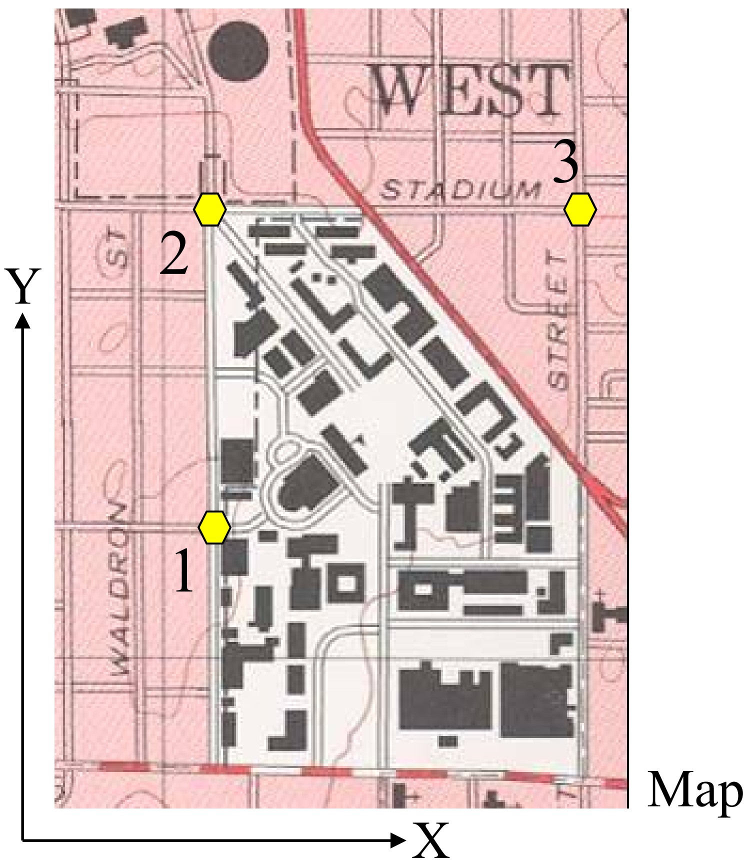
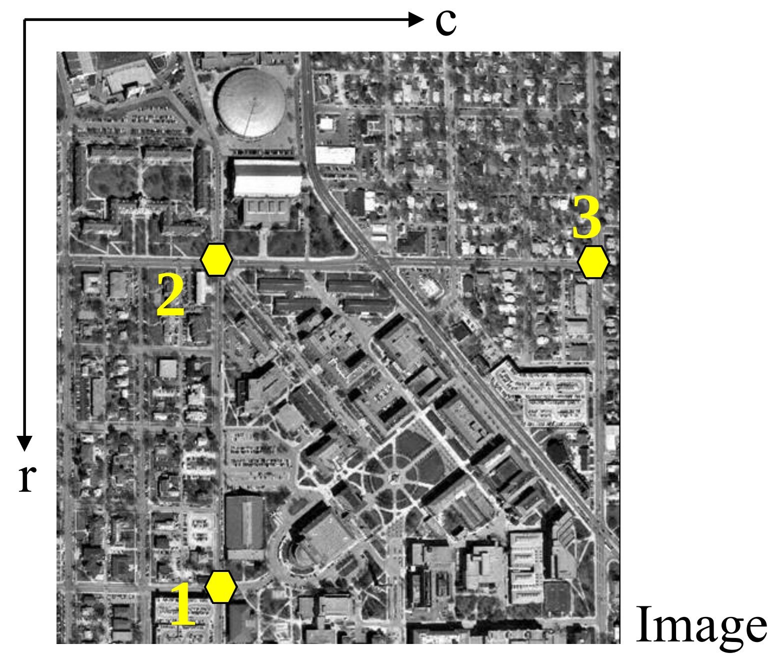
Sensor Modeling and Registration

Mapping Polynomials or Rubber Sheeting





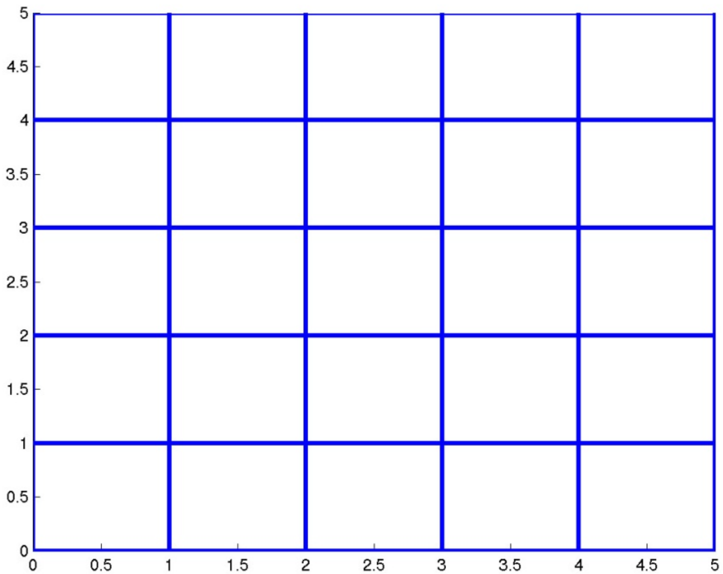
$$r = a_0 + a_1 X + a_2 Y + a_3 XY + a_4 X^2 + a_5 Y^2$$

