付け

るアプリケーションの試作を行なった (図 5)。

アプリケーション名の由来となっているオリジナルのアイデアでは、ブルーバックを利用したクロマキー合成で人物を CG 画像の背景上に登場させている。それに対して本アプリケーションでは人物の肌色領域を抽出するためブルーバックを必要とせず、環境を問わず動作するという長所を持つ。反面、肌色および肌色で囲まれる領域しか認識しないので、髪の毛が表示されないという問題がある。



図 5: デムパ

本アプリケーションの目的

本アプリケーションは、当初はただ楽しむための、あるいは MAlib 利用のサンプル的なソフトウェアとして開発された。しかし現在では他のアプリケーションで利用するパラメータを対話的に

調整するためのツールとして重要な役割を担うようになっている。

肌色認識や顔領域抽出のアルゴリズムは、前述の3次元ユーザインタフェースエージェントあるいは次に述べるバーチャル地球儀用に作成されたものを流用したものであり開発成果は相互に還元されている。これらのアプリケーションにおける肌色領域認識の基礎は、次のとおりである。

1. まず画素のRGB値を以下の式により変換しab平面に射影する ($R = G = B = 0$ のときは、 $a = b = 0$ とする)。

$$a = \frac{2R + G}{2(R + G + B)} \quad (1)$$

$$b = \frac{\sqrt{3}G}{2(R + G + B)} \quad (2)$$

2. 変換した a, b の値が、中心を (a_0, b_0) 、それぞれの半径を (r_a, r_b) とする楕円領域に含まれるか否かどうかを判定する (図6)。

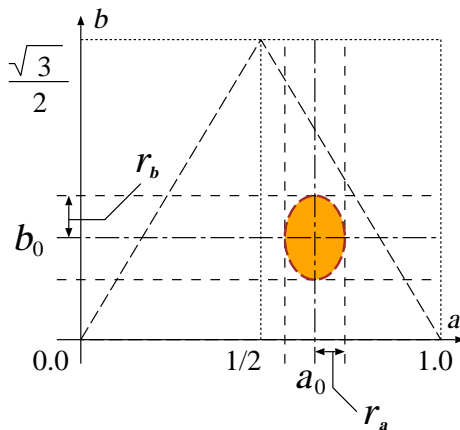


図 6: 肌色領域の判定

本アプリケーションは、この肌色領域を設定する4つのパラメータ a_0, b_0, r_a, r_b を対話的に調整する機能を備えており、設定情報をテキストファイルで与え、適宜プログラムから読み直すことで認識状況を確認することができる。現在この機能をGUIにより簡便に再設定することでパラメータの設定ツールとしての操作性を高める作業を行なっているところである。

4.4 バーチャル地球儀

3次元ユーザインタフェースエージェントの研究開発成果を応用し、仮想空間に配置された3D-CGオブジェクトを投影する視点を、ユーザの頭

部位置の移動により操作することで、直感的な観測を可能とするアプリケーション [4] を試作した。観測する3D-CDオブジェクトは何でもよいが、本アプリケーションでは、簡単なモデルでありながらデモンストレーション効果の高い「地球」を表示させている (図7)。

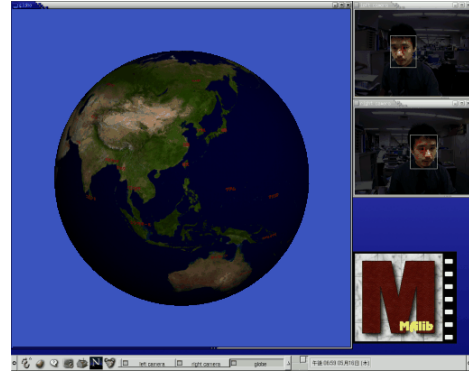


図 7: バーチャル地球儀

概要

図7の右上に示す2枚のウィンドウでは、3次元ユーザインタフェースエージェントと同様にユーザの顔位置を認識している状況が表示されている。また左側に大きく表示されている地球が、ユーザの視点から観測された3D-CGオブジェクトのレンダリング結果である。地表面のイメージは、球体へのテキストチャマッピングである。

基本的なアイデアは、ユーザの顔の実空間上における位置 (以後、観測点とする) を仮想空間上で投影変換を行なう際の視点位置 (以下、視点とする) に対応付けてから投影することにより、仮想オブジェクトを投影した像の生成を観測点の実空間上における位置でコントロールする、というものである (図8)。

本アプリケーションではやはり各カメラにおける顔位置の認識部分についてMALibをベースに実装し、認識と3次元位置の決定までのスレッドを仮想オブジェクトのレンダリングと並行に行なう設計となっている。

