

2011 年課題 多眼ステレオプログラミング

先端表現情報学基礎 IV

コンピュータビジョン

東京大学生産技術研究所 池内研究室

阪野 貴彦

2011 年 5 月 12 日

1 はじめに

ステレオとは、位置や姿勢、焦点距離などのパラメータが全て分かっているカメラによって撮影した画像を複数枚用いて、撮影された対象物体の 3 次元形状を求める手法である。このステレオ視は立体視とも呼ばれ、人間の 2 つの眼による 3 次元認知も同じメカニズムであると考えられている。コンピュータビジョンの分野では、ステレオ研究の歴史は古く、現在でもなお盛んに研究が行われている。

ステレオ視の基本は、2 枚の画像から視差を求め、カメラ位置による三角計測で対象物体までの奥行きを推定することである。人間は 2 つの眼による 2 つの画像によって奥行きの感知を行っているが、コンピュータでは 3 枚以上の画像を用いて奥行きを推定することができる。一般に枚数を増やすことによって、復元される 3 次元形状の精度が上がる事が知られている。

本資料ではまず、2 枚の画像を用いたステレオによる 3 次元形状復元手法の基礎的事項を説明する。次に、複数枚のステレオ (多眼ステレオ) への拡張法について説明する。

2 三角計測による 3 次元推定

2 つのカメラを用いた三角計測の原理を説明する。

ここでは、同じ姿勢をした全く同じカメラを左右に配置する。左側のカメラを基準カメラ、右側のカメラを参照カメラと名付ける。

座標系として、基準カメラの位置を原点とし、 XYZ の各座標軸を図 1 のように設定する。参照カメラの位置は $(B, 0, 0)$ で、両カメラ間の距離 B はベース長と呼ばれる。

カメラによって得られる画像は、3 次元中の対象物体が平面 $Z = f$ 上に投射投影される 2 次元像で、 f を焦点距離と呼んでいる。

画像座標 (x, y) は、画像中心を原点とした 2 次元座標であり、通常はピクセル単位で表す。画像座標軸 x, y は、それぞれ 3 次元空間の X 軸、 Y 軸に平行となるよう設定する。

いま、3 次元空間中の点 (X_p, Y_p, Z_p) をこの 2 つのカメラで撮影したとする。このとき、基準カメラで得られた画像中では位置 (x_L, y_L) に写っており、参照カメラによる画像では点 (x_R, y_R) に写っているものとする。ただし、図 2 のように 2 つの同じカメラを平行に配置したときには、 $y_L = y_R$ となる。この 2 つの点 (x_L, y_L) と (x_R, y_R) とは対応点ペアと呼ばれる。

先述したようにステレオとは、位置姿勢とパラメータ (焦点距離 f) が既知であるカメラによって撮影された複数枚の画像から、写っている 3 次元物体の形状を復元することである。ここでは、 $B, f, (x_L, y_L), (x_R, y_R)$ を用いて、点 (X_p, Y_p, Z_p) の座標を求めることになる。