$$c = b_0 + b_1 X + b_2 Y + b_3 XY + b_4 X^2 + b_5 Y^2$$

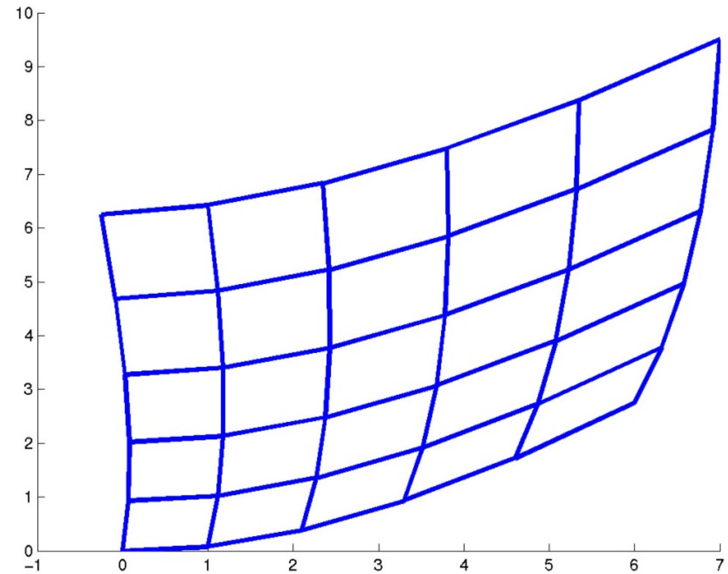
For each point we create two equations. We need at least as many equations as unkowns. If more, then we use least squares. It is like a regression problem: linear, easy. But we are confounding the effects of sensor, platform motion, and terrain relief. What should be the order of the polynomial ?

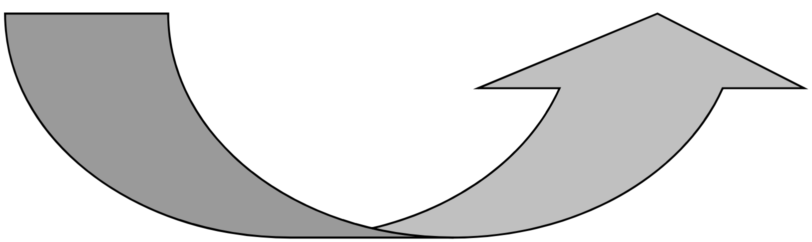
Graphical View of Rubber Sheet Transformation (2^{nd} order, 12-parameter)

Reference grid



Transformed grid





Examples of mapping polynomials from prominent textbooks in remote sensing

2.4.1.1 Mapping Polynomials and Ground Control Points

Since explicit forms for the mapping functions in (2.8) are not known they are generally chosen as simple polynomials of first, second or third degree. For example, in the case of second degree (or order)

$$u = a_0 + a_1 x + a_2 y + a_3 xy + a_4 x^2 + a_5 y^2 \qquad (2.9\,a)$$
$$v = b_0 + b_1 x + b_2 y + b_3 xy + b_4 x^2 + b_5 y^2 \qquad (2.9\,b)$$

Sometimes orders higher than third are used but care must be taken to avoid the introduction of worse errors than those to be corrected, as will be noted later.

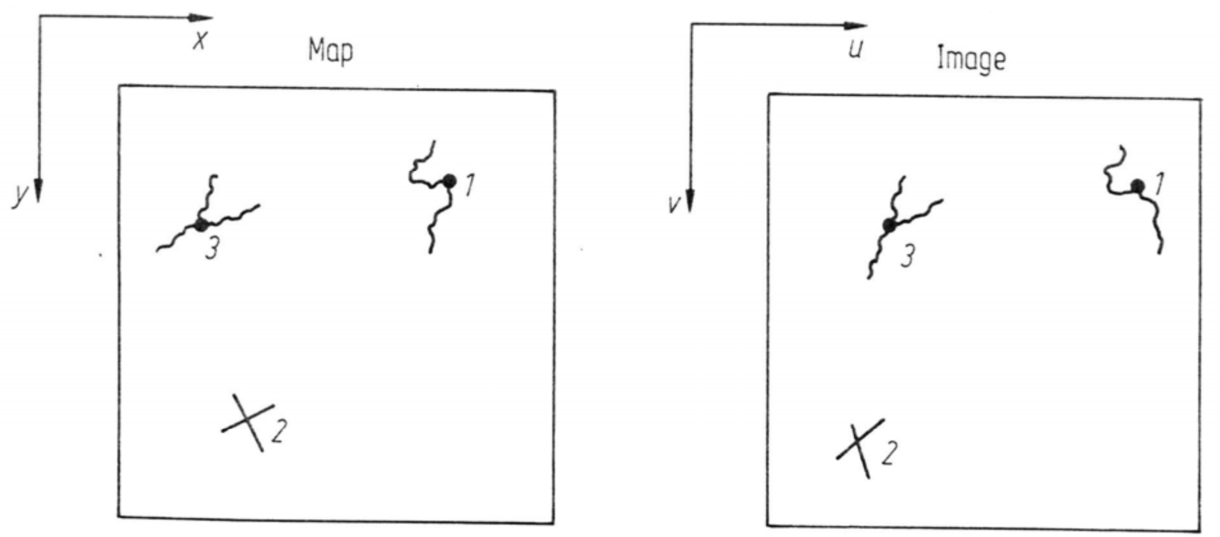


Fig. 2.12. Coordinate systems defined for the image and map, along with the specification of ground control points

