最適化手法を用いた画像指令値の誤差低減に関する提案

2017/1/17

M2 李 尭希
藤本 博志

はじめに：FF指令値を用いた動作再現

- 今までの発表：① 動作再現で指令値に遅れなく追従するにはFFが必要。
  ② FF指令値を正しく作るにはイメージヤコビアンの推定が必要。
  ③ 時不変イメージヤコビアンだと推定が容易。

■ Control Scheme

$$\underbrace{\boldsymbol{V}_{ref}(t)}_{\text{3D robot speed ref}} = -\lambda \boldsymbol{J}^{+} \left(\boldsymbol{\xi}_{ref}(t) - \boldsymbol{\xi}_{c}(t)\right) + \underbrace{\boldsymbol{J}^{+} \frac{d}{dt}\boldsymbol{\xi}_{ref}(t)}_{\text{Feedforward must be accurate}}$$

Fe informati

Correct camera move from current image

Fe t

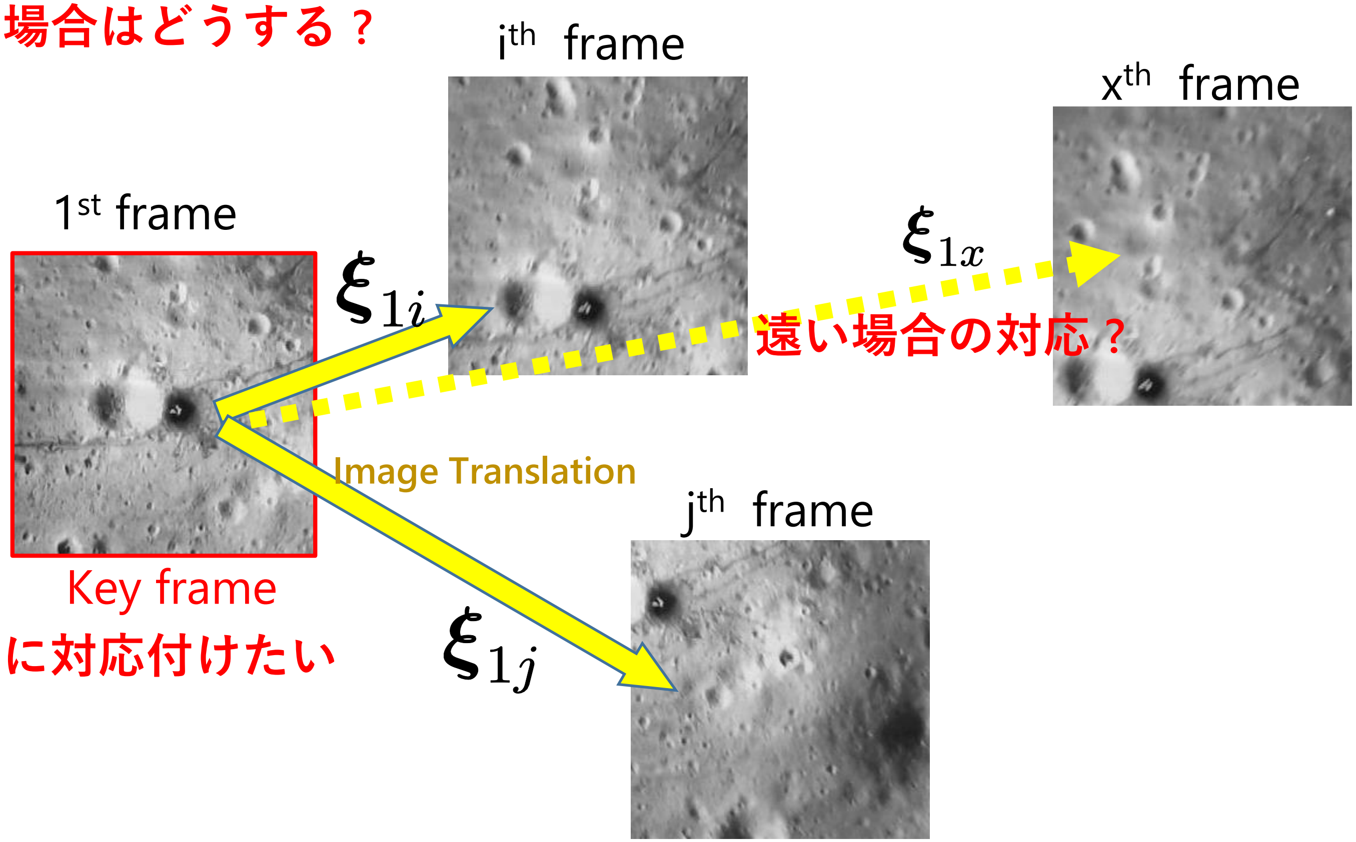
Predict next necessary move from video

- **現状の課題：① 距離の推定**
  ② 指令値の作成方法 → 今回はこれについて話します。
  ③ 指令値のノイズ処理

指令値の作成時における問題点

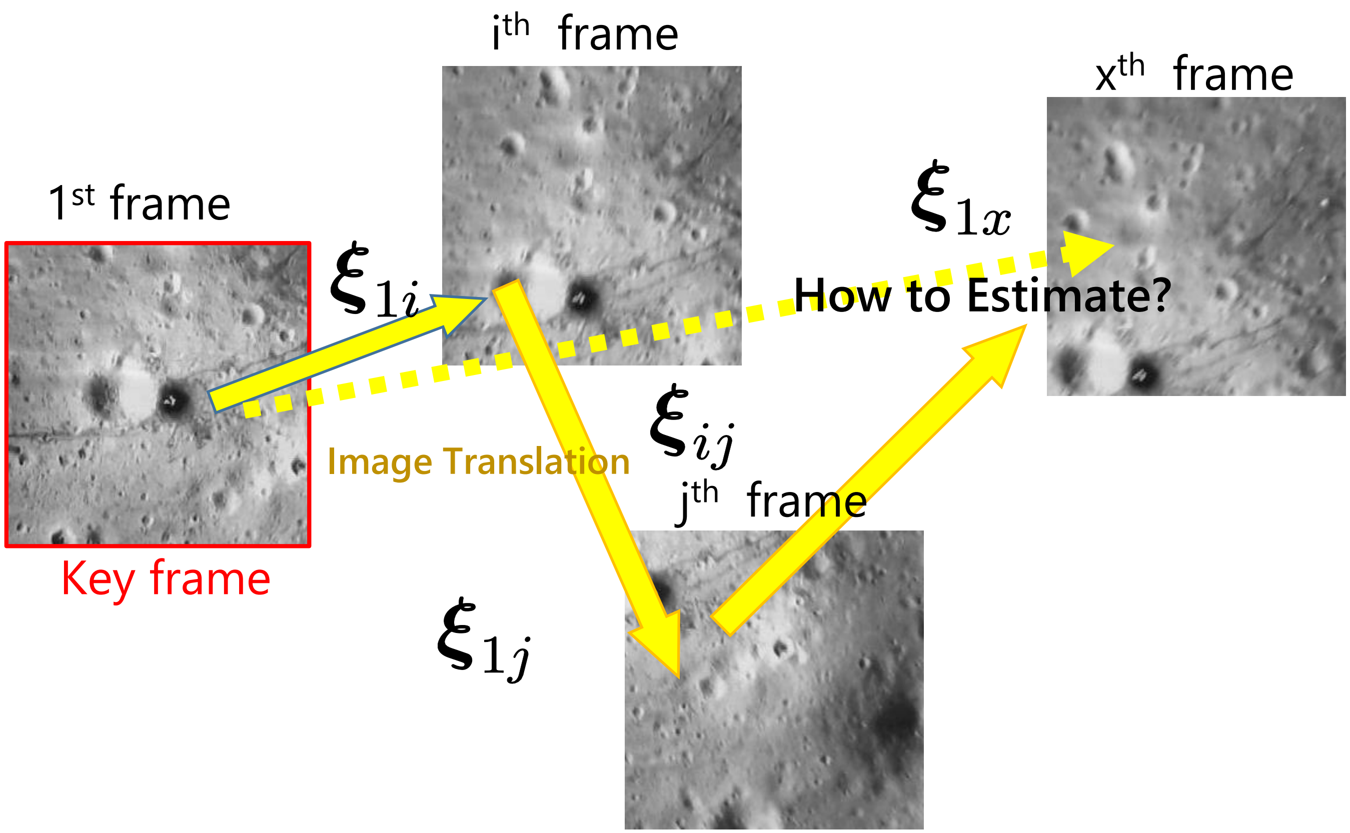
- FF指令値：ビデオ画像から再現した画像の変位から作る
- **全ての画像を一番目の画像（Key frame)に関連付けたい。**
  → ∵時不変イメージヤコビアンが使える＝FF指令値が簡単に作れる

