

IUE (画像理解用標準ソフトウェア) と Calibrated Image Database の現状

松山隆司 (京都大学) 谷口倫一郎 (九州大) 浅田尚紀 (広島市大)
 荒川賢一 (NTT) 佐藤宏介 (奈良先端大) 全炳東 (東京商船大)
 田中弘美 (立命館大) 寺内睦博 (広島市大) 松尾啓志 (名工大)
 和田俊和 (京都大学)

米国を中心として設計、開発が進められてきた画像理解用標準ソフトウェア IUE (Image Understanding Environment) と、IUE プロジェクトに対する日本からの貢献として進められてきた CIDB (Calibrated Image Database: 正確なキャリブレーション-物理パラメータ付きの画像データセット) の成果がまとめられ、現在一般ユーザが利用可能なソフトウェア、データベースとしてほぼ整備された状態になっている。これらは、今後も改良・内容の充実が図られていく予定であるが、本稿では、IUE プロジェクトの現状とその成果を利用して何ができるのかについて述べる。

Image Understanding Environment and Calibrated Image Database: Status Report

T. Matsuyama (Kyoto Univ.) R. Taniguchi (Kyushu Univ.)
 N. Asada (Hiroshima City Univ.) K. Arakawa (NTT)
 K. Sato (Nara Inst. of Sci. & Tech.) H. Zen (Tokyo Univ. of Mercantile Marine)
 H. Tanaka (Ritsumeikan Univ.) M. Terauchi (Hiroshima City Univ.)
 H. Matsuo (Nagoya Inst. of Tech.) T. Wada (Kyoto Univ.)

This paper gives an overview of the current status and products of

1. Image Understanding Environment (IUE): A comprehensive software system for computer vision and image understanding developed under US initiative.
2. Calibrated Image Database (CIDB): A set of images taken under well calibrated environments developed as the original Japanese contribution to the IUE project.

After several years intensive efforts and international collaborations, IUE Version 2.0 is released for practical use in various computer vision and image understanding applications. While the size of CIDB is limited, it includes useful calibration information and image data to be used for testing the performance of various computer vision and image understanding algorithms. The paper describes what can be done using current IUE and CIDB products.

1 はじめに

米国を中心として設計，開発が進められてきた画像理解¹用標準ソフトウェア IUE (Image Understanding Environment) と，IUE プロジェクトに対する日本からの貢献として，CVIM 研究連絡会が RWCP (新情報処理開発機構) の援助を受けて作成してきた CIDB (Calibrated Image Database: 正確なキャリブレーション-物理パラメータ付きの画像データセット) の成果がまとめられ，現在一般ユーザが利用可能なソフトウェア，データベースとしてほぼ整備された状態になっている。本稿では，IUE プロジェクトの現状とその成果を利用して何ができるのかについて述べる。

IUE および CIDB 開発の動機，目的，意義についてはすでに文献 [1] で述べた通りであるが，要約すると次のように言える。

長年に渡る画像理解研究の成果を基に，画像理解を確固たる基盤を持った科学技術分野として確立し，研究者間での情報交換，研究機関から企業への技術移転を促進するために，

1. 画像理解の基本となる概念の整理・体系化
2. 画像理解ソフトウェアの標準化
3. 画像理解アルゴリズム評価のための画像データベースの開発

を行う。

IUE は，オブジェクト指向言語 C++²を用いて，世界中の研究者，技術者が共通に実行，評価，再利用可能な画像理解ソフトウェア開発環境を作ろうというプロジェクトである。IUE は，SPIDER[2] や SLIP[3]，VIEW-Station[4]，PIK[5] といった従来の画像処理ソフトウェアと異なり，画像データだけでなく，多次元ベクトル，マトリクス，座標系，点列，グラフなどを含む豊富なデータ型を提供するクラスライブラリ，それらを用いて作成された基本処理ツール群であるタスクライブラリ，Visual Programming インターフェイスを含む実行環境から構成される。特に，クラスライブラリには，画像の構造的解析および 3 次元シーンの認識・理解のために必要となる多種多様な基本概念を表す多数の型定義が含まれており，多くの部分はグラフィックスや CAD，地理情報システムといった空間データを処理する他の研究分野でも利用可能である。

一方，CIDB は，SIDBA[7] や通常の画像データベースとは異なり，各画像データには，撮影に用いたカメラ較正用のデータ，撮影対象の 3 次元形状や分光スペクトル特性といった精密なキャリブレーションデータ