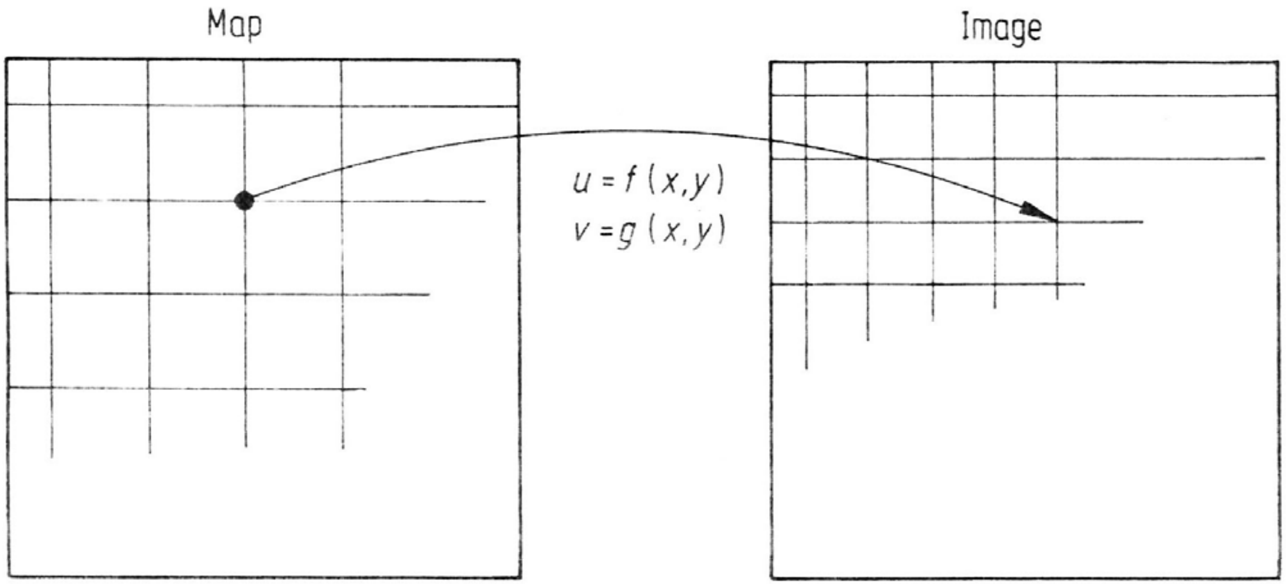


Fig. 2.13. Use of the mapping polynomials to locate points in the image corresponding to display grid positions

isters with the reference image. To accomplish this, we need to know between the two coordinate systems. We assume that the systems can be app ed by a least-squares fit to a polynominal of the form

$$x = a_0 + a_1 x' + a_2 y' + a_3 x' y' + a_4 x'^2 + a_5 y'^2 \cdots + \varepsilon_x$$

$$y = b_0 + b_1 y' + b_2 x' + b_3 y' x' + b_4 y'^2 + b_5 x'^2 \cdots \varepsilon_y$$

where ε_x and ε_y are the residual errors after the transform. For simple dist the two images, a good transform can be achieved with a fairly low-order example, only the zero-th–order terms a_0 and b_0 are needed for a simple shi and the first two terms (a_1, a_0, b_1, and b_0) are needed for a combined scale shifting of the origin. Higher-order terms are required to account for rota keystoning effects due to acquisition and perspective differences (cf. Schow

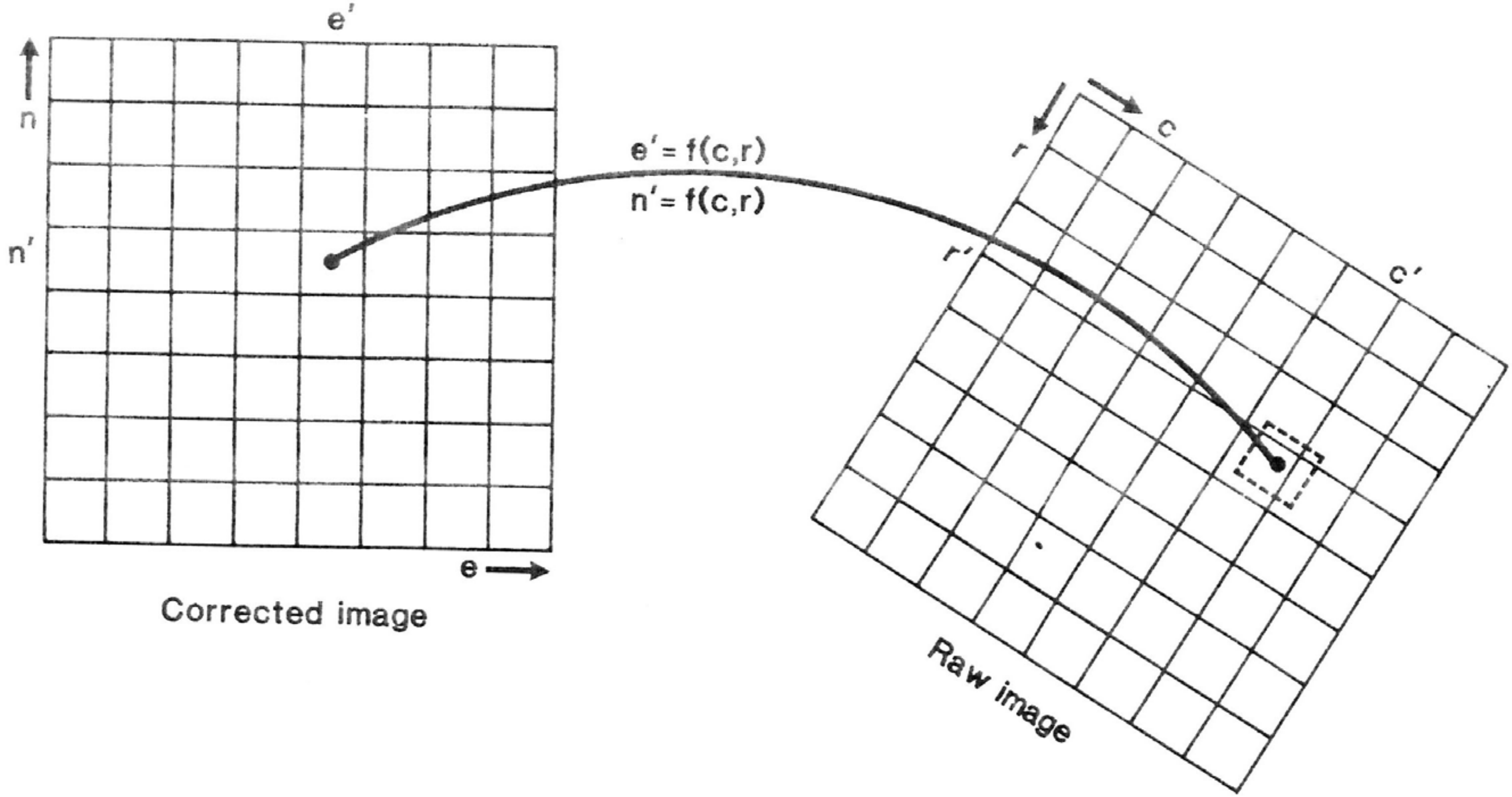
$$x = f_1(X, Y)$$
$$y = f_2(X, Y)$$

where

(x, y) = distorted image coordinates (column, row)