遠い場合はどうする？



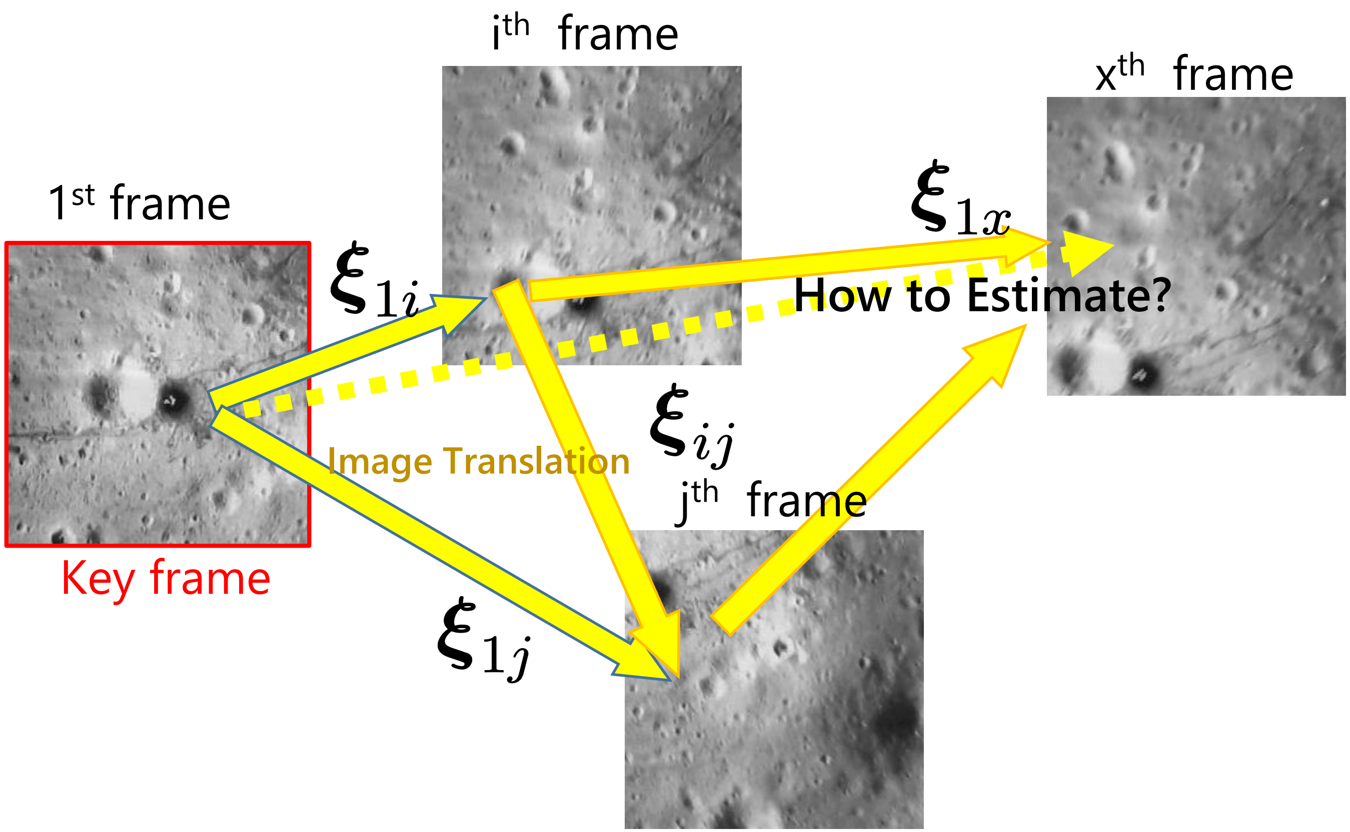
従来法：数珠つなぎに対応

- 従来：近い対応を一つずつ辿っていく
- **誤差が蓄積してしまう問題**がある。

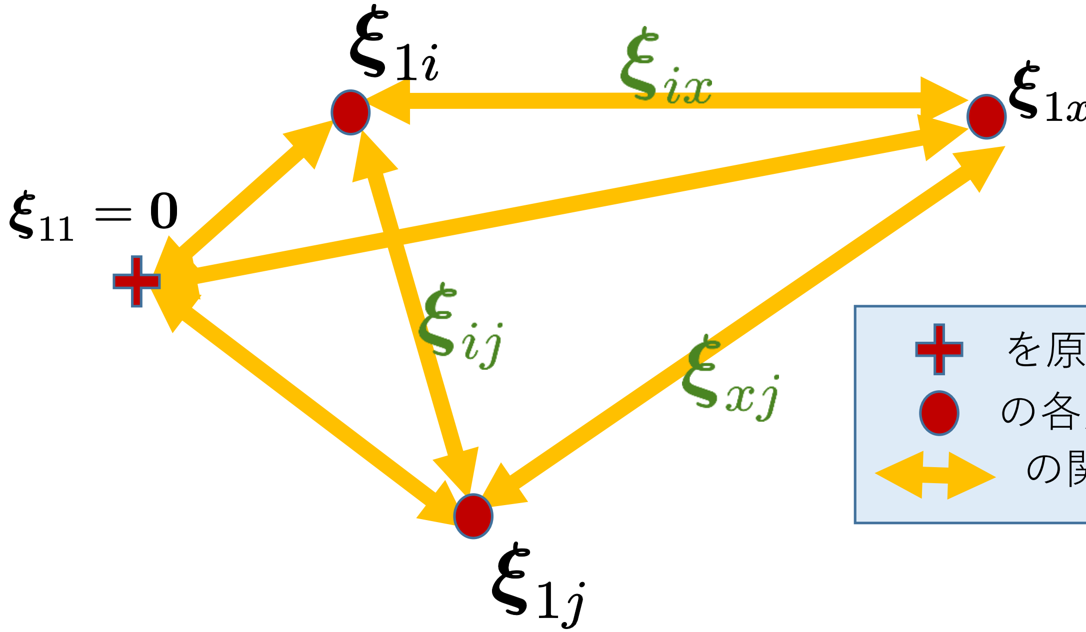


提案法：すべての関係を用いる

- 提案法：使える対応を全て使う。
- どの関係を用いる？どのようにして最適な値を出すのか？

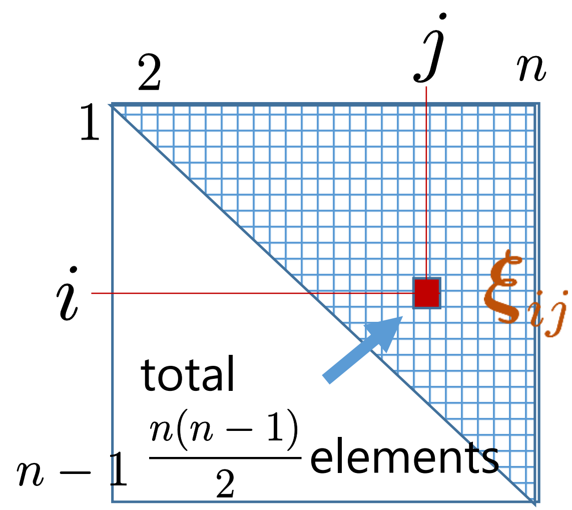


問題の定式化



$\boldsymbol{\xi}_{ij}$ i番目から見たj番目のフレームの変位

を原点と考えた時
の各点の座標を
の関係性から求める。



網掛けの部分にのみ情報が入る

原点含めn個の点がある場合,
n(n-1)/2個の関係性がある。

これを行列で管理する。

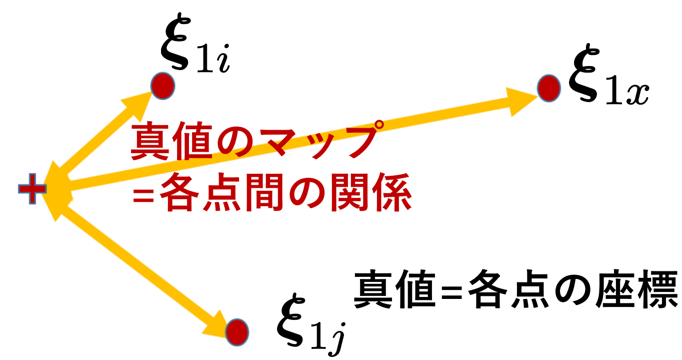
これを，対応付け行列と呼ぶことにする。

最小化問題への帰着

目的：１番目のフレームとの全てのフレームとの対応を求める　$\boldsymbol{\xi}_{1x} \quad (x = 2, 3, ..., n)$

手段：予め計測を繰り返して作ったマップを用いて
尤もらしい対応を求める

→「尤もらしい」をきちんと定義する

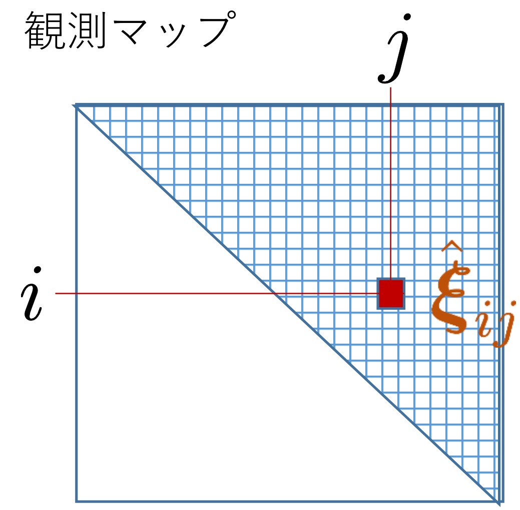


誤差の二乗和 ＝ (真値-計測値)^2 の最小化

真値：　$\boldsymbol{\xi}_{1x} \quad (x = 2, 3, ..., n)$

真値=各点の座標から
対応値を求めることで　$\boldsymbol{\xi}_{ij} = f(\boldsymbol{\xi}_{1i}, \boldsymbol{\xi}_{1j})$
真値のマップを作成

対応分を埋めたマップ ＝
観測マップ



観測値：　$\hat{\boldsymbol{\xi}}_{ij}$　＝ **観測値のマップ**

