

# 1 幾何補正

## 1.1 地上基準点と画像基準点

リモートセンシングによって得られるデータは、基本的に画像座標であり、それを地上座標に変換する必要がある。これを**幾何補正** (geometric correction) と呼んでいる。実際に画像データに座標変換を適用したい場合、変換式における変換係数を求める必要がある。そのためには基準点データが必要になる。**地上基準点** (ground control point) とは、あらかじめ地上での座標の値が解っている点  $(x_i, y_i)$  のことを言う。それらの点が画像上に投影されている場合、対応する画像座標  $(u_i, v_i)$  が存在し、これを**画像基準点** (image control point) という。これら、地上座標と画像座標を一組とするデータセットを基準点データという。これらの基準点データを変換式に代入すると、基準点の数だけ変換係数を変数とする方程式が立てられる。それらの方程式を連立させれば、変換係数が求められる。

例えば、アフィン変換の変換式を導く場合、一つの変換式において3つの変換係数があるため、少なくとも3つの基準点データが必要になる。3つを越える数の基準点を利用して変換係数を導くには、最小二乗法を利用して解く。

誤差の調整という観点で考えると、基準点の数は多い方が良い。また、基準点の配置は重要で、狭い範囲に偏った基準点の場合、基準点から離れた場所においては誤差が大きくなる。また、一直線上に基準点が配置される場合も変換式に含まれる誤差が大きくなる。したがって、対象範囲において広い範囲をまんべんなくカバーするような基準点データが必要である。

基準点の精度が、座標変換の精度に直接関わるので、正確な基準点データが必要である。画像基準点として、1画素程度の精度では十分ではなく、1/5画素以上の精度が望ましい。対応する地上基準点も同様で、1画素あたり何mの画像かによるが、1画素あたり10mの衛星画像の場合、やはりその1/5の2mの精度での地上基準点が必要である。地上基準点は、測量等により正確に測ることは比較的容易である。しかし、対応する画像座標は、測量した地点を画像において1/5画素の精度で特定することは困難である。したがって、画像マッチング等の画像処理手法を用いて画像座標を求めることが望まれる。

## 1.2 最小二乗法による線形変換

$n$  個の地上基準点のデータ  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  とそれに対応する画像基準点のデータ  $(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)$  があるとき、最小二乗法を用いて変換式を導く手法について解説する。

### 1.2.1 アフィン変換

アフィン変換 (式??) は、 $x$  と  $y$  に関する変換係数が独立なので、 $x$  と  $y$  は別々に解くことができる。ここでは  $x$  に関する式を例に挙げる。まず、 $i$  番目の基準点のデータをアフィン変換の式に代入すると、 $x_i = au_i + bv_i + c$  となるが、導かれる変換式には誤差が存在するために、この式は成立しない。その誤差量  $\delta_i$  は、単純に右辺から左辺を引いたものとなり、 $\delta_i = au_i + bv_i + c - x_i$  といえる。

ここでは  $n$  個の基準点データがあるので、 $n$  個の誤差量に関する式を立てることが出来る。最小二乗法は、それらの誤差量の二乗和を最小にする式を導くものなので、二乗和の誤差関数  $\Phi$  は、以下の式で表すことが出来る。

$$\begin{aligned}\Phi &= \sum_{i=1}^n \delta_i^2 \\ &= \sum_{i=1}^n (au_i + bv_i + c - x_i)^2\end{aligned}\quad (1)$$

この関数  $\Phi$  は二次関数なので、この関数が最小となる  $a, b, c$  を求めるには、まずこの式を  $a, b, c$  で偏微分する。偏微分すると、誤差関数の勾配を表す関数となるので、偏微分した関数が 0 となるところが極小値であり、それは最小値でもある。下の式は、偏微分した式である。

$$\frac{\partial \Phi}{\partial a} = \sum_{i=1}^n 2u_i(au_i + bv_i + c - x_i) = 0 \quad (2)$$

$$\frac{\partial \Phi}{\partial b} = \sum_{i=1}^n 2v_i(au_i + bv_i + c - x_i) = 0 \quad (3)$$

$$\frac{\partial \Phi}{\partial c} = \sum_{i=1}^n 2(au_i + bv_i + c - x_i) = 0 \quad (4)$$

