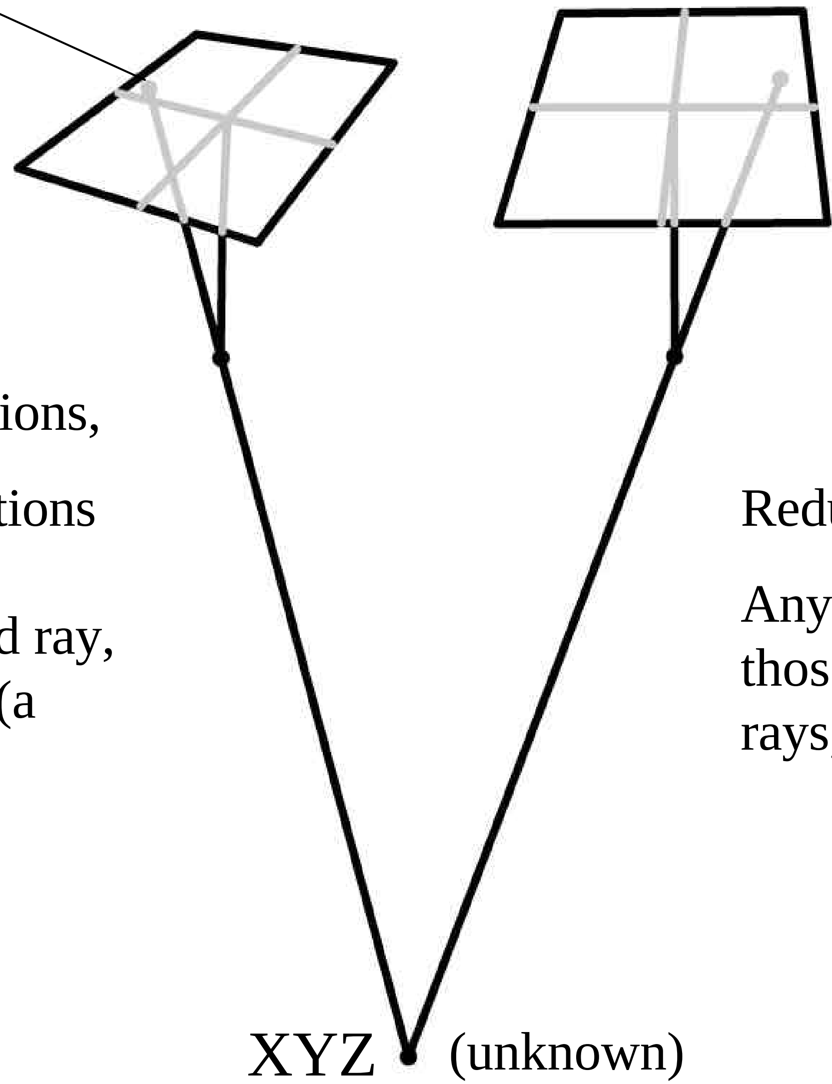
Space Intersection

Image coordinates, usually considered observations (with uncertainty)

Camera interior and exterior orientation, often considered as constants, can also be considered as observations (with uncertainty)



Unique Solution:

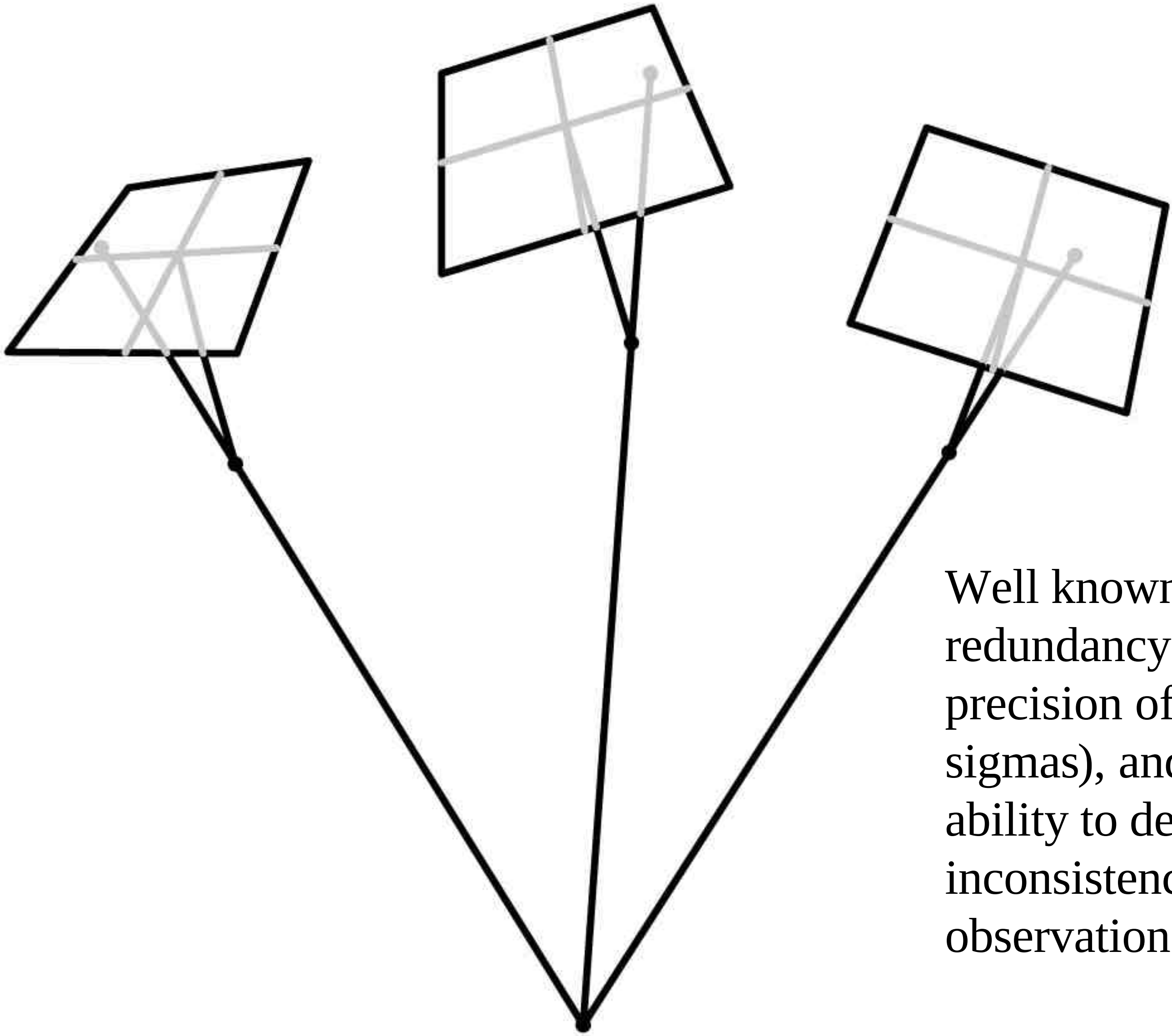
3 unknowns, 3 equations,

For example: 2 equations from first ray and 1 equation from second ray, or 1 ray and a plane (a ground plane, etc.)

Redundant Solution:

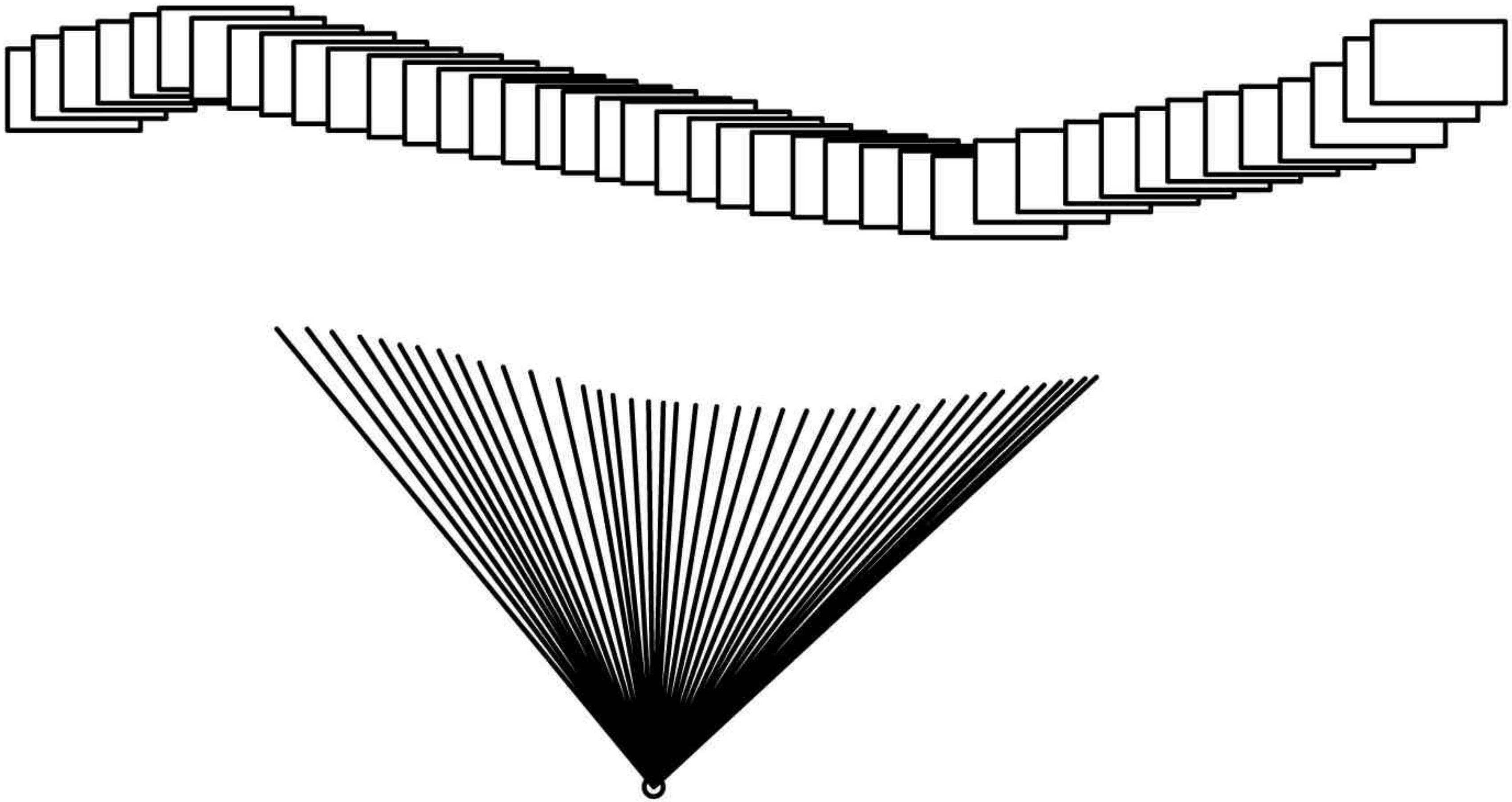
Anything in excess of those listed at left, i.e. 2 rays, 3 rays, …, n rays, etc.

Multi-Image Intersection



Well known benefits of redundancy: (a) increased precision of results (smaller sigmas), and (b) enhanced ability to detect blunders and inconsistencies among observations.

What about intersection from a many image sequence, for example from video frames? Can we drive uncertainty of the ground point to a negligible quantity? Probably not, if the errors include a non-random component (i.e. a bias) then increased redundancy will reach a point of diminishing returns, and the bias component will dominate



Development of the Collinearity Equations

$$\begin{bmatrix} x_c \\ y_c \\ -f \end{bmatrix} = l\mathbf{M}\begin{bmatrix} X - X_L \\ Y - Y_L \\ Z - Z_L \end{bmatrix} = l\begin{bmatrix} U \\ V \\ W \end{bmatrix}$$

dividing to remove scale factor,

$$x_c = (-f)\frac{m_{11}(X - X_L) + m_{12}(Y - Y_L) + m_{13}(Z - Z_L)}{m_{31}(X - X_L) + m_{32}(Y - Y_L) + m_{33}(Z - Z_L)}$$

$$y_c = (-f)\frac{m_{21}(X - X_L) + m_{22}(Y - Y_L) + m_{23}(Z - Z_L)}{m_{31}(X - X_L) + m_{32}(Y - Y_L) + m_{33}(Z - Z_L)}$$

or, written as condition equations,

$$F_x = x_c + f\frac{U}{W} = 0$$

$$F_y = y_c + f\frac{V}{W} = 0$$

Common Stochastic Assumptions for Intersection

$(x_c, y_c)_i$: refined image observations, image i

(X, Y, Z) : unknown ground point

$(X_L, Y_L, Z_L, w, j, k, f)$: constants

This can be solved as an indirect observation problem, with two equations per image