(X, Y) = correct (map) coordinates

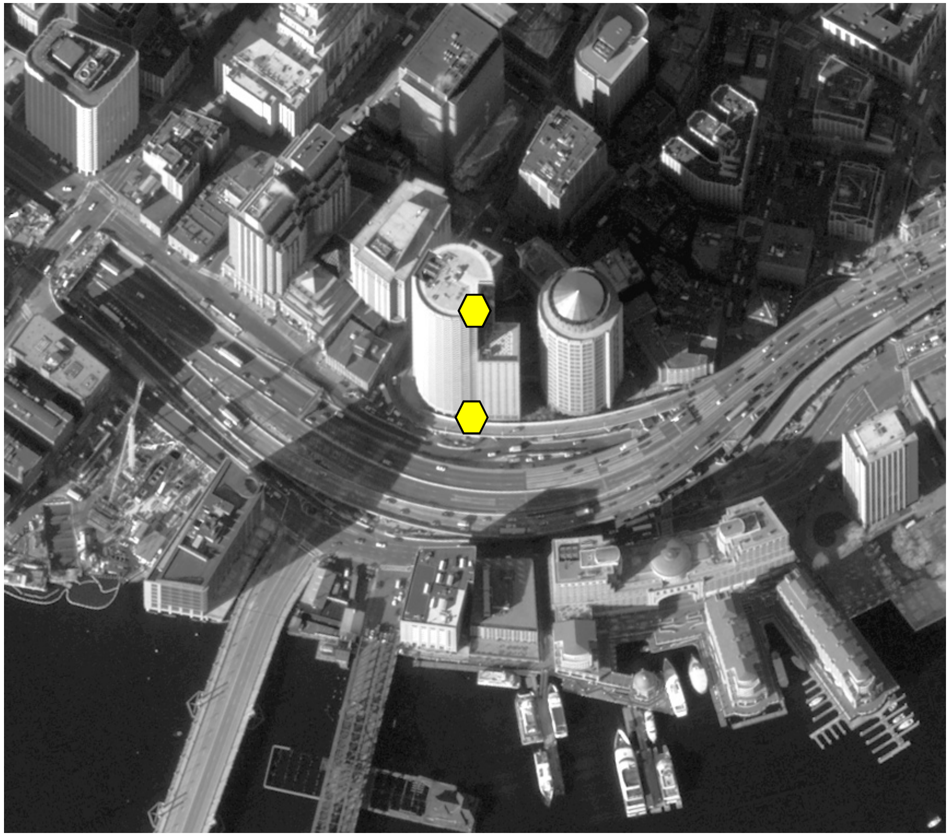
f_1, f_2 = transformation functions


e′
n
n′
e
Corrected image
e′ = f(c,r)
n′ = f(c,r)
r
c
r′
c′
Raw image

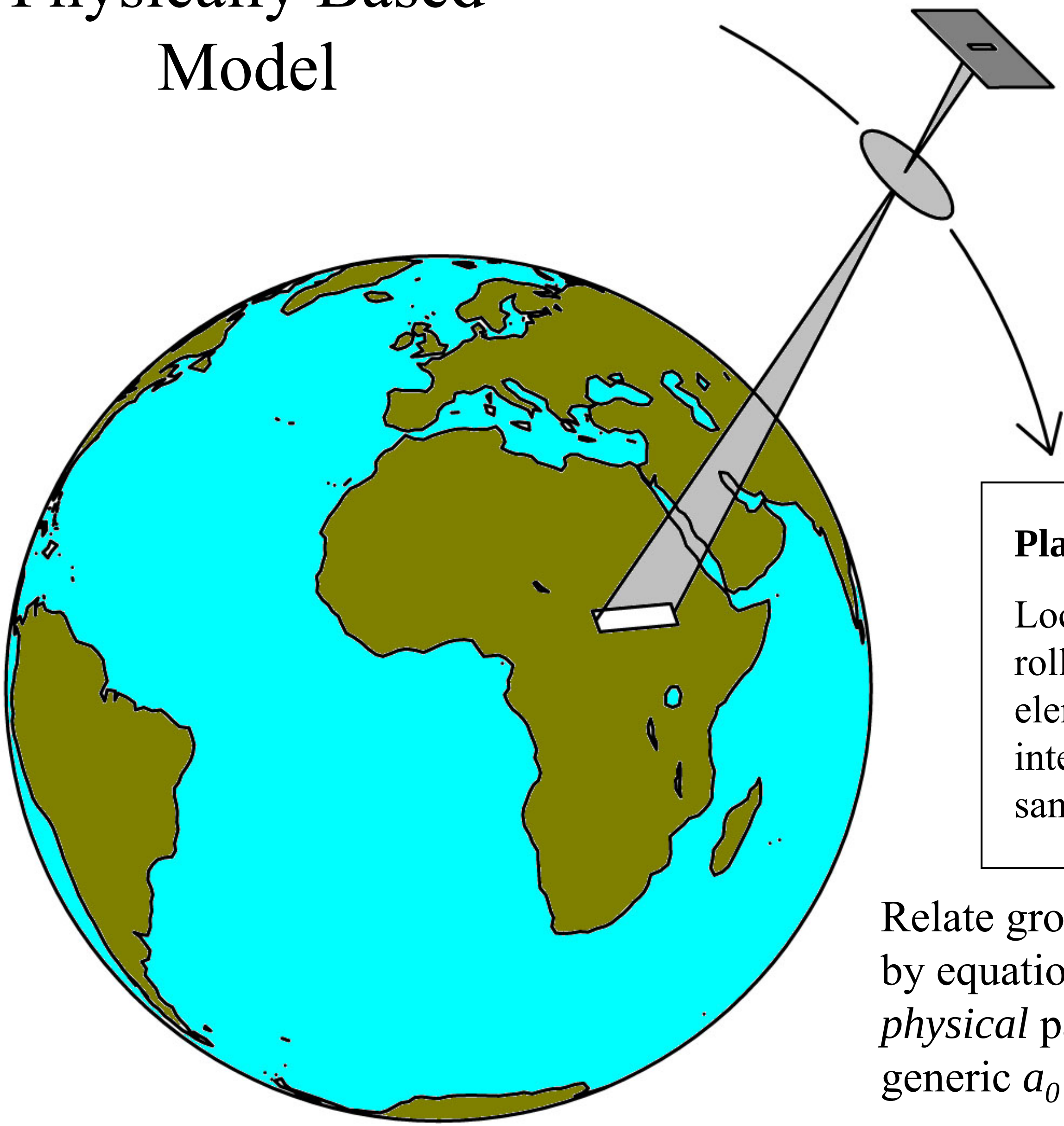
Mapping Polynomials or Rubber Sheeting

If the terrain is flat, the sensor has narrow field of view, the sensor is nadir looking, and the ground sample distance is large, then *you can get reasonable results using the approach of mapping polynomials.*

The accompanying Quickbird image (0.61m pixel) shows the pitfalls of mapping polynomials when the above conditions do not apply. The two marked points have the same XY and they