観測点と視点の対応

仮想空間と実空間の対応付けの一方法としては、ディスプレイに実際に表示されている位置へ仮想オブジェクトを配置することが考えられる。

仮想オブジェクトをこのように配置すると、ディスプレイに表示された仮想オブジェクトを覗きこむように顔を近づけることで仮想オブジェクト

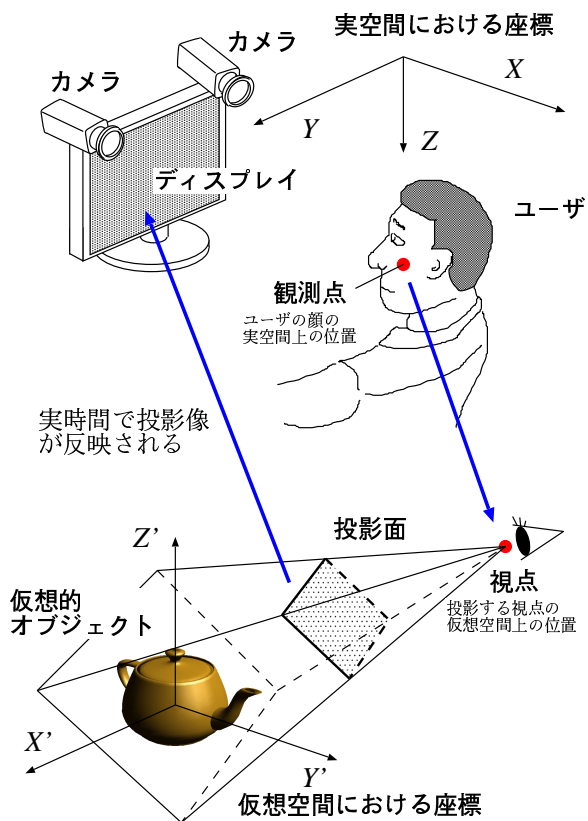


図 8: 観測点と視点の対応

が大きく表示される、したがって詳細な表示が得られるようになる。また上から覗きこめば仮想オブジェクトの上部を観測することができ、左右から観察することで仮想オブジェクトの左右の面を確認することができる。

バーチャル地球儀のアプリケーションにおいては、仮想オブジェクトが地球であることを考慮し、以下の 2 点の技巧を加えている。

- 上下左右方向に関しては、観測点の実空間上における移動量を、仮想空間上において仮想オブジェクトを中心とする極座標を考慮した際の視点の角度の変化量に対応させる。
- 上下方向にはさらに、一定の角度以上は変化しない制限を課す。

前者により、オブジェクトの裏側を観測することが可能となる。仮想オブジェクトをディスプレイ位置に仮想的に配置した場合、幾何学的にオブジェクトの裏側を観測することは不可能である。地球儀の場合は、くるくると回してみることが多いと考えられ、この制限は適切ではない。

そこで観測点の中心からの移動量を視点の角度の変化量に比例させるような変換を施す。この変

換により、観測点を十分に移動させることで角度の変化量を 180 度以上にすることが可能となる。

バーチャル地球儀のアプリケーションでは、ユーザがディスプレイの正面に正対した場合まず日本を中心として表示するように設定してあるが、この機構により左側から覗くことによりヨーロッパを中心として観察することができ、また右側から覗きこむことによりアメリカ大陸を表示させることが可能となっている。

後者の制限は、上下方向に必要な以上の回転を加えないようにするものである。通常の地球儀では北極と南極を結ぶ直径を軸としているので、上下方向には回転しない。しかし観測点と視点の変化量を線形に対応させただけでは極を中心とする観測状態をうまく表現できない。そこで、上下方向からの自然な観測を可能としつつ過度の回転を防ぐために、一定の角度以上は変化させないような制限を加えた。

本技術の想定応用例

本開発の成果として得られた技術で強調したいことは、3D モデルの観察と操作に関して、直感的なユーザインタフェースを実現できるということである。

これまでの一般的な 3D モデリングソフトウェアでは、高価な専用の 3 次元操作デバイスを利用しない限りはユーザに苦痛を強いてきた。すなわちそれはマウスでの 3 次元オブジェクト操作に無理があるためであり、XYZ 方向および各軸の回転を含む空間上の 6 自由度の変更を、2 次元上の自由度しかないマウスによる操作で対応する点に大きな問題がある。

さらに今までのユーザインタフェースでは、視点の変更とオブジェクトの変型などオブジェクトそのものに対する操作を同時に実現できないため、マウスのボタンを押しながら操作するとか、メニューで切り替えるといったモードの意識を伴わなければならない。