$$\mathbf{v} + \mathbf{B}\Delta = \mathbf{f}$$

$$\begin{bmatrix} v_x \\ v_y \end{bmatrix} + \begin{bmatrix} \frac{\partial F_x}{\partial X} & \frac{\partial F_x}{\partial Y} & \frac{\partial F_x}{\partial Z} \\ \frac{\partial F_y}{\partial X} & \frac{\partial F_y}{\partial Y} & \frac{\partial F_y}{\partial Z} \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix} = \begin{bmatrix} -F_x^0 \\ -F_y^0 \end{bmatrix}$$

For prototyping and fast development, use numerical approximations to partials,

$$\frac{\partial F}{\partial p} \approx \frac{F(p + \Delta p) - F(p)}{\Delta p}$$

What if we want to consider camera location and attitude as observations with uncertainty? Revise stochastic assumptions.

$$(x_c, y_c, w, j, k, X_L, Y_L, Z_L)_i : \text{observations, image i}$$

$$(X, Y, Z) : \text{unknown ground point}$$

$$f : \text{constant}$$

Now it becomes a general LS problem,

$$\mathbf{Av} + \mathbf{B\Delta} = \mathbf{f}$$

Still with two equations per image.

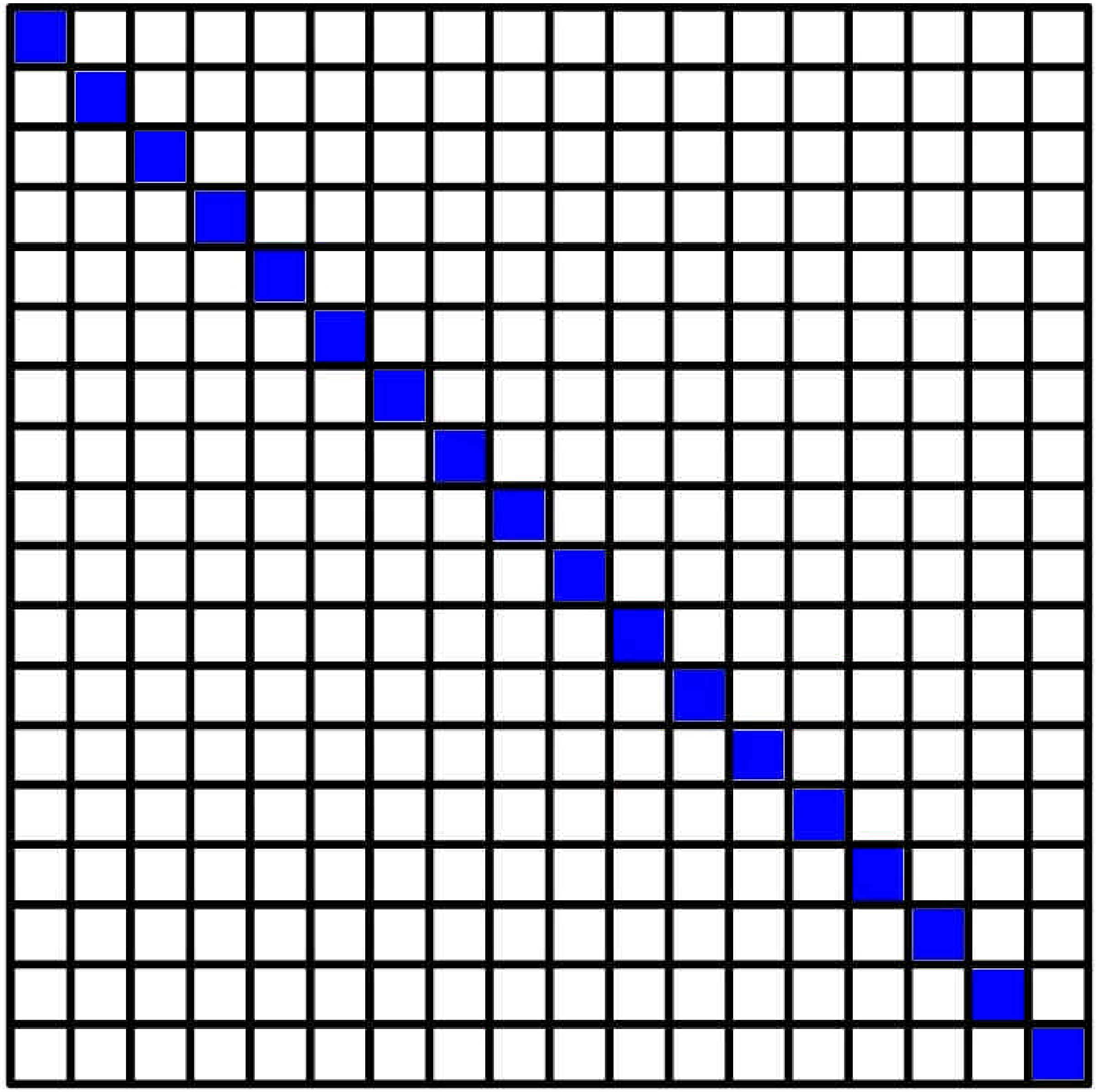
Writing out the matrix elements for contribution of one image