would get mapped into the same (row, col), but clearly that is wrong. You could expand the polynomial by adding some Z-terms. *Modeling the actual physical imaging process is the best way. Otherwise you are the same polynomial to model sensor geometry, scan motion, platform motion, and terrain relief.*



Physically Based Model



Sensor parameters:

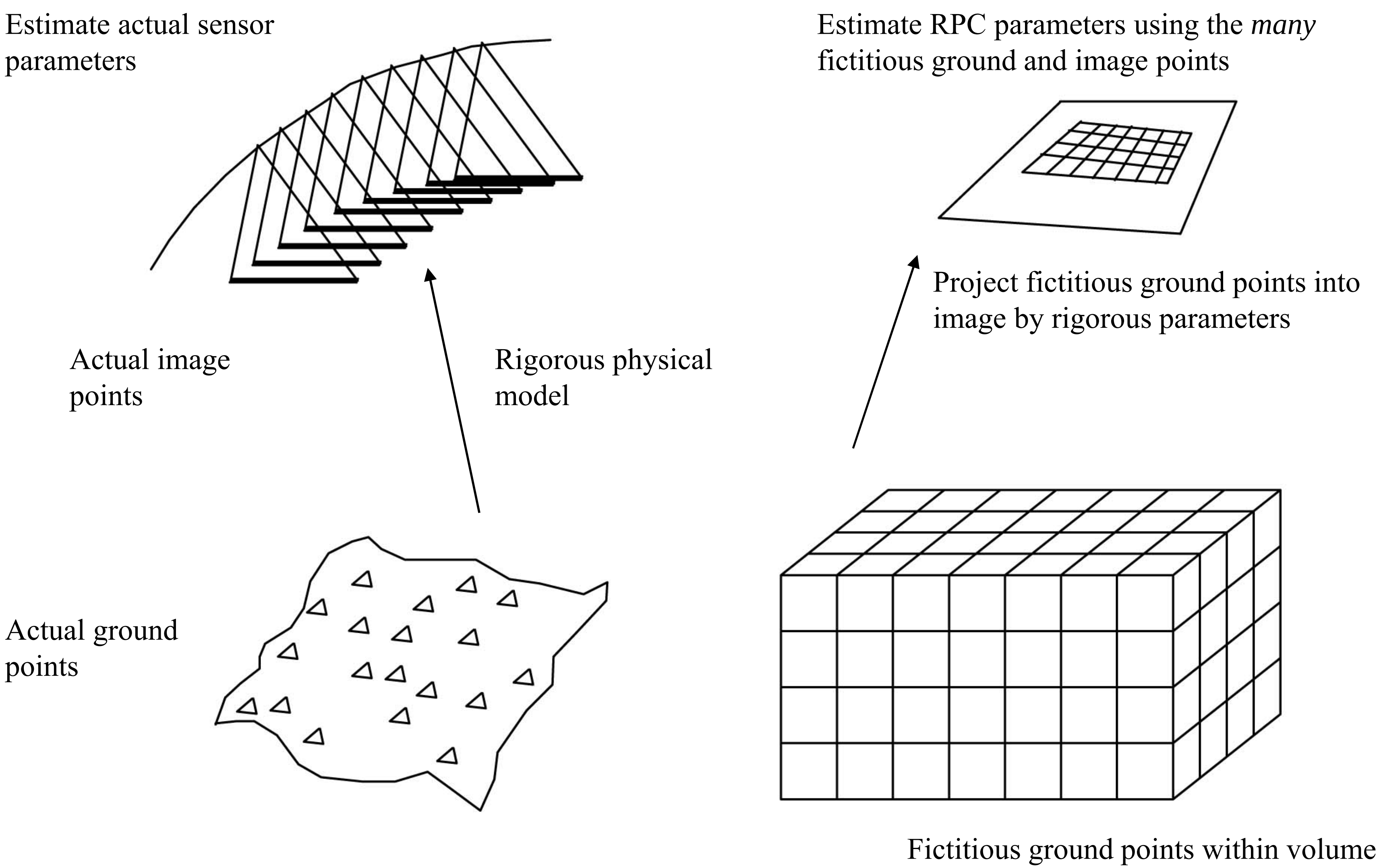
Focal length, principal point location, lens distortion, line rate, detector (pixel) size

Platform parameters:

Location X,Y,Z, time, attitude roll, pitch, yaw, kepler orbit elements (a, e, i, Ω, ω, ν) – or interpolate between densely sampled trajectory points.

