Geometric Transformation

8.1.1 The Spatial Transformation

general form $g(x,y) = f(x',y') = f[a(x,y),\ b(x,y)]$

issues: continuity & connectivity of objects in image

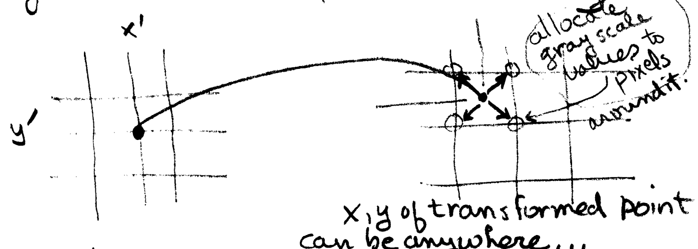
8.1.2. Gray level interpolation

necessary because moving pixels tends to stretch and/or compress them.

integer input pixels map to fractional (non integer) coordinates

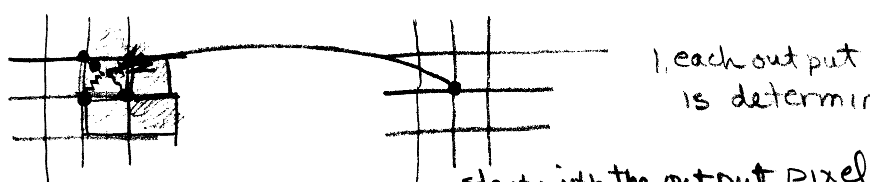
8.1.3. pixel carry over (forward mapping) approach.

transform input pixel to output and split up among four output pixels according to interpolation rule



problems
1. pixels mapping to locations outside image
2. multiple addressing of output pixels.
3. missing of output pixels

pixel-filling (backward mapping) approach



1. each output pixel is determined.