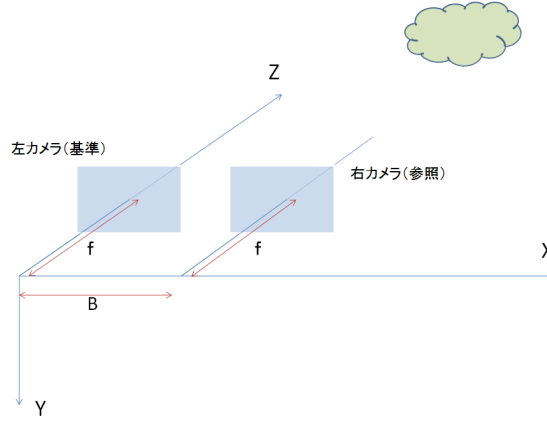


図 1: 座標軸，カメラの配置

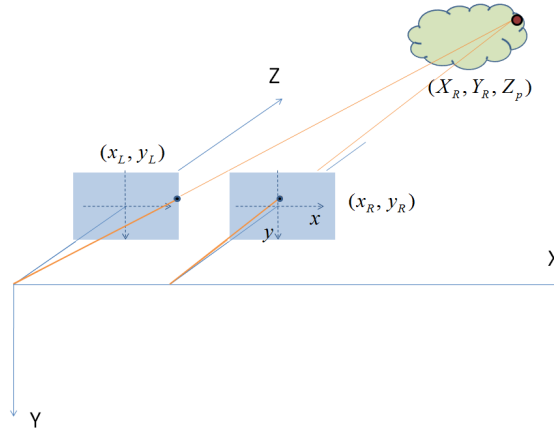


図 2: ステレオ視

三角測量による奥行き Z は，3 角形の相似より求める．図 3 は図 2 の様子を上から Y 軸方向に見た図である．3 次元空間中の対象点 (X_p, Y_p, Z_p) と 2 つのカメラ位置が作る 3 角形と，基準カメラ位置とその画像上の視差が作る 3 角形が相似であることから，次の関係式が成り立つ．

$$|x_L - x_R| : f = B : Z \quad (1)$$

このとき用いられる焦点距離 f はピクセル換算された数値を用いる．

ここで，基準画像中の点 (x_L, y_L) が参照画像では $(x_R, y_R (= y_L))$ に移動していることから，移動量 $x_R - x_L$ を基準画像における視差と呼び， $d = x_R - x_L$ で表す．視差 d を用いて，上式を整理すれば，奥行き Z_p は，

$$Z_p = -\frac{Bf}{d} \quad (2)$$

として求めることができる．図 1 のようなカメラ配置では必ず $x_L \geq x_R$ であり，視差 d を $x_R - x_L$ で定義しているため，上式にマイナスの符号がついている．

奥行きが求められれば，スケーリングによって点 (X_p, Y_p, Z_p) は，次のようにして求められる．

$$(X_p, Y_p, Z_p) = \left(\frac{Z_p}{f} x_L, \frac{Z_p}{f} y_L, -\frac{Bf}{d} \right) = -\frac{B}{d} (x_L, y_L, f) \quad (3)$$

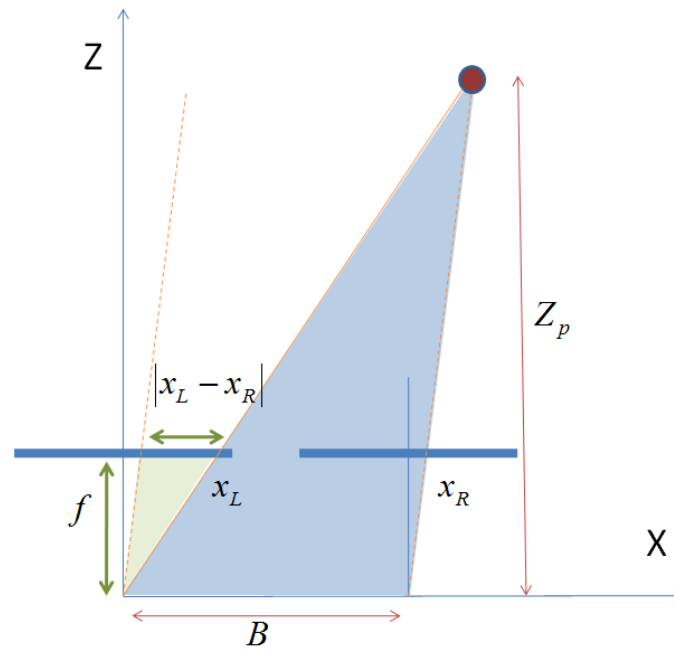


図 3: 奥行きの計算

3 対応点探索

基本画像と参照画像とからそれぞれ対応点ペアを求めて、視差が得られれば、その3次元空間中の座標値を求めることができた。

それでは、その対応点ペアはどのように求めれば良いのか？現在のステレオ研究では、この対応点ペアの如何にして精度良く密に求めるかが最大の課題となっている。本資料では、最も基本的で実装の簡単なブロックマッチング法について述べる。

図4のような図が与えられたとする。対応点探索とは、左側の基本画像中にある点 u_L を取り出したとき、これに対応する参照画像中の点はどこか？を探し出すことである。

実は、ステレオでは、カメラ間のベース長 B に対して、カメラから物体までの距離は十分大きいと仮定している。そのため、”2枚の画像の対応点どうしは、3次元空間中の同一点を見ているので、2つの画像では対応点周辺の見え方が似ているはずである”，ということを暗に仮定している。この仮定に従えば、基本画像中の点 u_L を中心としたブロックを取り出し、このブロックと似たブロックを参照画像から探し出し、そのブロック中心が対応点 u_R となるはずである。このように、2つの画像からそれぞれブロックを取り出し、最も良いマッチングを探し出すことが対応点探索となる。さらに、今回扱っている平行ステレオの場合には、点 u_L の対応点は、同じ高さのスキャンライン上に必ず存在することが分かっているため、このラインに沿って探索を行えば良いことになる。