これらの式は、連立一次方程式なので、これらを行列を用いて整理すると、以下の式を得る。なお総和記号  $\sum$  は、簡単に表記するためガウスの総和記号  $\llbracket$  で代用した。

$$\begin{pmatrix} \llbracket u_i^2 \rrbracket & \llbracket u_i v_i \rrbracket & \llbracket u_i \rrbracket \\ \llbracket u_i v_i \rrbracket & \llbracket v_i^2 \rrbracket & \llbracket v_i \rrbracket \\ \llbracket u_i \rrbracket & \llbracket v_i \rrbracket & n \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} \llbracket x_i u_i \rrbracket \\ \llbracket x_i v_i \rrbracket \\ \llbracket x_i \rrbracket \end{pmatrix} \quad (5)$$

この連立方程式を解けば、最確値  $a, b, c$  が求まる。 $y$  についても同様に解けば良い。

### 1.2.2 ヘルマート変換

ヘルマート変換 (式??) においては、二つの式を別々に解いてしまうと、 $a, b$  の値が別々に求まり、同じ値を示さない。これではアフィン変換と変わらなくなってしまう。そこで、このような場合には  $x$  と  $y$  のそれぞれの二乗和の誤差関数の和を最小とするように解かなければならない。したがって、二乗和の誤差関数  $\Phi$  は、以下の式で表すことが出来る。

$$\Phi = \sum_{i=1}^n \{(au_i - bv_i + c - x_i)^2 + (bu_i + av_i + d - y_i)^2\} \quad (6)$$

この式を  $a, b, c, d$  で偏微分し、それが 0 となる方程式を立てる.

$$\frac{\partial \Phi}{\partial a} = \sum_{i=1}^n 2\{u_i(au_i - bv_i + c - x_i) + v_i(bu_i + av_i + d - y_i)\} = 0 \quad (7)$$

$$\frac{\partial \Phi}{\partial b} = \sum_{i=1}^n 2\{-v_i(au_i - bv_i + c - x_i) + u_i(bu_i + av_i + d - y_i)\} = 0 \quad (8)$$

$$\frac{\partial \Phi}{\partial c} = \sum_{i=1}^n 2(au_i - bv_i + c - x_i) = 0 \quad (9)$$

$$\frac{\partial \Phi}{\partial d} = \sum_{i=1}^n 2(bu_i + av_i + d - y_i) = 0 \quad (10)$$

アフィン変換の場合と同様に、これらを行列を用いて整理すると、以下の式を得る.

$$\begin{pmatrix} [u_i^2 + v_i^2] & 0 & [u_i] & [v_i] \\ 0 & [u_i^2 + v_i^2] & -[v_i] & [u_i] \\ [u_i] & -[v_i] & n & 0 \\ [v_i] & [u_i] & 0 & n \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} [x_i u_i + y_i v_i] \\ [-x_i v_i + y_i u_i] \\ [x_i] \\ [y_i] \end{pmatrix} \quad (11)$$

この連立方程式を解けば、最確値  $a, b, c, d$  が求まる.

この他、射影変換においては分数関数であるが、分母の式を両辺にかけて分母をはらい、線形化した後に誤差関数を立てれば、同様に解くことができる.

### 1.3 最小二乗法による非線形変換

#### 1.3.1 二次元回転変換

画像の座標ではなく、測量した値を座標変換する場合は、縮尺を考慮する必要がないので、回転と原点移動だけの変換で十分である. これを式で表すと以下のようになる.

$$\begin{cases} x = u \cos \kappa - v \sin \kappa + x_0 \\ y = u \sin \kappa + v \cos \kappa + y_0 \end{cases} \quad (12)$$

したがって、未知数の個数は 3 個となり、最低 2 個の基準点データで変換式を決定できることになる. しかし、アフィン変換やヘルマート変換が線形の式であったのに対し、座標回転を導入すると三角関数が含まれる非線形の式となる、基準点 2 個のデータより回転角を求めるのは簡単であるが、2 個を越える基準点データから最小二乗法を用いて回転角を求めるには、非線形方程式なので通常の連立方程式で解くことは出来ない. したがって、テイラー展開を使って逐次計算をすることにより近似値を求めることになる.

