地形无关的控制方案建立RPC模型

武汉大学
遥感信息工程学院 赵双明
smzhao@whu.edu.cn

旋转矩阵（Euler Angle）

rotation matrix to transform **column 3-vectors** from one cartesian coordinate system to another. Final system is formed by rotating original system about its own axis by angle ϕ(counterclockwise as viewed from the +axis direction):

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\omega & -sin\omega \\ 0 & sin\omega & cos\omega \end{bmatrix} \begin{bmatrix} X_\omega \\ Y_\omega \\ Z_\omega \end{bmatrix} \qquad \begin{bmatrix} X_\omega \\ Y_\omega \\ Z_\omega \end{bmatrix} = R_1\ (\omega) \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \qquad R_1(\omega) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\omega & sin\omega \\ 0 & -sin\omega & cos\omega \end{bmatrix}$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} cos\varphi & 0 & sin\varphi \\ 0 & 1 & 0 \\ -sin\varphi & 0 & cos\varphi \end{bmatrix} \begin{bmatrix} X_\varphi \\ Y_\varphi \\ Z_\varphi \end{bmatrix} \qquad \begin{bmatrix} X_\varphi \\ Y_\varphi \\ Z_\varphi \end{bmatrix} = R_2\ (\varphi) \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \qquad R_2(\varphi) = \begin{bmatrix} cos\phi & 0 & -sin\phi \\ 0 & 1 & 0 \\ sin\phi & 0 & cos\phi \end{bmatrix}$$

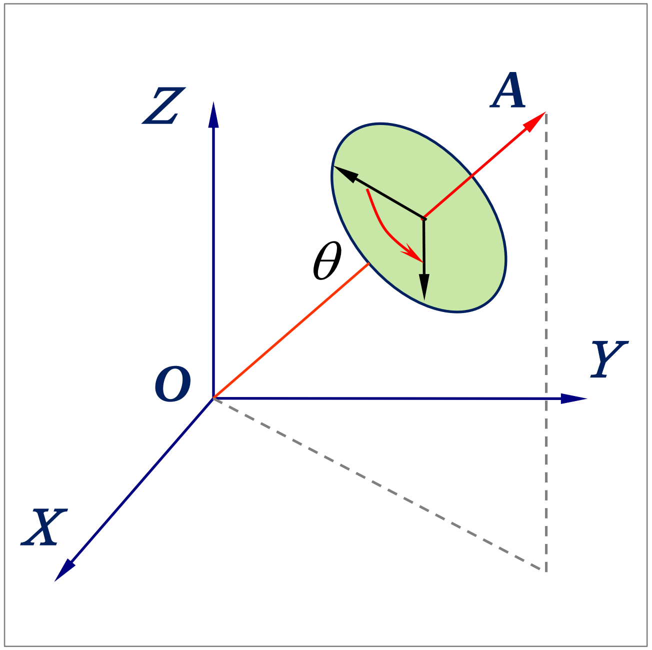
$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} cos\kappa & -sin\kappa & 0 \\ sin\kappa & cos\kappa & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_\kappa \\ Y_\kappa \\ Z_\kappa \end{bmatrix} \qquad \begin{bmatrix} X_\kappa \\ Y_\kappa \\ Z_\kappa \end{bmatrix} = R_3\ (\kappa) \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \qquad R_3(\kappa) = \begin{bmatrix} cos\kappa & sin\kappa & 0 \\ -sin\kappa & cos\kappa & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

四元数（Quaternion）描述姿态

空间任意一个旋转由一个*旋转轴*和一个*转角*进行描述

旋转轴：矢量表示 $\overrightarrow{OA} = [X_A,\quad Y_A,\quad Z_A]$

旋　转：旋转角　θ



分别定义四个元素：

$$x = c \bullet X_A \quad y = c \bullet Y_A$$

$$z = c \bullet Z_A \quad w = s$$

$$c = \sin(\theta/2) \quad s = \cos(\theta/2)$$

$$Q = [x,\quad y,\quad z,\quad w]$$

$$Q = w + xi + yj + zk$$

$$Q = [q_1 \quad q_2 \quad q_3 \quad q_4]$$

称为为四元数(Quaternion)

$$N(Q) = x^2 + y^2 + z^2 + w^2 = 1$$

单位四元数

旋转矩阵（ Quaternion ）

$$\boldsymbol{R}=\begin{bmatrix} 1-2(\boldsymbol{y}^2+\boldsymbol{z}^2) & 2(\boldsymbol{xy}-\boldsymbol{wz}) & 2(\boldsymbol{xz}+\boldsymbol{wy}) \\ 2(\boldsymbol{xy}+\boldsymbol{wz}) & 1-2(\boldsymbol{x}^2+\boldsymbol{z}^2) & 2(\boldsymbol{yz}-\boldsymbol{wx}) \\ 2(\boldsymbol{xz}-\boldsymbol{wy}) & 2(\boldsymbol{yz}+\boldsymbol{wx}) & 1-2(\boldsymbol{x}^2+\boldsymbol{y}^2) \end{bmatrix}$$

$$\boldsymbol{R}_E=\boldsymbol{R}_\varphi\boldsymbol{R}_\omega\boldsymbol{R}_\kappa=\begin{bmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 & \boldsymbol{a}_3 \\ \boldsymbol{b}_1 & \boldsymbol{b}_2 & \boldsymbol{b}_3 \\ \boldsymbol{c}_1 & \boldsymbol{c}_2 & \boldsymbol{c}_3 \end{bmatrix}$$

$$\boldsymbol{R}_E=\boldsymbol{R} \Rightarrow$$

$$\varphi=-\mathrm{arctg}\left(\frac{\boldsymbol{a}_3}{\boldsymbol{c}_3}\right)$$

