

MATLAB 言語による 3 次元計測の試み

全 炳 徳* ・ 後 藤 惠之輔*

A New Trial of 3-Dimensional Analysis for Photogrammetry Using MATLAB Language

by

Byungdug JUN* and Keinosuke GOTOH*

The informationized society needs so many real time analog and digital information. The CCD camera is good device to catch for image information. In this study, we used MATLAB language to get 3-dimensional information using digital stereo camera images.

MATLAB is a technical computing environment for high-performance numeric computation and visualization. And also, there are so many Toolbox like a image processing required to research for photogrammetry. Especially, the Image Processing Toolbox is a collection of fuctions bult on MATLAB's powerful numeric computing environment. The Toolbox functions support a wide range of image processing operation, from image display to filtering, analysis and image transforms.

We estimate bias (RMS) of 3-dimensional information using CCD images with C and MATLAB language in laboratory work. As a result, we can obtain the most probable to apply MATLAB language to catch 3-dimensional information.

1. はじめに

デジタル機器の普及に伴い、写真測量がアナログからデジタルへと進歩し続けている。デジタルカメラによるリアルタイム計測などが試みられている¹⁾のは、このような時代の影響が大きい。このように、進む一方であるデジタル写真測量は、データのリアルタイム計測や CAD データとの結合、画像の加工処理が行えるなどの利点を備えており、建設現場における形状計測作業にも有効であることが提案されている²⁾。

本研究では、デジタルカメラによる写真測量を試みるために、室内での 3 次元計測を行った。まず、写真測量の理論を検討し、検討した理論に基づきプログラムを作成した。作成したプログラムは C 言語および MATLAB 言語を併用した。特に、行列計算や画像処

理がより簡単にできる MATLAB 言語を写真測量に適用し、その可能性を試みた。最後に、室内実験を行い、両プログラム(C 言語及び MATLAB 言語による)の精度を比較するとともに、デジタルカメラにおける 3 次元計測の実用性について検討した。

2. MATLAB 言語と写真測量

MATLAB 言語は、第 4 世代言語と呼ばれるコンピュータ言語に分類される³⁾。この種の言語は、専門のプログラマやコンピュータに詳しい人ではなしに、普通の利用者向けを目指しているものである。このため、次のような特徴を持っている。①言語そのものは最小限に押さえ、利用者の馴染みのある表現法を採用している。②複雑なプログラミングを関数として用意し、必要な計算や画像解析を簡単な命令文だけでサポート

する。③豊富なビジュアル関数を用意しており、必要な表現法を簡単に実現させる。

写真測量では、画像上の写真情報を複雑な行列計算を通して、必要とする情報を含んだ画像として表現する必要がある。すなわち、写真測量関連のプログラムは大きく分けて、行列計算部と画像処理部として2分化される。特に、画像処理においては、各コンピュータシステムにより異なっており、グラフィックスボードごとに相当の差が生じる。写真測量関連のプログラマーにとっては、MATLABのような統一した画像処理関数が望ましく、効率が良い。

MATLABは現在、ほとんどのOSに対応しており、フリーOS(LINUXなど)にまで利用の幅を拡大している。

特に、本研究では、写真測量において伝統的な、C言語によるプログラムと、MATLAB言語によるプログラムを比較・検討することにより、写真測量分野へのMATLAB言語の導入を試みた。

3. 立体写真測量

写真測量とは、写真を用いて、写されている被写体の物理量を測定することである⁴⁾。測定の対象となる物理量には、位置、量、質などがあるが、本研究では空間的な位置、すなわち3次元座標を求めることを扱うことにする。応用目的あるいは写真の種類の違いなどにより、空中写真測量、地上写真測量、天文写真測量、レントゲン写真測量、顕微鏡写真測量など、いろいろな写真測量の種類があるが、その原理はほぼ同じである。

この原理とは、「写真の像は、被写体から反射された光がレンズ中心を直進してフィルムあるいはCCDの平面に投影されてできたもの」である。このような投影は中心投影と呼ばれる。一言でいえば、写真は中心投影像である。

3. 1 立体写真の原理

立体写真とは、同じ対象物を異なる位置から平衡な光軸で撮影した1組の写真の対を言う⁵⁾。

今、図-1に示すように、上からビルの立体写真を写したとする。左の写真と右の写真のなかでは、ビルが写されている位置が異なる。

中心からの位置のずれは、高い所ほど(レンズに近い所ほど)大きくなっている。物体から2つのレンズに向かう2本の光線の作る角度も、レンズに近いものほど大きくなる。これは、人間の目が2つあって、物の遠近を計るのに物までの視覚の違いを利用しているのと