そのためにはまず、変換式を次のように関数として置き換える.

$$F_x(x_0, \kappa) = u \cos \kappa - v \sin \kappa + x_0 - x \quad (13)$$

$$F_y(y_0, \kappa) = u \sin \kappa + v \cos \kappa + y_0 - y \quad (14)$$

各未知係数の近似値を  $x_{00}, y_{00}, \kappa_0$  とし、各補正量を  $\Delta x_0, \Delta y_0, \Delta \kappa$  とすると、未知係数は、以下の式で表すことが出来る。

$$x_0 = x_{00} - \Delta x_0 \quad (15)$$

$$y_0 = y_{00} - \Delta y_0 \quad (16)$$

$$\kappa = \kappa_0 - \Delta \kappa \quad (17)$$

関数  $F_x, F_y$  において、近似値の周りにテイラー展開する。つまりテイラー級数（式??）において、 $x = x_0, a = x_{00}$  とおく。すると、 $x_0 - x_{00} = -\Delta x_0$  となる。他の変数  $y_0, \kappa$  についても同様に考え、多変数の関数なので偏微分を用いると以下の式が得られる。なお、級数における二階導関数の項以降は無視できるものとしている。

$$F_x(x_0, \kappa) \approx F_x(x_{00}, \kappa_0) - \frac{\partial F_x}{\partial x_0} \Delta x_0 - \frac{\partial F_x}{\partial \kappa} \Delta \kappa \quad (18)$$

$$F_y(y_0, \kappa) \approx F_y(y_{00}, \kappa_0) - \frac{\partial F_y}{\partial y_0} \Delta y_0 - \frac{\partial F_y}{\partial \kappa} \Delta \kappa \quad (19)$$

テイラー展開による微係数は、以下のように計算される。

$$\frac{\partial F_x}{\partial x_0} = 1 \quad (20)$$

$$\frac{\partial F_y}{\partial y_0} = 1 \quad (21)$$

$$\frac{\partial F_x}{\partial \kappa} = -u \sin \kappa - v \cos \kappa \quad (22)$$

$$\frac{\partial F_y}{\partial \kappa} = u \cos \kappa - v \sin \kappa \quad (23)$$

したがって、

$$F_x(x_0, \kappa) \approx F_x(x_{00}, \kappa_0) - \Delta x_0 + (u \sin \kappa + v \cos \kappa) \Delta \kappa \quad (24)$$

$$F_y(y_0, \kappa) \approx F_y(y_{00}, \kappa_0) - \Delta y_0 - (u \cos \kappa - v \sin \kappa) \Delta \kappa \quad (25)$$

さて、 $n$  個の基準点データ  $(u_1, v_1, x_1, y_1), \dots, (u_n, v_n, x_n, y_n)$  を用いて、最小二乗法と逐次計算により  $x_0, y_0, \kappa$  を求める手順について解説する。

まず、 $x_{00}, y_{00}, \kappa_0$  の初期値と基準点データを式 22, 23 に代入する。なお微係数の式にも  $x_{00}, y_{00}, \kappa_0$  の初期値を代入する。すると、 $F_x, F_y$  それぞれについて、 $n$  個の式が出来る。基準点に誤差がなければ、 $F_x = 0, F_y = 0$  となるが、誤差は必ず存在する。そして誤差の値は、計算された  $F_x, F_y$  の値そのものとなる。この誤差の二乗和が最小となるような  $\Delta x_0, \Delta y_0, \Delta \kappa$  を求めなければならない。ここで、この誤差関数  $E$  は、以下の式となる。

$$E = \sum_{i=1}^n \{F_x(x_{00}, \kappa_0) - \Delta x_0 + (u_i \sin \kappa_0 + v_i \cos \kappa_0) \Delta \kappa\}^2 + \sum_{i=1}^n \{F_y(y_{00}, \kappa_0) - \Delta y_0 - (u_i \cos \kappa_0 - v_i \sin \kappa_0) \Delta \kappa\}^2 \quad (26)$$

したがって、この誤差関数を  $\Delta x_0, \Delta y_0, \Delta \kappa$  で偏微分し、偏微分した関数の値が 0 となる  $\Delta x_0, \Delta y_0, \Delta \kappa$  を連立方程式を解くことによって求めれば良い。

$$\frac{\partial E}{\partial \Delta x_0} = -2 \sum_{i=1}^n \{F_x(x_{00}, \kappa_0) - \Delta x_0 + (u_i \sin \kappa_0 + v_i \cos \kappa_0) \Delta \kappa\} = 0 \quad (27)$$