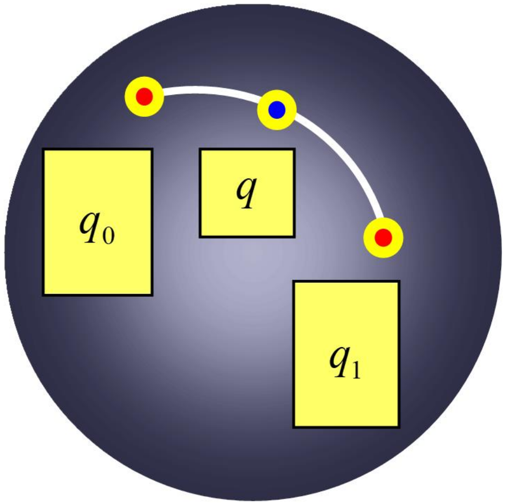
$$\omega=-\arcsin(\boldsymbol{b}_3)$$

$$\kappa=-\mathrm{arctg}\left(\frac{\boldsymbol{b}_1}{\boldsymbol{b}_2}\right)$$

四元数插值

三维单位矢量定义了一个球状上点。单位四元数则定义了一个四维超球面上的一个点。沿着超球状上两点之间的大弧进行插值就可得到平滑轨迹

$$
\begin{array}{lllllll}
t_0 & q_0 = [x_0 & y_0 & z_0 & w_0] \\
t & q_t = [x_t & y_t & z_t & w_t] \\
t_1 & q_1 = [x_1 & y_1 & z_1 & w_1]
\end{array}
$$



四元数球面线性内插

$$q_t = \eta_0(t) q_0 + \eta_1(t) q_1$$

$$\eta_0 = \frac{\sin(\theta(t_1 - t)/(t_1 - t_0))}{\sin(\theta)}$$

$$\eta_1 = \frac{\sin(\theta(t - t_0)/(t_1 - t_0))}{\sin(\theta)}$$

$$\cos(\theta) = q_0 q_1$$

$|q_0 q_1| \to 1 \Rightarrow$ 退化为线性内插

$q_0 q_1 < 0 \Rightarrow \theta > \frac{\pi}{2}$，$q_0$或$q_1$取反

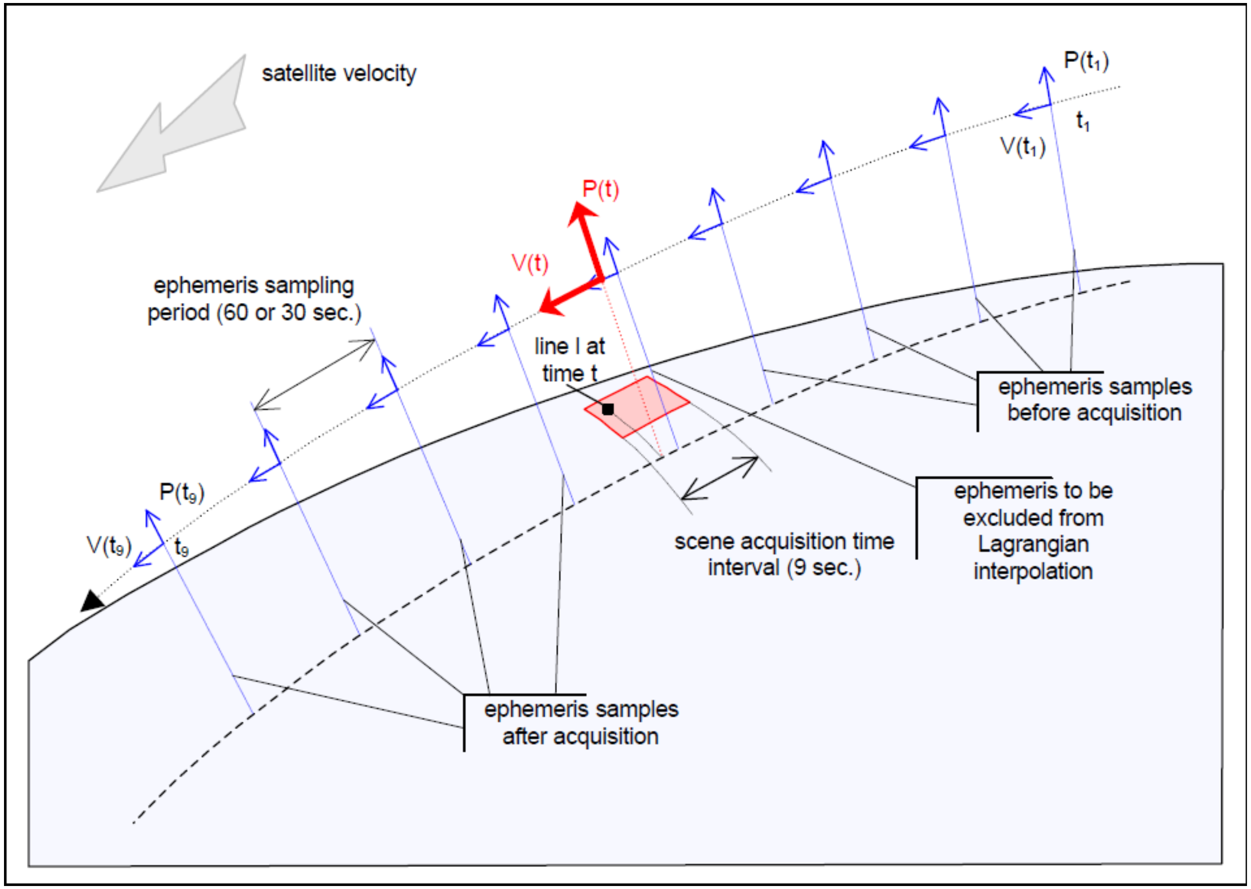
使得插值的角距离最小化

姿态数据

```
attData_00 =
{
    timeCode = 97499270.0736999959 ;//卫星累积秒，从2009-01-01-0:0:0开始计
    dateTime = "2012 02 03 11:07:50.073700" ;           //格林尼治时间
    roll  = -0.00001245 ;               // roll/pitch/yaw  卫星本体到轨道坐标系姿态
    pitch = -0.00000509 ;
    yaw   = 0.03356850 ;
    roll_velocity  = 0.00000000 ;
    pitch_velocity = 0.00000000 ;
    yaw_velocity   = 0.00000000 ;
    q1 = -0.76259744 ;   // q1/q2/q3/q4  卫星本体到J2000坐标系姿态;
    q2 = 0.58472162 ;
    q3 = 0.25788319 ;
    q4 = 0.23561892;

}
```