同じ原理である。次に、立体写真を撮影された時の状態と同じようにおいて、図-2に示すように、上から光をあてて眺めてみることにする。

左右の写真に写されている同じ対象物の対応する像の光線は、互いに交わり立体像を作る。左右の対応する写真上の点からでた光線が交わる角度は、写真上の

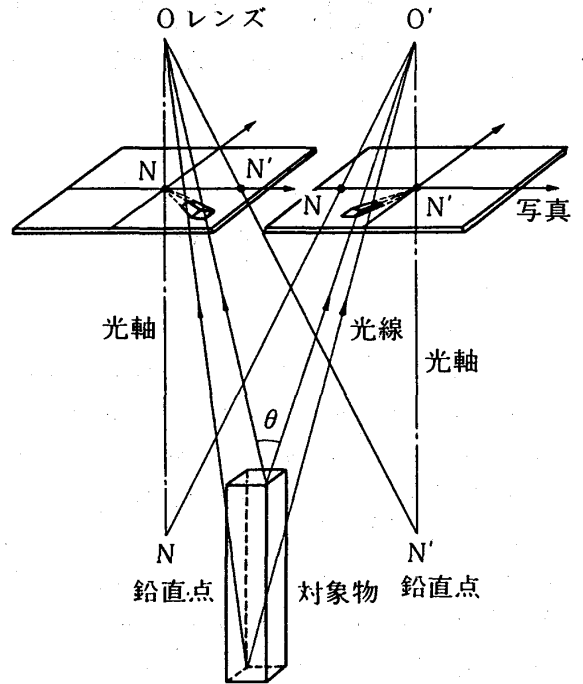


図-1 立体写真上の物体の位置と形

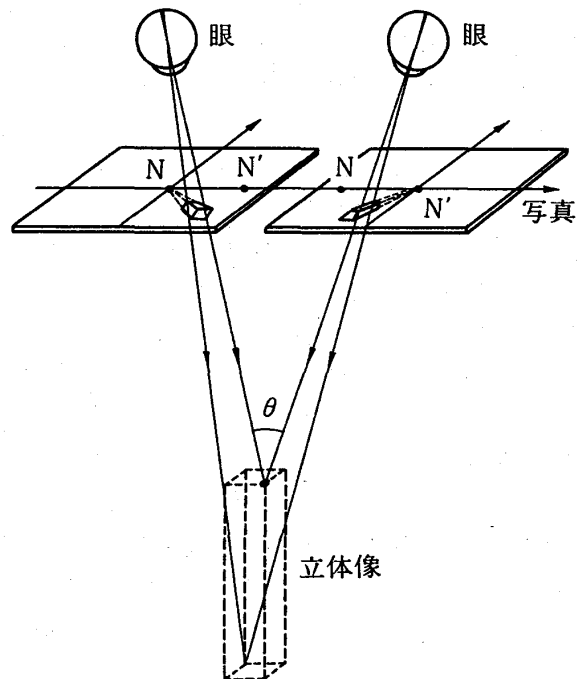


図-2 立体写真からつくられる立体像

位置のずれが大きいものほど大きくなるために、人間の目には立体的な像としてみえる。このような原理をもとに立体写真の 2 枚の写真の像の座標や長さを測定して、対象物の三次元的な位置や形を求めることを写真測量と言う。

3. 1. 1 内部標定

内部標定とは、画面距離と主点(中心)の位置を求めるものであり、機械座標系から、写真座標系に変換することである。内部標定には次に示すような共線条件を妨げる要因に対する補正がある。

- (1) 指標座標の誤差
- (2) 指標中心と主点との位置のずれ
- (3) レンズディストーション
- (4) フィルムおよび CCD の平面度
- (5) 大気屈折の補正

3. 1. 2 相互標定

相互標定とは、基準点の地上座標と、対応する写真座標上の像の表面座標とから、撮影時のカメラの位置と傾きを求めることである。相互標定方法は、次に示す 3 種類に分けられる。

① 共面条件による方法

2 本の光軸が交会するためには、図-3 に示すように、2 本の空間直線が同一平面に含まれるという条件が成立することを利用した方法である。このような条件を共面条件と呼ばれる。

② 縦視差消去法

図化機の投射機の機能を数学的に再現しようとした方法で、傾きのある写真を傾きのない鉛直写真にしたとき得られる投影像の縦視差(左写真の y の値と右の写真の y の値の差)が、図-4 に示すようにゼロとなる条件を利用した方法である。

①の方法が回転行列を用いるのに対し、この方法では、偏歪修正に用いられるような射影変換式が用いられる。この方法は原理的には、共面条件を用いる方法の 1 つとみなすこともできる。

③ 共線条件による方法

オランダの van der Hout が開発した方法で、未知変量に方向余弦を用いた 5 つの独立な変量を導入し、かつ共線条件から出発したユニークな方法である。

条件式が上記の変量に対して一次式となるので、①および②の方法のように非線形となる方法より、逐次計算の収束が良い利点である。しかし、日本では、殆ど①および②による方法が用いられ、この方法は用いられない。

