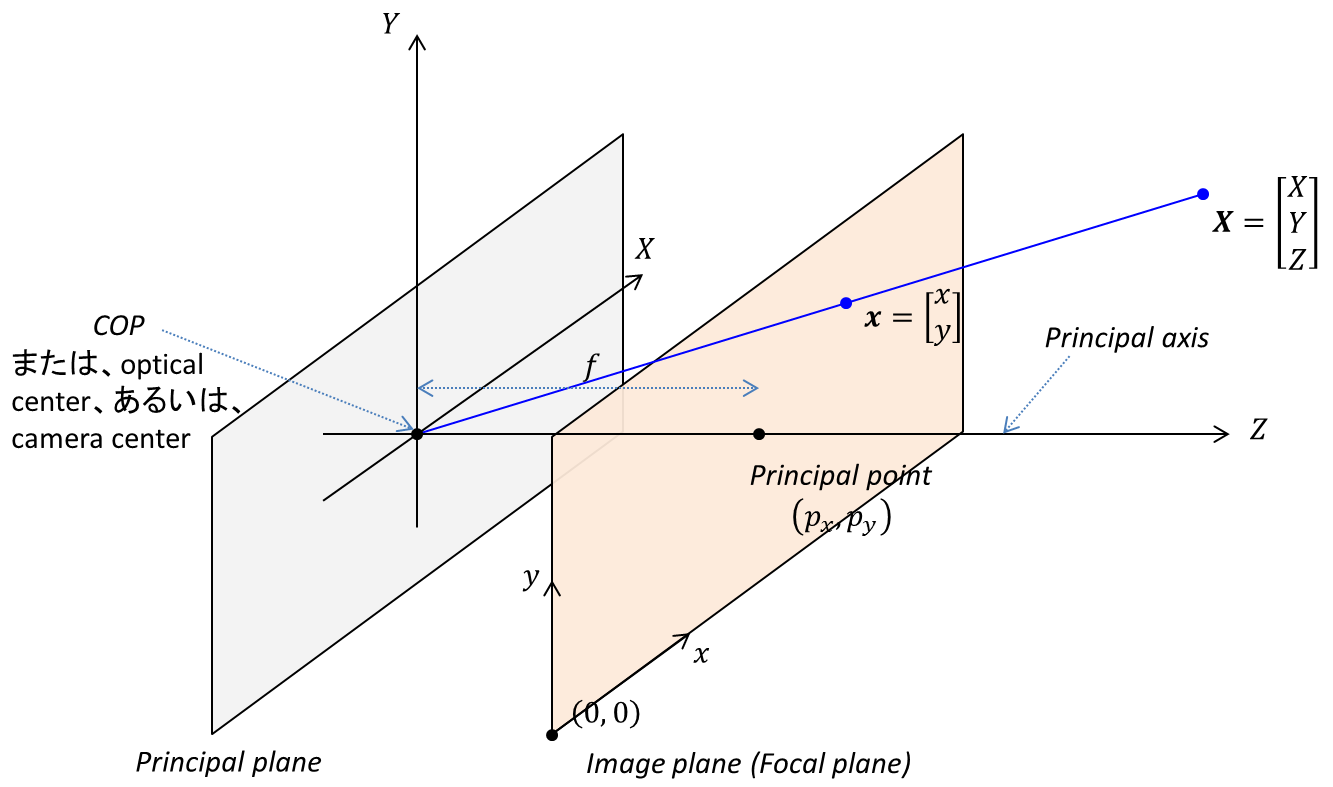
# Camera Calibration

## Intrinsic Calibration Matrix



まず、カメラのレンズの中心は**Center of Projection、つまりCOP**と呼ぶ。オブジェクトの、カメラ座標系での座標をとし、そのイメージプレーンへの写像点の座標をとする。は焦点距離で、COPからイメージプレーンへの距離だ。また、**COP**からイメージプレーンへ下ろした垂線の足を、**principal point**と呼ぶ。Principal pointのカメラ座標系での座標をとする。この時、

これを、例によってHomogeneousなんとかの書き方にしてやると、

(1)

ここで、左辺を**、**、そして、

とおくと、

となる。**注意すべきは、ここで言うは、カメラ座標系でのオブジェクトの座標だ。より一般的な、ワールド座標系での話しは、1.2で説明する。**

ここでがカメラの**内部パラメータ行列**だ。

## Extrinsic calibration matrix

カメラ座標系とワールド座標系との関係を表す行列が、外部パラメータ行列だ。カメラが１個なら、カメラ座標系とワールド座標系を同じと見なしても良いんだけど、例えばカメラが２個ある場合などは、そうもいかないよね。