generate output pixels one by one

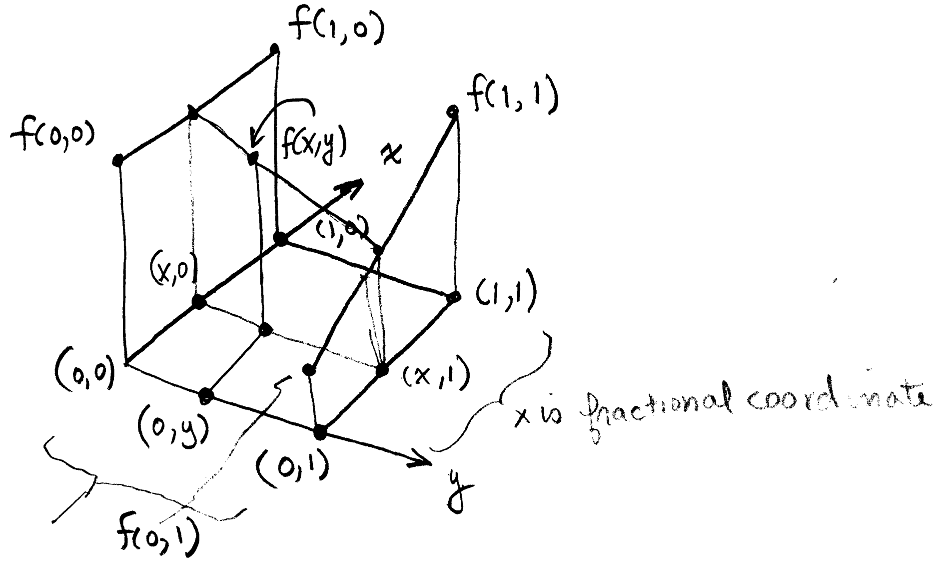
8.2 gray level interpolation

8.2.1. nearest neighbor

gray level of output is that of nearest pixel in input image

can produce edge artifacts where gray levels change rapidly

8.2.2. Bilinear interpolation



can't fit plane through four points

fit hyperbolic paraboloid $f(x,y) = ax + by + cxy + d$

fit to values at each corner by simple algorithm

1) linearly interpolate between upper two points

$$f(x,0) = f(0,0) + x[f(1,0) - f(0,0)] \qquad (1)$$

2) linearly interpolate between lower two points

$$f(x,1) = f(0,1) + x[f(1,1) - f(1,0)] \qquad (2)$$

3) interpolate vertically

$$f(x,y) = f(x,0) + y[f(x,1) - f(x,0)]. \qquad (3)$$

Combine all 3 equations

$$f(x,y) = [f(1,0) - f(0,0)]x + [f(0,1) - f(0,0)]y + [f(1,1) + f(0,0) - f(0,1) - f(1,0)]xy + f(0,0)$$

5 additions
+4 multiplications
+3 additions

8 additions plus 4 ~~additions~~ multiplications efficient

surfaces produced by bilinear interpolation match in amplitude at neighborhood boundaries, but do not match in slope, $\Rightarrow$ generated surface is continuous but derivatives are discontinuous at boundaries

8.2.3. Higher order interpolation

bilinear gray level interpolation
- smooths image losing fine level detail
- slope discontinuities may cause undesirable effects

higher order functions
- cubic splines
- Legendre functions
- $\sin(\alpha x)/\alpha x$

8.3 Spatial transformation

transforms x and y to x', y'

$$g(x,y) = f(x', y') = f[a(x,y),\ b(x,y)]$$

if $\underline{a(x,y) = x}$ $\quad$ $\underline{b(x,y) = y}$ $\Rightarrow$ identity operation

if $a(x,y) = x + x_0$ $\quad$ $b(x,y) = y + y_0$

translates point (x_0, y_0) to origin by amount $\sqrt{x_0^2 + y_0^2}$

<u>homogeneous coordinates</u>

new coords.

$$\begin{bmatrix} a(x,y) \\ b(x,y) \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$\left.\begin{aligned} a(x,y) &= x + x_0 \\ b(x,y) &= y + y_0 . \end{aligned}\right\}$ translation

extra coordinate

new coords.

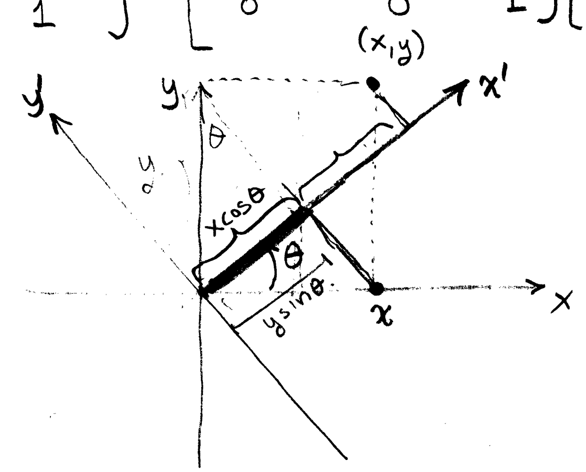
$$\begin{bmatrix} a(x,y) \\ b(x,y) \\ 1 \end{bmatrix} = \begin{bmatrix} 1/c & 0 & 0 \\ 0 & \frac{1}{d} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$\left.\begin{aligned} a(x,y) &= \frac{x}{c} \\ b(x,y) &= \frac{y}{d} \end{aligned}\right\}$ scaling

can do reflections, rotations, etc,

$$\begin{bmatrix} a(x,y) \\ b(x,y) \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

rotation θ about origin



$$x' = a(x,y) = x\cos\theta - y\sin\theta$$

$$y' = b(x,y) = x\sin\theta + y\cos\theta .$$

homogeneous coordinates make it easy to do compound transformations

$$\begin{bmatrix} a(x,y) \\ b(x,y) \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

translate origin to (x_0, y_0) | rotate θ about (x_0, y_0) | translate origin back to original origin

$\longrightarrow$ sequence of operations

Separable implementation

$$a(x,y) = x\cos\theta - y\sin\theta \qquad (13)$$

$$b(x,y) = x\sin\theta + y\cos\theta \qquad (14)$$

solve for $x = \dfrac{a(x,y) + y\sin\theta}{\cos\theta}$ (17)

substitute into (14)

$$b(x,y) = \frac{a(x,y)\sin\theta + y}{\cos\theta} \qquad (18)$$

(1) do first transform (compresses features in x).