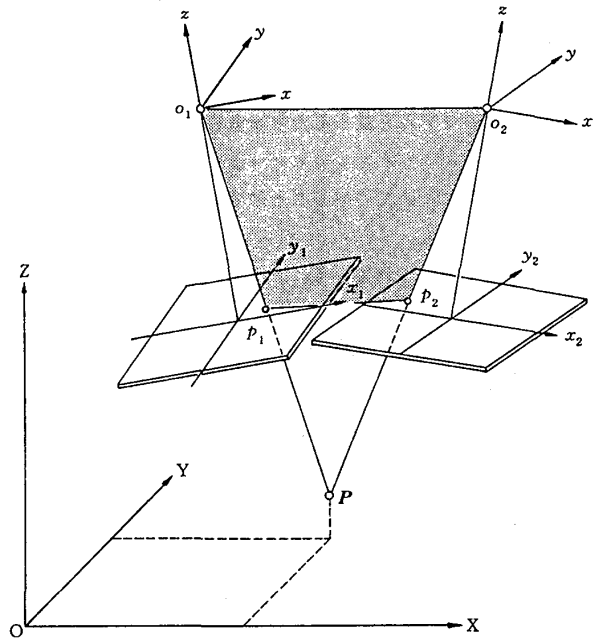


図-3 共面条件

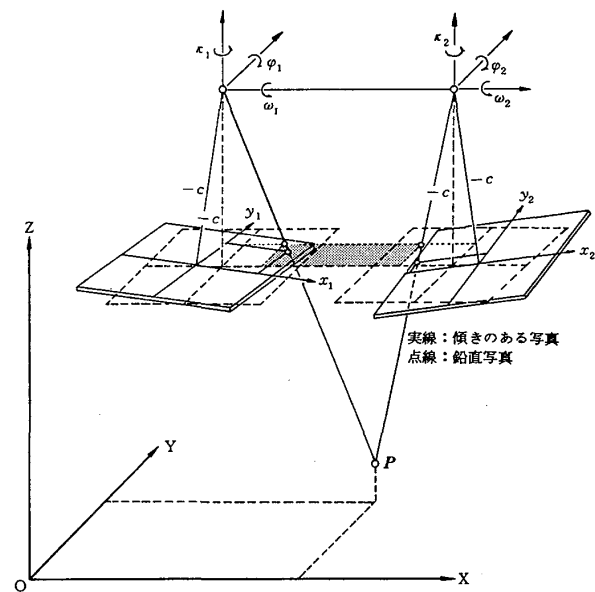


図-4 鉛直写真投影像の縦視差

相互標定において選択される座標系とそれに対応する変量のとり方により、次の 2 つの方法がある。

- (1) 投影中心を結ぶ方法
- (2) 左側のカメラ座標を固定する方法

本研究では、以上の理論を検討した結果、基本的な所で MATLAB 言語による試ができる共面条件式の左側のカメラ座標を固定する方法を使い、プログラム作成を行った。その理論については、参考文献の 4)

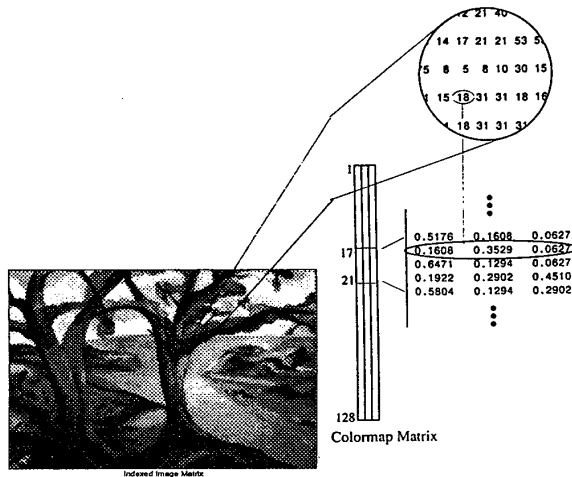


図-5 MATLAB 内部での画像の色表現方法

-7)までの文献に詳しく説明しているので、本稿では省略する。

4. MATLAB 言語による写真測量プログラム

4.1 MATLAB 内部での画像表現方法⁸⁾

前述したように、MATLAB の基本的なデータタイプは、長方形の行列で、順序付けされた実数または複素数要素からなるものである。このオブジェクトは、イメージや実数値、順序付けされた色や強度データを表現するのに適している。

カラーイメージは、一般に2つの行列、カラーマップ行列とイメージ行列を必要とする。カラーマップ (colormap) は、イメージの中で色を表現する順序付けされた複数の値からなる。各々のイメージピクセルに対し、イメージ行列はカラーマップの中に対応するインデックスを含んでいる。

カラーマップ行列の大きさは、 n 色を含むイメージに対し n 行3列になる。カラーマップの各々の行は、3つの要素、赤、緑、青 (RGB) となる。これにより、その色の赤、緑、青の成分の強度を指定する。R, G, B は0.0(黒)から1.0(最強)までの範囲の実数カラー値で、MATLAB は、表現しようとするイメージとカラーマップに対し、これらの値を対応する強度に変換している。

MATLAB がインデックス付きのイメージを表現するとき、カラーマップの中の希望する色を選ぶために、イメージ行列の中の値を使う。例えば、図-5のように、イメージ行列が行列の位置 (86,198) の部分で値 18 であるとするれば、ピクセル (86,198) の色は、カラーマップの18行目に対する色になる。

MATLAB を離れて、 n 色を持ったインデックス付きのイメージは、しばしば0から($n-1$)までの値を含んでいる。これらの値は、その最初のインデックスとして0のカラーマップの中のインデックスになる。

ツールボックスの関数でイメージを作るには、MATLAB 行列はインデックス1から始まるので、イメージの中で各々の値を増加させるか、イメージをシフトアップするかになる。

4.2 MATLAB による画像表示⁸⁾⁻⁹⁾

ここでは、MATLAB により写真測量用の画像を、コンピュータのディスプレイ上に表示するための必要な、幾つかのテクニックを述べておく。

① カラーマップの表示

まず、カラーマップしたイメージを見るには次のステートメントを使う。

```
imshow (X, map)
```

ここで、 X は画像行列で、 map は色の種類を指す。

② 複数のイメージの表現

表示したいイメージが同じカラーマップを持っている場合、関数 subplot を使って、単一 figure ウィンドウにこれらを表示することができる。このコマンドはサブエリアに表示部を分割する。例えば、

```
subplot (m, n, 1)
```