ワールド座標系からカメラ座標系への変換は、

と表せる。ここで、は、カメラ中心の、ワールド座標系での座標だ。実際に上の式にを代入すれば、となるので、確かにつじつまが合っていることが分かる。これを、例によってhomogeneous coordinatesで表現すると、

これを、1.1で得たの式と合体させると、より一般的な式が得られる。

ここで、一般にとおいて、

これが、より一般的な射影変換の式だ。ここで、が、カメラの**外部パラメータ行列**だ。

なお、この結果から、カメラの射影変換行列の自由度は9であることが分かる。が自由度3（、、）、が自由度3、が自由度3だ。

## ピクセル座標

これまで、image plane上での座標は、何らかのユニット単位で議論してきた。実際のCCDカメラにおいて、1ユニットあたりのピクセル数を軸方向が、軸方向がとすると、ピクセル座標は、以下の式で計算できる。

これに、カメラの内部パラメータ行列を合わせると、

これが、ある意味で、本当のカメラ内部パラメータ行列で、スライド「Camera Calibration」p.30の行列だ！

最後に、もう一度、射影変換をまとめると、

**カメラの射影変換行列の自由度は、9から10に増えた。**なぜなら、のが、になったので、の自由度が3から4に増えたから。

## Skew Parameter

さらにより一般的なカメラ内部パラメータ行列ではskew parameterを加えて

とする。これにより、カメラの射影変換行列の自由度は、11になる。なお、このようなを**「Finite projective Camera」**と呼ぶ。

## Optimization

内部パラメータ行列と外部パラメータ行列をまとめると、

(3)

まず、大きく分けて２ステップでoptimizationを行う。ステップ１では、、を画像の中心、とみなすことで、線型方程式に変換し、すばやく行列をestimateする。ステップ２では、元の非線型方程式を使って、estimateした行列を初期値として、非線形最小二乗法で最適解を探す。ポイントは、スライドp.49にある通り、**「非線形最小二乗法の収束は、初期値の精度に依存する」**だ。つまり、ステップ１で良い精度で初期値を求めることが重要となる。

**ステップ１（回転行列と、の推定）**

まず、とみなす。式(1)より、

ここで、、とおくと、

(4)

１行目から２行目を割ると、

パッド上の点の座標は0としても一般性は失わないので、。よって、

よって、

これを行列の形で表すと、

個の点があれば、この式を個積み上げていけば良い。この線型方程式の形は定数項がないので、**Homogeneous Linear System**と呼ばれ、左側の行列に対しての最小固有値に対応する固有ベクトルが、右側のベクトルの解となる。

スライドp.39では、**Cholesky分解**を使って解くと書いてあるが、Cholesky分解は正定値実対称行列にしか使えない。一般に、上の行列は縦に長い（overdetermined）ので、Cholesky分解は使えない。一般的に、**OverdeterminedのHomogeneous Linear Systemを解くには、SVD分解をし、行列の最後の列が解**となる。

ここで、ポイント！Homogeneous Linear Systemの解が得られたら、任意の実数でスケールしたやつも解だ。じゃあ、どうやってユニークな解を得るのか？それには、回転行列の各行がnormalizeされているという条件と、直交条件を使用すれば良い。つまり、SVDで得られた解に対して、

(5)

最初の２つの式を変形して、

(6)

上と下の式を掛けると、

(7)

また、３つ目の式を変形して、

両辺を二乗して、

(8)

式(7)から(8)を引くと、

(9)

ところで、式(6)より、

これを式(9)に代入して、

整理して、

(10)

SVDにより、、、は求まっているので、式(10)からを計算できる。

が求まったら、式(6)を使ってとを計算できる。後は、それぞれで割れば、回転行列が得られる。

なお、、については、式(5)を見ると分かるように二乗しているので、符号が＋か－か不明だ。式(5)の３つ目の式より、との積の符合は決定する。つまり、

しかし、これでもまだ２つの候補が残っていしまう。例えば、が正なら、とが共に正、または、共に負という２つの候補がある。これを見極めるためには、実際に3Dの座標を射影してみて、2Dの座標に近い方を選ぶしかない。なんだか、ヘタレだねぇ。。。具体的には、個の点について以下を計算し、正ならＯＫ。もし負なら、もう一方の符号にする。