本技術を応用すれば、視点の移動に関してはオブジェクトの操作と確実に分離させることが可能となるので、その操作はより直感的になることが期待される。例えば 3D のモデリングを行なう際に、好きな位置から観察しつつオブジェクトを修正する、といった作業ができるようになる。

またマウスのような機器を用いることなく覗きこむだけで自然な角度から 3 次元オブジェクトを

観察できるというインタフェースは、情報機器の操作に不馴れなユーザであっても容易に扱える可能性を持つ。その応用には、博物館におけるバーチャルオブジェクトの観察装置、遠隔操作システムのビジョン部分への適用、立体表示装置の提示方向の検知など、様々な例が考えられる。

5 今後の予定

最後に MALib が現在抱えている課題を示し、今後の展開について述べる。

5.1 MALib 開発の課題

以前からの課題として残されているものとしては、映像系アプリケーションへの対応を想定した音声・画像同期機構の実現がある。開発するアプリケーションがどうしてもコンピュータビジョンよりのものであるため、課題として残されておりながらもなかなか手を付けられていない。

また現状の MALib では適切なタイミング制御が実装されておらず、入力ソースにおいてはオンデマンドでその時点の画像を取得するのみである。この点に関しても、計算資源の乏しい環境では適切なコマ落ちを許し、あるいは高性能な環境では適度なウェイトを与えることができるようにするなどの機能が望まれる場合もある。

今後、サンプルアプリケーションのひとつとして映像系アプリケーションを作成し、これらの機能を実装していきたい。我こそはと思いき方はぜひ開発に協力していただきたいところである。

5.2 各アプリケーションの実用化

さらに、我々 MALib 開発グループの今後の展望としては、MALib を増強すると同時に各アプリケーションの実用化にも注力していきたいと考えている。

とくにバーチャル地球儀で示した直感的なユーザインタフェースの実現は、他分野に応用できる有望な技術として更なる研究開発を行なう予定である。具体的には認識の安定化、ウォークスルー型への適用、先に述べたような他アプリケーションへの実装などを検討中である。

6 まとめ

本論文で示したように、現在では、動画像処理、コンピュータビジョン関連の有用なライブラリがオープンソースとして利用できる恵まれた

環境が用意されつつある。ほんの数年前と比較しても格段の進歩である。

このような状況を鑑みると、これからは動画像処理を応用したアプリケーションが益々発展することが期待される。MALib がこれらの研究開発に少しでも寄与し、本論文で紹介したようなアプリケーションが多々生み出されることを望み、まとめとしたい。

謝辞

MALib 開発に関わる全ての皆様に感謝します。

参考文献

- [1] 飯尾, 谷田部, 比屋根: “汎用動画像処理ソフトウェアライブラリ -MALib-”, LC2001, <http://lc.linux.or.jp/lc2001/papers/malib-paper.pdf>, 2001.
- [2] 飯尾, 谷田部, 比屋根: “組み込み Linux 向け動画像処理基盤ソフト「MALib」”, IPA 平成 12 年度成果論文集 CD-ROM, 2001.
- [3] 飯尾, 谷田部, 比屋根, 米元, 谷口, 長島, 土原: “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”, IPA 平成 13 年度成果論文集 CD-ROM(掲載予定), 2002.
- [4] 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”, SSII2002, pp.573-576, 2002.
- [5] 村上, 豊高, 村田: “マシビジョン開発ツール FusionChecker ~アカデミックの研究成果の産業界への還元を試みるツール~”, SSII2002, pp.585-590, 2002.
- [6] Kato H., Billingham M.: “Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System”, IWAR'99, pp.85-94, 1999.
- [7] 加藤, Billingham M., 浅野, 橘: “マーカー追跡に基づく拡張現実感システムとそのキャリブレーション”, 日本バーチャルリアリティ学会論文誌, Vol.4, No.4, pp.607-616, 1999.
- [8] 1394 Trade Association: “IIDC 1394-based Digital Camera Specification Version 1.30”, http://www.1394ta.org/Download/Technology/Specifications/2000/IIDC_Spec_v1.30.pdf, 2000.
- [9] 大谷: “携帯機器向け GUI 環境「式神」の実装”, LC2001, <http://lc.linux.or.jp/lc2001/papers/sikigami-paper.pdf>, 2001.
- [10] Tsai: “A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses”, IEEE Journal of Robotics and Automation, Vol.RA-3, No.4, pp.323-344, 1987.
- [11] 川戸, 鉄谷: “顔領域抽出を目的とした肌色モデルと肌色領域抽出”, PRMU2001-59, pp.143-148, 2001.