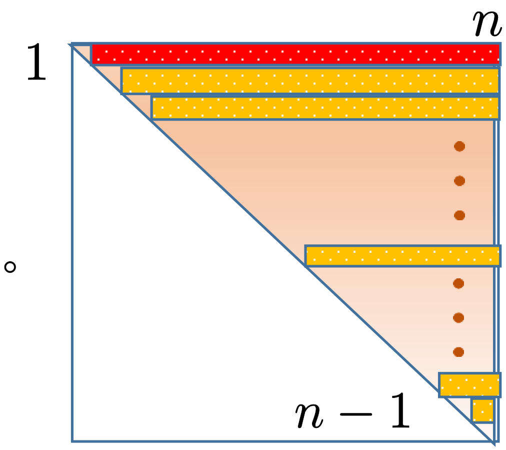
マップを用いた最小化問題

- 一行目の対応の真値からマップを作成できる。

$$\boldsymbol{\xi}_{ij} = f(\boldsymbol{\xi}_{1i}, \boldsymbol{\xi}_{1j})$$

真値：赤を決めると下のオレンジの領域全てを埋められる。

すなわち，真値のマップを作成可能。



定義：

誤差の二乗和 ＝ (真値のマップ-観測値のマップ)^2 の最小化！

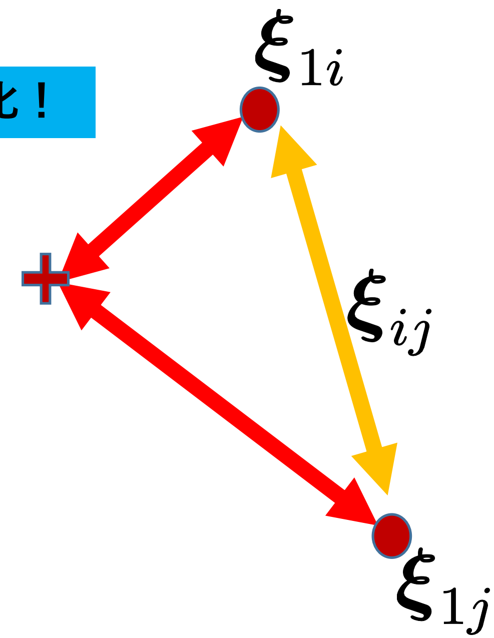
真値　観測値

$$\arg\min \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(\boldsymbol{\xi}_{ij} - \hat{\boldsymbol{\xi}}_{ij} \right)^2$$

更に，観測値の信頼性を示す重み係数をつける

重み　真値　観測値

$$\arg\min \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \omega_{ij} \left(f(\boldsymbol{\xi}_{1i}, \boldsymbol{\xi}_{1j}) - \hat{\boldsymbol{\xi}}_{ij} \right)^2$$



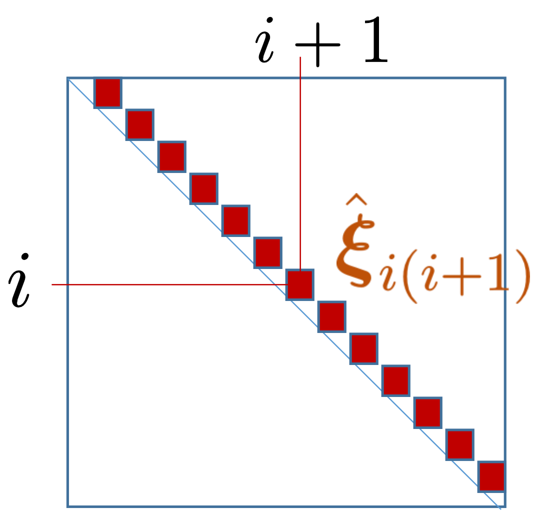
提案法と従来法の定義

- **目的**：観測値のマップが与えられた時に，最適な真値を求めること。

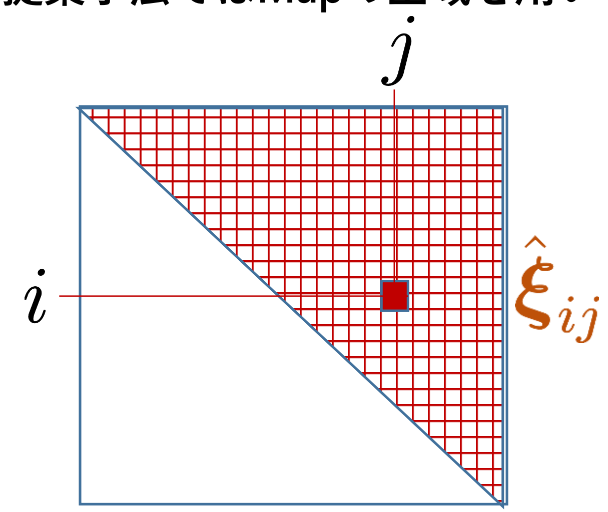
$$\arg\min \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \omega_{ij} \left(f(\boldsymbol{\xi}_{1i}, \boldsymbol{\xi}_{1j}) - \hat{\boldsymbol{\xi}}_{ij} \right)^2$$