星历内插（GPS）



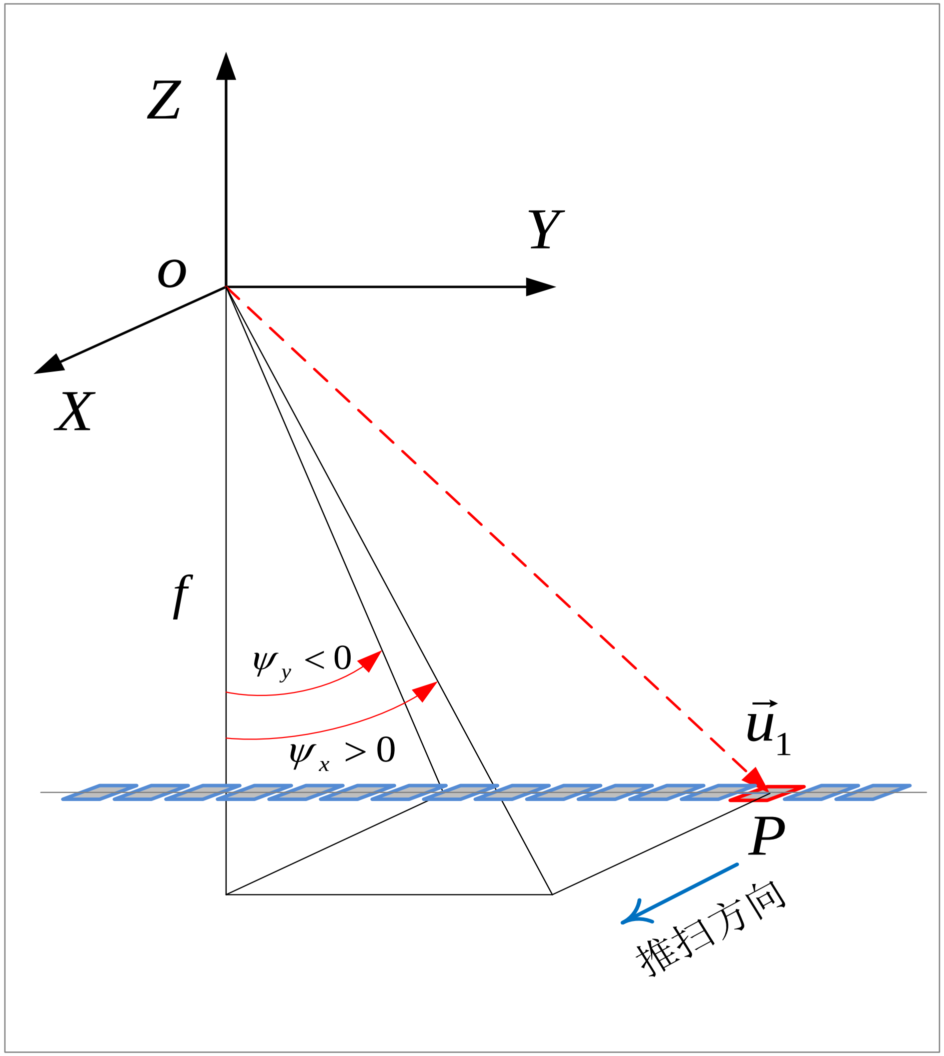
$$\vec{P}(t)=\sum_{j=1}^{8}\frac{\vec{P}(t_j)\times\prod_{\substack{i=1\\i\neq j}}^{8}(t-t_i)}{\prod_{\substack{i=1\\i\neq j}}^{8}(t_j-t_i)}$$

$$\vec{V}(t)=\sum_{j=1}^{8}\frac{\vec{V}(t_j)\times\prod_{\substack{i=1\\i\neq j}}^{8}(t-t_i)}{\prod_{\substack{i=1\\i\neq j}}^{8}(t_j-t_i)}$$

轨道数据

```
coordinateType = WGS84;//GPS测量卫星位置、速度矢量所在坐标系
dataType = GPS;
groupNumber = 542 ;        //测量记录总数
gpsData_00 =
{
    timeCode = 97499268.0000000000 ;//卫星累积秒，从2009-01-01-0:0:0开始计
    dateTime = “2012 02 03 11:07:48.000000”；//格林尼治时间
    PX = -1912991.5000000000 ;     //卫星位置（X,Y,Z），单位米
    PY = 3092157.1000000001 ;
    PZ = 5828414.6000000006 ;
    VX = -1475.4500000000 ;          //卫星速度（VX,VY,VZ），单位米/秒
    VY = 6456.2100000000 ;
    VZ = -3905.8700000000 ;
}
```

相机坐标系\像元指向角



$$\frac{x - x_0 - \Delta x}{f} = \tan\psi_y$$

$$\frac{y - y_0 - \Delta y}{f} = \tan\psi_x$$

Look direction Angles

ψ_x / ψ_y

焦平面上像元的指向角
描述像元的位置

SPOT/IKONOS/ZY3等

相机校正文件

像元ID	垂轨指向角	沿轨指向角
0.000000	0.050668614594336	-0.000791106890569
1.000000	0.050664505955175	-0.000791109449488
2.000000	0.050660397314306	-0.000791112008175
3.000000	0.050656288671731	-0.000791114566631
4.000000	0.050652180027448	-0.000791117124855
5.000000	0.050648071381458	-0.000791119682847
6.000000	0.050643962733761	-0.000791122240609
7.000000	0.050639854084357	-0.000791124798139
8.000000	0.050635745433247	-0.000791127355437
9.000000	0.050631636780430	-0.000791129912504
10.000000	0.050627528125907	-0.000791132469341
11.000000	0.050623419469678	-0.000791135025946