とすれば、figure を m 行 n 列のサブエリアに分割し、最初のサブエリアをアクティブにする。

③ イメージのズーム

関数 imzoom は、マウスを使って縮尺を行ったり、拡大を行ってイメージの詳細を調べることができる。命令は次のようなコマンドを用いる。

```
imzoom on, imzoom off
```

imzoom on のコマンドが実行されると、ズームを適用したいイメージエリアがコマンド待ちとなり、マウスによりクリックすることで、拡大・縮小が自由に行われる。その時、マウスの左ボタンは拡大、右ボタンは縮小のコマンドとなる。ズーム状態を解除するには imzoom off のコマンドを使う。

④ マウスを用いた座標入力とキーボード入力

画面上の点をマウスでカーソルを選ぶことにより、一連の座標データ (X, Y) をベクトルとして入力するための関数 ginput が用意されている。点の座標は、画面の左下隅が (0, 0)、右上隅が (1, 1) となる。関数 ginput の指定の仕方は次のいずれかである。

- 1) `[x, y]=ginput` : 入力の終わりはリターンキー.
- 2) `[x, y]=ginput (n)` : n 個の座標の入力.
- 3) `[x, y, button]=ginput` : キーボードから整数を入力し, `button` に入れる.

⑤ 問い合わせ入力

プログラムの中からメッセージを画面に出力してそれに応える形でデータをキーボードから入力することができる。例えば、

```
k=input('number of data=')
```

と実行すると、

```
number of data=
```

というメッセージを出力して入力待ちになる。キーボードから10を入力して改行キーを打ち込むと、プログラム中の次のコマンドを実行することになる。kとしては行列やベクトルを指定してもよい。

⑥ メニューの選択

関数 `menu` を使うと、画面にメニュー一覧を写し出し、メニュー番号を入力することにより、プログラム中でこの番号に応じた処理をすることができる。例えば、コマンド

```
k=menu('Close a color', 'Red', 'Green', 'Blue')
```

を実行すると

```
----- Choose a color -----
                               1) Red
                               2) Green
                               3) Blue
```

```
Select a menu number:
```

の画面が出力され、番号の入力待ちとなる。

⑦ 2 値化処理

写真測量での必要とする画像処理の1つは、マッチングである。マッチングとは、2つの写真から同じポイントを抜き出すことである。この処理のために MATLAB では 2 値化処理を可能にしている。

参考のために長崎大学工学部 1 号館の屋上から撮った中庭の様子を、2 値化処理を行い図-6 に示している。

以上の 7 項目の処理については、写真測量関連のプログラムに有効に使えるもので、本研究のメインプログラムにおいても用いられている。

5. 室内実験とプログラミング

5. 1 室内実験装置とプログラム

本研究では、デジタルカメラによる写真撮影と MATLAB 言語のプログラム検証のために、付録-A のように実験用ボードを壁の上に制作し、実験用ボードに向かって直角方向を撮影したもの (VP シリーズ)、角度を入れて撮影したもの (SP シリーズ) 及び、ボードに近づいて撮影したもの (LP シリーズ) に分けて撮影を行い、分析した。

プログラムは C 言語のものと MATLAB 言語のものに分けて作成し結果を比較した。

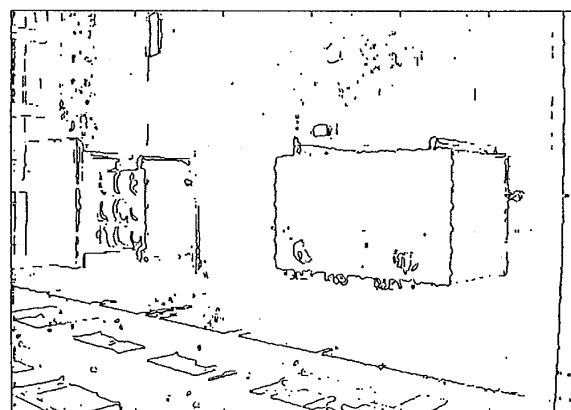
プログラムのフローチャートを図-7 に示す。また、プログラムの内容は付録-B (MATLAB 言語によるもの) に付けておく。

5. 2 実験結果

室内実験用ボードを撮影した画像の内、本研究で分析に用いたものは、総10ポイントである。その中でも、本研究により開発されたプログラムは、相互標定とし



(a) 左からとった写真



(b) 右からとった写真

図-6 MATLABでのコマンドによる 2 値化処理結果

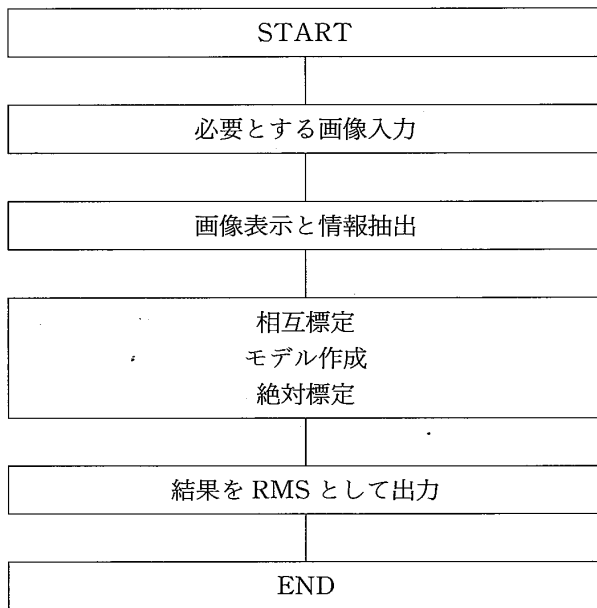


図-7 プログラムのフローチャート

て左側のカメラ座標を固定する方法を用いたため、撮影ポイント VP1 を固定したカメラ座標とし、右カメラにより得られた画像として、SP1, SP5, SP10, SP15, SP18, SPN, VP2, VP3, LP2 が用いられた。