Relate ground point and image point by equations with the above *actual physical* parameters, rather than the generic a_0 , a_1 , a_2 , … parameters.

Rigorous Sensor Model Parameter Estimation & RPC Parameter Estimation



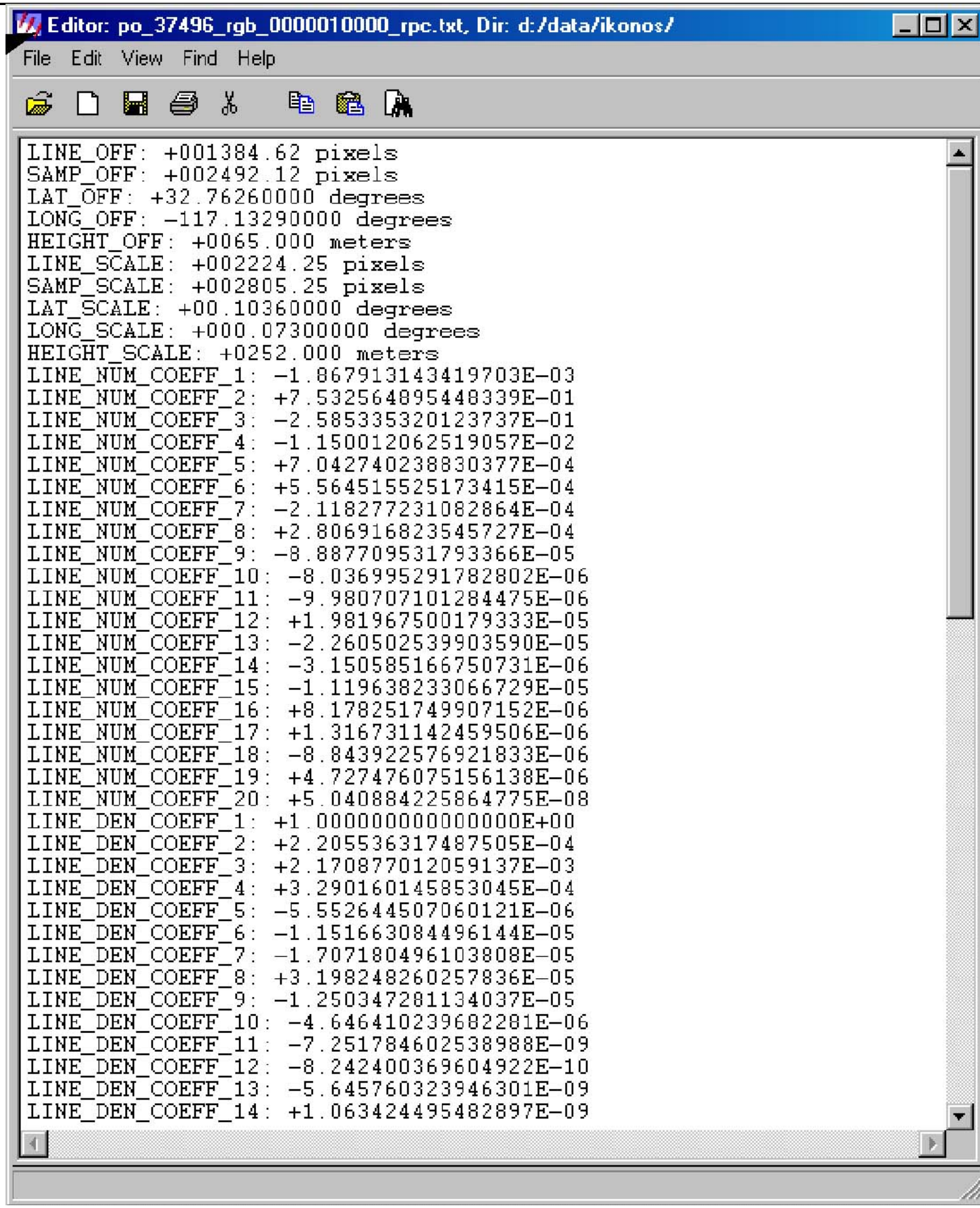
RPC Model

$$r = \frac{p1(X,Y,Z)}{p2(X,Y,Z)} = \frac{\sum_{i=0}^{m1}\sum_{j=0}^{m2}\sum_{k=0}^{m3} a_{ijk} X^i Y^j Z^k}{\sum_{i=0}^{n1}\sum_{j=0}^{n2}\sum_{k=0}^{n3} b_{ijk} X^i Y^j Z^k}$$

$$c = \frac{p3(X,Y,Z)}{p4(X,Y,Z)} = \frac{\sum_{i=0}^{m1}\sum_{j=0}^{m2}\sum_{k=0}^{m3} c_{ijk} X^i Y^j Z^k}{\sum_{i=0}^{n1}\sum_{j=0}^{n2}\sum_{k=0}^{n3} d_{ijk} X^i Y^j Z^k}$$

For the third order model, only terms with i+j+k <= 3 are allowed. Those terms are shown below.