……

$$\psi(p)=\psi(i)+\left(\psi(p_{i+1})-\psi(p_i)\right)\times\frac{p-p_i}{p_{i+1}-p_i}\qquad (p_i<p<p_{i+1})$$

偏置矩阵文件

starttime = 0.000000000000000

pitch = -0.000511776876952

Vpitch = -0.000000000125194

roll = 0.001828916699906

Vroll = 0.000000000892101

yaw = 0.003770429577750

Vyaw = 0.000000000000000

$$R = \begin{bmatrix} cos\varphi & 0 & sin\varphi \\ 0 & 1 & 0 \\ -sin\varphi & 0 & cos\varphi \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\omega & -sin\omega \\ 0 & sin\omega & cos\omega \end{bmatrix} \begin{bmatrix} cos\kappa & -sin\kappa & 0 \\ sin\kappa & cos\kappa & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

时间标签文件

RelLine	Time	deltaTime
0	131862405.000371930000000000	131862405.000371930000000000
1	131862405.000743870000000000	0.000371932983398437500
2	131862405.00111580000000000	0.000371932983398437500
3	131862405.001487730000000000	0.000371932983398437500
4	131862405.001859660000000000	0.000371932983398437500
5	131862405.002231600000000000	0.000371932983398437500
6	131862405.002603530000000000	0.000371932983398437500
7	131862405.002975460000000000	0.000371932983398437500
8	131862405.003347400000000000	0.000371932983398437500
9	131862405.003719330000000000	0.000371932983398437500
10	131862405.004091260000000000	0.000371932983398437500
11	131862405.004463200000000000	0.000371932983398437500
12	131862405.004835130000000000	0.000371932983398437500

……

相机严格模型

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{WGS84} = \begin{bmatrix} X_{GPS} \\ Y_{GPS} \\ Z_{GPS} \end{bmatrix} + R_{J2000}^{WGS84}(t) R_{star}^{J2000} \left(R_{star}^{body}\right)^{T} \left[\begin{bmatrix} D_x \\ D_y \\ D_z \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} + \lambda \cdot R_{camera}^{body} \begin{bmatrix} tan(\psi_y) \\ tan(\psi_x) \\ -1 \end{bmatrix} \cdot f \right]$$

$$\Downarrow$$

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix}_{WGS84} = \begin{bmatrix} X_{GPS} \\ Y_{GPS} \\ Z_{GPS} \end{bmatrix} + \lambda \cdot R_{J2000}^{WGS84}(t) \cdot R_{body}^{J2000} \cdot R_u \cdot R_{camera}^{body} \begin{bmatrix} tan(\psi_y) \\ tan(\psi_x) \\ -1 \end{bmatrix} \cdot f$$

$$\Downarrow \qquad \qquad \Downarrow$$

$$R_u$$

http://hpiers.obspm.fr/eop-pc/index.php?index=matrice_php&lang=en

EOP TIME SERIES
FTP products
Reference C04 series Each day since 1962
Reference C01 series Each 0.05 year from 1846
EOP series & analysis
EOP series & comparison
Bulletins B, C, D
Last days for EOP
Rotation matrix/vector
WEB Service
EOP series: synoptic
THEORY AND MODELLING
GEOPHYSICAL EXCITATION
LINKS

Compute the Earth rotation matrix

updated on December 2007

Include combined EOP C04 (1962 - today + 6 months prediction) :
Celestial pole offsets ☑ UT1 - UTC ☑ Polar motion ☑

Real time
stop ◉ **yes** ○

☐ Include diurnal and semidiurnal variations produced by ocean tides in polar motion and UT1 (IERS conventions 2000)

Submit request

Civil date : year 2020 month 4 day 17
UTC : 15 h 36 min 22 s

2020 4 17 15 h 36 min 22 s UTC

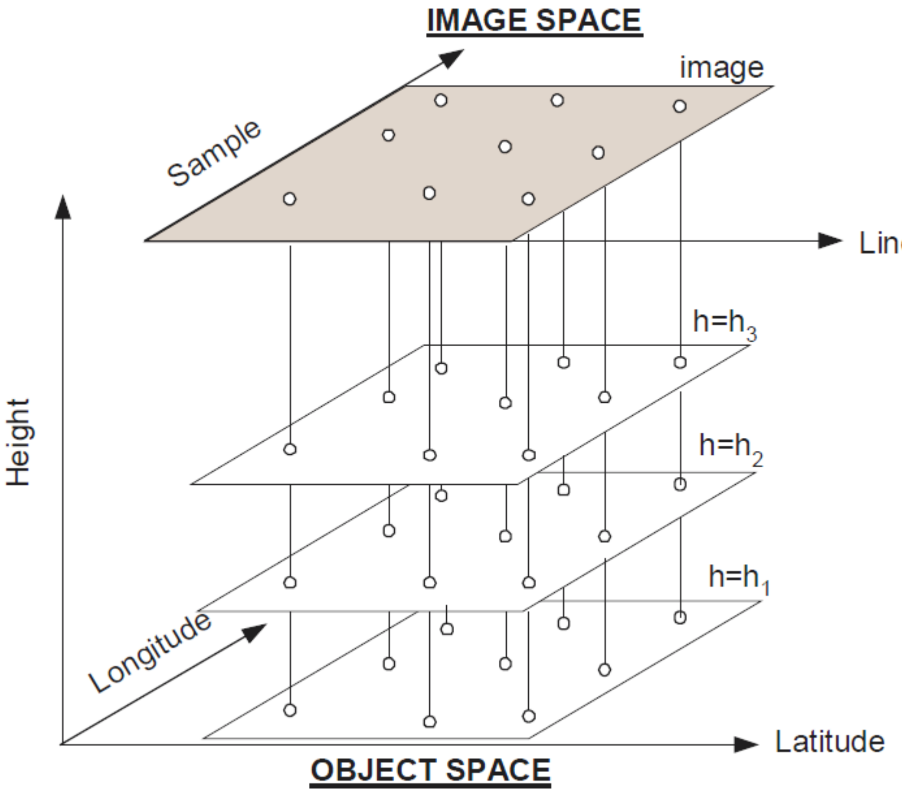
matrice:

0.172639552573	**-0.984983163469**	**0.001937155147**
0.984985012993	**0.172639868400**	**-0.000004241506**
-0.000330252398	**0.001908801039**	**0.999998123704**

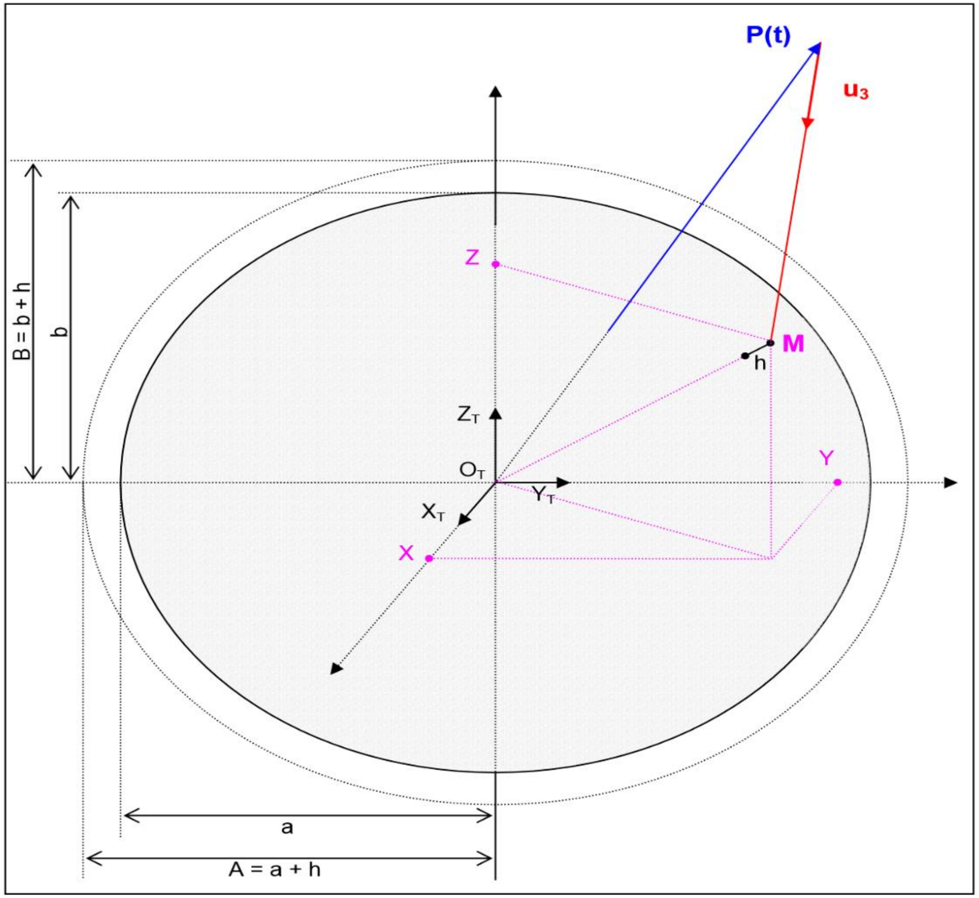
Transformation coordinate M from the international terrestrial reference system (ITRF) to the international celestial reference frame (ICRF) : Celestial coordinates (X Y Z) = M x Terrestrial coordinates (x y z)

- from 1962 to the current week the matrix can include the EOP of the IERS combined series C04 (time resolution for EOP fluctuations is about 6 days), as well as diurnal and semidiurnal variations in UT1 and polar motion produced by oceanic tides (IERS 2000 conventions). Accuracy is 0.0001", corresponding to 5 10^{-10} rad.

Intersection of look direction with the Earth ellipsoid



Getting the look direction u3 from the satellite at the position P(t), we now compute the intersection with the ellipsoid located at an altitude h above the standard ITRF ellipsoid (see fig. below).



Intersection of look direction with the Earth ellipsoid

Let $M = (X, Y, Z)$ the geocentric coordinates to be found, point M is involved within the two following equations:

$$\overrightarrow{O_3M} = \vec{P}(t) + \mu \times \vec{u}_3 \Rightarrow \begin{cases} X = X_P + \mu \times (u_3)_X \\ Y = Y_P + \mu \times (u_3)_Y \\ Z = Z_P + \mu \times (u_3)_Z \end{cases} \qquad Eq.1$$

$$\frac{X^2 + Y^2}{A^2} + \frac{Z^2}{B^2} = 1 \quad with \begin{cases} A = a + h \\ B = b + h \end{cases} \qquad Eq.2$$