- **従来**：最も簡単には，隣り合うフレーム同士の対応だけで行ってきた。
  すなわち観測マップのごく一部の情報のみを用いて推定をしていた。

従来法と定義（仮）

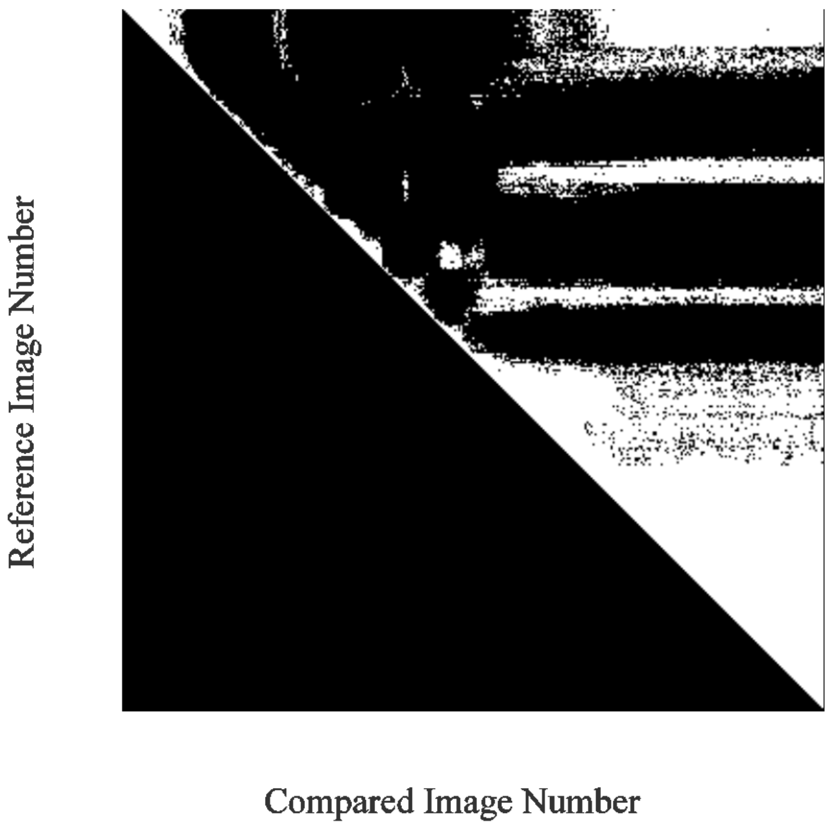


提案手法ではMapの全域を用いる

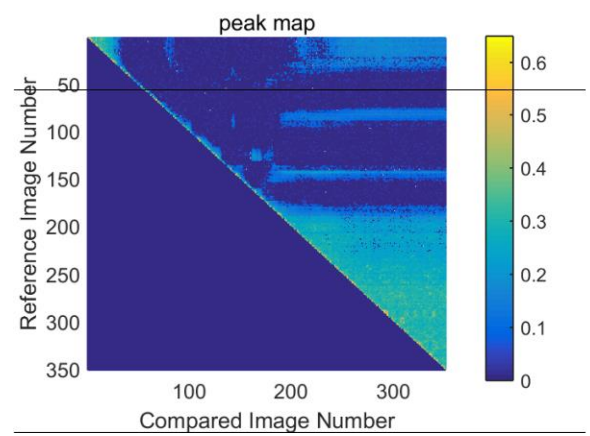


具体的な値はどうなっているのか

- 実際にビデオから作成した重み行列を下に示す。（351*351)
- 白が対応付けのある領域 ω=1，黒が対応付の取れなかった領域ω=0
- 白いところの情報を全て用いて推定する。



もともとは位相相関のピーク値の行列からしきい値を設定して作る。



<１>回転量の最適化

- 後は偏微分から極小値を探していく。
- 画像の変形パラメータは 回転，拡大縮小，平行移動の４自由度あるがそれぞれ**独立な変数なので別々に最適化する。**

- **<1> 回転量の最適化**
- 回転量において最適化する関数Jは次の通り。

$$J(\theta_{12}, ..., \theta_{1n}) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \omega_{ij} \left(\theta_{ij} - \hat{\theta}_{ij}\right)^2$$

回転量は足し引きで計算できるため：

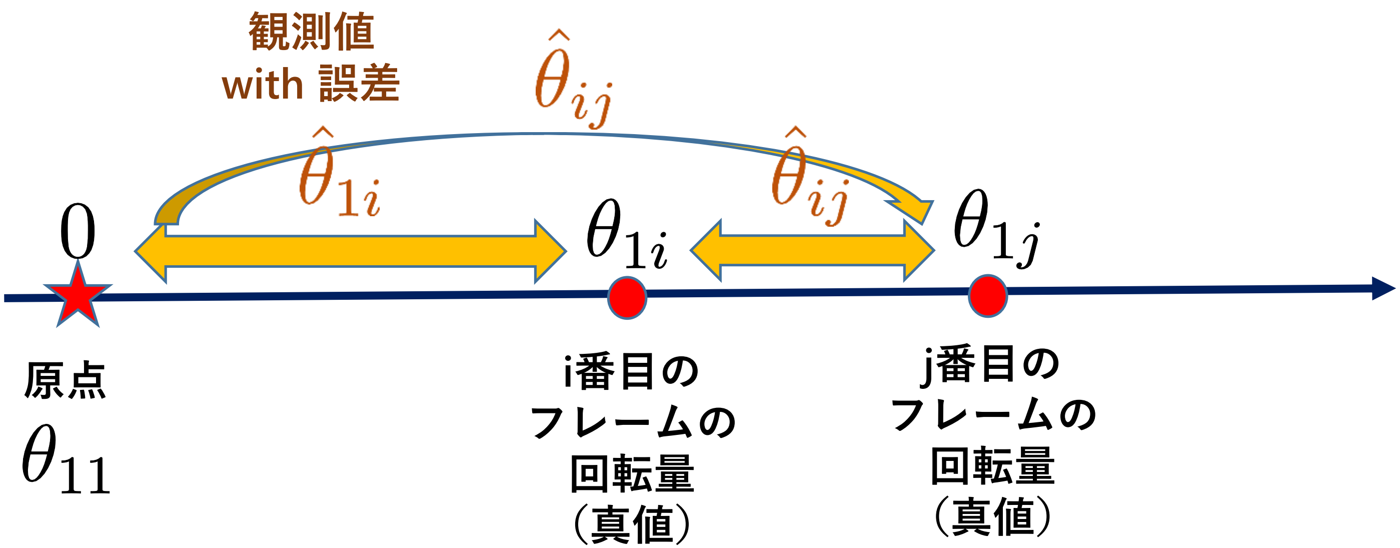
$$\theta_{ij} = \theta_{1j} - \theta_{1i}$$

最小化する関数

$$J(\theta_{12}, ..., \theta_{1n}) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \omega_{ij} \left(\theta_{1j} - \theta_{1i} - \hat{\theta}_{ij}\right)^2$$

回転量の最適化：図解

- １次元なので線分上で表せる。



- 線分上の座標と観測した点間の距離の差を最小化する。
- **赤い点（真値）を仮定して関係量を計算したマップと実際に計測した黄色い関係式のマップとの差が最小になるようにする！**

多変数関数の極小値の条件

- 最小化する二乗和は真値のベクトル $\theta_{1x}(x = 2, 3, ..., n)$ の多変数関数。
- 極小値の条件：① **偏微分に代入した時0**
  ② **ヘッセ行列が正定値行列**

Theorem 1 多変数関数 $f(x_1, x_2, ..., x_n)$ において点 $(a_1, a_2, ..., a_n)$ が $f_{x_1}(a_1, ..., a_n) = 0, ..., f_{x_n}(a_1, ..., a_n) = 0$ をみたすとき，$f(a_1, ..., a_n)$ が極値となるための判定条件は次の通りである．
但し，ヘッセ行列として ij 成分が $h_{ij} = \dfrac{\partial^2 f}{\partial x_i \partial x_j}$ となる行列 $\boldsymbol{H}$ を定義する。

(i). ヘッセ行列 $\boldsymbol{H}$ が正定値行列の時，極小値をとる．

(ii). ヘッセ行列 $\boldsymbol{H}$ が負定値行列の時，極大値をとる．

② ヘッセ行列の導出

- 一階と二階の偏微分は紙面を参照ください。
- ヘッセ行列は以下のようになる。

$$\boldsymbol{H} = 2\begin{pmatrix} \omega_{12} + \sum_{j=3}^{n} \omega_{2j} & \cdots & -\omega_{2x} & \cdots & -\omega_{2n} \\ \vdots & \ddots & \vdots & & \vdots \\ -\omega_{2x} & \cdots & \sum_{i=1}^{x-1} \omega_{ix} + \sum_{j=x+1}^{n} \omega_{xj} & \cdots & -\omega_{xn} \\ \vdots & & \vdots & \ddots & \vdots \\ -\omega_{2n} & \cdots & -\omega_{xn} & \cdots & \sum_{i=1}^{n-1} \omega_{in} \end{pmatrix}$$

- **ポイント： ヘッセ行列の値に影響を与えるのは重み付けの値のみ！**
- **最適化問題が一意に求まるかどうかは，重み付けのとり方に依る。**

- **ヘッセ行列が正定値 = 観測の結果「他のフレームと対応付けられない画像が無い」だと思うのですが…証明の手段を持たず…**

① 平衡点の条件

- 偏微分からの条件式から真値ベクトルに対する線形方程式を作ることが出来る。
- 偏微分=0 ：

$$\frac{\partial J}{\partial \theta_{1x}} = 2\left\{\left(\sum_{i=1}^{x-1}\omega_{ix} + \sum_{j=x+1}^{n}\omega_{xj}\right)\theta_{1x} - \sum_{i=2}^{x-1}\omega_{ix}\theta_{1i} - \sum_{j=x+1}^{n}\omega_{xj}\theta_{1j} - \left(\sum_{i=1}^{x-1}\omega_{ix}\hat{\theta}_{ix} - \sum_{j=x+1}^{n}\omega_{xj}\hat{\theta}_{xj}\right)\right\} \quad (x \neq 2 \cap x \neq n)$$

- 上の式の係数を集めて線形方程式 $\boldsymbol{A\theta} = \boldsymbol{b}$ を得る。

bは当然観測値の値（略します）

真値のベクトル

$$\boldsymbol{\theta} = \begin{pmatrix} \theta_{12} \\ \vdots \\ \theta_{1x} \\ \vdots \\ \theta_{1n} \end{pmatrix}$$