ただし、は番目の観測された2D座標、は番目の射影された2D座標。

**※はまだ求まっていないので、適当な値を使うのか？**

以上で、回転行列の最初の２行のベクトルとが求まった。最後に、３つ目の行ベクトルは、単純にcross productで計算できる。つまり、

以上から、回転行列が得られる。なお、スライドでは、ステップ２以降の説明がない！？

**ステップ２（との推定）**

次に、とを計算する。まず、式(4)より、

式(2)を代入して、、を消去すると、

整理して、

思い出して欲しい。既に回転行列は得られているので、右辺は既知だ。つまり、との線形方程式になっている。一応、行列の形にしてあげると、

個の点があれば、この式を積み上げれば良い。の形になるので、ヤコビ法なり、ガウス・サイデル法でとを計算できる。

ここまでのステップにより、焦点距離と外部パラメータ行列が求まった。

**ステップ３（principal pointとの推定）**

最後に、estimateした行列と焦点距離を初期値とし、、を画像の中心、を初期値とし、以下の非線形方程式を使って、最小二乗法で最適解を探す。

なお、は番目の射影された2D座標で、式(3)により計算できる。

※この最適化は、たぶん、Minpackのlmdifを使って解くんだと思うけど。。。

## Zhangのアルゴリズム

まず、以下のモデルを使用する。

なお、

(1)

は、カメラ内部パラメータ行列だ。

**ステップ１（Homography行列を解く）**

パッドは、ワールド座標系の平面上にあると見なしても一般性を失わないので、上のモデルでとおく。すると、

つまり、

(2)

よって、

となる。ここで、を**ホモグラフィ行列**と呼ぶ。ホモグラフィとは、２次元の座標から２次元の座標への変換のことだ。あとは、一般的なホモグラフィの解法を使って、ホモグラフィ行列を求めれば良い。具体的には、まず、上の式を以下のように分解する。

そして、３番目の式を上の２つの式に代入してを消去してやると、

これを、以下のように行列の形に変形してやる。

思い出して欲しい。は、パッド上の点だったよね。１つの点につき、上のような式が１つできる。パッド上の個の点について、上のような式を積み重ねてやると、以下のようになる。

これは、の線形方程式だ。このような形を**Homogeneous Linear System**と呼ぶ。Overdeterminedの場合は、左の行列に対して、の最小固有値に対応する固有ベクトルが、右のの解となる。実際には、**SVD（特異値分解）**を行った時に、行列の最後の列が解となる。（assignmentでは、これを実装するとextra creditだと言っている。）

※ちなみに、このアルゴリズムを**DLT(Direct Linear Transform)**とも呼ぶらしい。

**ステップ２（行列を解く）**

まず、式(2)で、、は互いに直交し、ノルム（長さ）が同じだということから、制約条件を生成したい。式(2)で、とおいたよね。なので、

とおくと、

よって、、は互いに直交し、ノルム（長さ）が同じだということから、

式を変形して、

(3)

(4)

ここで、とおくと、行列は式(1)で定義しているので、実際に力ずくで計算できて、以下のように求まる。

(5)

これを見ると分かるとおり、行列は対称行列だ。なので、を以下のようにおく。

また、式(3)、(4)にを代入すれば、

ここで、一般に、について考えて見る。

これは、さらに以下のように変形できる。

(6)

左側のベクトルを、右側をとおくと、

と表される。これを、式(3), (4)に適用すれば、

行列の形にしてやると、

(8)

式(6)を思い出して欲しい。ベクトルは、homography行列の要素で構築されている。そして、それはステップ１で計算済みだ。つまり、上の式で、ベクトルだけが未知なのだ。

一応、行列の形にしておくと、

各画像について１つのhomography行列が求まる。つまり、上の式が１つできる。個の画像があれば、上の式(8)を個積み上げれば良い。そして、SVDを使ってベクトルを解く。ただし、画像が１個しかない場合は、式が２つということ。なので、２つの未知数しか求めることができない。一般的に、、とは画像の中心とみなし、残りのパラメータとを求めることになる。一方、画像が２個しかない場合は、式が４つということなので、４つの未知数しか求めることができない。一般的に、とみなし、残りの４つのパラメータを求めることになる。（assignmentでは画像を最低２個と言っているので、と見なして良いということだ）。ということは、ということ。さらに、、を画像の中心とみなし、以下の式を追加してあげることで、式が７個となり、行列を求めることが出来る。