¹ここでは，英語名との対応から「画像理解」という言葉を統一的に使うが，その内容はむしろ「コンピュータビジョン」と言うのが適切である。

²当初は Lisp も基本言語として考えられていたが，現在では実用的有用性の高い C++ のみとなっている。

が添付されており，処理結果の妥当性や精度を客観的，定量的に評価できる画像理解アルゴリズム評価用画像データベースである。

以下では，本年リリースされた IUE Version 2.0 の技術的内容および，データの公開準備が整った CIDB に格納されている画像データとその利用法について述べる。IUE および CIDB についての技術的詳細やソフト，画像データの入手法については，ホームページ (<http://vision.kuee.kyoto-u.ac.jp/IUE/>) を参照して頂きたい。

2 画像理解プログラミングと IUE

ここでは，画像理解プログラムにおいてどのような機能が必要であり，IUE ではそれをどのように実現しているのかについて具体例を上げながら説明した後，IUE Version 2.0 の概要について述べる。

2.1 画像理解プログラミングの課題

一般に，入力画像から 3 次元シーンの記述を生成する画像理解システムを構築するには，次に示すように様々なデータを扱う必要がある。

画像データ フィルタリングやエッジ検出等の画像処理段階で扱う。

幾何学的データ 線分，領域，輪郭，曲面，ボクセルなど，画像データを解析することによって得られる。

記号データ 幾何学的データ間の関係記述や知識表現など，認識・理解の処理段階で扱う。

画像理解においては，このように多様なデータ型を扱うプログラミングが不可欠である。ここでは，画像理解において扱うデータ型を 3 種類に大別したが，これらはさらに細かく分類される。例えば画像データ一つとっても，2 値画像，濃淡画像，RGB カラー画像，マルチスペクトル画像，複素画像など，画素の値を表現するための「要素型」として，整数・実数，N 次元の整数・実数ベクトル，複素数など，実に多様なものが考えられる。それらの要素型から成る配列，木構造などの「データ構造」の多様性を考えると，画像データを表現するための実際の型としては膨大な種類のものが有り得ることが分かる。特に幾何学的データの場合には，型定義の任意性が高いため，型の種類は非常に多くなる。

データ型が多様である場合，それら全てを扱うことができる単一の画像理解プログラムを作成するためには，データの型に応じて適切な演算を行わなければならない。例えば単に画素値の加減乗除を行う場合でも，整数・実数，整数・実数ベクトル，複素数，など型が異なる場合には演算の内容も異なるため，各型に応じ

た演算ルーチンを用意しておき、与えられたデータの型に応じた演算ルーチンを呼び出す機能が必要である。しかし、C言語のような通常のプログラミング言語にはこのような機能があらかじめ用意されていないため、多様なデータ型を扱うプログラムを作成することは困難である。この結果、特殊なデータしか扱えない一般性のないプログラムが氾濫し、異なるアルゴリズムに基づくプログラム同士の効率や精度を比較することができないだけでなく、研究が終わるとプログラムが捨てられてしまうことになる。

以上のように、画像理解プログラミングにおいては「型に依存しないプログラミング」の実現が強く望まれており、それによって

- 異なるアルゴリズムに基づくプログラムの比較
- プログラムの再利用
- アルゴリズムに近い抽象度の高いプログラム作成

が可能になるものと考えられる。

異なるアルゴリズムに基づく複数のプログラムの比較ができれば、各アルゴリズムによって何ができ、何が出来ないのかを明確にすることができる。また、プログラムの再利用ができれば、既知のアルゴリズムに基づく処理を自分の作成するシステムの一部で用いる場合に、新たにプログラムを作成する作業が不要になり、研究の本質的な部分に集中することができる。さらに、抽象度の高いプログラミングを行うことができれば、プログラムが行う処理の本質的な部分が読み取りやすくなる。

2.2 IUE におけるプログラミング

IUE では、オブジェクト指向言語 C++ の持つ 1) データ抽象化、2) 型継承、3) テンプレート、4) 仮想関数、等の機能を駆使し、型に依存しない画像理解プログラミングを実現するための枠組みを提供している。

データ抽象化とは、データと、データに対する操作関数を 1 つのまとまりのある実体として扱うことである。この場合、データの型定義には、データの構成要素（メンバ）の型とそれに対する操作関数（メンバ関数）の 2 種類の定義が含まれる。例えば 2 次元画像の型定義では、初期化、ファイルからの読み出し、座標値を指定した画素値の読み書き、コピー、型変換などのメンバ関数が考えられる。