係数行列 ： ヘッセ行列の1/2！

→ **ヘッセ行列が正定値なら逆行列がある。**
最小値があるなら必ず求まる。ということ。

$$\boldsymbol{A} = \begin{pmatrix} \omega_{12} + \sum_{j=3}^{n}\omega_{2j} & \cdots & -\omega_{2x} & \cdots & -\omega_{2n} \\ \vdots & \ddots & \vdots & & \vdots \\ -\omega_{2x} & \cdots & \sum_{i=1}^{x-1}\omega_{ix} + \sum_{j=x+1}^{n}\omega_{xj} & \cdots & -\omega_{xn} \\ \vdots & & \vdots & \ddots & \vdots \\ -\omega_{2n} & \cdots & -\omega_{xn} & \cdots & \sum_{i=1}^{n-1}\omega_{in} \end{pmatrix}$$

まとめ

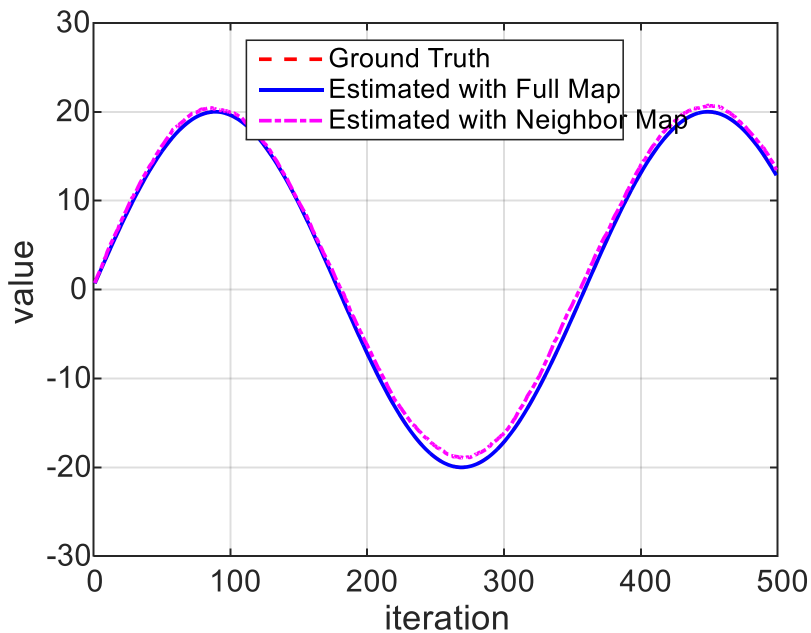
- 真値を仮定して，そこから作成した関係のマップと
  実際に観測した関係値のマップとの誤差の二乗和を最小化。
- 重み行列が解を握る。= 実際ほとんどの場合において解はある。

- 次章では,
- 実際取得したデータに対して試す前に，シミュレーションにて妥当性を確認。

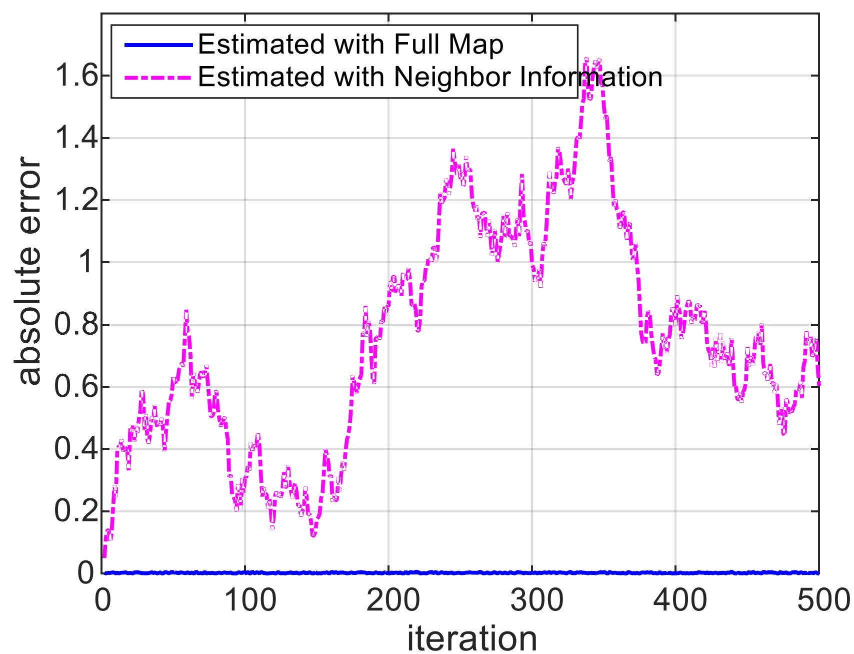
シミュレーション・角度の推定

- 条件：真値を用意し，ノイズを載せた観測データのマップを作成，
  従来法：隣り合ったフレームの情報のみ
  提案法：全ての情報を利用
- 角度の推定の誤差が減ったことを確認。

真値と推定値



推定値の誤差



なお，ベクトルの最適化なので横軸は要素番号であり時間ではありません。

＜2＞ 拡大縮小量の最適化

- 拡大縮小量の最適化関数もほとんど同様

$$\kappa_{ij} = \frac{\kappa_{1j}}{\kappa_{1i}}$$

フレーム間対応： 倍率なので積商の関係 → 非線形

対数をとることで，和差の関係に変更

$$\log \kappa_{ij} = \log \kappa_{1j} - \log \kappa_{1i}$$

- 変数を置き換えることで同様の式になる $\log \kappa_{ij} = \mu_{ij}$

$$J_{log} = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \omega_{ij} \left(\log \kappa_{ij} - \log \hat{\kappa}_{ij}\right)^2$$

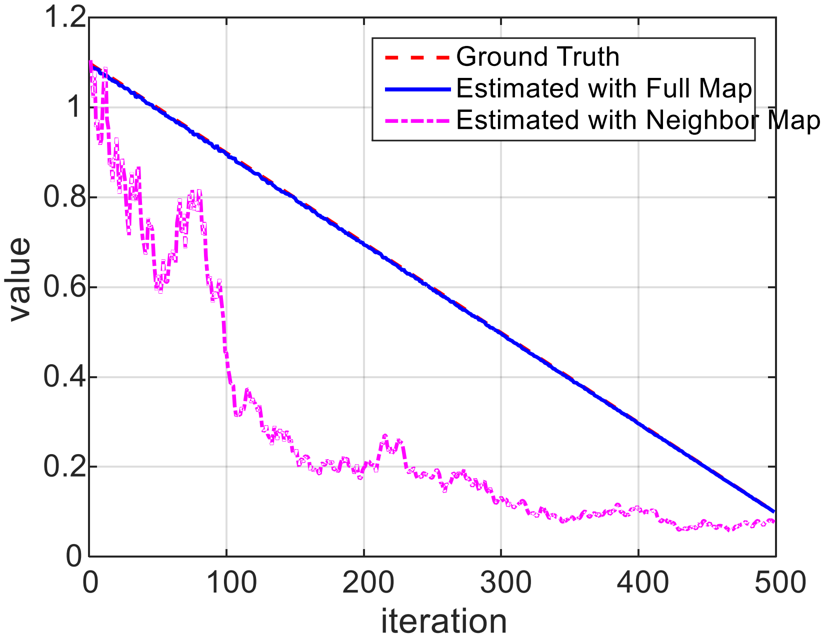
回転量の時と全く同じ手順！（よって省略）

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \omega_{ij} \left(\mu_{1j} - \mu_{1i} - \hat{\mu}_{ij}\right)^2$$

