(1 hour) CE 503 Photogrammetry 1 22 Oct 03

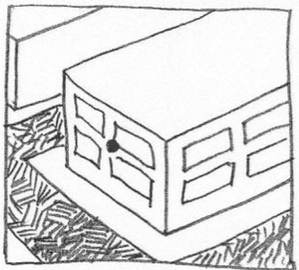
(1 page of notes allowed)

Exam I

____

Name

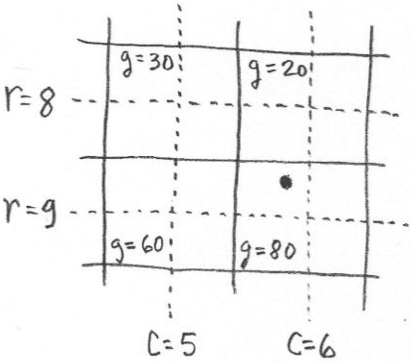
[1.]



frame photograph

You are given X_L, Y_L, Z_L, the orientation matrix, M, and the principal point and focal length, x_0, y_0, f. At the indicated dot, we measure the image coordinates, (x, y) in the fiducial system of a window corner on the building. We know the plane of the building face is at $Y = Y_B$. Show how you would find the object space coordinates (X, Y, Z) of that point.

[2.]



The indicated point for resampling falls at a fractional pixel location in an image.

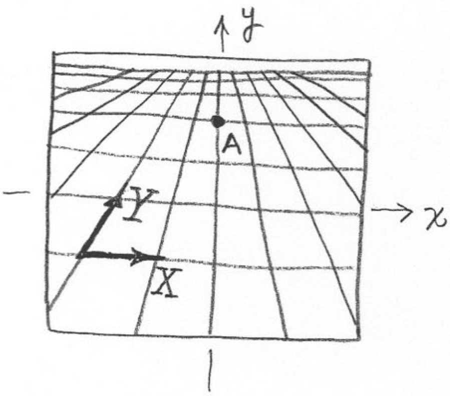
$r = 8.6$

$c = 5.8$

The gray values for the surrounding pixels are given. Interpolate a gray value by (a) nearest neighbor, and (b) bilinear

(Note: whole number coordinates fall at center of pixel)

[3.]

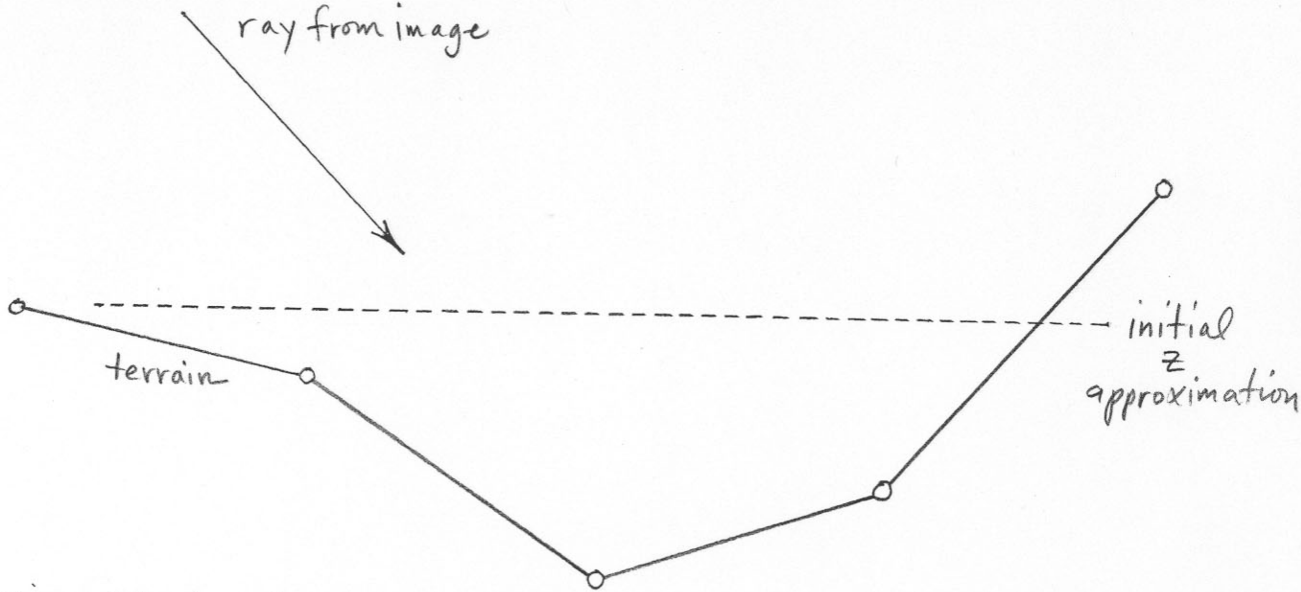


The ground coordinates (X, Y) are related to the image coordinates (x, y) of an oblique image by,

$$x = \frac{a_0 + a_1 X + a_2 Y}{1 + c_1 X + c_2 Y}$$

$$y = \frac{b_0 + b_1 X + b_2 Y}{1 + c_1 X + c_2 Y}$$

Derive an expression for (a) the scale in the x-direction at A, and (b) the scale in the y-direction at A



4. Five points in a terrain profile are shown with linear interpolation between the points. The initial z approximation is shown by the dashed (----) line. A ray projected from an image comes from the upper left. Sketch the sequence of intermediate points as we iterate to find the ray-terrain intersection. Label by sequence number.