$$\frac{\partial E}{\partial \Delta y_0} = -2 \sum_{i=1}^n \{F_y(y_{00}, \kappa_0) - \Delta y_0 - (u_i \cos \kappa_0 - v_i \sin \kappa_0) \Delta \kappa\} = 0 \quad (28)$$

$$\begin{aligned} \frac{\partial E}{\partial \Delta \kappa} = & -2 \sum_{i=1}^n \{(u_i \cos \kappa_0 - v_i \sin \kappa_0)(F_x(x_{00}, \kappa_0) - \Delta x_0 + (u_i \sin \kappa_0 + v_i \cos \kappa_0) \Delta \kappa)\} \\ & - 2 \sum_{i=1}^n \{(u_i \sin \kappa_0 + v_i \cos \kappa_0)(F_y(y_{00}, \kappa_0) - \Delta y_0 - (u_i \cos \kappa_0 - v_i \sin \kappa_0) \Delta \kappa)\} = 0 \end{aligned} \quad (29)$$

ここで、問題となるのが初期値の値である。  $x_{00}, y_{00}$  については、最初の基準点座標の差を利用することが出来る。また  $\kappa$  は、二つの基準点座標から、その線分の傾きの差を利用する。

$$x_{00} = x_1 - u_1 \quad (30)$$

$$y_{00} = y_1 - v_1 \quad (31)$$

$$\kappa_0 = \tan^{-1} \frac{y_2 - y_1}{x_2 - x_1} - \tan^{-1} \frac{v_2 - v_1}{u_2 - u_1} \quad (32)$$

初期値を代入して、最小二乗法により求めた補正量  $\Delta x_0, \Delta y_0, \Delta \kappa$  が求めれば、初期値から補正量を差し引くことで、1 回目の近似値が求まる。さらにこの近似値を用いて 2 回目の補正量を求め、さらに近似値を求めて行く。この繰り返し計算を求める解の精度を満足するまで行う。

### 1.3.2 三次元回転変換

三次元データ  $(u, v, w)$  を  $(x, y, z)$  に変換するには、以下のように三次元アフィン変換のような線形変換でも表現できる。

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} \quad (33)$$

この変換式には、 $a_{11} \sim a_{33}$  と  $(x_0, y_0, z_0)$  の 12 個の未知数が存在する。これらの未知数を導くためには、式が 3 つなので、最低 4 個の基準点データが必要となる。しかし、 $(u, v, w)$  と  $(x, y, z)$  が共に同じ単位の地上座標で与えられている場合、つまり縮尺を考慮しなくて良い場合、 $a_{11} \sim a_{33}$  の行列は、座標回転によって得られる。 $(u, v, w)$  と  $(x, y, z)$  は、それぞれレーザースキャナや写真測量において得られたが、それぞれローカルな座標系であった場合を想定する。この時、 $x$  軸回りの回転を  $\omega$ ,

$y$  軸回りの回転を  $\varphi$ ,  $z$  軸回りの回転を  $\kappa$  とすると, 以下の式で回転行列を表すことができる.

$$\begin{aligned} \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \omega & -\sin \omega \\ 0 & \sin \omega & \cos \omega \end{pmatrix} \begin{pmatrix} \cos \varphi & 0 & \sin \varphi \\ 0 & 1 & 0 \\ -\sin \varphi & 0 & \cos \varphi \end{pmatrix} \begin{pmatrix} \cos \kappa & -\sin \kappa & 0 \\ \sin \kappa & \cos \kappa & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \cos \varphi \cos \kappa & -\cos \varphi \sin \kappa & \sin \varphi \\ \cos \omega \sin \kappa + \sin \omega \sin \varphi \cos \kappa & \cos \omega \cos \kappa - \sin \omega \sin \varphi \sin \kappa & -\sin \omega \cos \varphi \\ \sin \omega \sin \kappa - \cos \omega \sin \varphi \cos \kappa & \sin \omega \cos \kappa + \cos \omega \sin \varphi \sin \kappa & \cos \omega \cos \varphi \end{pmatrix} \end{aligned} \quad (34)$$

したがって, 未知数の個数は 6 個となり, 最低 2 個の基準点データで変換式を決定できることになる. しかし, 三次元アフィン変換が線形の式であったのに対し, 三軸の座標回転を導入すると三角関数が含まれる非線形の式となり, 通常連立方程式で解くことは出来ない. したがって, テイラー展開を使って逐次計算をすることにより近似値を求めることになる.