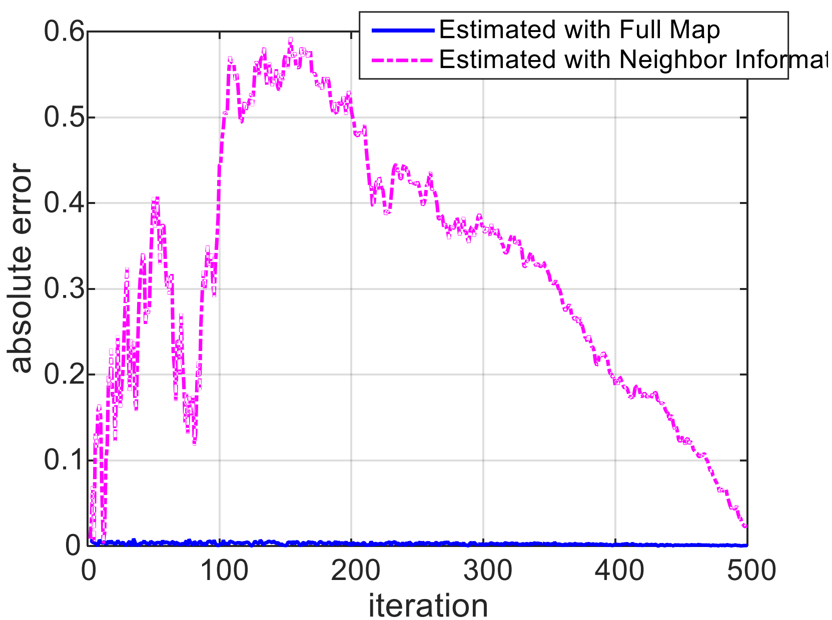
シミュレーション・拡大縮小量の検出

- 誤差モデルは同様に，拡大縮小量の式もシミュレーションにて確認を行った
- **従来法**：隣り合ったフレームの情報のみ
  提案法：全ての情報を利用

真値と推定値



推定値の誤差



＜３＞平行移動量の最適化

- そのままだと対応付けが複雑！（干渉等）

$$\begin{pmatrix} \xi_{x_{ij}} \\ \xi_{y_{ij}} \end{pmatrix} = \frac{1}{\kappa_{1i}} \begin{pmatrix} \cos\theta_{1i} & -\sin\theta_{1i} \\ \sin\theta_{1i} & \cos\theta_{1i} \end{pmatrix} \left(\begin{pmatrix} \xi_{x_{1j}} \\ \xi_{y_{1j}} \end{pmatrix} - \begin{pmatrix} \xi_{x_{1i}} \\ \xi_{y_{1i}} \end{pmatrix} \right)$$

- 適切な座標変換を用いる

$$\begin{pmatrix} x_{ij} \\ y_{ij} \end{pmatrix} = \kappa_{1i} \begin{pmatrix} \cos\theta_{1i} & \sin\theta_{1i} \\ -\sin\theta_{1i} & \cos\theta_{1i} \end{pmatrix} \begin{pmatrix} \xi_{x_{ij}} \\ \xi_{y_{ij}} \end{pmatrix}$$

- 結果，回転量の時と同様の式が得られる。以降の手順略

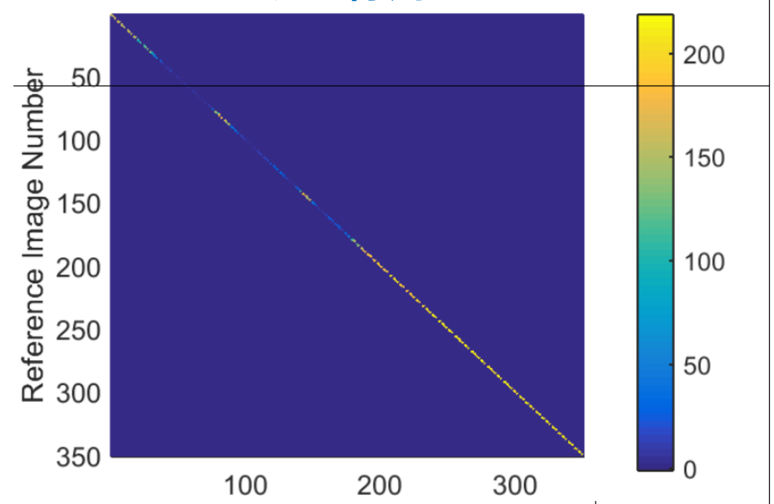
$$\begin{pmatrix} x_{ij} \\ y_{ij} \end{pmatrix} = \begin{pmatrix} x_{1j} \\ y_{1j} \end{pmatrix} - \begin{pmatrix} x_{1i} \\ y_{1i} \end{pmatrix}$$