CE 503 Exam I Solution 22-Oct-03

1. $$\begin{bmatrix} x-x_0 \\ y-y_0 \\ -f \end{bmatrix} = \lambda M \begin{bmatrix} X-X_L \\ Y-Y_L \\ Z-Z_L \end{bmatrix}$$

First, we must rearrange the equations,

$$\frac{1}{\lambda} M^T \begin{bmatrix} x-x_0 \\ y-y_0 \\ -f \end{bmatrix} = \begin{bmatrix} X-X_L \\ Y-Y_L \\ Z-Z_L \end{bmatrix}$$

let's call the product,

$$M^T \begin{bmatrix} x-x_0 \\ y-y_0 \\ -f \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

rewrite: $$\frac{1}{\lambda} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} X-X_L \\ Y-Y_L \\ Z-Z_L \end{bmatrix}$$

since Y is known, divide 1st and 3rd equations by the 2nd

$$\left.\begin{aligned} \frac{u}{v} &= \frac{X-X_L}{Y-Y_L} \\ \frac{w}{v} &= \frac{Z-Z_L}{Y-Y_L} \end{aligned}\right\} \text{now solve for } X \,\&\, Z$$

$$\boxed{\begin{aligned} X &= X_L + (Y-Y_L)\, u/v \\ Z &= Z_L + (Y-Y_L)\, w/v \end{aligned}}$$

(note: Y was given as Y_B)

---

2. (a) nearest neighbor (by inspection) $\boxed{g = 80}$

(b) bilinear: first interpolate in horizontal direction, twice

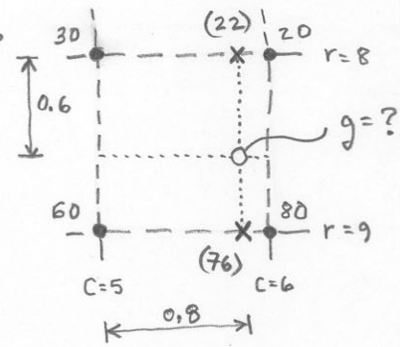
upper X: $.8 \times 20 + (1-.8) \times 30 = 22$

lower X: $.8 \times 80 + (1-.8) \times 60 = 76$

now interpolate in the vertical direction,

O: $.6 \times 76 + (1-.6) \times 22 = 54.4$

round it to $\boxed{g_{\text{new point}} = 54}$



---

3. In general, $\boxed{\text{Scale} = \dfrac{\text{image length}}{\text{object length}}}$ usual interpretation requires same units for both lengths

In vertical image, scale is the same in any direction, In an oblique image the scale is different in each direction

for point A x parallel with X, and y parallel with Y (on the sketch)

$\Rightarrow \text{Scale}_x = \dfrac{\Delta x}{\Delta X}$ and $\text{Scale}_y = \dfrac{\Delta y}{\Delta Y}$ If we have a known object we can measure $\Delta x, \Delta X, \Delta y, \Delta Y$

or we can get by calculus,

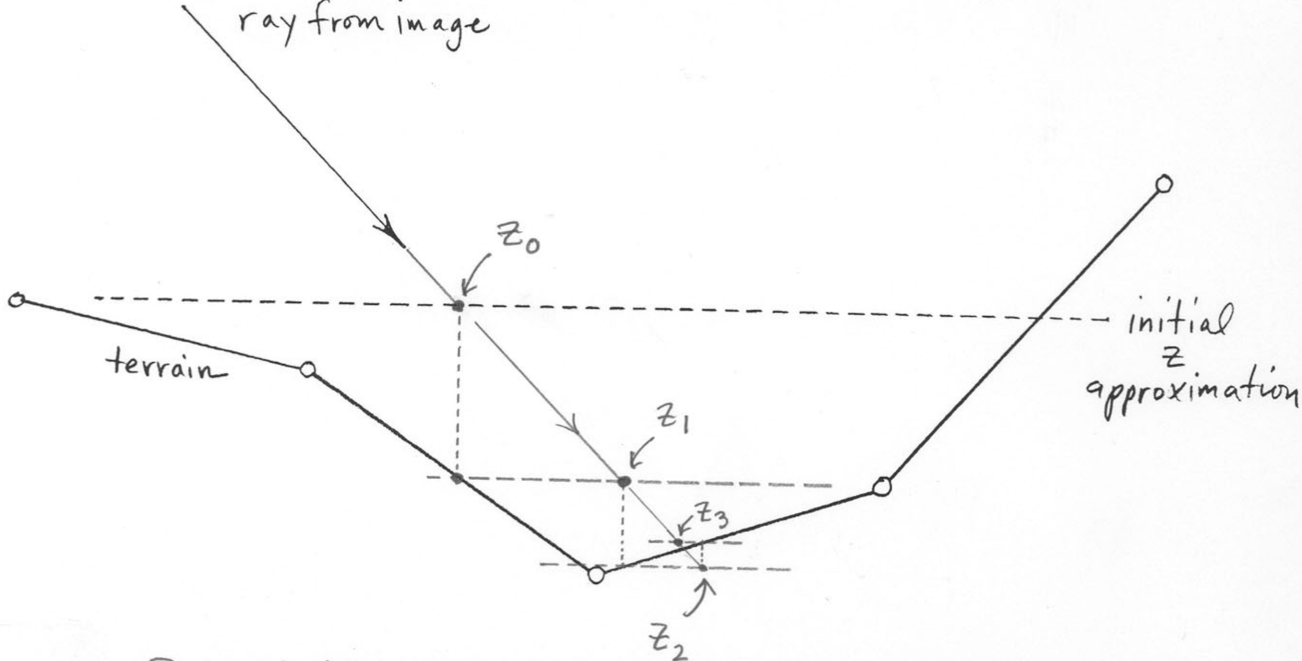
$$\frac{\Delta x}{\Delta X} \approx \frac{\partial x}{\partial X} = \frac{(1+c_1X+c_2Y)\,a_1 - (a_0+a_1X+a_2Y)\,c_1}{(1+c_1X+c_2Y)^2}$$