そのためにはまず, 変換式を次のように関数として置き換える.

$$\begin{aligned} F_x(x_0, \omega, \varphi, \kappa) &= a_{11}u + a_{12}v + a_{13}w + x_0 - x \\ &= (\cos \varphi \cos \kappa)u - (\cos \varphi \sin \kappa)v + (\sin \varphi)w + x_0 - x \end{aligned} \quad (35)$$

$$\begin{aligned} F_y(y_0, \omega, \varphi, \kappa) &= a_{21}u + a_{22}v + a_{23}w + y_0 - y \\ &= (\cos \omega \sin \kappa + \sin \omega \sin \varphi \cos \kappa)u + (\cos \omega \cos \kappa - \sin \omega \sin \varphi \sin \kappa)v \\ &\quad + (-\sin \omega \cos \varphi)w + y_0 - y \end{aligned} \quad (36)$$

$$\begin{aligned} F_z(z_0, \omega, \varphi, \kappa) &= a_{31}u + a_{32}v + a_{33}w + z_0 - z \\ &= (\sin \omega \sin \kappa - \cos \omega \sin \varphi \cos \kappa)u + (\sin \omega \cos \kappa + \cos \omega \sin \varphi \sin \kappa)v \\ &\quad + (\cos \omega \cos \varphi)w + z_0 - z \end{aligned} \quad (37)$$

各未知係数の近似値を  $x_{00}, y_{00}, z_{00}, \omega_0, \varphi_0, \kappa_0$  とし, 各補正量を  $\Delta x_0, \Delta y_0, \Delta z_0, \Delta \omega, \Delta \varphi, \Delta \kappa$  とすると, 未知係数は, 以下の式で表すことができる.

$$x_0 = x_{00} - \Delta x_0 \quad (38)$$

$$y_0 = y_{00} - \Delta y_0 \quad (39)$$

$$z_0 = z_{00} - \Delta z_0 \quad (40)$$

$$\omega = \omega_0 - \Delta \omega \quad (41)$$

$$\varphi = \varphi_0 - \Delta \varphi \quad (42)$$

$$\kappa = \kappa_0 - \Delta \kappa \quad (43)$$

関数  $F_x, F_y, F_z$  において, 近似値の周りにテイラー展開する. つまりテイラー級数 (式 1) において,  $x = x_0, a = x_{00}$  とおく. すると,  $x_0 - x_{00} = -\Delta x_0$  となる. 他の変数  $y_0, z_0, \omega, \varphi, \kappa$  についても同様に考え, 多変数の関数なので偏微分を用いると以下の式が得られる. なお, 級数における二階導

関数の項以降は無視できるものとしている.

$$F_x(x_0, \omega, \varphi, \kappa) \approx F_x(x_{00}, \omega_0, \varphi_0, \kappa_0) - \frac{\partial F_x}{\partial x_0} \Delta x_0 - \frac{\partial F_x}{\partial \omega} \Delta \omega - \frac{\partial F_x}{\partial \varphi} \Delta \varphi - \frac{\partial F_x}{\partial \kappa} \Delta \kappa \quad (44)$$

$$F_y(y_0, \omega, \varphi, \kappa) \approx F_y(y_{00}, \omega_0, \varphi_0, \kappa_0) - \frac{\partial F_y}{\partial y_0} \Delta y_0 - \frac{\partial F_y}{\partial \omega} \Delta \omega - \frac{\partial F_y}{\partial \varphi} \Delta \varphi - \frac{\partial F_y}{\partial \kappa} \Delta \kappa \quad (45)$$

$$F_z(z_0, \omega, \varphi, \kappa) \approx F_z(z_{00}, \omega_0, \varphi_0, \kappa_0) - \frac{\partial F_z}{\partial z_0} \Delta z_0 - \frac{\partial F_z}{\partial \omega} \Delta \omega - \frac{\partial F_z}{\partial \varphi} \Delta \varphi - \frac{\partial F_z}{\partial \kappa} \Delta \kappa \quad (46)$$

テイラー展開による微係数は、以下のように計算される.