座標変換で，和差の関係に変更

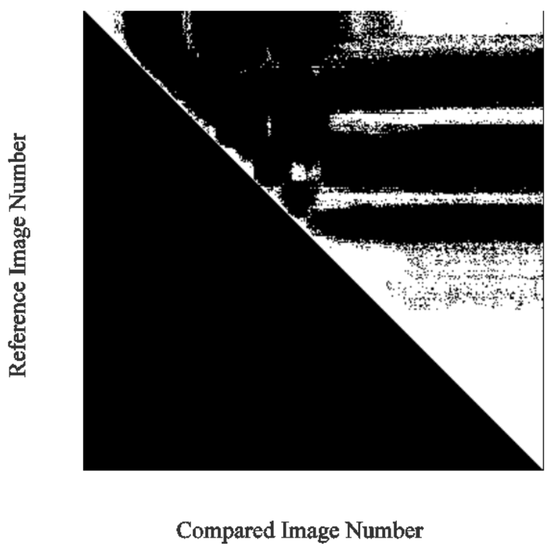
実験データの最適化

- まずはヘッセ行列から極小値のある条件を出す。
- 固有値が全て正より正定値行列。

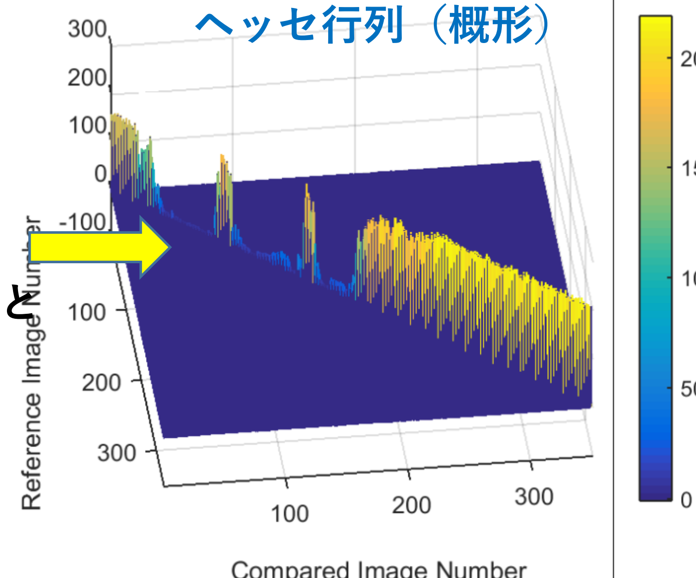
ヘッセ行列



重み行列 ω



ヘッセ行列（概形）

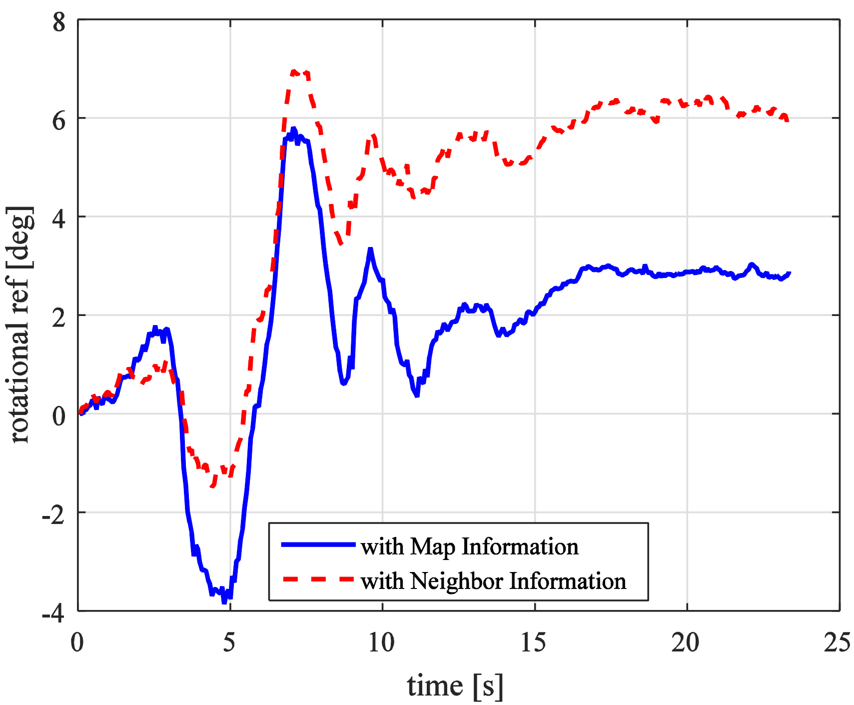


対角の値が小さい
部分は他のフレームと
の対応が少なかった
と見ることが出来る

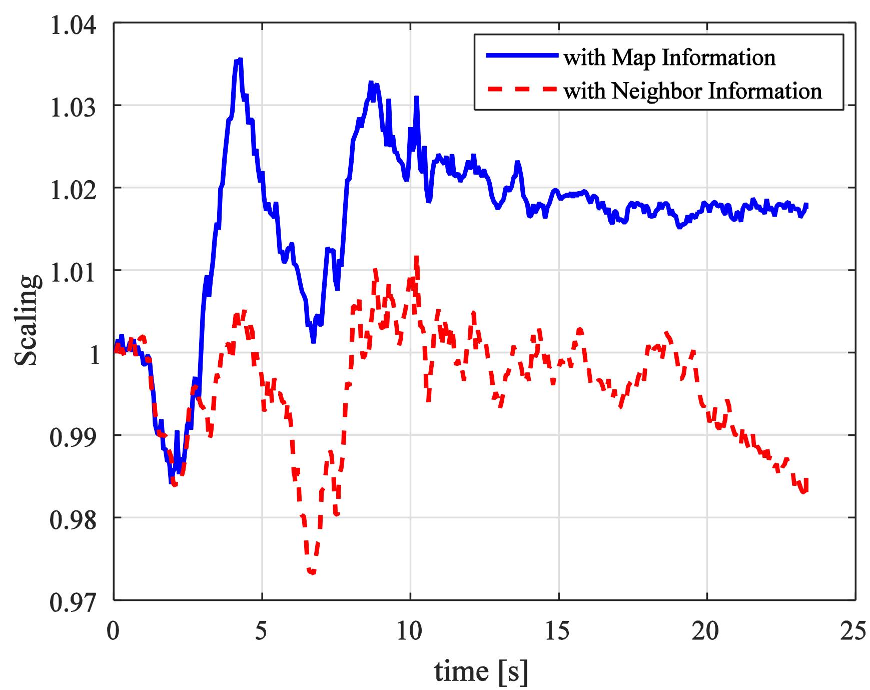
実験データの最適化 ①回転・拡大縮小量

- 青：提案法
  赤：従来法
- 比較的大きな差異が見られる。

回転量を最適化した結果



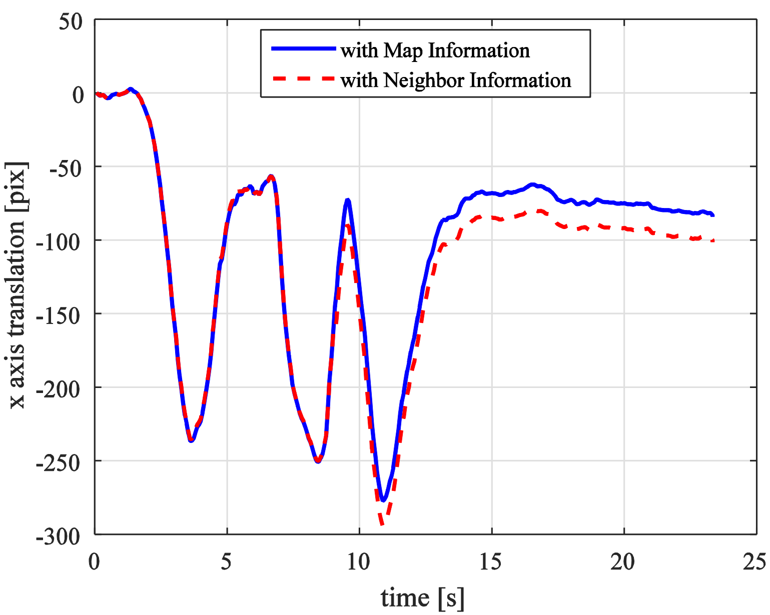
拡大縮小量を最適化した結果



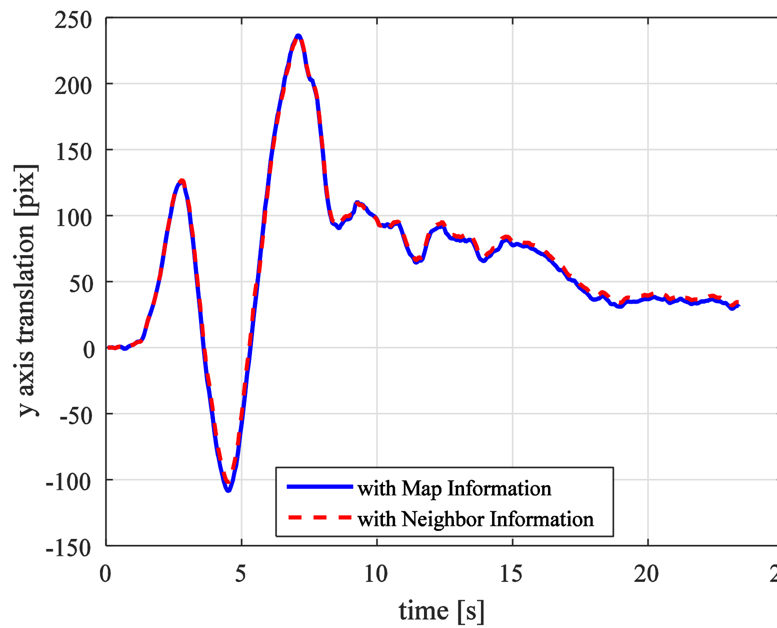
実験データの最適化 ②平行移動量

- 青：提案法
  赤：従来法
- 主に回転量の変化によりx軸に変位が確認される。

X方向を最適化した結果



Y方向を最適化した結果

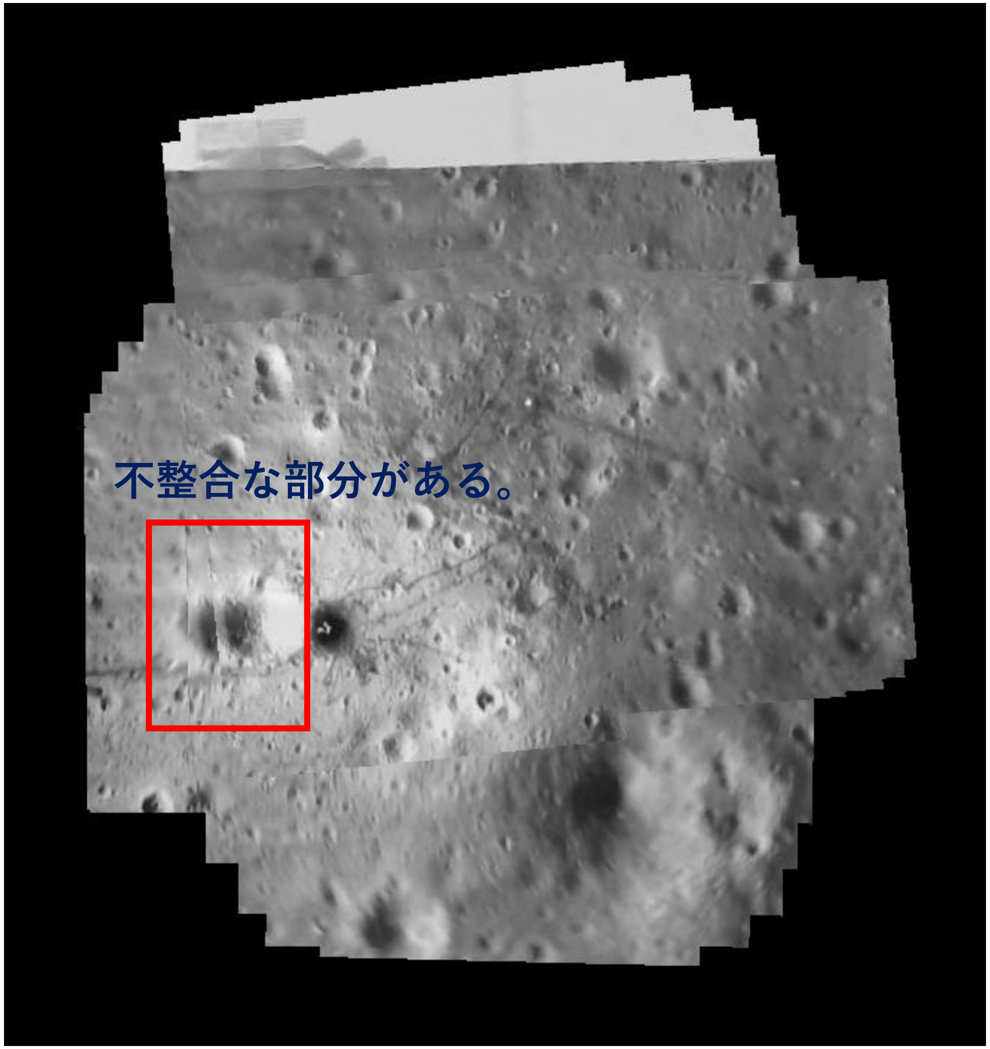


結果の比較：従来法

- 得られた指令値を元に画像を
重ねる事ができる。
（パノラマ画像合成の原理）
- 従来の繋げ方では徐々に
検出のズレが積算していってしまい
同じ箇所が同様に重ならない。