型継承とは、あるデータの型定義に含まれているメンバやメンバ関数に新たな定義を付け加えて新しい型を定義する機能である。メンバおよび、メンバ関数は継承元の型から引き継がれるので、型定義を行う際に全てを記述する必要はなく、ユーザはすでに定義された型を利用して新たな型を簡単に定義することができる。前述のデータ抽象化の機能と型継承の機能を持つ

```
//倍精度実数型の画素値を持つ画像
IUE_scalar_image_2d_of<IUE_DOUBLE> DImage(640,480);
//符合なし 8bit 整数型の画素値を持つ画像
IUE_scalar_image_2d_of<IUE_UINT8> U8Image(640,480);
//符合つき 16bit 整数型の画素値を持つ画像
IUE_scalar_image_2d_of<IUE_INT> IImage(640,480);
```

図 1: テンプレートを利用した画像オブジェクトの宣言の例

データは「オブジェクト」と呼ばれ、オブジェクトの型は「クラス」と呼ばれる。

テンプレートとは、クラスや関数の族を定義するために用いられる一種のマクロ機能であり、例えば、画素の型をテンプレートを用いて一般的に表した 2 次元画像のクラスを定義しておき、図 1 に示すように、実際のオブジェクトを定義する際に画素の型を指定するというように利用される。この機能は型定義だけでなく、関数定義にも用いることができる。この機能を用いれば、関数呼び出しの実引数の型が事前に分かる場合には、単一の関数定義で多数の型に対する計算が効率よく記述できる。

仮想関数とは、多様な型に対応する関数を同じ名前でも重に定義しておき、実際の関数呼び出しが起きた際に、実引数に合った適切な関数を動的に起動する「多様型」関数を定義する際に用いられる機能である。仮想関数は、関数テンプレートと同じように、引数の型が多様である場合の関数定義に利用される。しかし、仮想関数の場合には、与えられるデータの型が事前に決まっていない場合にも、実引数の型チェックを行い動的に起動すべき関数を決定するため、呼び出すべき関数がコンパイル時に静的に決まってしまう関数テンプレートとは異なる。

IUE における実際のプログラミング例として、Sobel フィルタを用いてエッジの強度と方向を表す画像を生成する関数 IUE_Sobel_Example を図 2 に示す。関数 IUE_Sobel_Example の第 1 引数 image_in は入力画像を表しており、画素値としてスカラ値を持つ。第 2、第 3 引数の magnitude_image, direction_image は、それぞれエッジの強度と方向を表す画像であり、実数の要素型を持つ。

プログラム中では、これら 3 つの画像の画素に対するアクセスを行うために、ウィンドウアクセス関数 w と、画素アクセス関数 p_m, p_d が用いられている。ウィンドウアクセス関数は、第 1、第 2 引数で指定された (x,y) 座標に、第 3、第 4 引数で指定されたオフセットを加えた位置の画素値を読み出す (get)、変更する (set) という操作を行うものである。画素アクセス関数は、第 1、第 2 引数で指定された (x,y) 座標の位置にある画素に対して get, put を行うものである。これらのアクセス関数の型は、図 2 のようにテンプレートを利用して

```

typedef IUE_scalar_image_2d_of<IUE_DOUBLE> ImageTypeF;
typedef IUEh_image_window_accessor_2d<IUE_DOUBLE,IUE_image_accessor_boundnearest<IUE_DOUBLE> > WindowTypeF;
typedef IUE_image_point_accessor_2d<IUE_DOUBLE,IUE_image_accessor_boundnocheck<IUE_DOUBLE> > PointTypeF;

static void
IUE_Sobel_Example(IUE_scalar_image_2d& image_in, ImageTypeF& magnitude_image, ImageTypeF& direction_image)
{
    WindowTypeF w(image_in, 1, 1); //実数値を返すウィンドウアクセス関数
    PointTypeF p_m(magnitude_image), p_d(direction_image); //実数値を返す画素アクセス関数
    IUE_DOUBLE gx, gy, mag, theta;

    for (IUE_INT y = min_y; y <= max_y; ++y){
        for (IUE_INT x = min_x; x <= max_x; ++x){
            gx=((w.get(x,y, 1,-1)+2*w.get(x,y, 1, 0)+w.get(x,y, 1, 1))
                -(w.get(x,y,-1,-1)+2*w.get(x,y,-1, 0)+w.get(x,y,-1, 1)));
            gy=((w.get(x,y,-1, 1)+2*w.get(x,y, 0, 1)+w.get(x,y, 1, 1))
                -(w.get(x,y,-1,-1)+2*w.get(x,y, 0,-1)+w.get(x,y, 1,-1)));
            mag = sqrt(gx * gx + gy * gy);
            if (IUE_epsilon::less_than(mag,edge_threshold)){ p_m.set(x, y, 0.0); p_d.set(x, y, 0.0);}
            else { p_m.set(x, y, mag); p_d.set(x, y, atan2(gy, gx));}
        }
    }
    p_m.flush(); p_d.flush(); //アクセス関数の破棄
}