$$1, x, y, z, x^2, y^2, z^2, xy, xz, yz, x^2y, xy^2, x^2z, xz^2, y^2z, yz^2, x^3, y^3, z^3, xyz$$




```
LINE_OFF: +001384.62 pixels
SAMP_OFF: +002492.12 pixels
LAT_OFF: +32.76260000 degrees
LONG_OFF: -117.13290000 degrees
HEIGHT_OFF: +0065.000 meters
LINE_SCALE: +002224.25 pixels
SAMP_SCALE: +002805.25 pixels
LAT_SCALE: +00.10360000 degrees
LONG_SCALE: +000.07300000 degrees
HEIGHT_SCALE: +0252.000 meters
LINE_NUM_COEFF_1: -1.867913143419703E-03
LINE_NUM_COEFF_2: +7.532564895448339E-01
LINE_NUM_COEFF_3: -2.585335320123737E-01
LINE_NUM_COEFF_4: -1.150012062519057E-02
LINE_NUM_COEFF_5: +7.042740238830377E-04
LINE_NUM_COEFF_6: +5.564515525173415E-04
LINE_NUM_COEFF_7: -2.118277231082864E-04
LINE_NUM_COEFF_8: +2.806916823545727E-04
LINE_NUM_COEFF_9: -8.887709531793366E-05
LINE_NUM_COEFF_10: -8.036995291782802E-06
LINE_NUM_COEFF_11: -9.980707101284475E-06
LINE_NUM_COEFF_12: +1.981967500179333E-05
LINE_NUM_COEFF_13: -2.260502539903590E-05
LINE_NUM_COEFF_14: -3.150585166750731E-06
LINE_NUM_COEFF_15: -1.119638233066729E-05
LINE_NUM_COEFF_16: +8.178251749907152E-06
LINE_NUM_COEFF_17: +1.316731142459506E-06
LINE_NUM_COEFF_18: -8.843922576921833E-06
LINE_NUM_COEFF_19: +4.727476075156138E-06
LINE_NUM_COEFF_20: +5.040884225864775E-08
LINE_DEN_COEFF_1: +1.000000000000000E+00
LINE_DEN_COEFF_2: +2.205536317487505E-04
LINE_DEN_COEFF_3: +2.170877012059137E-03
LINE_DEN_COEFF_4: +3.290160145853045E-04
LINE_DEN_COEFF_5: -5.552644507060121E-06
LINE_DEN_COEFF_6: -1.151663084496144E-05
LINE_DEN_COEFF_7: -1.707180496103808E-05
LINE_DEN_COEFF_8: +3.198248260257836E-05
LINE_DEN_COEFF_9: -1.250347281134037E-05
LINE_DEN_COEFF_10: -4.646410239682281E-06
LINE_DEN_COEFF_11: -7.251784602538988E-09
LINE_DEN_COEFF_12: -8.242400369604922E-10
LINE_DEN_COEFF_13: -5.645760323946301E-09
LINE_DEN_COEFF_14: +1.063424495482897E-09
```

Erdas Imagine / Orthobase support for IKONOS RPC data – note the line_numerator coefficients go up to #20, this implies a 3rd order polynomial

$$r = \frac{a_0 + a_1 x + a_2 y}{1 + c_1 x + c_2 y}$$

$$c = \frac{b_0 + b_1 x + b_2 y}{1 + c_1 x + c_2 y}$$

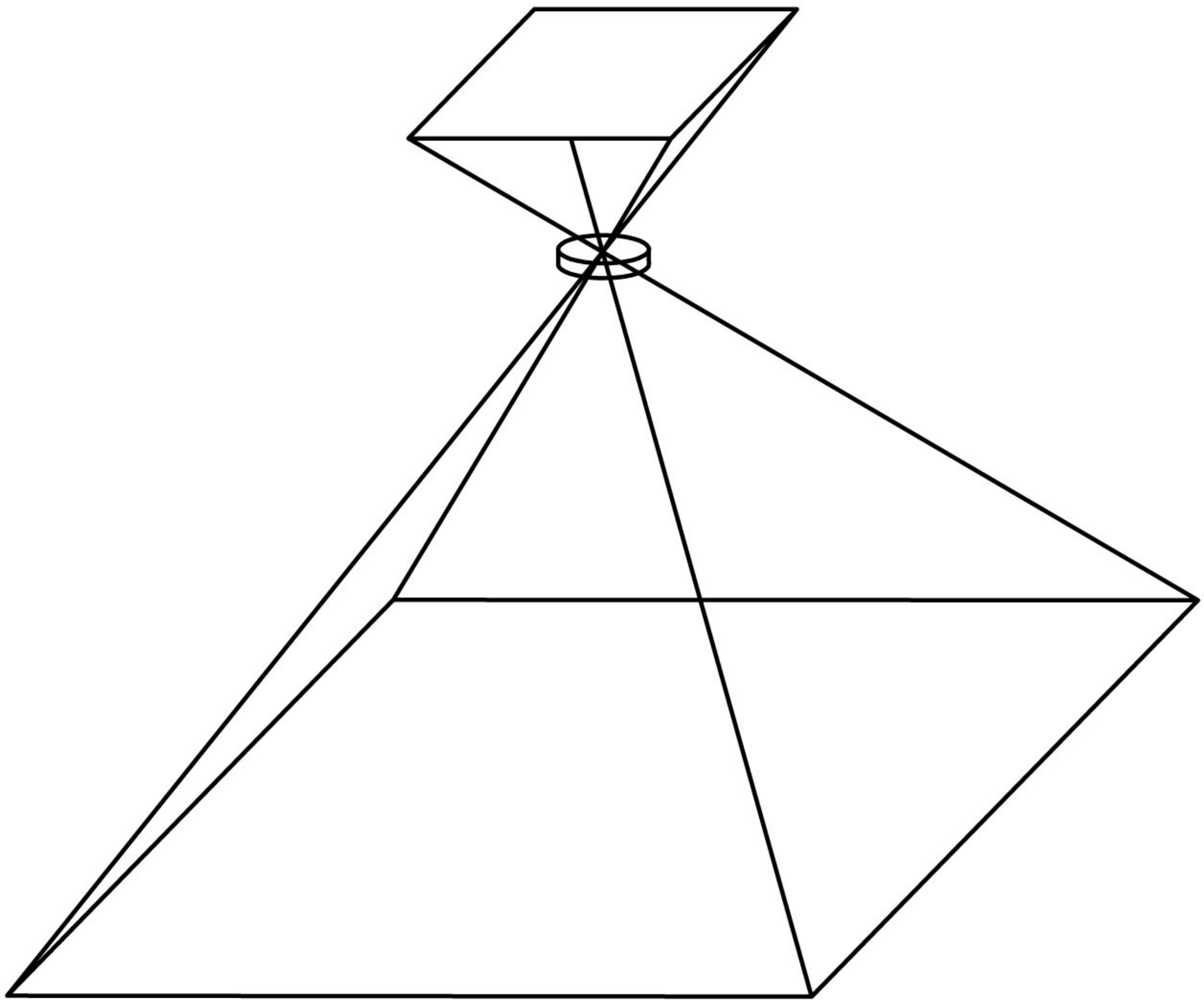
$$r + rc_1 x + rc_2 y = a_0 + a_1 x + a_2 y$$
$$c + cc_1 x + cc_2 y = b_0 + b_1 x + b_2 y$$