計算機によって対応点ペアを求めるためには、2つのブロックのマッチング度合いを数値的に評価する必要がある。”最も良いマッチング”とは、この評価値が最大（あるいは最小）となることを意味している。マッチング度合いを数値化する指標として、代表的なもの3つを挙げておく。なお、式中出现する $I(x, y)$ は画像中の点 (x, y) における輝度値やRGB値である。

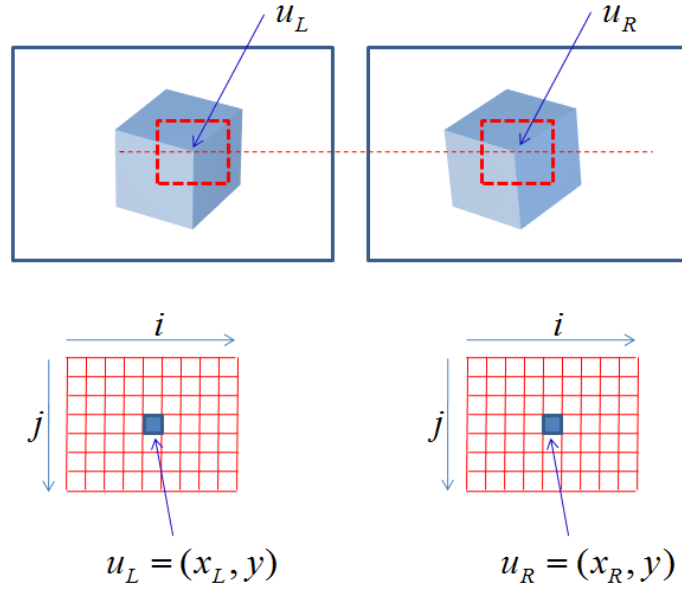


図 4: ブロックマッチング

SAD (Sum. of Absolute Difference)

$$M(x_L, x_R) = \sum_{i=-I}^I \sum_{j=-J}^J |I_L(x_L + i, y + j) - I_R(x_R + i, y + j)| \quad (4)$$

SSD (Sum. of Squared Difference)

$$M(x_L, x_R) = \sum_{i=-I}^I \sum_{j=-J}^J \left(I_L(x_L + i, y + j) - I_R(x_R + i, y + j) \right)^2 \quad (5)$$

SAD や SSD によって計算された値は、小さいほど 2 つのブロックが似ていることを意味する。

NCC (Nomalized Cross Correlation)

$$M(x_L, x_R) = \frac{\sum_{i=-I}^I \sum_{j=-J}^J \left(I_L(x_L + i, y + j) - I_R(x_R + i, y + j) \right)^2}{\sqrt{\sum_{i=-I}^I \sum_{j=-J}^J \left(I_L(x_L + i, y + j) - \hat{I}_L(x_L, y) \right)^2} \sqrt{\sum_{i=-I}^I \sum_{j=-J}^J \left(I_R(x_R + i, y + j) - \hat{I}_R(x_R, y) \right)^2}} \quad (6)$$

ここで、 $\hat{I}(x, y)$ はブロック内での平均値を表し、

$$\hat{I}_L(x_L, y) = \frac{1}{(2I+1)(2J+1)} \sum_{i=-I}^I \sum_{j=-J}^J I_L(x_L + i, y + j) \quad (7)$$

$$\hat{I}_R(x_R, y) = \frac{1}{(2I+1)(2J+1)} \sum_{i=-I}^I \sum_{j=-J}^J I_R(x_R + i, y + j) \quad (8)$$

である．NCC は正規化相関係数で，-1 から 1 の間の値をとり，1 に近いほど 2 つのブロックが似ていることを意味する．

以上の処理によって，基本画像中のある点 u_L について，参照画像中の対応点 u_R が求められる．したがって，基本画像中の全てのピクセルを点 u_L として順次代入していくことで，基本画像全ての点の視差が求められる．