結果は表-1のように、C言語と MATLAB 言語を対比し、基準となる3次元情報と計算結果との RMS 誤差(mm)を出力した。

結果から見ると、VP シリーズが SP シリーズより誤差が大きい。しかし、LP シリーズは分析に使った例が一つしかないが、SP シリーズとほぼ一致した結果となっている。この結果から、次のようなことが言える。

- ①左側のカメラ座標を固定する方法を用いる場合、片一方のカメラに角度を入れて撮影した方が、3次元情報を抽出する際、良い精度が得られる。
- ②左側のカメラ座標を固定する方法を用いる場合、適当な角度を維持することによって、誤差1センチ以下の3次元情報を得ることができる。
- ③カメラは撮影面において、同じ高さを維持しなくても、3次元情報の精度を悪くすることはない。
- ④C言語と MATLAB 言語との精度を比較した結果、MATLAB 言語の方が若干精度を上げている(VP 1 と SP10を用いた場合)。これについてはまだ、検討の余地がある。

6. ま と め

MATLAB 言語には高度な行列計算と多彩な画像処理の能力を備えている。本研究の結果からみれば、

表-1 Cと MATLAB 言語によるプログラムの RMS 誤差(mm)

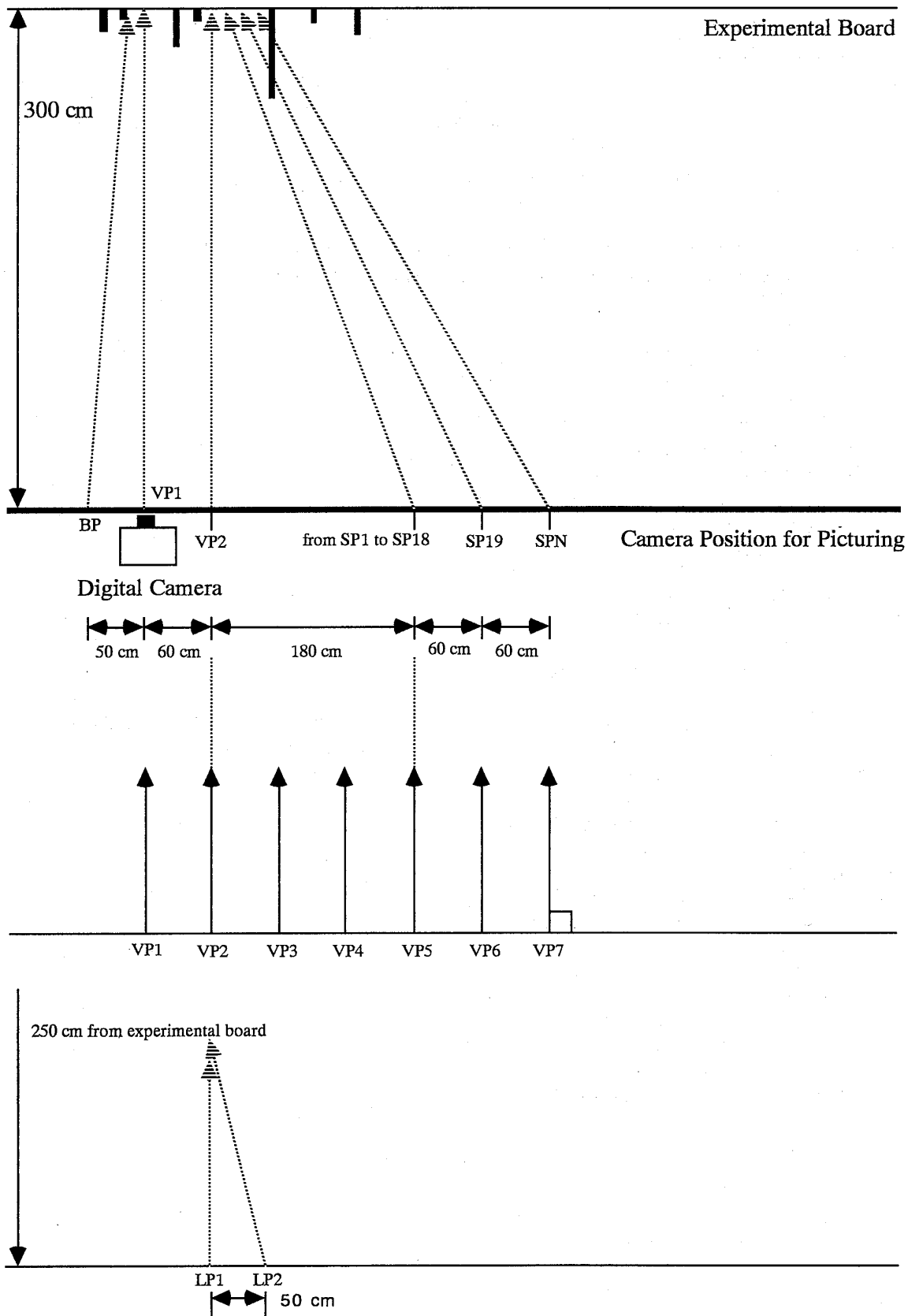
左カメラ	右カメラ	C 言語(RMS)	MATLAB 言語 (RMS)
VP1	SP1	8.9	8.4
	SP5	10.1	10.1
	SP10	6.5	6.0
	SP15	4.1	4.1
	SP18	1.6	1.6
	SPN	3.1	3.1
	VP2	11.8	11.8
	VP3	13.4	13.4
	LP2	7.9	7.9