$$\frac{\partial F_x}{\partial x_0} = 1, \frac{\partial F_y}{\partial y_0} = 1, \frac{\partial F_z}{\partial z_0} = 1, \frac{\partial F_x}{\partial \omega} = 0 \quad (47)$$

$$\frac{\partial F_x}{\partial \varphi} = -(\sin \varphi \cos \kappa)u + (\sin \varphi \sin \kappa)v + (\cos \varphi)w \quad (48)$$

$$\frac{\partial F_x}{\partial \kappa} = -(\cos \varphi \sin \kappa)u - (\cos \varphi \cos \kappa)v \quad (49)$$

$$\frac{\partial F_y}{\partial \omega} = (-\sin \omega \sin \kappa + \cos \omega \sin \varphi \cos \kappa)u + (-\sin \omega \cos \kappa - \cos \omega \sin \varphi \sin \kappa)v - (\cos \omega \cos \varphi)w \quad (50)$$

$$\frac{\partial F_y}{\partial \varphi} = (\sin \omega \cos \varphi \cos \kappa)u + (\sin \omega \cos \varphi \sin \kappa)v + (\sin \omega \sin \varphi)w \quad (51)$$

$$\frac{\partial F_y}{\partial \kappa} = (\cos \omega \cos \kappa - \sin \omega \sin \varphi \sin \kappa)u + (-\cos \omega \sin \kappa - \sin \omega \sin \varphi \cos \kappa)v \quad (52)$$

$$\frac{\partial F_z}{\partial \omega} = (\cos \omega \sin \kappa + \sin \omega \sin \varphi \cos \kappa)u + (\cos \omega \cos \kappa - \sin \omega \sin \varphi \sin \kappa)v - (\sin \omega \cos \varphi)w \quad (53)$$

$$\frac{\partial F_z}{\partial \varphi} = (-\cos \omega \cos \varphi \cos \kappa)u + (\cos \omega \cos \varphi \sin \kappa)v - (\cos \omega \sin \varphi)w \quad (54)$$

$$\frac{\partial F_z}{\partial \kappa} = (\sin \omega \cos \kappa + \cos \omega \sin \varphi \sin \kappa)u + (-\sin \omega \sin \kappa + \cos \omega \sin \varphi \cos \kappa)v \quad (55)$$

$x_{00}, y_{00}, z_{00}, \omega_0, \varphi_0, \kappa_0$  の初期値と基準点データを式 20~21 に代入し、最小二乗法を適用すれば、補正量  $\Delta x_0, \Delta y_0, \Delta z_0, \Delta \omega, \Delta \varphi, \Delta \kappa$  が求まる. ここで、問題となるのが初期値の値である. もし三次元データ  $(u, v, w)$  と  $(x, y, z)$  が、共に整準された計測機器によって測られたものであれば、 $\omega_0 = 0, \varphi_0 = 0$  とおくことが出来る.

## 1.4 平均二乗誤差による変換式の評価

### 1.4.1 基準点データによる評価

最小二乗法によって得られた変換式については、確からしい変換が可能かどうか評価しなければならぬ. 一般に、変換式にどの程度誤差が含まれているのかは、**平均二乗誤差** (root mean square error) で評価することができる. 画像基準点の座標  $(u_i, v_i)$  を変換式に代入すると、地上座標  $(f_x(u_i, v_i), f_y(u_i, v_i))$  が得られる. しかしこの座標は、実際の地上基準点の座標  $(x_i, y_i)$  とは若干異なる. 基準点のデータが正しいとするならば、これらの差は誤差といえる. 全基準点を用いて以下の

式により平均二乗誤差  $RMSE_x, RMSE_y$  を求めることが出来る。

$$RMSE_x = \sqrt{\sum_{i=1}^n (x_i - f_x(u_i, v_i))^2} \quad (56)$$

$$RMSE_y = \sqrt{\sum_{i=1}^n (y_i - f_y(u_i, v_i))^2} \quad (57)$$

この平均二乗誤差を計算した上で、画素の単位にして 1 画素以上の誤差が発生しているようであれば、基準点データの信頼性を疑わなければならない。信頼できる基準点データにも関わらず誤差が大きい場合は、変換式自体に問題がある可能性があるため、十分な検討が必要である。

#### 1.4.2 検証点データによる評価

基準点データによる平均二乗誤差に問題がなかったとしても、それで十分とは言えない場合がある。そこで、基準点データ以外に検証点を用いて評価することが好ましい。基準点から離れたところに検証点を配置させ、誤差の状況を確認すべきである。このときの誤差の評価方法は、検証点を使った平均二乗誤差により可能である。