この式(8)も**Homogeneous Linear System**なので、**SVD**を使って解くことが出来る。行列が求まれば、式(5)を使って、、、、、を計算すれば良い。つまり、

こうして、カメラ内部パラメータ行列が求まった。

次に、外部パラメータ行列を求めよう。思い出してくれ。Homography行列は、

よって、

つまり、

ただし、

もも既に求まっているので、、、は簡単に求まるよね。また、回転行列の３つ目の列については、

で、普通に解ける。

**ステップ３（推定値の改善）**

ここまでで、基本的には内部パラメータ行列、外部パラメータ行列は得られたが、iterativeな最適化により、更に推定値を改善できる。

個の画像を使用し、各画像内に個の点がある時、最適化に使用する目的関数は、

(10)

ただし、は番目の画像内での番目の点の座標、は3D座標から写像により得られた二次元座標だ。これを、**Levenberg-Marquardtアルゴリズム**（**Minpack**の**lmdif関数**に実装されている）で解けば良い。

**ステップ４（レンズ歪みの推定）**

まず、レンズ歪みのモデルは、

ただし、は、歪みなしの二次元座標、は、歪みより実際に観測された二次元座標。また、

ただし、は正規化された座標で、以下で求められる。

後は、**Minpack**を使って、以下の目的関数を最小化すれば良い。

(14) (10)

## Bundle Adjustment

Wikipediaによると、Camera calibrationやら3D Reconstructionやらにおいて、必ず最後のステップとして実施されるプロセスとのこと。コンピュータビジョンの世界ではめちゃくちゃ有名らしい。

基本的には、3D座標を射影した2D座標と、実際に観測された2D座標との差を最小化する処理だ。Levenberg-Marquardtアルゴリズムが、この処理によく使用される。簡単に説明すると、現在の値の近傍について、目的関数がほぼ線形であるという前提に基づき、勾配を使って極小値へ向かう。**（Gradient Descentと何が違うの？）**

## Algebraic Error vs Geometric Error

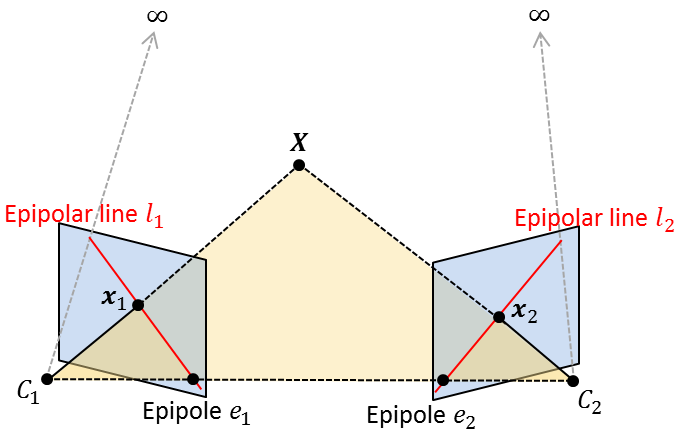
例えばZhangのアルゴリズムで、Homogeneous Linear Systemを解くところはAlgebraicアプローチと呼ばれる。なぜなら、というHomogeneous Linear Systemを解く際に、は最小化したい何らかの距離ではない。つまり、特に意味が無く、ただ制約条件に基づいて機械的に代数を使って解いているだけなのだ。

一方、最後のnon-linear optimizationのステップは、意味のある何らかの距離を最小化するよう最適化している。

一般的に、**「まずは線型方程式を使ってalgebraic errorを最小化し、得られたパラメータを初期値として、次にnon-linear optimizationを使って目的の距離を最小化する」**というのがよく使われる流れだ。

# Stereo Reconstruction

## Epipolar Geometry



左のカメラにとって、点はへ写像される。一方、右のカメラにとっては、点はへ写像される。つまり、

点は、と表せるので、左のカメラ中心とを通る直線上の点は、次の式で表される。

**※なぜ、じゃないの？**

左のカメラ中心との、右のカメラへの写像は、それぞれ、だ。なので、この２つの点を通る線、つまり、epipolar line は、

一方、はepipole なので、

となる。ここで、

と定義すると、となり、epipolar lineの定義が得られた。

また、は上にあるので、だ。を代入すると、

これが**「epipolar constraints」**だ！は**Fundamental Matrix**と呼ばれる。*、*は観測データなので、これを使えばを求めることが出来る。