MATLAB は写真測量分野に十分な能力を発揮する可能性があると言える。

本研究を進めるにあたっては、韓国関東大学の崔勝弼副教授ならびに卒論着手者であった坪内秀平君に協力を得た。ここに記し、感謝の意を表する次第である。

参 考 文 献

- 1) 近津博文, 中野一也, 穴井哲治, 村井俊治: リアルタイムフォトグラメトリのための CCD カメラキャリブレーション, 写真測量とリモートセンシング, Vol. 35, No. 2, pp. 4-10, 1996.
- 2) 桜井 浩: デジタルスチルカメラを用いた3次元形状計測システム, 写真測量とリモートセンシング, Vol. 32, No. 6, pp. 71-74, 1993.
- 3) 小国 力: MATLAB と利用の実際—現代の応用数学と CG—, サイエンス社, pp. 1-4, 1995.
- 4) (株)日本写真測量学会: 解析写真測量—改定版—, pp. 1-9, 1989.
- 5) (株)日本写真測量学会編: 立体写真のみかた・とりかた・つくりかた, pp. 3-4, 1988.
- 6) 米谷榮二, 森 忠次: 測量学, (株)丸善, 1993.
- 7) 田島 稔, 小牧和雄: 最小二乗法の理論とその応用—改訂版—, 東洋書店, 1996.
- 8) The MATH WORKS Inc.: Image Processing TOOLBOX for Use with MATLAB, サイバネットシステム(株), 1994.
- 9) The MATH WORKS Inc.: MATLAB High-Performance Numeric Computation and Visualization Software, サイバネットシステム(株), 1994.



付録-A 室内実験ポートレイアウト

```

%-----
%解析写真測量プログラム（相互標定—共面条件式利用—）
%左写真を固定した場合
%RGBファイルのためのプログラム（1996.10.31追加）

%- using Matlab language -----(1996.10.20)-----

% made by Byungdug JUN (Nagasaki University in Japan)
%
%-----
%---- RGB 左画像を読み込み、ヘッダを読み飛ばし、Rl, Gl, Blを読み込む。---
%---- RGB 右画像を読み込み、ヘッダを読み飛ばし、Rr, Gr, Brを読み込む。---
%---- 左写真のデータ -----
fid=fopen('c:\matdata\%nlp2.rgb');
status=fseek(fid, 23040, 'bof');
Rl=fread(fid, [1, 304640], '1*char', 2);
status=fseek(fid, 23041, 'bof');
Gl=fread(fid, [1, 304640], '1*char', 2);
status=fseek(fid, 23042, 'bof');
Bl=fread(fid, [1, 304640], '1*char', 2);
%---- 右写真のデータ -----
fid=fopen('c:\matdata\%nlp2.rgb');
status=fseek(fid, 23040, 'bof');
Rr=fread(fid, [1, 304640], '1*char', 2);
status=fseek(fid, 23041, 'bof');
Gr=fread(fid, [1, 304640], '1*char', 2);
status=fseek(fid, 23042, 'bof');
Br=fread(fid, [1, 304640], '1*char', 2);

%---- 読み込んだ Rl, Gl, Bl データを r_l.rgb, g_l.rgb, b_l.rgb ファイルに保管--
%---- 読み込んだ Rr, Gr, Br データを r_r.rgb, g_r.rgb, b_r.rgb ファイルに保管--
%---- 左写真のデータ -----
fid=fopen('c:\matdata\r_l.rgb', 'w');
count=fwrite(fid, Rl, '304640*char');
fid=fopen('c:\matdata\g_l.rgb', 'w');
count=fwrite(fid, Gl, '304640*char');
fid=fopen('c:\matdata\b_l.rgb', 'w');
count=fwrite(fid, Bl, '304640*char');
%---- 右写真のデータ -----
fid=fopen('c:\matdata\r_r.rgb', 'w');
count=fwrite(fid, Rr, '304640*char');
fid=fopen('c:\matdata\g_r.rgb', 'w');
count=fwrite(fid, Gr, '304640*char');
fid=fopen('c:\matdata\b_r.rgb', 'w');
count=fwrite(fid, Br, '304640*char');

%----- 画像表示 -----
load trees
% ①画像データをオープンする。
% ②640X476画像を読み込む。（A, B画像）
fid=fopen('c:\matdata\r_l.rgb');
Rl=fread(fid, [640, 476]);
fid=fopen('c:\matdata\g_l.rgb');
Gl=fread(fid, [640, 476]);
fid=fopen('c:\matdata\b_l.rgb');