$$\begin{bmatrix} \frac{\partial F_x}{\partial x_c} & \frac{\partial F_x}{\partial y_c} & \frac{\partial F_x}{\partial w} & \frac{\partial F_x}{\partial j} & \frac{\partial F_x}{\partial k} & \frac{\partial F_x}{\partial X_L} & \frac{\partial F_x}{\partial Y_L} & \frac{\partial F_x}{\partial Z_L} \\ \frac{\partial F_y}{\partial x_c} & \frac{\partial F_y}{\partial y_c} & \frac{\partial F_y}{\partial w} & \frac{\partial F_y}{\partial j} & \frac{\partial F_y}{\partial k} & \frac{\partial F_y}{\partial X_L} & \frac{\partial F_y}{\partial Y_L} & \frac{\partial F_y}{\partial Z_L} \end{bmatrix} \begin{bmatrix} v_{xc} \\ v_{yc} \\ v_w \\ v_j \\ v_k \\ v_{XL} \\ v_{YL} \\ v_{ZL} \end{bmatrix} + \begin{bmatrix} \frac{\partial F_x}{\partial X} & \frac{\partial F_x}{\partial Y} & \frac{\partial F_x}{\partial Z} \\ \frac{\partial F_y}{\partial X} & \frac{\partial F_y}{\partial Y} & \frac{\partial F_y}{\partial Z} \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix} = \begin{bmatrix} -F_x \\ -F_y \end{bmatrix} - \mathbf{A}(\mathbf{l} - \mathbf{l_0})$$

The values in the weight matrix will govern how any misclosure, or failure of the rays to actually intersect, will be distributed among the corrections to each of the observations. That weight matrix often comes as a result of a prior bundle block adjustment.