Use (13) $\left.\begin{aligned} a(x,y) &= x\cos\theta - y\sin\theta \\ b(x,y) &= y. \end{aligned}\right\}$ transform only x which is linear

intermediate image

(2) then do $a(x,y) = x$ (expand features in y)

$$b(x,y) = \frac{a(x,y)\sin\theta + y}{\cos\theta}$$

might be more efficient

Bilinear interpolation

$$G(x,y) = F(x',y') = F(ax+by+cxy+d,\ ex+fy+gxy+h)$$

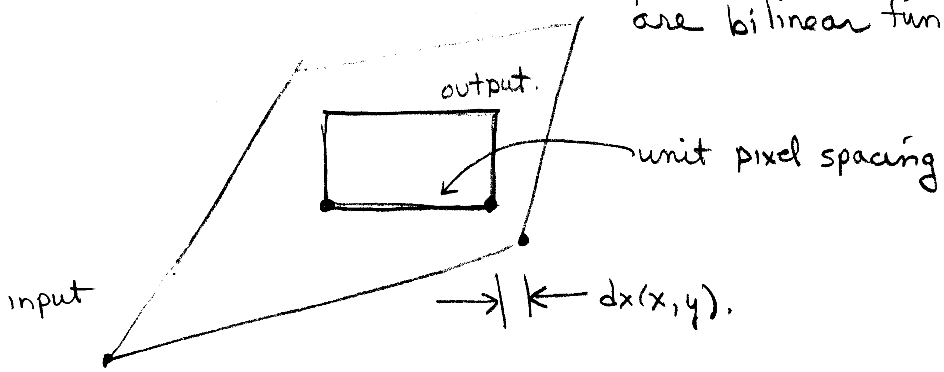
transform defined by a through h

if a quadrilateral maps to a quadrilateral
specifying vertices gives 2 sets of 4 linear
equations in four unknowns.

Better way to define it

$$G(x,y) = F\left[x + dx(x,y),\ y + dy(x,y)\right].$$

pixel displacements that
are bilinear functions of x and y.



dx (x,y) & dy (x,y) bilinear in x & y.
$\Rightarrow$ linear in x along each horizontal line in output

for each output line $\quad dx(x+1,y) = dx(x,y) + \Delta x$

where Δx varies for each line

8-4 Applications of Geometric operations

8.4.1. Geometric calibration

remove camera induced geometric distortion from images

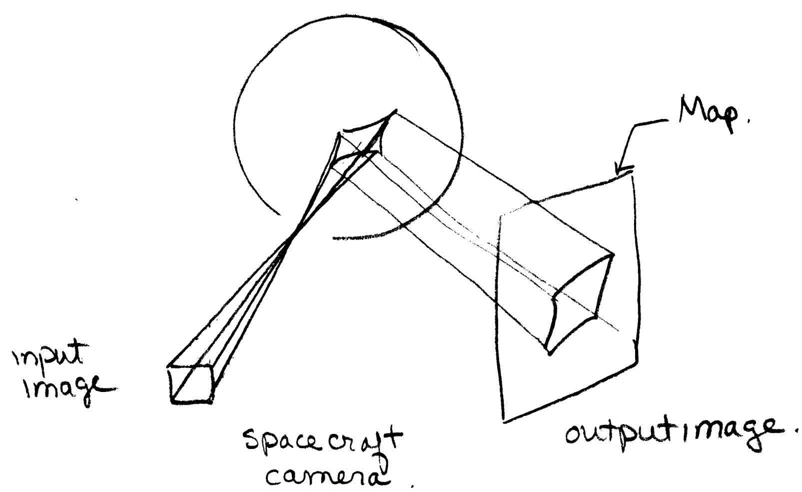
8.4.2. Image Rectification – transform into rectangular pixel coordinates