$$\frac{\Delta y}{\Delta Y} \approx \frac{\partial y}{\partial Y} = \frac{(1+c_1X+c_2Y)\,b_2 - (b_0+b_1X+b_2Y)\,c_2}{(1+c_1X+c_2Y)^2}$$

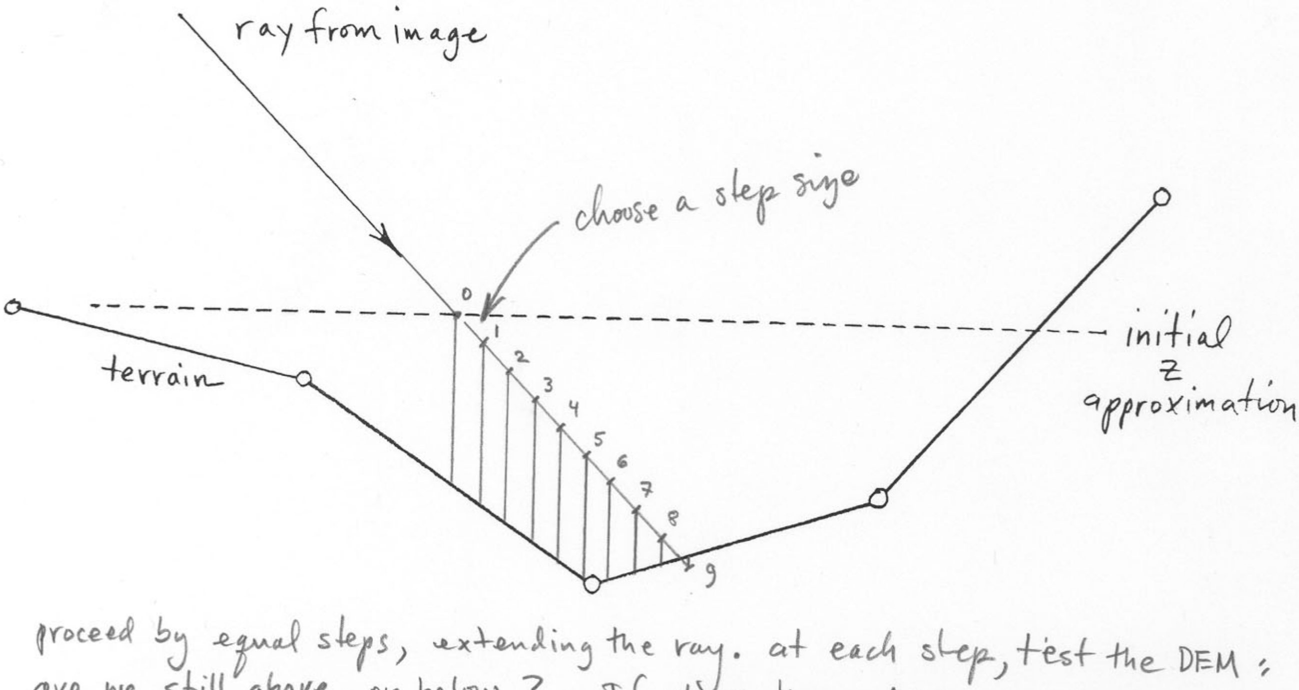
assume: transf. parameters use same units for image and object space

use: $$\frac{d}{dx}\frac{u}{v} = \frac{v\,du - u\,dv}{v^2}$$

4. By the algorithm given in class for homework #4,



By the algorithm in Ch. 5 of text (p. 114)



proceed by equal steps, extending the ray. at each step, test the DEM: are we still above, or below? If above keep going, If below then you have just passed through the terrain surface – now refine as in the first algorithm