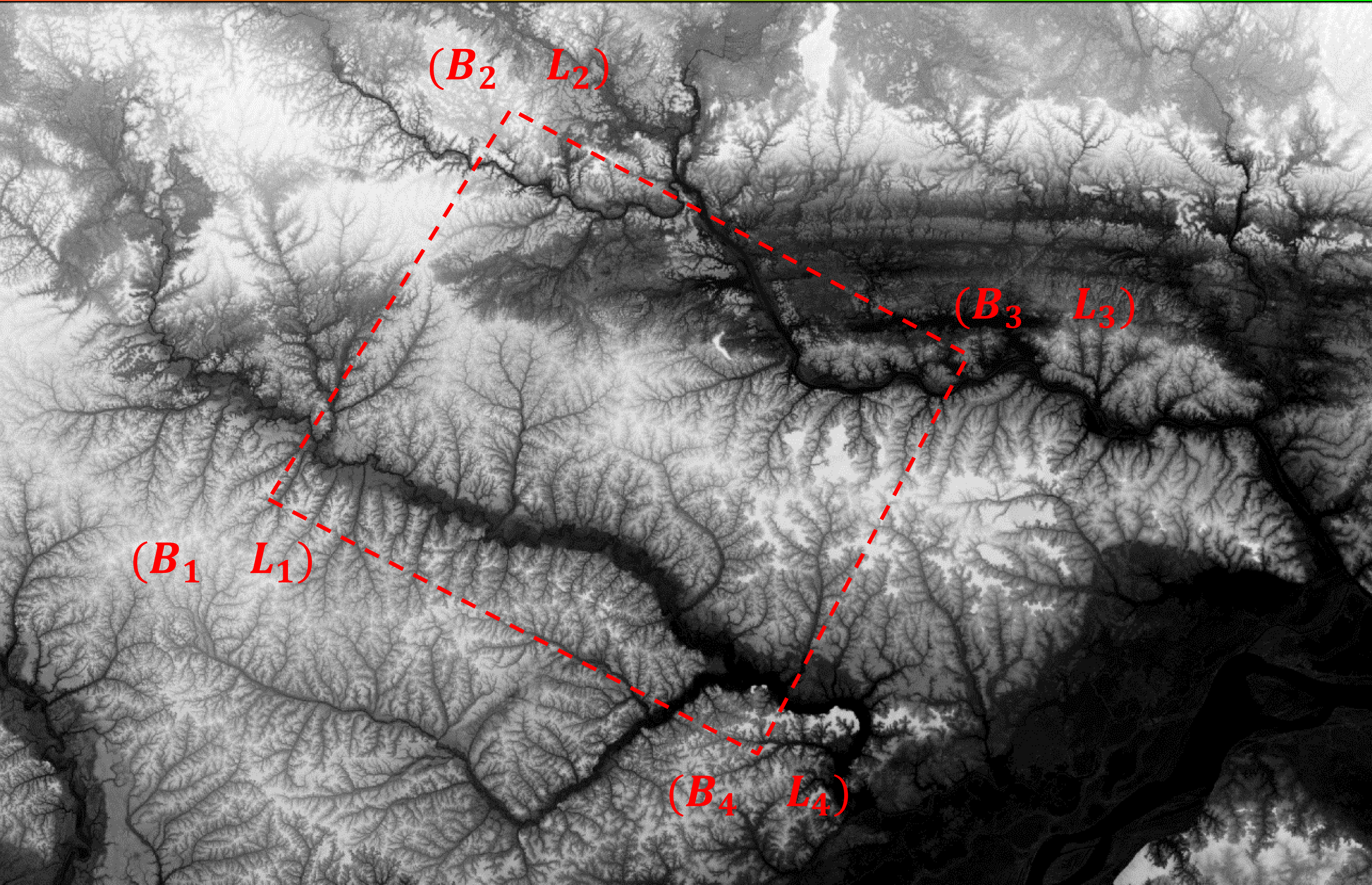
Leading to solve the 2nd degree equation:

$$\left[\frac{(u_3)_X^2 + (u_3)_Y^2}{A^2} + \frac{(u_3)_Z^2}{B^2}\right] \times \mu^2 + 2 \times \left[\frac{X_P(u_3)_X + Y_P(u_3)_Y}{A^2} + \frac{Z_P(u_3)_Z}{B^2}\right] \times \mu + \left[\frac{X_P^2 + Y_P^2}{A^2} + \frac{Z_P^2}{B^2}\right] = 1$$

This equation has necessarily two distinct solutions (μ_1, μ_2). The smallest one (μ_{min}) shall be kept. Re-introducing this value within equation 1 gives the geocentric coordinates (X,Y,Z) of point M. The geodetic coordinates (λ,ϕ,h) may be computed.

视向量$\vec{u}_3$与椭球面求交算法

SRTM-DEM



物方到像方投影变换

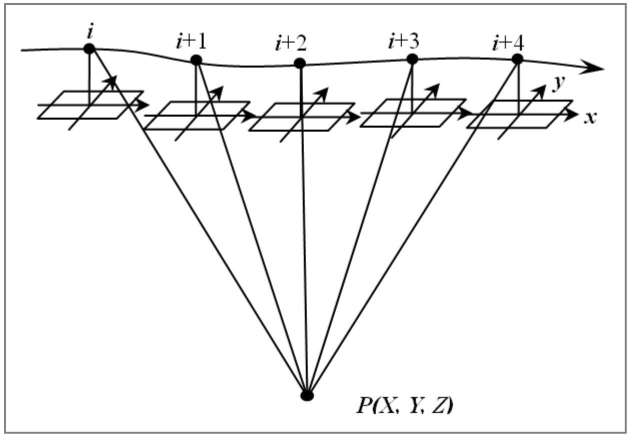
- 实现最佳扫描线搜索
- 基本思想

扫描线搜索是一个迭代过程;

选取扫描线对应的 6 EO计算像点坐标;

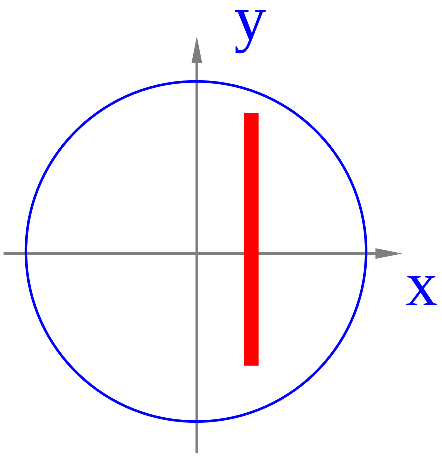
比较*计算的坐标值*与焦平面上像元*标定坐标*?