```



```

Bl=fread(fid,[640,476]);

fid=fopen('c:\matdata\Yr_r.rgb');
Rr=fread(fid,[640,476]);
fid=fopen('c:\matdata\Yg_r.rgb');
Gr=fread(fid,[640,476]);
fid=fopen('c:\matdata\Yb_r.rgb');
Br=fread(fid,[640,476]);

% ①一つのウィンドウに二つの画像を表示する。
% ②読み込んだ画像を逆行列方式で表示する。(色はクーパーに)
% ③画像の上にタイトルを書く。(左、右画像)

subplot(1,2,1),imshow(Rl',map),title('left photo')
subplot(1,2,2),imshow(Rr',map),title('right photo')

% ①メニューを作成する。タイトルは'Do you chose this scene?'で
% ボタンの順番は,'Yes','No'の二つにして表示する。
k=menu('Do you chose this scene?','Yes','No')

% もし、ボタンの1番を選んだ場合。ボタン画像をクローズし、
% 画像を一つのウィンドウに一つ作成する。
% (色はクーパーで、タイトルを付ける)
if k==1
close
imshow(Rl',map),title('left picture')

% 表示している画像から、情報をマウスにより取り込む。
n=input('number of point=')
[x1,y1]=ginput(n)

% 画像をクローズし、次の画像を表示する。
close
imshow(Rr',map),title('right picture')

% 表示している画像から、情報をマウスにより取り込む。
[x2,y2]=ginput(n)
close

% もし、ボタン2番を選んだ場合。
elseif k==2

% 312320バイトを読み込んで、二つの画像をウィンドウに表示する。
subplot(1,2,1),imshow(Gl',map),title('left photo')
subplot(1,2,2),imshow(Gr',map),title('right photo')
k=menu('Do you chose this scene?','Yes','No')

if k==1
close
imshow(Gl',map),title('left picture')
n=input('number of point=')
[x1,y1]=ginput(n)
close
imshow(Gr',map),title('right picture')
[x2,y2]=ginput(n)

```

```

close

elseif k==2
close
subplot(1, 2, 1), imshow(Bl', map), title(' left photo')
subplot(1, 2, 2), imshow(Br', map), title(' right photo')
k=menu('Do you chose this scene ?', 'Yes', 'No')

if k==1
close
imshow(Bl', map), title(' left picture')
n=input('number of point=')
[x1, y1]=ginput(n)
close
imshow(Br', map), title(' right picture')
[x2, y2]=ginput(n)
close

elseif k==2
close
end

end

%-----相互評定-----

Xg=[0;0;0.3;0.6;0.6;0.9];
Yg=[0;0.5;0.25;0;0.5;0.25];
Zg=[10.088;10.137;10.142;10.162;10.221;10.248];

b2=10;c=700;n=6;zzl=c;a1=1;

for i=1:1:6;
    for j=1:1:1;
        p(i, j)=y2(i, j)-y1(i, j);
    end;
end;

b=(1.0/6.0)*(((x1(1)-x2(1))+(x1(2)-x2(2))+(x1(3)-x2(3))+(x1(4)-x2(4))+(x1(5)-x2(5))+(x1(6)-x2(6))));
d=(1.0/8.0)*(((abs(y1(3))+abs(y2(3)))+(abs(y1(4))+abs(y2(4)))+(abs(y1(5))+abs(y2(5)))+(abs(y1(6))+abs(y2(6))))));
k3=(1.0/(3.0*b))*((p(1)-p(2)+p(3)-p(4)+p(5)-p(6)));
p3=(c/(2.0*b*d))*((p(3)-p(4)-p(5)+p(6)));
w3=-(c/(4.0*d*d))*((2.0*p(1)+2.0*p(2)-p(3)-p(4)-p(5)-p(6)));
bx=b;
by=-(1/3)*(p(2)+p(4)+p(6))+(c/3)*(3+(2*((d/c)*(d/c))))*w3;
bz=(c/(2*d))*(p(4)-p(6));

for i=1:1:6;
    for j=1:1:1;
        xx1(i, j)=x1(i, j);
        yy1(i, j)=y1(i, j);
    end;
end;
end;

```

```

for i=1:1:5;
    for j=1:1:1;
        h_mat(i,j)=0;
    end;
end;

while a1<b2
    bx=bx;by=by+h_mat(1,1);bz=bz+h_mat(2,1);w3=w3+h_mat(3,1);p3=p3+h_mat(4,1);k3=k3+h_mat(5,1);
    sw=sin(w3);sp=sin(p3);sk=sin(k3);cw=cos(w3);cp=cos(p3);ck=cos(k3);

    for i=1:1:6;
        for j=1:1:1
            xx2(i,j)=(cp*ck*x2(i,j)-cp*sk*y2(i,j))+sp*c+bx;
            yy2(i,j)=(sw*sp*ck+cw*sk)*x2(i,j)+(-sw*sp*sk+cw*ck)*y2(i,j)-sw*cp*c+
by;
            zz2(i,j)=(-cw*sp*ck+sw*sk)*x2(i,j)+(cw*sp*sk+sw*ck)*y2(i,j)+cw*cp*c+
bz;
            by1(i,j)=-xx1(i,j)*zz2(i,j)+xx2(i,j)*zz1;
            bz1(i,j)=xx1(i,j)*yy2(i,j)-xx2(i,j)*yy1(i,j);
            wl(i,j)=(yy1(i,j)*yy2(i,j)+zz1*zz2(i,j))-by*xx1(i,j)*yy2(i,j)-bz*xx1
(i,j)*zz2(i,j);
            pl(i,j)=-yy1(i,j)*xx2(i,j)+by*(xx1(i,j)*xx2(i,j)+zz1*zz2(i,j))-bz*(y
yl(i,j)*zz2(i,j));
            kl(i,j)=-xx2(i,j)*zz1-by*yy2(i,j)*zz1+bz*(xx1(i,j)*xx2(i,j)+yy1(i,j)
*yy2(i,j));
            ff(i,j)=-((yy1(i,j)*zz2(i,j)-yy2(i,j)*zz1)-by*(xx1(i,j)*zz2(i,j)-xx2
(i,j)*zz1)+bz*(xx1(i,j)*yy2(i,j)-xx2(i,j)*yy1(i,j)));
        end;
    end;

    for i=1:1:6
        a_mat(i,1)=by1(i,1);a_mat(i,2)=bz1(i,1);a_mat(i,3)=wl(i,1);a_mat(i,4)=pl(i,1
);a_mat(i,5)=kl(i,1);
    end;

    for i=1:1:6;
        for j=1:1:1;
            f_mat(i,j)=ff(i,j);
        end;
    end;

    b_mat=a_mat';c_mat=b_mat*a_mat;d_mat=inv(c_mat);g_mat=b_mat*f_mat;h_mat=d_mat*g_mat;
    a1=a1+1;

end

%----- モデル座標 -----

for i=1:1:6;
    for j=1:1:1
        xx3(i,j)=(cp*ck*x2(i,j)-cp*sk*y2(i,j))+sp*c;
        yy3(i,j)=(sw*sp*ck+cw*sk)*x2(i,j)+(-sw*sp*sk+cw*ck)*y2(i,j)-(sw*cp*c);
        zz3(i,j)=(-cw*sp*ck+sw*sk)*x2(i,j)+(cw*sp*sk+sw*ck)*y2(i,j)+(cw*cp*c);
    end;
end;