```

図 2: IUE を用いた Sobel フィルタの関数定義の例

定義されている。

このプログラムでは、入力画像 `image_in` の型が `IUE_scalar_image_2d`、すなわち画素値はスカラ型の範囲であれば何でも良いようにプログラミングされているという点に注目されたい。この場合には、ウィンドウアクセス関数によって、入力画像の画素値をプログラム内部の計算に用いる実数型に合わせるという方法で、入力画像のデータ型に依存しないプログラムが実現されている。これは、関数 `IUE_Sobel_Example` が呼び出される際の実引数の要素型が事前に決まっていなからであり、仮想関数として実現されたウィンドウアクセス関数を介して画素値の型変換が行われている。

入力画像の型に依存しないプログラムを作成する別の方法として、全ての計算に仮想関数を用いる方法も考えられるが、その場合には仮想関数の呼び出し回数が増え、呼び出し時の「型チェック」のオーバーヘッドが大きくなるため、実行効率が著しく低下する。

`IUE_scalar_image_2d` にも画素単位のアクセスを行うメンバ関数が埋め込まれているが、ウィンドウアクセス関数を用いた場合、内部にバッファを持ち、アクセスに先行して型変換が行われるため、実行効率が改善されており、しかもアクセス関数の内部構造はこの関数の中では見えない。

IUE では、仮想関数と、テンプレートをうまく使い分け、上述のように実行時の効率を重視しながら、型に出来るだけ依存しない抽象度の高いプログラミングを実現している。以上のようなプログラミング技法は、他の画像処理システムでも採用されているが、IUE の

最大の特長は後述するように、2次元画像だけでなく、3次元画像、点、線、領域、輪郭、平面、曲面、座標系、など非常に豊富なデータ型の定義を含んでいるということである。

2.3 IUE Version 2.0

ここでは、以上で述べた IUE の機能がソフトウェアシステムとしてどのように実現されているのかについて述べる。IUE Version 2.0 は、画像理解プログラムで用いられる多数の型定義を含んだクラスライブラリとそれを用いて作成された応用プログラム集であるタスクライブラリ、および、ユーザインタフェースから構成されている。

2.3.1 クラスライブラリ

前述のデータ抽象化、型継承、テンプレート、仮想関数、等の機能は、型に依存しないプログラムを作成するためだけでなく、大規模なソフトウェアシステムを複数のプログラムで分担して作成するためにも欠かすことができない有用な機能である。しかし、このような特長を生かすためには、「何をオブジェクトと考え、それらの間にどのような関係が成立するか」ということを共通認識として各プログラマが持たなければならない。この共通認識に基づくオブジェクトの表現の枠組み、すなわちオブジェクトの型が「クラス」である。各プログラマがばらばらのクラス定義に基づいて作成したプログラムは、部品として再利用できないばかり

か、通常の言語によって書かれたプログラムよりも理解し難いものとなる。一般にクラス定義の仕方には任意性があり、幅広い問題領域を対象とした一般性のあるクラスを定義するためにはその問題領域に関する深い洞察が不可欠となる。

IUEの骨格をなすクラスライブラリ³は画像理解における様々な概念を数理的に体系立てて定義することを第一の目的としており、画像理解システムの仕様記述や教育等にも利用できるように設計されている。このクラスライブラリが「画像処理用」ではなく、「画像理解用」と呼ばれる理由は、2次元の画像だけでなく、3次元の幾何学的対象や対象間の関係記述に必要となる、ベクトル、座標系、マトリックス、点列、グラフなど豊富なデータ構造を提供しているからである。実際のライブラリには、約600のクラス定義、400のテンプレート・インスタンスが含まれており、画像理解プログラムの作成に必要なと思われるクラスの多くが網羅されている。IUEクラスライブラリに含まれる様々な定義は、グラフィックス、CADなどの分野でも利用可能な部分を含んでおり、他分野の研究者にとっても一見の価値はあると思われる。