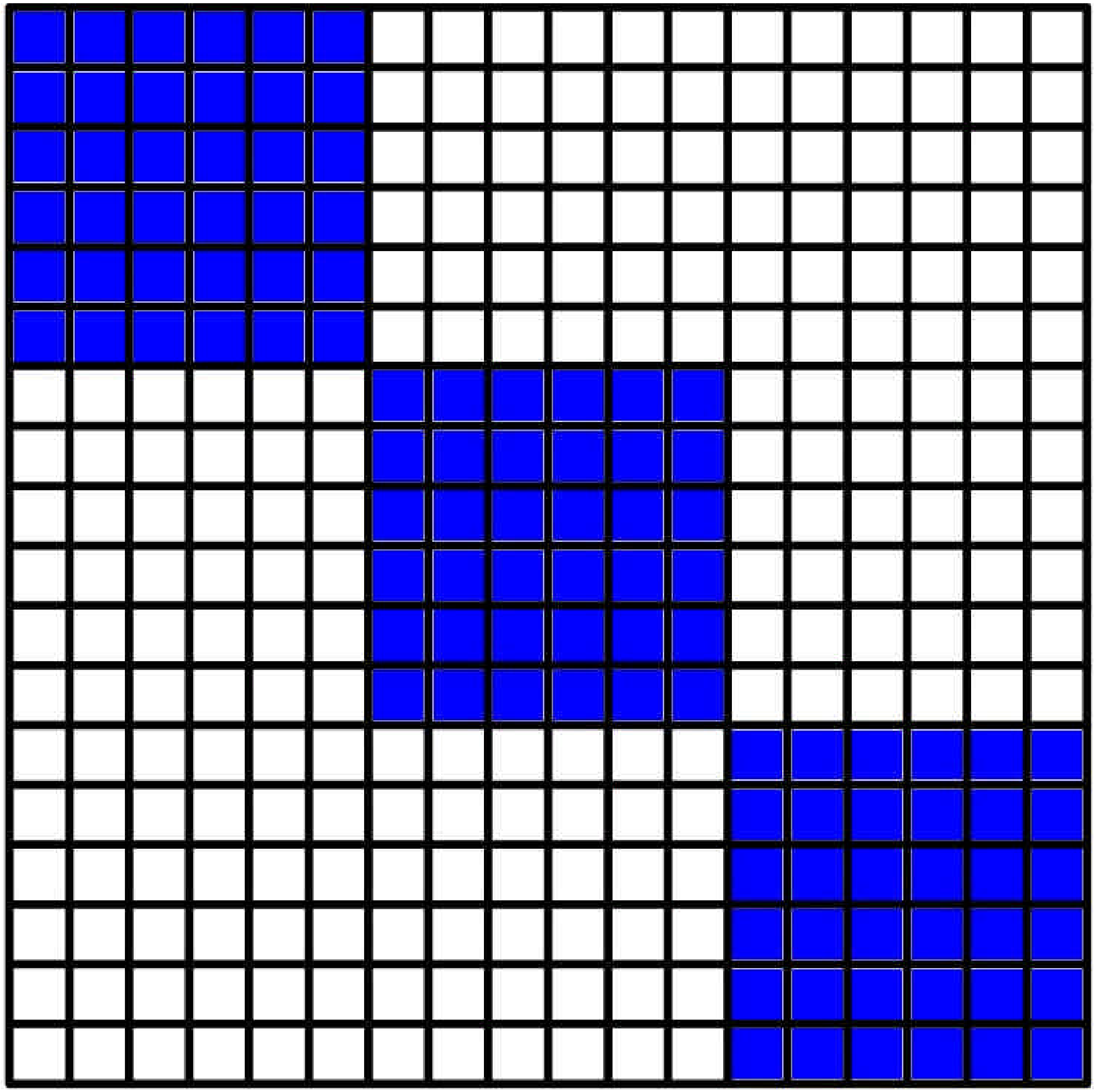
If you are lucky, your triangulation program may give you:

$\Sigma_{\mathbf{EO}} =$



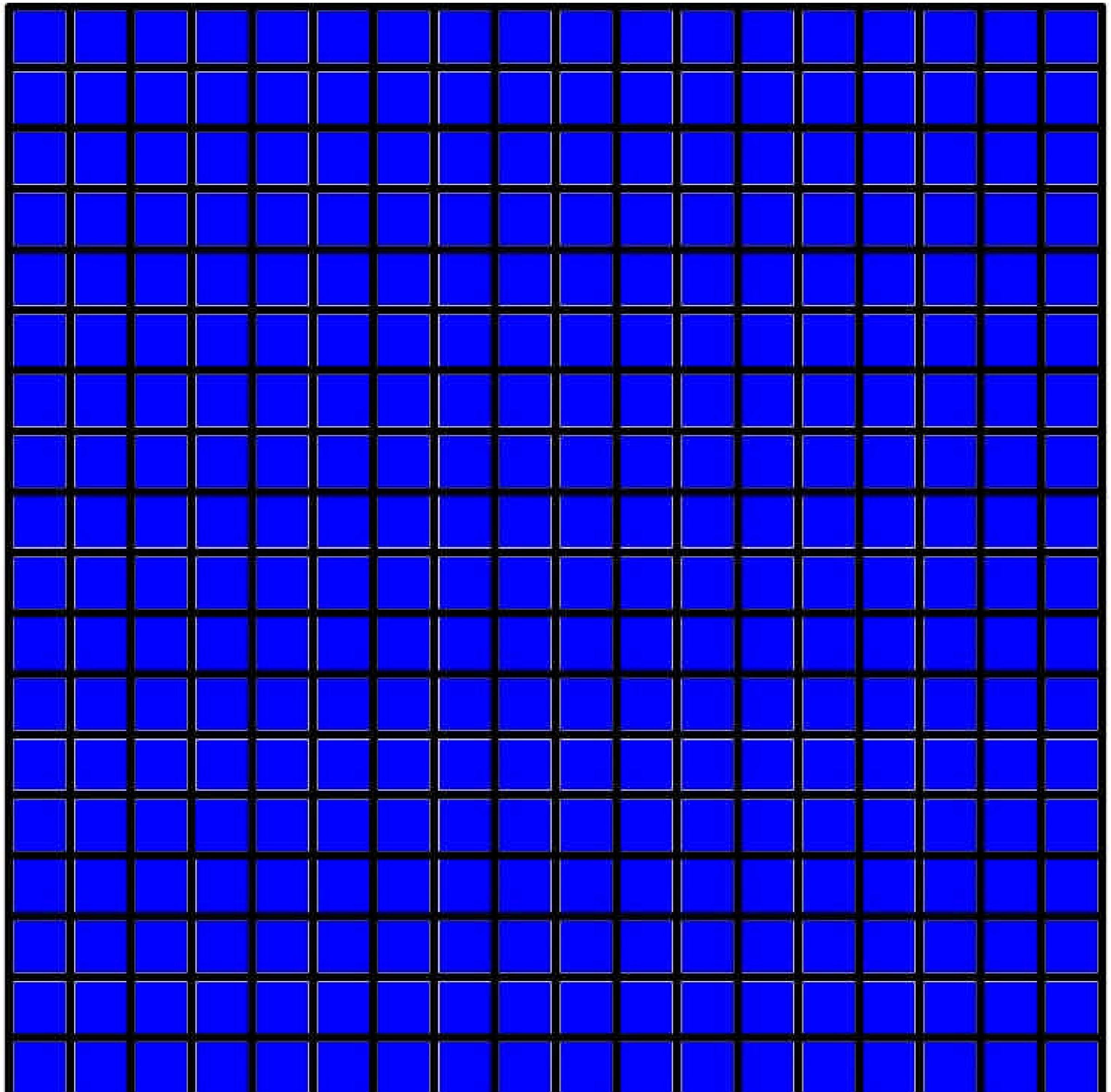
If you are very lucky, your triangulation program may give you:

$$\Sigma_{\mathbf{EO}} =$$



If you program it yourself then you can get:

$$\Sigma_{\mathbf{EO}} =$$



Prior estimation model was nonlinear – How to get initial approximations? Combine unknowns to produce a linear version of the equations which is *functionally* correct but *stochastically* incorrect.

$$\begin{bmatrix} x \\ y \\ -f \end{bmatrix} = l\mathbf{M}\begin{bmatrix} X - X_L \\ Y - Y_L \\ Z - Z_L \end{bmatrix}$$

$$\mathbf{M}^{\mathbf{T}}\begin{bmatrix} x \\ y \\ -f \end{bmatrix} = l\begin{bmatrix} X - X_L \\ Y - Y_L \\ Z - Z_L \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = l\begin{bmatrix} X - X_L \\ Y - Y_L \\ Z - Z_L \end{bmatrix}$$

$$c_1 = \frac{u}{w} = \frac{X - X_L}{Z - Z_L}$$

$$c_2 = \frac{v}{w} = \frac{Y - Y_L}{Z - Z_L}$$

$$c_1(Z - Z_L) = X - X_L$$
$$c_2(Z - Z_L) = Y - Y_L$$

$$c_1 Z - X = -X_L + c_1 Z_L$$
$$c_2 Z - Y = -Y_L + c_2 Z_L$$

Linear Version of Intersection Equations

$$\begin{bmatrix} -1 & 0 & c_1 \\ 0 & -1 & c_2 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} -X_L + c_1 Z_L \\ -Y_L + c_2 Z_L \end{bmatrix}$$

- 2 “linear” equations per image
- 2n “linear” equations for n images
- But, matrix elements (c_i) are not constants, and
- Elements of right-hand side vector are not observations
- Therefore “Least Squares” is really pseudo least squares
- However if data is reasonably good then it works well enough to generate good initial approximations. Then nonlinear model with proper stochastic assignment can be iterated to convergence.
- We see this strategy on several occasions – use linear model to bootstrap yourself into the nonlinear model without agonizing over approximations (8-parameter transformation, DLT, etc.)