- 誤差が蓄積していることがわかる。
（動画の方がよくわかります）

MappingImage with117Images

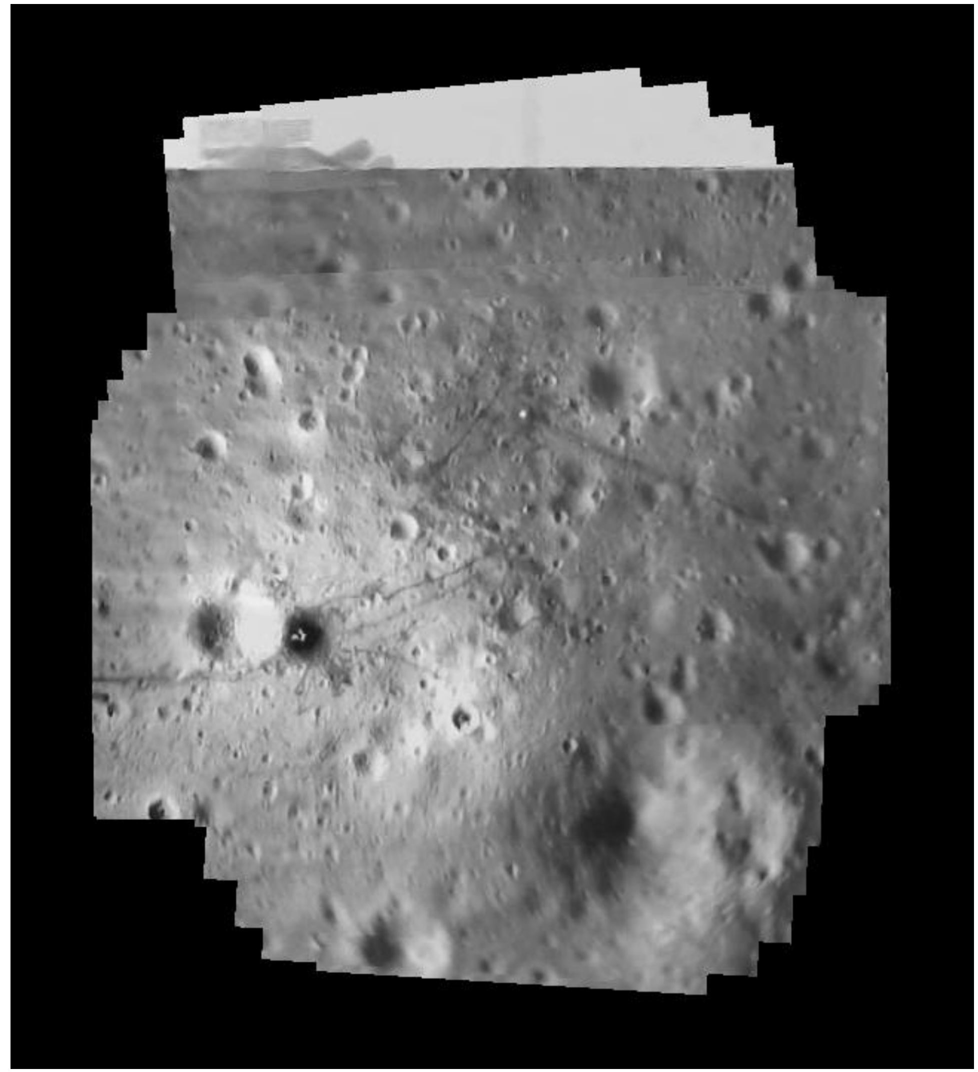


結果の比較：提案法

- ブレはあるが画像面が滑らかにつながるようになった。

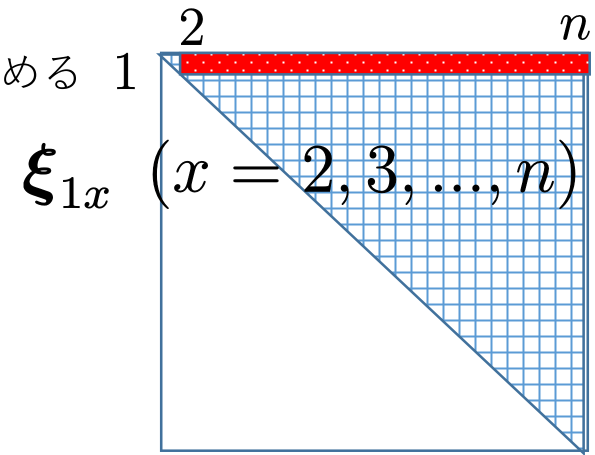
- 詳細は動画にて。

MappingImage with117Images



補足：マップを用いた最適化

目的： １番目のフレームとの全てのフレームとの対応を求める
（マップで言う所の１行目）

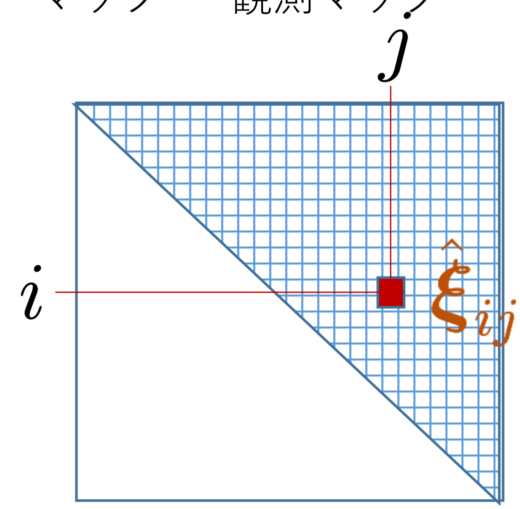


手段： 予め計測を繰り返して作ったマップを用いて
尤もらしい対応を求める

→「尤もらしい」をきちんと定義する

誤差の二乗和 ＝ (真値-計測値)^2 の最小化

画像処理で対応分を埋めた
マップ ＝ 観測マップ



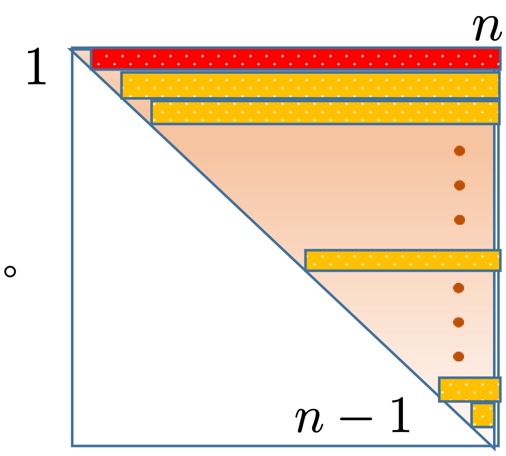
真値： $\boldsymbol{\xi}_{1x} \ (x = 2, 3, ..., n)$

真値からマップを作る $\boldsymbol{\xi}_{ij} = f(\boldsymbol{\xi}_{1i}, \boldsymbol{\xi}_{1j})$

観測値： $\hat{\boldsymbol{\xi}}_{ij}$ ＝ 観測値のマップ

マップを用いた最適化

- 一行目の対応の真値からマップを作成できる。



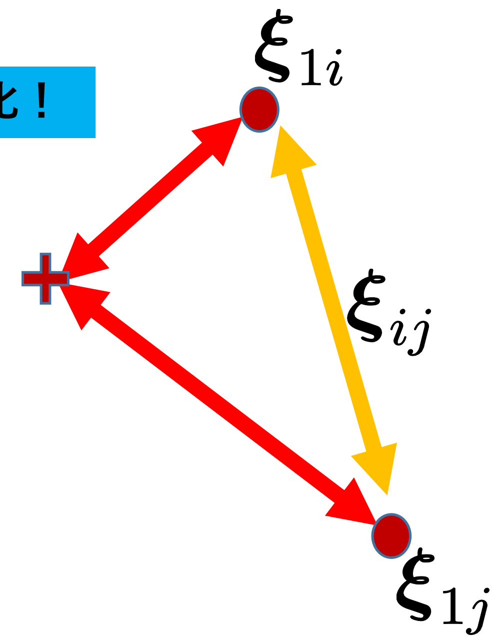
$$\boldsymbol{\xi}_{ij} = f(\boldsymbol{\xi}_{1i}, \boldsymbol{\xi}_{1j})$$

真値：赤を決めると下のオレンジの領域全てを埋められる。

すなわち，真値のマップを作成可能。

定義：

誤差の二乗和 ＝ (真値のマップ-観測値のマップ)^2 の最小化！



真値　観測値

$$\arg\min \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(\boldsymbol{\xi}_{ij} - \hat{\boldsymbol{\xi}}_{ij} \right)^2$$

真値　観測値

$$\arg\min \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left(f(\boldsymbol{\xi}_{1i}, \boldsymbol{\xi}_{1j}) - \hat{\boldsymbol{\xi}}_{ij} \right)^2$$