2.3.2 タスクライブラリ

IUEタスクライブラリ⁴とは、クラスライブラリを用いて作成されたプログラム群を指す。タスクライブラリには、現在のところ31のプログラム（特徴点検出、エッジ検出、点群への線・面などの当てはめ、エッジ点追跡による曲線生成、領域分割、ラベリング、フィルタリングなど）が収集されているだけであり、まだまだ充実しているとは言いがたい。しかし、収集されているプログラムは、アルゴリズムの考案者自身、もしくは専門のプログラマの手によって書かれたものであり、質は良いと言える。現時点でのタスクライブラリに含まれるプログラムの数が少ないことの原因は、IUEクラスライブラリの整備が遅れたことに起因しており、今後はより多くのプログラムが収録されていくものと考えられる。

2.3.3 プログラムの開発・実行環境

IUE Version 2.0は、Solaris 2.5, SunOS4.1.3, およびLinux 2(ELF)上で実行させることができ、コンパイルにはGNU g++が用いられている。また、Khoros画像処理システムのVisual Programming Language: Cantata[6]を利用することができる。Cantataは、図3に示すように処理モジュールを表すアイコンを利用して解析過程のデータフローを記述するプログラミング

³IUEのクラスライブラリに関する詳細は、<http://www.aai.com/AAI/IUE/spec/IUEClassHierarchy.html>を参照されたい。

⁴IUEのタスクライブラリに関する詳細は、<http://www.aai.com/AAI/IUE/IUETasks.html>を参照されたい。

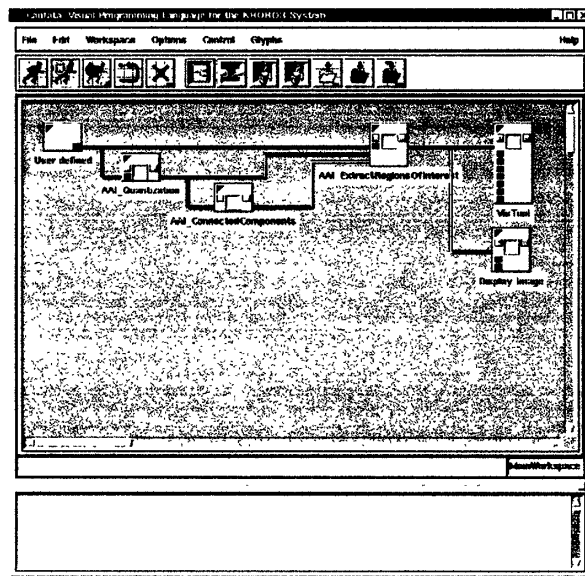


図 3: cantata を用いた visual programming の様子

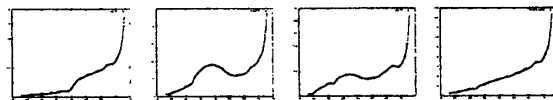
環境であり、IUEの画像処理アルゴリズムや表示ツールなどを利用した処理を視覚的に構成することができる。タスクライブラリの整備が進めば、画像解析・理解システムの試行錯誤的な構成が行え、一般ユーザーによる画像解析（医師によるレントゲン画像の解析、学生による画像解析実験など）が容易に実現可能となる。

IUE Version 2.0は、それ以前のVersion 1.3.1と比べて大幅な速度向上、オブジェクトプログラムサイズのコンパクト化が達成されており、道具としてのソフトウェアとしても利用価値が生まれつつある。

3 Calibrated Image Database

IUEに対する日本独自の貢献として我々は、Calibrated Image Database (CIDB) の設計・構築を進めてきた[1][8][9][10]。CIDBには、現在のところ次のような内容の画像が収録されており、様々な画像理解アルゴリズムの評価実験に用いることができる。

(1) マンセル色標分光放射輝度データベース



色標:10R/7-6 色標:10G/7-6 色標:10B/7-6 標準白色版

図 4: マンセル色標分光放射輝度データベースの一部：横軸は波長 [nm], 縦軸は放射輝度 [W/m^2sr]

概要：1016枚の色の異なるマンセル色標（米国マクベス社製）を分光照度が既知の標準光源によって照ら

し、分光放射輝度計で計測したスペクトルデータ (380nm~730nm まで 2nm 間隔:図 4) と、同じ色標を通常の 3CCD カメラで撮影した画像の RGB 値を収録

撮影対象 :マンセル色標 (米国マクベス社製, 市販品)