$$r = a_0 + a_1 x + a_2 y - rc_1 x - rc_2 y$$
$$c = b_0 + b_1 x + b_2 y - cc_1 x - cc_2 y$$

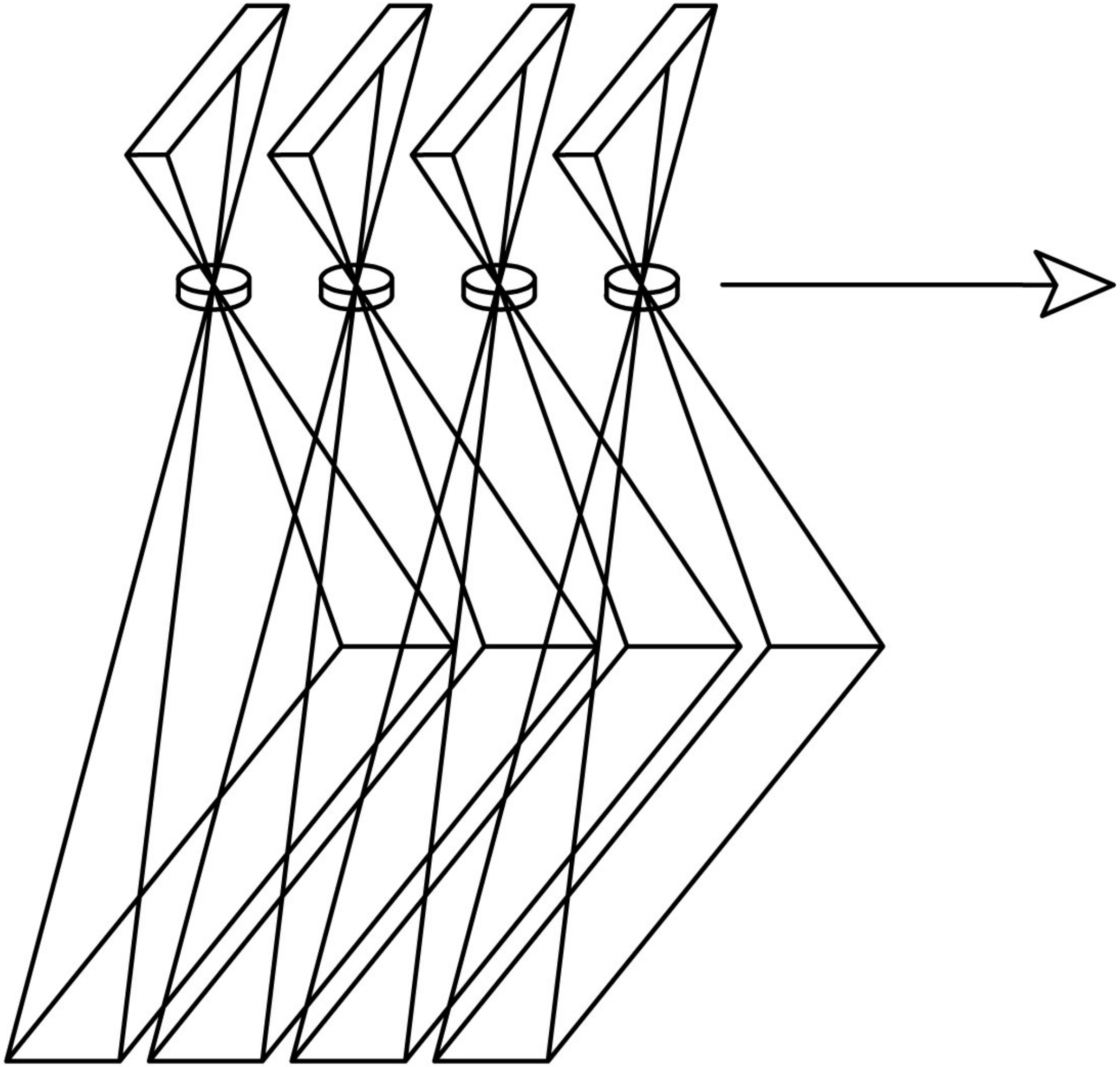
$$\begin{bmatrix} r \\ c \end{bmatrix} = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 & -rx & -ry \\ 0 & 0 & 0 & 1 & x & y & -cx & -cy \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ b_0 \\ b_1 \\ b_2 \\ c_1 \\ c_2 \end{bmatrix}$$

Show handling of low order rational polynomials as pseudo-linear problem. This is a good way to get approximations for the parameters, then final estimates can be obtained by rigorous non-linear estimation. Note that we will scale both the object and image coordinates into the range: -1 to +1.

Frame (Pinhole Camera) Sensor Geometry



Pushbroom Sensor Geometry



Panoramic Sensor Geometry