本資料では，対応点探索として最も単純なブロックマッチングを取り上げた．ただし，最近のステレオ研究では，Graph Cut や Belief Propagation といった手法が主流になりつつある．また，ブロックマッチングにしても，矩形型以外の領域マッチングや，動的計画法 (DP マッチング) を組み合わせるなどの手法が多く用いられている．

サブピクセル推定について

以上述べてきた手法をそのまま適用しただけでは，求められる視差 d はピクセル単位であり，整数値しか取ることができない．整数値のみによる視差から 3 次元形状を復元すると，離散的な奥行 Z しか持たない形状が出てくる．したがって，小数点以下が表現できる実数値で視差を求めることが必要になる．

サブピクセル精度の視差推定を行うための手法はいくつか提案されているが，ここでは簡単な 2 次曲線フィッティングによる手法を説明する．例として，SSAD あるいは SSSD を用いた場合，計算された $M(x_L, x_R)$ が最小となる視差を求めることになる (図 5 参照)．ブロックマッチングによって，最小となる視差が整数値で d^* であり，マッチング指標が M^* であったとする．このとき， d^* 周辺でのマッチング指標の変化を 2 次関数によって近似することで，サブピクセル推定を行う．具体的には，まず視差 $d^* - 1$ での指標値 M_-^* と， $d^* + 1$ での指標値 M_+^* を求める．次に，これら 3 点 $(d^* - 1, M_-^*)$ ， (d^*, M^*) ， $(d^* + 1, M_+^*)$ を通る 2 次関数 $y = ad^2 + bd + c$ の各係数を求める．すると，指標値が極小となる視差は， $d = -\frac{b}{2a}$ として，実数値で得られる．なお，NCC の場合にも，2 次関数の極大をとる視差を求めることで，同様にサブピクセル推定ができる．

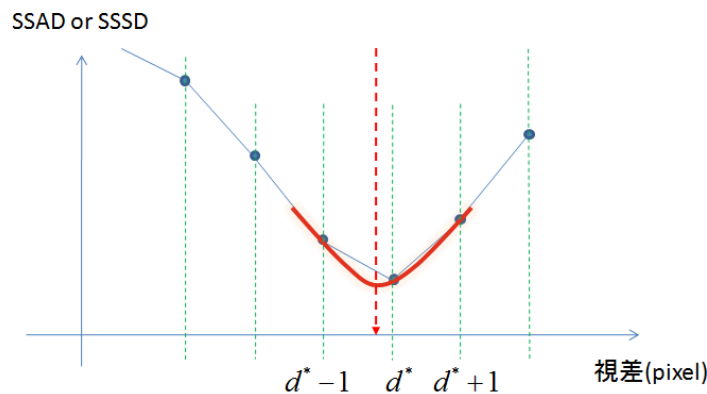


図 5: SSAD，SSSD におけるサブピクセル推定

4 多眼ステレオへの拡張

最後に、これまで説明してきた基礎的事項を用いて、複数枚の画像からの形状推定について述べる．本資料では、2 種類の手法例を示すが、これ以外にも多眼ステレオへの拡張方法は数多く存在する．

4.1 手法その 1(視線ベクトルの交点として)

複数枚の画像それぞれから、ブロックマッチング等により、対応ペアを抽出する．図 6 では、3 枚の画像において、点 u_0, u_1, u_2 が対応している．このとき、3 つの視線ベクトル $\vec{a}_0, \vec{a}_1, \vec{a}_2$ の交点が、求めるべき 3 次元点の座標となる．また、それぞれの画像 $i, i = 0, 1, 2, \dots$ 上において、対応点が (x_i, y_i) にあれば、式 (3) から分かるように、3 次元視線ベクトル \vec{a}_i は $(x_i, y_i, f)^\top$ に平行 ($\vec{a}_i \propto (x_i, y_i, f)^\top$) である．

ただし、この手法で実装する場合、ノイズ等の影響により複数視線ベクトルが 1 点で交わらないことがある．そのため、全ての視線からの距離が最も小さくなるような 3 次元点を交点とみなす手法が、多く用いられる．