データ数 :1016 種類の分光スペクトルデータ

キャリブレーションデータ :標準光源照明下での標準白色板のスペクトル特性および RGB 値のデータ, 照明光の分光照度データ

用途 :カメラの分光感度特性, 物体の分光反射特性, 光源の分光照度特性などの推定といったカラーキャリブレーション, カラー画像処理アルゴリズムの性能評価や人間の色知覚特性の分析のための参照基準データ

(2) 標準物体画像

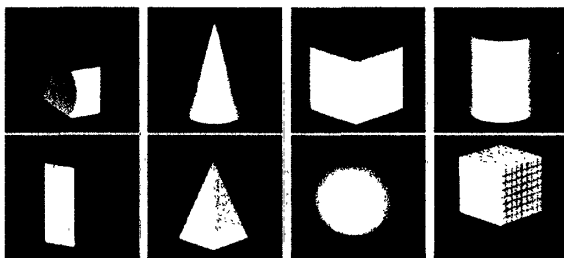


図 5: 標準物体画像の一部

概要 :形状や表面の反射特性が既知の物体を, 絞り, ズーム, フォーカスなどのカメラパラメータや光源の位置を変えながら撮影した画像データベース.

撮影対象 :カメラキャリブレーション用の 4 種類の物体 (標準白色板, 下振, マクベスカラーチャート, 較正用立方体), 光源キャリブレーション用の 2 種類の物体 (垂直軸付白色板, 大型白色板), 7 種類の標準物体 (石膏:球, 円錐, 円柱, 角錐, 角柱, 立方体, 凹物体:図 5), 2 種類のテクスチャ物体 (コルク:円錐, 角柱)

光源 :写真撮影用ハロゲンランプ

画像枚数 :112 枚

キャリブレーションデータ :画像データの大半 (85 枚) がカメラおよびレンズ, 光源をキャリブレーションするためデータで, それらを用いて利用者がキャリブレーションを行う.

用途 :光源の位置, CCD のノイズレベル, ズームによる FOE の移動, レンズ歪み, ホワイトバランスなどのカメラ, レンズのキャリブレーションア

ルゴリズムおよび, Shape from Shading/Texture, Depth from (De)Focus/Zooming, 多重絞り画像を用いた高精度カラー画像の合成などの解析アルゴリズム評価.

(3) イメージスキャナ画像



書籍画像

断面形状

図 6: イメージスキャナ画像の一部

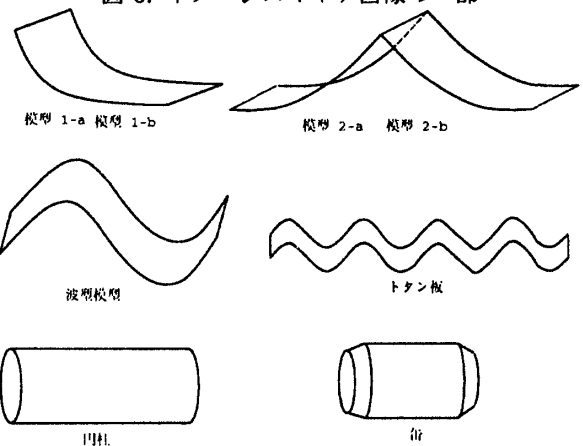


図 7: イメージスキャナ画像の撮影対象の形状

概要 :厚みのある書籍の見開き面や缶, トタン板などの 3 次元曲面を, モノクロイメージスキャナで撮影し, 得られた画像から元の 3 次元曲面形状を復元することや材質の違いによる反射特性の分析を目的としたもの.

撮影対象 :厚い書籍の見開き面 (図 6) および図 7 に示した曲面物体に 10 種類の紙を貼った対象

画像枚数 :対象の向きやスキャナの解像度を変えて撮影した合計 2585 枚

キャリブレーションデータ :対象曲面の 3 次元形状データ, スキャナ面上での対象の傾き角度, スキャナ面に対して異なった傾きを持ち, 上記の 10 種類の紙を貼った平面を撮した画像 (光源の位置や方向, 紙の材質に依存した反射特性較正用)

用途 :各種の紙の反射特性分析, 画像中の陰影やテクスチャからの物体の 3 次元形状復元問題

(4) 高精細高濃淡分解能白黒・カラー画像

概要 :ベルチエ素子電子冷却型 CCD 素子を搭載した高感度 (14bit/画素), 高空間分解能 (2048 × 2048) カメラを用いて, 様々なカラー物体を様々な照明光の下で撮影したカラー画像 (図 8)