判断是否一致?



$$x = -f\frac{a_1(X - X_S) + b_1(Y - Y_S) + c_1(Z - Z_S)}{a_3(X - X_S) + b_3(Y - Y_S) + c_3(Z - Z_S)}$$

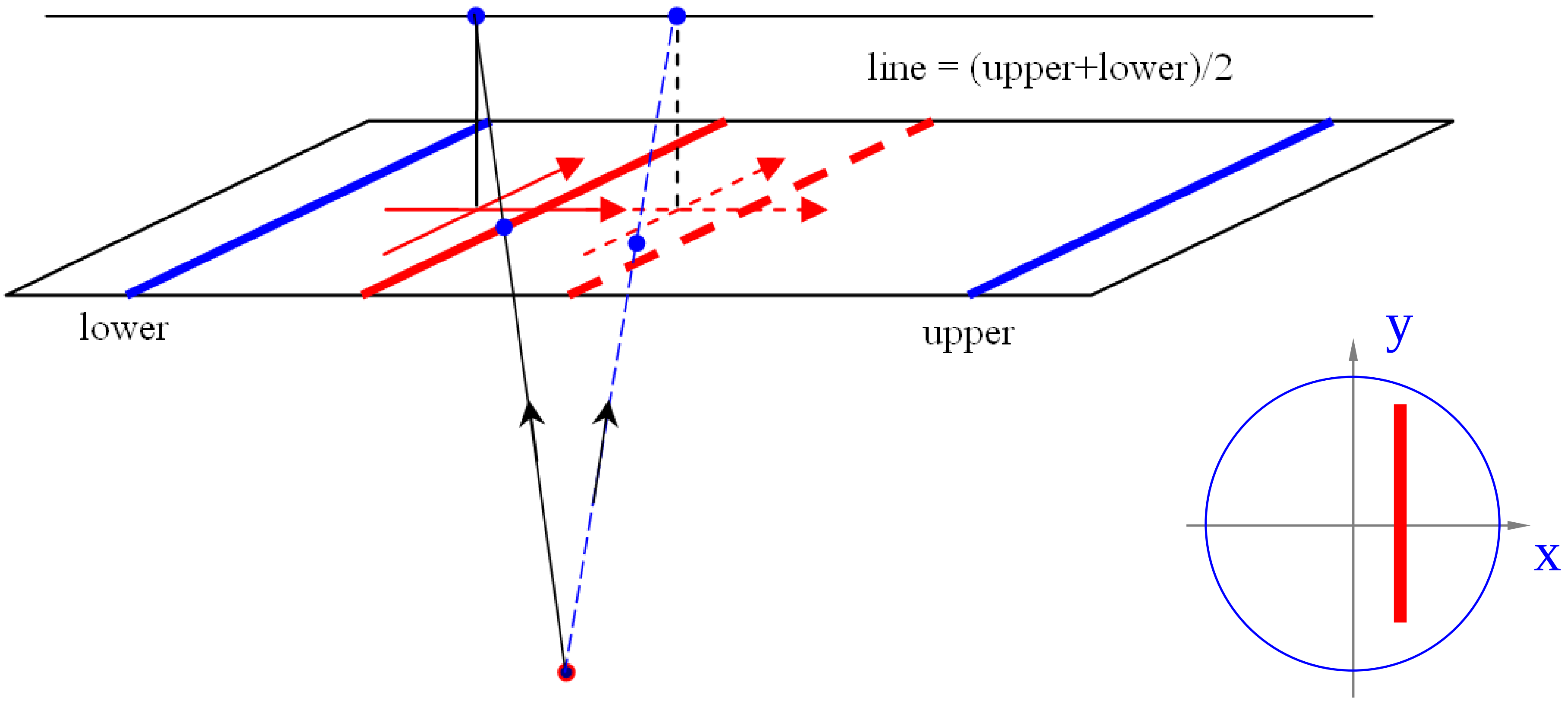
$$y = -f\frac{a_2(X - X_S) + b_2(Y - Y_S) + c_2(Z - Z_S)}{a_3(X - X_S) + b_3(Y - Y_S) + c_3(Z - Z_S)}$$



经典的像方搜索采用顺序搜索的策略

窗口二分法

迭代时，将搜索窗口二等分；分别将搜索窗口的中间、首或尾扫描行的6EO代入共线方程；基于像方搜索判据判断并取窗口的二分之一作为新的搜索窗口，快速、有效地缩小搜索范围，提高了算法效率。



1、最佳扫描行和搜索窗口的迭代初始值

设影像由 N条扫描行构成。

$L=N/2$，初始搜索窗口取为[1,N]， $L_S=1$，$L_E=N$。

2、计算像点坐标

根据 L_S 、L、L_E ，计算像点坐标x_S、 x、 x_E。

3、计算$x_S \cdot x$及 $x \cdot x_E$，并判断其符号

如果$x_S \cdot x \le 0$，则二分L_S及L之间的窗口；

如果 $x \cdot x_E \le 0$，则二分L及L_E之间的窗口；

4、判断搜索窗口大小$|L_S - L_E|$

如果$|L_S - L_E| \le \delta$，则进行下一步；否则转2步

5、搜索窗口$[L_S, L_E]$内确定最佳扫描行

搜索窗口内，依次取扫描线计算像点坐标，与焦平面上CCD线阵距离最近的像点对应最佳扫描线。

RPC模型

$$\left.\begin{aligned} r_n &= \frac{r - r_0}{r_s} \\ c_n &= \frac{c - c_0}{c_s} \end{aligned}\right\}$$

$$\left.\begin{aligned} X_n &= \frac{X - X_0}{X_s} \\ Y_n &= \frac{Y - Y_0}{Y_s} \\ Z_n &= \frac{Z - Z_0}{Z_s} \end{aligned}\right\}$$