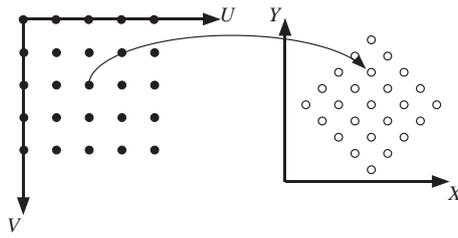
### 1.5 画像の再配列

変換式に従って、原画像を幾何学的に変換させて新しい画像を作成することを**再配列** (resampling) と呼んでいる。この再配列は、通常地上座標系に従って画像を生成し、他の画像や計測データと重ね合わせが可能となるために行う。例えば、たくさんの衛星画像は、それぞれの画像座標を持ち、直接重ね合わせても位置がずれているので、再配列が必要なのである。再配列の際には、まず地上座標において画像化する範囲と 1 ピクセルあたりの大きさを決める必要がある。そしてこれらの決められた情報は、作成された画像とともに正確に保存しておかなければならない。このような情報は**メタデータ** (meta data) と呼ばれており、第十章の地理情報システムで、改めて解説する。通常、リモートセンシングのソフトウェアや地理情報システムのソフトウェアで扱う画像でなければ、画像には地上座標の情報が付与されていないので、自作ソフトで画像を再配列させるときには、メタデータを独自で保存しておくべきである。

それではここで、アフィン変換を用いた場合の再配列法について解説する。アフィン変換は、これまで次式で表してきた。

$$\begin{cases} x = au + bv + c \\ y = du + ev + f \end{cases} \quad (58)$$

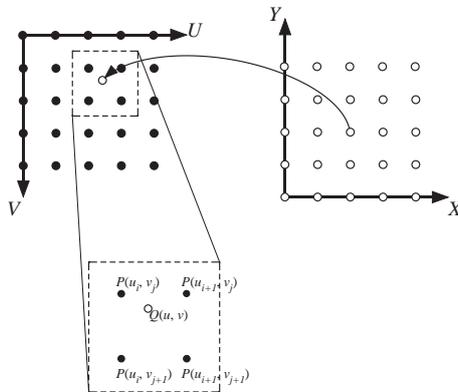
基準点データより変換係数である  $a, b, c, d, e, f$  が求めれば、画像座標  $(u, v)$  から地上座標  $(x, y)$  を計算することが出来る。この式に従って、各画像の値を地上座標上にプロットすると、下図のようなイメージで変換される。



したがって、地上座標に沿って等間隔の画像とはならず、斜めに歪んだ画像になってしまう。そこで地上座標系に従った画像を生成しようとした場合、地上座標  $(x, y)$  を画像座標  $(u, v)$  に変換する式を利用する方が簡単である。つまり、次式のような変換式を用いるのが一般的である。

$$\begin{cases} u = ax + by + c \\ v = dx + ey + f \end{cases} \quad (59)$$

単に  $(x, y)$  と  $(u, v)$  を入れ替えただけである。この式のもと、基準点データより変換係数を求める。変換式が立てられれば、地上座標  $(x, y)$  は、画像座標のどこに対応するかが計算できるのである。下図は、再配列のイメージを表したもので、地上座標系に沿って等間隔に画素を配置させ、それぞれの画素が画像ではどの位置に来るかを示している。



ここで問題となるのが、地上座標での画素の位置と画像座標での画素の位置が一一に対応せず、中途半端な位置に来ってしまうことである。この問題を解決するためにいくつかの手法が存在する。

### 1.5.1 Nearest Neighbor 法

Nearest Neighbor 法は、**最近隣法**と訳されている。この手法は、位置の最も近い画素の値で代用する手法である。ある地上座標  $(x, y)$  が画像においては  $(u, v)$  で与えられたとき、その画像上の周りの4つの画素の値がそれぞれ  $P(u_i, v_j), P(u_{i+1}, v_j), P(u_i, v_{j+1}), P(u_{i+1}, v_{j+1})$  だったとき、地上座標  $(x, y)$  の画素の値  $Q(x, y)$  は、最も近い値で表す。上図の場合では、 $P(u_i, v_j)$  となる。式では、次の

ように表すことが出来る.