標準光源 ハロゲンランプ 蛍光灯 白色電球
 図 8: 高精細高濃淡分解能白黒・カラー画像の一部

撮影対象 : 色鉛筆 36 色セット, 果物・野菜の店舗デコレーション用プラティック模型, 紙製色玉, 色較正用カラーチャート, 色較正用マンセル色標, 標準白色板

光源 : 電球型蛍光灯 (昼光色, 電球色), 写真撮影用ハロゲンランプ (1500, 100V 500W), 一般電球 (シリカなし白, 黄, 赤, 緑, 青)

画像枚数 : 18 枚

キャリブレーションデータ : 標準白色板上での分光放射輝度データ (390nm~730nm, 2nm 間隔: 照明の分光強度の推定に用いる). (1) のマンセル色標分光放射輝度データベースと画像中のマンセル色標の RGB カラーデータからカメラの分光感度特性が推定できる.

用途 : カラーキャリブレーション, カラー画像領域分割, エッジ検出, ハイライト領域抽出, 相互反射分析などの各種カラー情報処理

(5) 多視点ステレオ画像

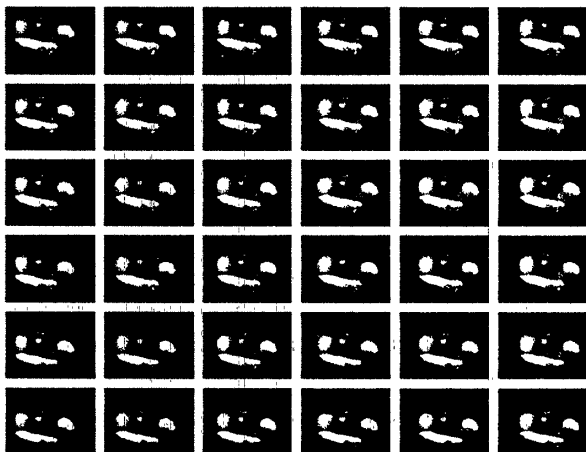


図 9: 多視点ステレオ画像 (果物模型) の一部

概要 : X-Z ステージを用いてカメラを水平/垂直に移動させながら撮影した多視点 (15 × 15=225 眼, 36+36-1=71 眼, 71+71-1=141 眼) 画像

撮影対象 : プラスティック製果物模型 (図 9), 建物模型, カラー/テクスチャ付き標準物体をそれぞれ複数配置

光源 : ハロゲンランプ

画像枚数 : 1294 枚

キャリブレーションデータ : X-Z ステージによるカメラ位置座標, 3次元位置計測装置による物体表面上の特徴点群の 3次元座標データおよび, 3次元位置が既知の格子点を撮影した幾何学的キャリブレーション用画像データ

用途 : 多視点ステレオマッチング, イメージベースドレンダリング, 動画像 (オプティカルフロー) 解析, 陰影, 相互反射解析

(6) 動きによるボケを伴う動画像



1/60[秒] 1/100[秒] 1/250[秒] 1/1000[秒]

図 10: 動きによるボケを伴う動画像の一部

概要 : LP レコードプレーヤのターンテーブル上にマークをつけ, 一定速度で回転させ, シャッタースピードを変化させて撮影した動画像 (図 10)

撮影対象 : マークを付けたターンテーブル

光源 : 日光による間接光

画像枚数 : シャッタースピードを 1/60, 1/100, 1/250, 1/1000 にした 30 フレーム/秒のビデオ画像 184 枚

キャリブレーションデータ : カメラの幾何学的キャリブレーション用格子点画像, ターンテーブルの回転速度, シャッター速度, カメラと物体までの距離など

用途 : 動画像 (オプティカルフロー) 解析, 動きによるブレ (motion blur) の解析

上述の我々が撮影した画像以外に, わが国の標準画像データベースとして長年利用されてきた SIDBA[7] および, 画像理解研究でよく実験に用いられてきた画像データ群を収集し, 画像データとそれを用いた実験が行われている論文リストを合わせて Standard Image Database (図 11) を構築した. また, 正確なキャリブレーションに拘らず, 同じ意味を持った (たとえば, 人



図 11: Standard Image Database の一部

間の顔) 画像を集めたデータベースを Sematic Image Database と呼んでおり, CIDB とこれらを合わせた全体の画像データベースを Real World Image Database と称している。

以上述べたように, CIDB では様々なキャリブレーションデータが添付されており, ある画像データを用いて行った実験の結果が妥当であるか否かが正確に評価できる。したがって,