**HZテキストP.245 9.2.4Properties of the fundamental matrixの解説**

まず、1つ目の画像上の点に対応する、2つ目の画像上のepipolar line は、で表される。逆に、2つ目の画像上の点に対応する、1つ目の画像上のepipolar line は、で表される。

次に、2つ目の画像上のepipole は上にあるので、だ。よって、

1つ目の画像上の全てのでこれを満たすので、を満たす必要がある。つまり、は、の「left nullベクトル」と呼ばれる。（の左側にあって、掛けたものが0になるから）。転置してやれば、となる。

同様に、1つ目の画像上のepipole は上にあるので、だ。よって、

2つ目の画像上の全てのでこれを満たすので、を満たす必要がある。転置してやれば、となる。はの**「right nullベクトル」**と呼ばれる。（の右側にあって、掛けたものが0になるから）

**HZテキストP.244 Example 9.2の解説**

ワールド座標系をどのように取っても一般性は失われないので、カメラ１の座標系をワールド座標系とする。つまり、

この時、カメラ２について、

とおくと、1.2で述べたとおり、カメラ２の中心である。

この時、

同様に、

**HZテキストP. 246 9.2.5 The epipolar line homographyの解説**

１つ目の画像上の、epipole を通らない直線がある時、との交点は、

で表される。ところで、だったから、

同様に、

で、だから、

また、としてを選ぶと、。一般的に、は0でないよね。一方、ある点が直線上にある時、だ。逆に、直線上にないとき、だ。今、なので、直線はepipole を通らないということ。つまり、先ほど述べた直線の条件を満たしている。なので、をとして選ぶ。すると、

同様に、

**HZテキスト P.250 9.4 Geometric reprensetation of the fundamental matrixの解説**

あほほ

**HZテキストP.253 9.5 Retrieving the camera matricesの解説**

は２つの画像の点、の対応にのみ依存していて、ワールド座標系には依存していない。また、カメラ行列、に対して、ある投影変換行列を適用したカメラ行列、も、3次元座標上の点を２つの画像上の同じ点、に投影する。つまり、解がいくつもあるということ。これを**「projective ambiguity」**と呼ぶ。

**HZテキストP.257 9.6 The essential matrixの解説**

カメラ座標系での座標。これを、**「normalized coordinates」**と呼ぶ。

これを使うと、Fundamental matrixの式は、

変形して、

ここで、

(2.5)

とおくと、

となる。

## テキスト（Multiple View Geometry in Computer Vision）10章

まず、2.1でも挙げたが、以下の３つの式がとても重要な式だ。

(2.6)

(2.7)

(2.8)

**ステップ１（Fundamental matrix を求める）**

、とすると、式(2.6)より、

これを整理すると、

後は、たくさんの観測点について、上の式を積み上げてやれば、のhomogeneous linear systemの形になるので、例によってSVDでを解くことが出来る。

※テキスト11章に詳細が説明されている。

**ステップ１のおまけ（Sampson approximation）**

が求まったら、それに基づいて観測された2D座標を修正できる。特に、目視で2D座標を計算した時など、かなり誤差があるので、この修正は必要だろう。HZテキストP.314に記載されている。

※実際に実装して試したが、わずかな修正（0.1とか0.2）なので、結果にはほとんど影響なし。むしろ、bundle adjustmentの方が重要だと思う。

**ステップ２（Essential matrix の計算）**

式(2.5)より、を計算する。

**ステップ３（カメラ行列の計算）**

まず、射影行列、は一意に決まらないというのは、なんとなく理解できると思う。

数学的にちゃんと書くと、

に対して、次のように変形しても良いよね。

これは、点が、射影行列、によって点、に射影されるということだ。点がたくさんあったとしても、すべて同じで点は変わっちゃう。つまり、3D座標は、点の数に関わらず、一意に求まらないということだ。

**え！じゃあ、どうやって3D座標を求めれば良いの？？？**

答えは、ワールド座標系を、自分の都合に合わせて決めちゃえば良い。つまり、

(2.10)

とおく。この時、

(2.11)

となる。行列、は既に分かっているので、射影行列も計算できる。

ただし、行列はrank 2の行列で、このままではまずい（？）そこで、テキストResult9.14に記載されている通り、適当なベクトルを使って、以下のように変形する。