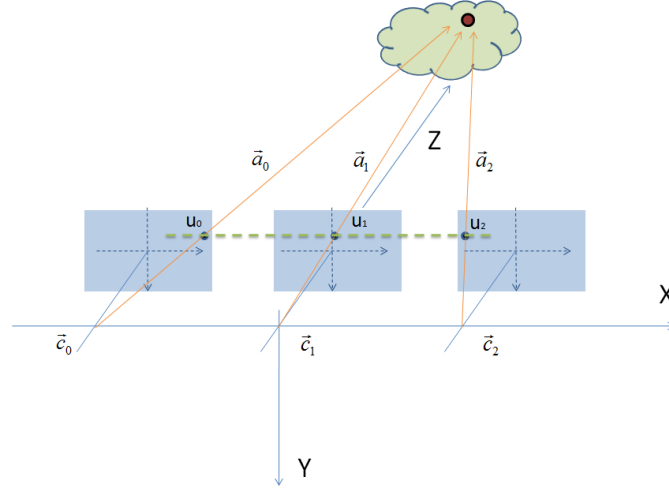


図 6: 視線ベクトルの交点

ヒント

画像 i を撮影したカメラ位置を \vec{c}_i 、単位視線ベクトルを $\vec{a}_i, \|\vec{a}_i\| = 1$ とする．このとき求める点の位置を $(X, Y, Z) = \vec{X}$ とすると、全ての視線からの距離の総和は次のように求められる．

$$f(X, Y, Z, s_i) = \sum_i \left(s_i \vec{a}_i + \vec{c}_i - \vec{X} \right)^2 \quad (9)$$

ここで、スカラー s_i は、点 \vec{X} と各カメラ中心との距離である．点 \vec{X} は、全ての視線からの距離が最も小さくなるような 3 次元点であるので、式 (9) が最小になるような (X, Y, Z) と s_i を求めれば良い．最も簡単な方法としては、式 (9) は凸関数なので、関数値が極値をとるような変数 (X, Y, Z, s_i) を選べば良く、 $\frac{\partial f}{\partial X} = \frac{\partial f}{\partial Y} = \frac{\partial f}{\partial Z} = \frac{\partial f}{\partial s_0} = \dots = 0$ の解として求められる．

4.2 手法その2(各点の奥行き探索を通して)

複数枚の画像のうちの1枚を基準画像として考える．図7では中央の画像を基準とする．基準画像中のある点 $u_1 = (x_1, y_1)$ は3次元空間中では，点 \vec{c}_i を通り視線ベクトル veca_1 に平行な直線上のどこかに必ず存在する．したがって，この点の3次元座標は，未知のパラメータ t を用いると， $(tx_1 ty_1 tf)^\top + \vec{c}_1$ として表すことができる．3次元形状を復元するためには，このパラメータ t を求めれば良い，ということになる．

もし，仮定した t の値が正しければ，3つの点 u_0, u_1, u_2 を中心とした3つのブロックのマッチングがとれるはずである．

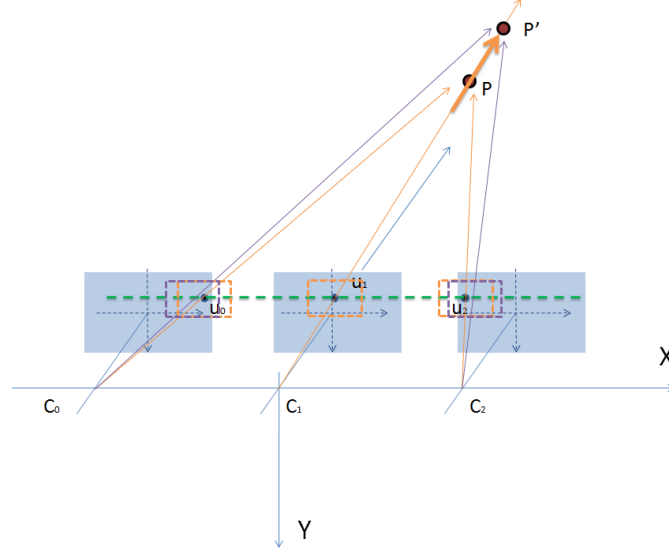


図7: 奥行き探索

このように，この手法では基準画像の全ての点に関して，奥行パラメータ t のとり得る値の範囲をあらかじめ設定しておき，この範囲内で最良のマッチングがとれる（例えば，SSDの総和が最小になる等）ような t を順次探索していけば良い．なお，奥行パラメータ t を離散的に探索したときにも，前章で説明した2次関数フィッティングによって，連続的な実数値で奥行きが推定できる．