$$\boxed{\begin{aligned} r_n &= \frac{p_1(X_n, Y_n, Z_n)}{p_2(X_n, Y_n, Z_n)} \\ c_n &= \frac{p_3(X_n, Y_n, Z_n)}{p_4(X_n, Y_n, Z_n)} \end{aligned}}$$

$$\left\{\begin{aligned} X_0 &= \frac{\sum X}{m} \\ Y_0 &= \frac{\sum Y}{m} \\ Z_0 &= \frac{\sum Z}{m} \end{aligned}\right.$$

$$\left\{\begin{aligned} r_0 &= \frac{\sum r}{m} \\ c_0 &= \frac{\sum c}{m} \end{aligned}\right.$$

$$\left\{\begin{aligned} X_s &= \max\left(\left|X_{\max} - X_0\right|, \quad \left|X_{\min} - X_0\right|\right) \\ Y_s &= \max\left(\left|Y_{\max} - Y_0\right|, \quad \left|Y_{\min} - Y_0\right|\right) \\ Z_s &= \max\left(\left|Z_{\max} - Z_0\right|, \quad \left|Z_{\min} - Z_0\right|\right) \end{aligned}\right.$$

$$\left\{\begin{aligned} r_s &= \max\left(\left|r_{\max} - r_0\right|, \quad \left|r_{\min} - r_0\right|\right) \\ c_s &= \max\left(\left|c_{\max} - c_0\right|, \quad \left|c_{\min} - c_0\right|\right) \end{aligned}\right.$$

多项式形式

多项式形式 关于坐标的3次多项式 $(i+j+k) \leq 3$

$$p = \sum_{i=0}^{m_1}\sum_{j=0}^{m_2}\sum_{k=0}^{m_3} a_{ijk} X^i Y^j Z^k$$

$$= a_0 + a_1 Z + a_2 Y + a_3 X +$$

$$a_4 ZY + a_5 ZX + a_6 YX + a_7 Z^2 + a_8 Y^2 + a_9 X^2 +$$

$$a_{10} ZYX + a_{11} Z^2 Y + a_{12} Z^2 X + a_{13} Y^2 Z + a_{14} Y^2 X +$$

$$a_{15} ZX^2 + a_{16} YX^2 + a_{17} Z^3 + a_{18} Y^3 + a_{19} X^3$$

$$a_i (i = 0, 1, \cdots 19)$$

多项式系数

a_{ijk} are polynomial coefficients called rational function coefficients (RFCS)

线性化

$$r_n = \frac{p_1(X_n,Y_n,Z_n)}{p_2(X_n,Y_n,Z_n)}$$

$$c_n = \frac{p_3(X_n,Y_n,Z_n)}{p_4(X_n,Y_n,Z_n)}$$

$$B = \begin{pmatrix}1 & Z & Y & X & \cdots & Y^3 & X^3\end{pmatrix} \cdot \begin{pmatrix}1 & b_1 & \cdots & b_{19}\end{pmatrix}^T$$

$$J = \begin{pmatrix}a_0 & a_1 & \cdots & a_{19} & b_1 & b_2 & \cdots & b_{19}\end{pmatrix}^T$$

$$D = \begin{pmatrix}1 & Z & Y & X & \cdots & Y^3 & X^3\end{pmatrix} \cdot \begin{pmatrix}1 & d_1 & \cdots & d_{19}\end{pmatrix}^T$$

$$K = \begin{pmatrix}c_0 & c_1 & \cdots & c_{19} & d_1 & d_2 & \cdots & d_{19}\end{pmatrix}^T$$

$$v_r = \begin{bmatrix}\frac{1}{B} & \frac{Z}{B} & \frac{Y}{B} & \frac{X}{B} & \cdots & \frac{Y^3}{B} & \frac{X^3}{B} & \frac{-rZ}{B} & \frac{-rY}{B} & \cdots & \frac{-rY^3}{B} & \frac{-rX^3}{B}\end{bmatrix} \cdot J - \frac{r}{B}$$

$$v_c = \begin{bmatrix}\frac{1}{D} & \frac{Z}{D} & \frac{Y}{D} & \frac{X}{D} & \cdots & \frac{Y^3}{D} & \frac{X^3}{D} & \frac{-cZ}{D} & \frac{-cY}{D} & \cdots & \frac{-cY^3}{D} & \frac{-cX^3}{D}\end{bmatrix} \cdot K - \frac{c}{D}$$

误差方程

假定有n个地面控制点及对应像点

$$\begin{bmatrix} v_{r1} \\ v_{r2} \\ \vdots \\ v_{rn} \end{bmatrix} = \begin{bmatrix} \frac{1}{B_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{B_2} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \frac{1}{B_n} \end{bmatrix} \cdot \begin{bmatrix} 1 & Z_1 & \cdots & X_1^3 & -r_1 Z_1 & \cdots & -r_1 X_1^3 \\ 1 & Z_2 & \cdots & X_2^3 & -r_2 Z_2 & \cdots & -r_2 X_2^3 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & Z_n & \cdots & X_n^3 & -r_n Z_n & \cdots & -r_n X_n^3 \end{bmatrix} \cdot J$$

$$- \begin{bmatrix} \frac{1}{B_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{B_2} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \frac{1}{B_n} \end{bmatrix} \cdot \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix}$$

最小二乘解算

写成矩阵形式 $V_r = W_r M J - W_r R$

$$M = \begin{bmatrix} 1 & Z_1 & \cdots & X_1^3 & -r_1 Z_1 & \cdots & -r_1 X_1^3 \\ 1 & Z_2 & \cdots & X_2^3 & -r_2 Z_2 & \cdots & -r_2 X_2^3 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 1 & Z_n & \cdots & X_n^3 & -r_n Z_n & \cdots & -r_n X_n^3 \end{bmatrix} \quad W_r = \begin{bmatrix} \frac{1}{B_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{B_2} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \frac{1}{B_n} \end{bmatrix} \quad R = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix}$$

法方程形式 $M^T W_r^2 M \cdot J = M^T W_r^2 R$