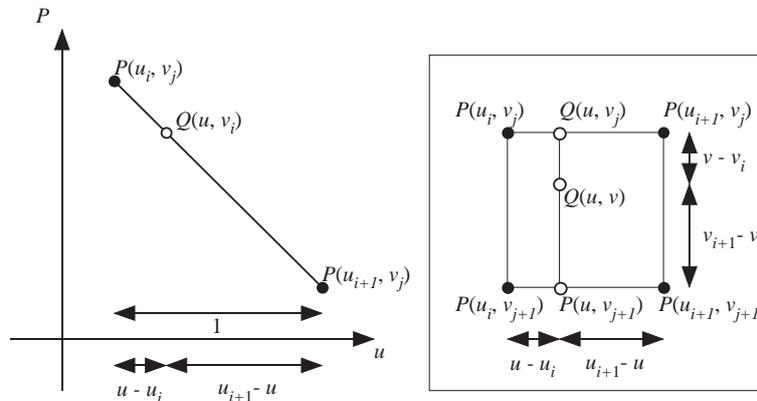
$$\begin{aligned} Q(x, y) &= P(u, v) \\ u &= \text{int}(ax + by + c + 0.5) \\ v &= \text{int}(dx + ey + f + 0.5) \end{aligned} \quad (60)$$

ここで,  $\text{int}()$  は, 計算された値から整数部のみを返す関数を表している. 変換式によって計算された値に 0.5 を加えることにより, 小数第一位で四捨五入して得られる値と等しくなるようにしている.

この手法は, 簡単なアルゴリズムであり, 元の画素の値が壊されないことから, 衛星画像の最配列には多用されている.

### 1.5.2 Bi-Linear 法

Bi-Linear 法は, **共一次内挿法**と訳されている. この手法は, 周囲の 4 つの画素を利用して, 線形近似を行うものである. 画像は二次元であるが, 簡単に考えるため  $u$  方向の一次元平面で考える. 下図は,  $v$  座標が  $v_i$  における  $P(u_i, v_j), P(u_{i+1}, v_j)$  を通る断面での状況をグラフに表したものである. 求めたい点  $Q(u, v_i)$  が,  $P(u_i, v_j)$  と  $P(u_{i+1}, v_j)$  の間にあるとき,  $P(u_i, v_j)$  と  $P(u_{i+1}, v_j)$  を直線で繋ぎ, その直線上の値でもって  $Q(u, v_i)$  の値として内挿することが出来る. このような内挿法は, 線形内挿とも呼ばれている.



$u, v$  ともに画素の単位なので, 画素間の距離は 1 とすることが出来る. そうすると  $u$  方向の距離を用いて, 比例配分に従うと,  $Q(u, v_j)$  は次式で表すことが出来る.

$$Q(u, v_j) = (u_{i+1} - u)P(u_i, v_j) + (u - u_i)P(u_{i+1}, v_j) \quad (61)$$

$v$  座標が  $v_{j+1}$  のときも同様で, 内挿後の値  $Q(u, v_{j+1})$  は, 次式で表すことが出来る.

$$Q(u, v_{j+1}) = (u_{i+1} - u)P(u_i, v_{j+1}) + (u - u_i)P(u_{i+1}, v_{j+1}) \quad (62)$$

これを二次元に拡張するため, 今度は  $Q(u, v_j)$  と  $Q(u, v_{j+1})$  間において  $v$  方向の内挿を考えると,

$Q(u, v)$  は次式で表すことができる.

$$\begin{aligned} Q(u, v) &= (v_{j+1} - v)Q(u, v_j) + (v - v_j)Q(u, v_{j+1}) \\ &= (u_{i+1} - u)(v_{j+1} - v)P(u_i, v_j) + (u - u_i)(v_{j+1} - v)P(u_{i+1}, v_j) \\ &\quad + (u_{i+1} - u)(v - v_j)P(u_i, v_{j+1}) + (u - u_i)(v - v_j)P(u_{i+1}, v_{j+1}) \end{aligned} \quad (63)$$

この手法により画像を再配列させると、最近隣法に比べて画像にシャープさが無くなるというデメリットがある。したがって衛星画像の最配列にはあまり利用されていない。画像ではなく、標高データの場合には、このような手法で内挿することが多い。

最配列は、ケースバイケースで手法を決定する必要がある。Bi-Linear 法の他にも線形内挿では無く、非線形の関数を用いた内挿手法もある。例えば、三次関数を利用した Cubic Convolution (三次たたみ込み法) が有名である。しかし衛星画像の解析においては、画素の値を出来るだけ壊さない方が良いと言えるので、ここでは Nearest Neighbor 法を勧める。Cubic Convolution 法については、他の文献を参考にしてほしい。