ヒント

まず t の値の範囲を適当に決め，範囲内を探索することで最も尤もらしい値 t を決めれば良い．3次元空間中のある1点がとりあえず $(tx_1 ty_1 tf)^\top + \vec{c}_1$ であると仮定すると，この点が他の画像でどの位置に写っているかを計算することができる．例えば，カメラ0では，点 $(tx_1 ty_1 tf)^\top + \vec{c}_1$ とカメラ中心 \vec{c}_0 とを通る直線と，平面 $Z = f$ との交点に写ることになる．ただし，カメラ0から観た座標系に変換する必要があるので，画像0に写った点の位置を $u_0 = (x_0, y_0)$ とすれば，適当なスカラー s を用いれば，

$$t \begin{pmatrix} x_1 \\ y_1 \\ f \end{pmatrix} + \vec{c}_1 = s \begin{pmatrix} x_0 \\ y_0 \\ f \end{pmatrix} + \vec{c}_0 \quad (10)$$

が成り立つので，これを満たす点 (x_0, y_0) が，画像0での対応点になる．

4.3 上記ステレオの問題点

本資料では、ブロックマッチングによって対応点を探索した。ブロック領域に模様 (テクスチャ) 等がない場合には、どの点が対応点なのか分からない場合がある。また、ある画像では見えている箇所が、他の画像では見えない、といった遮蔽 (オクルージョン) の問題があると、対応点自体が存在しないことになる。さらに、単純なブロックマッチングのみによって 3 次元復元を行った場合、前景物体の境界に背景領域が加わる膨張効果が起こる。

5 課題

3 枚の入力画像全てを用いた 3 眼ステレオによって、対象シーンの 3 次元形状を求める。

5.1 入力データ

Middlebury のデータセットは、コンピュータビジョンの分野でステレオアルゴリズムの評価を行うサイトで用意されている。世界中の研究者は、新たなステレオアルゴリズムを開発した場合、このサイトにあるデータセットを用いて、その性能を評価することが一般的になっている。ここでは、そのデータセットにある画像を用いる。

画像サイズは 584×466 であり、カメラは、X 方向 (水平方向) に等間隔に配置されたものである。なお、各画像上の 2 次元座標 (x, y) の原点としている画像中心は、画像の中央 $(292, 233)$ とする。また、焦点距離は $f = 500.0$ を用いることにする。3 次元座標系を図 1 のように設定し、各カメラ位置は $viewC$ を撮影したカメラが $(0, 0, 0)$ にあるものとし、 x 方向に 0.1 間隔で並べられたものとする。

$viewL = (-0.1, 0, 0) \quad \dots \text{Left}$

$viewC = (0, 0, 0) \quad \dots \text{Center}$

$viewR = (0.1, 0, 0) \quad \dots \text{Right}$

手法その 2 で解く場合、3 次元点を探索する範囲は、 $2.0 \leq Z \leq 7.5$ 程度で良い。



図 8: データセットの画像。左から $viewL$, $viewC$, $viewR$ 。

5.2 出力データ

デフォルトとして、Windows で動作するソフトを用意しておく。図 9 のようなテキスト形式ファイルを読み込むことで、3D 形状を表示できる。

表示ソフトでは「ファイル」「開く...」でテキストファイルを読み込むか、テキストファイルをソフトのウィンドウにドラッグ&ドロップすることで表示させる。画面上では、マウスの左ボタンで併進、右ボタンで回転、中央ボタンでズームできるようになっている。

復元した3D点数	x y zの座標値	その点のRGB値
36349	-0.996863 -4.34368 -0.843289	140 154 155
	-1.07544 -4.5921 -1.17308	85 95 89
	0.355353 -2.74553 0.256178	141 126 104
...		
...		
...		

図 9: ファイルフォーマット: 第 1 行に復元した 3 次元点の数, 第 2 行から各点の X,Y,Z 座標値とその点の RGB 値 (0 ~ 255) を並べる. したがって, 全体の行数は (復元点数 + 1) となる.



図 10: 付属ソフトにデータセットを 3 次元復元したモデルを表示した例

5.3 提出物

レポート (フォーマット自由), プログラム (使用言語自由), 復元形状データ (上記指定フォーマット) の 3 点を提出すること. なお, ステレオのプログラムはインターネット上にあるコードの使用も認める. ただしその場合, レポート内でどのようなアルゴリズムを使用しているか, 説明をつけること.

参考文献

- [1] 安居院猛, 長尾智晴, "C 言語による画像処理入門," 昭晃堂, 2000.
- [2] <http://vision.middlebury.edu/stereo/>