ただし、は適当な3次元ベクトルで、は適当な0でないスカラー値。

ただし、得られた行列は、**「canonic カメラ行列」**と呼ばれるものだ。

**ステップ５（3Dポイント座標の計算）**

テキスト12章に説明有り。

まず、射影行列が分かったので、

各要素も書くと、

つまり、

これはhomogeneous座標系なので、

整理すると、

行列の形にしてやると、

１個の点について２つの式が出来上がる。２つの画像があれば、４つの式だ。

(2.12)

なお、ここで、と表記している。後は、擬似逆行列を使ってこの線型方程式を解くだけだ。

なお、講義スライド「Stereo and 3D Reconstruction」のP.53では、代わりに以下の式が掲載されている。

式(2.12)と実は同じなんだけど、観測される画像の座標ということに基づいて式を組み上げている。が未知数と入るので、その分、計算量が増えてあまりうれしくない。式(2.12)を使うべきだ。また、P.51では、未知数がで、式がと言っているが、これは、個の画像を使う場合、未知数はの3つに加えて、各画像ごとにが異なるので、合わせて。一方、式は、上に書いたとおり、1つの画像について3つなので、全部でということ。なので、なら式が解けるということだ。これは、式(2.12)も同様だ。つまり、2つの画像があり、対応する点があれば、その点の3次元座標をtriangulationで求めることができるということ。なんだか当たり前のことを回りくどく言ってるような気が。。。

**ステップ６（Bundle Adjustment）**

しかし、この方法だと問題がある。Camera Calibrationの時と同様で、あくまでNumericalに解いただけなので、実際にreprojectすると結構な誤差がある。なんかいろいろHZテキストP.314に説明があるけど、要するに、次のコスト関数を使った方法が望ましいみたい。

まず、我々は、以下のコスト関数を最小化したい。

ただし、、はreprojectして得られた2D座標だ。また、最適化にあたって、を満たすようにする。（どうやって？）

これを、例によって**Levenberg-Marquardtアルゴリズム**（**Minpack**の**lmdif関数**に実装されている）で最適化すればよい。

講義スライド「Stereo and 3D Reconstruction」P.55で、がunknownと書いているけど、だけなら、bundleじゃなくて各点ごとにadjustすれば良い。なんでbundleするのか？？？

俺の予想：

カメラ２の行列も併せて最適化するのでは？だとすると、未知数は点の数とで、合計。一方、射影された点のX座標、Y座標についての式は、個。これを使って、最適化するのでは？

しかし、実はダイレクトに解く方法があるらしい。テキストp.318のAlgorithm 12.1に記載されているが、難しくて理解できない。。。。

## 補足

ところで、

という説明が、テキストブックにあるし、講義スライドにもある。でも、なんで？絶対おかしいと思うのだが。他の資料を見て見よう。

⇒分かった。これは、あくまで2D座標系での話しをしているのだ。2Dの線を、と表すと、線上の点は、

という制約を持つと言える。また、２つの点、を通る線は、

を満たす。本当か、確認しよう。右辺は、

となる。一方、、を通る線は、

変形して、

なので、上の式と一致するよね。

これらの性質は、すごく重要なので覚えておけ。

## 実装にあたって

取り急ぎ、分かったこと。

1. OpenCVのfindFundamentalMat関数でを直接求めちゃうと、得られた3D座標はフラットになっちゃう。なんでか分からないが、とりあえず、SfM-Toyサンプルの実装と同じように、以下の式でを計算することにする
2. HZテキスト P.259、Result9.19に記載されている通り、を使った場合でもについて4つの候補があり、実際にreprojectしないとどれが正しいか分からない。つまり、  
   とSVD分解できるとき、  
   ただし、はの3列目のベクトルで、また、  
   の時、は、以下の4つの候補がある。  
   どれが正しいかは、実際に射影してチェックするしかない。  
   を計算し、ほとんどの点についてならＯＫだ。そんなは1つしかないので、これで判断すれば良い。  
   ちなみに、このようにから求められた射影行列は、**「normalized camera matrix」**と呼ばれ、3D座標を以下で説明するnormalized coordinatesに変換する。
3. 2D座標のnormalization  
   HZテキストP.257 9.6節に記載されている**「normalized coordinates」**を使うことで、3D座標の計算は4つの候補に絞られることが分かった。では、このnormalized coordinatesをどうやって計算するか？単純に、  
   で計算できる。ちなみに、これは、俺が自流で考えた以下の式と同じだ。  
   実際に、計算すれば分かる。
4. Fundamental matrixを計算した後、2D座標をrefineできる。Sampson近似（HZテキストP.314 12.4）を使うか、optimal（HZテキスト P.315 12.5）を使う。