```

```

left(i, j)=(bx*zz3(i, j)-bz*xx3(i, j))/(xx1(i, j)*zz3(i, j)-xx3(i, j)*c);
right(i, j)=(bx*c-bz*xx1(i, j))/(xx1(i, j)*zz3(i, j)-xx3(i, j)*c);

Xl(i, j)=left(i, j)*xx1(i, j);Yl(i, j)=left(i, j)*yy1(i, j);Zl(i, j)=left(i, j)*c;

Xr(i, j)=right(i, j)*xx3(i, j)+bx;Yr(i, j)=right(i, j)*yy3(i, j)+by;Zr(i, j)=right(
i, j)*zz3(i, j)+bz;

Xm(i, j)=(0.5*(Xl(i, j)+Xr(i, j)))*0.00001;
Ym(i, j)=(0.5*(Yl(i, j)+Yr(i, j)))*0.00001;
Zm(i, j)=(0.5*(Zl(i, j)+Zr(i, j)))*0.00001;

end;
end;

mx=0;my=0;mz=0;gx=0;gy=0;gz=0;

for i=1:1:6;
    for j=1:1:1;
        mx=mx+Xm(i, j);my=my+Ym(i, j);mz=mz+Zm(i, j);gx=gx+Xg(i, j);gy=gy+Yg(i, j);gz=gz+
Zg(i, j);
    end;
end;

for i=1:1:6;
    for j=1:1:1;
        xm(i, j)=Xm(i, j)-mx/n;ym(i, j)=Ym(i, j)-my/n;zm(i, j)=Zm(i, j)-mz/n;
        xg(i, j)=Xg(i, j)-gx/n;yg(i, j)=Yg(i, j)-gy/n;zg(i, j)=Zg(i, j)-gz/n;
    end;
end;

u1=0;u2=0;u3=0;v1=0;v2=0;v3=0;

for i=1:1:6;
    for j=1:1:1;
        u1=u1+xm(i, j)*xg(i, j)+ym(i, j)*yg(i, j);u3=u3+ym(i, j)*xg(i, j)-xm(i, j)*yg(i, j);
        v1=v1+(xm(i, j)*xm(i, j))+ym(i, j)*ym(i, j);
        v2=v2+(xm(i, j)*xm(i, j));v3=v3+(ym(i, j)*ym(i, j));
    end;
end;

A=u1/v1;B=u3/v1;S=sqrt(A*A+B*B);

xh=0;yh=0;xym=0;

for i=1:1:6;
    for j=1:1:1;
        xh=xh+xm(i, j)*(zg(i, j)+S*zm(i, j));yh=yh+ym(i, j)*(zg(i, j)+S*zm(i, j));xym=xym+
xm(i, j)*ym(i, j);
    end;
end;

xym1=xym*xym;

```

```
P=(v3*xh-xym*yh)/(S*(v2*v3-xym1));W=(-v2*yh+xym*xh)/(S*(v2*v3-xym1));K=atanh(B/A);
X0=(gx-mx*A-my*B)/n;Y0=(gy-my*A+mx*B)/n;Z0=(gz+mz*S-S*mx*P+S*my*W)/n;
```

```
%----- 絶対座標 -----
```

```
for i=1:1:6;
    for j=1:1:1;
        XG(i, j)= A*Xm(i, j)+B*Ym(i, j)+X0;YG(i, j)=-B*Xm(i, j)+A*Ym(i, j)+Y0;
        ZG(i, j)= S*(P*Xm(i, j)-W*Ym(i, j)-Zm(i, j))+Z0;
    end;
end;
```

```
ZZ=Zg-ZG
```

```
% 最終残差の表示
```

```
RMS=sqrt((ZZ(1)^2+ZZ(2)^2+ZZ(3)^2+ZZ(4)^2+ZZ(5)^2+ZZ(6)^2)/n)
```

```
%----- プログラムの終了 -----
```

付録-B-6