- 画像理解研究者, 技術者にとっては様々な実験とともに, キャリブレーションデータに基づく実験結果の客観的評価ができる。
- CG 研究者, 技術者にとっては与えられた照明条件, カメラパラメータ, 物体形状などを基にして生成した画像が, 実写画像とどの程度違っているのかを定量的に評価することができ, よりリアルな映像生成アルゴリズムの研究に役立てることができる。
- 一般ユーザにとっては, 2章で述べた IUE ソフトウェアと組み合わせることにより, 各種画像解析アルゴリズムがどの程度の正確さを持つのかを評価することができ, 各アルゴリズムの使い分けの基準を知ることができる。

4 おわりに

本稿では, 画像理解のための標準的ソフトウェア IUE と, 画像理解アルゴリズム評価用画像データベース CIDB の現状について述べた。

IUE ソフトウェアの設計を行ってきた技術委員会は本年 5 月でその任務終え, 今後は Amerinex Applied Imaging (AAI) がソフトウェア開発作業を引き継ぐことになっている。AAI による IUE ソフトウェアの作成作業に対しては今後も ARPA からの資金的援助が継続される様子で, Version 2.0 の範囲内では, 2D および 3D オブジェクトの表示部の改善, ドキュメントの充実, が行われ, それ以降も実行速度の改善, タスクライブラリの充実, 座標系, トポロジ, マトリクスなどのクラスライブラリの改善が行われる予定である。重要な点は, 今後とも IUE ソフトウェアは, パブリックドメインソフトであり続けるということで, 現場の研究者・技術者からの建設的意見とともに, タスクライブラリへの積極的な貢献が期待されている。

一方, CIDB については, 本年度より新たに複数カメラを用いた多視点動画データベースの作成に取り組むことになっている。このデータベースが完成すれば, 多視点動画画像解析, イメージベースドレンダリングを用いた仮想空間の視覚ナビゲーションなどの実験, アルゴリズム評価に役立つ。これらの技術については, 画像理解だけでなく CG, VR の研究分野でも現在盛んに研究が行われているが, 解析対象となる多視点動画

像データを正確に撮影するには, 多数のカメラ相互間の 3 次元位置・視線方向を正確に定め, かつ, 複数のカメラのフレームタイミングの同期を取らなければならないという困難さが付きまとう。こうした問題を解決し, より多くの研究者, 技術者が利用可能なデータを提供するというのが, このデータベース開発の目的である。

従来, 研究グループ毎に個別に取り組みされてきたアルゴリズムのインプリメント作業, 特殊な機材や多くの手間が必要となる画像データの撮影作業を省くことは, 研究・開発の時間を短縮することにつながる筈であり, IUE および CIDB が少しでも役に立てば幸である。

最後に, CIDB の開発は, RWCP (新情報処理開発機構) の援助の下で行われたもので, 感謝の意を表したい。

参考文献

- [1] 松山, 谷口: Image Understanding Environment (IUE) プロジェクト, 情報処理, Vol.36, No.3, pp.203-211, 1995.
- [2] 田村ほか: ポータブル画像処理サブルーチンパッケージ SPIDER の開発, 情報処理, Vol.23, No.3, pp.321-328, 1982.
- [3] 鳥脇, 福村: 画像処理のためのサブルーチンライブラリ SLIP について, 情報処理, Vol.22, No.4, pp.353-359, 1981.
- [4] 佐藤ほか: 画像処理ワークステーション VIEW-Station のソフトウェアアーキテクチャ, 情報処理, Vol.31, No.7, pp.1015-1026, 1990.
- [5] Butler, T. and Krolak, P.: An Overview of the Programmer's Imaging Kernel (PIK) Proposed Standard, Computers and Graphics, Vol.15, pp.465-472, 1991.
- [6] Konstantinedes, K. and Rasure, J.: "The Khoros Software Development Environment for Image and Signal Processing", IEEE Trans. on Image Processing, Vol.3, No.3, pp.243-252, 1994
- [7] 尾上: 標準画像データベース, 第 8 回画像工学コンファレンス, pp.97-100, 1977.
- [8] CV 研究連絡会 IUE-WG 編: 画像データベースの構築調査報告書, イメージ情報科学研究所, 1995
- [9] CV 研究連絡会 IUE-WG 編: 画像データベースの構築に関する報告書, イメージ情報科学研究所, 1996
- [10] CVIM 研究連絡会 IUE-WG 編: Real World Image Database 構築のための画像データ収集・作成に関する報告書, イメージ情報科学研究所, 1997