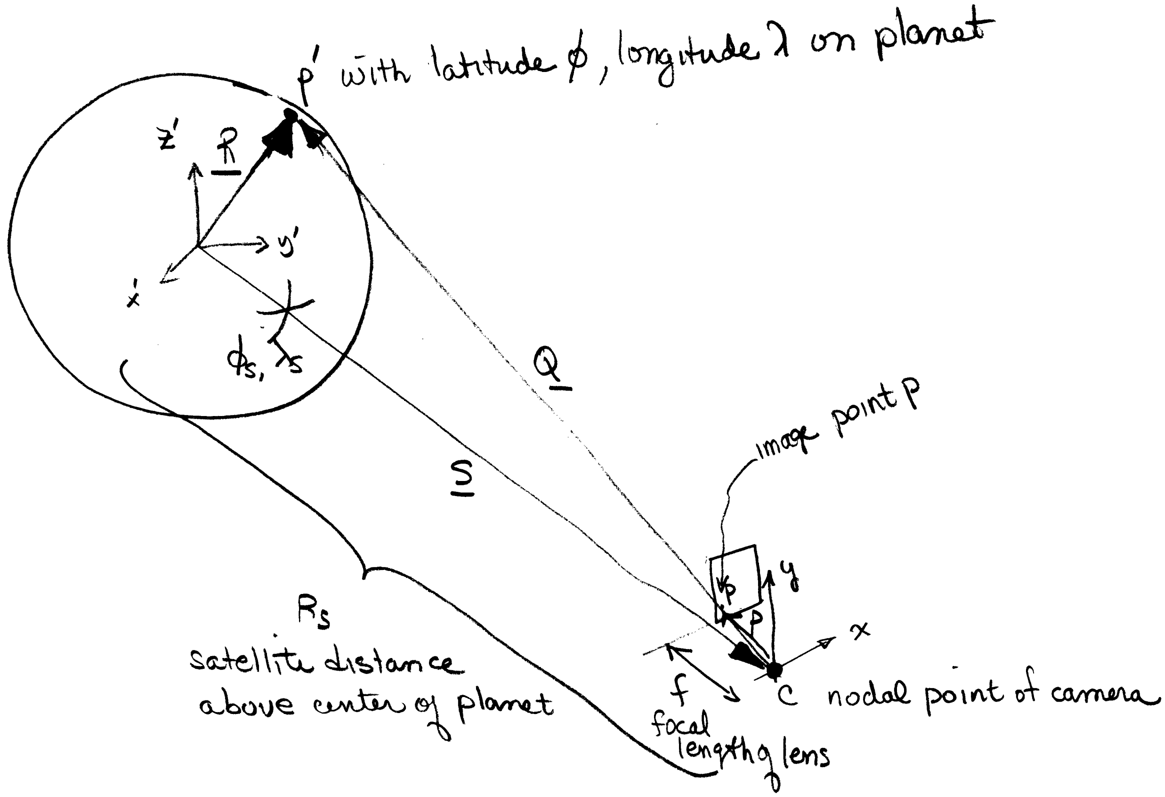
fisheye lens – 5th order polynomial warp.
implemented in polar coordinates

8.4.3. Image Registration

8.4.4. Image Format Conversion

8.4.5 Map Projection





satellite above latitude ϕ_s, longitude λ_s

$\underline{P} = \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix}$ camera pixel position coordinates

$\underline{P}$ & $\underline{Q}$ are collinear $\Rightarrow$ $\underline{P} = \underbrace{\frac{f}{Q_z}}_{\text{scale factor}} \underline{Q}$

implies

$$x_p = \frac{f}{Q_z} Q_x$$

$$y_p = \frac{f}{Q_z} Q_y$$

$\underline{Q} = \underline{R} - \underline{S}$

$\underline{R}$ position of object on planet in planet coordinates

$\underline{S}$ position of camera in planet coordinates.

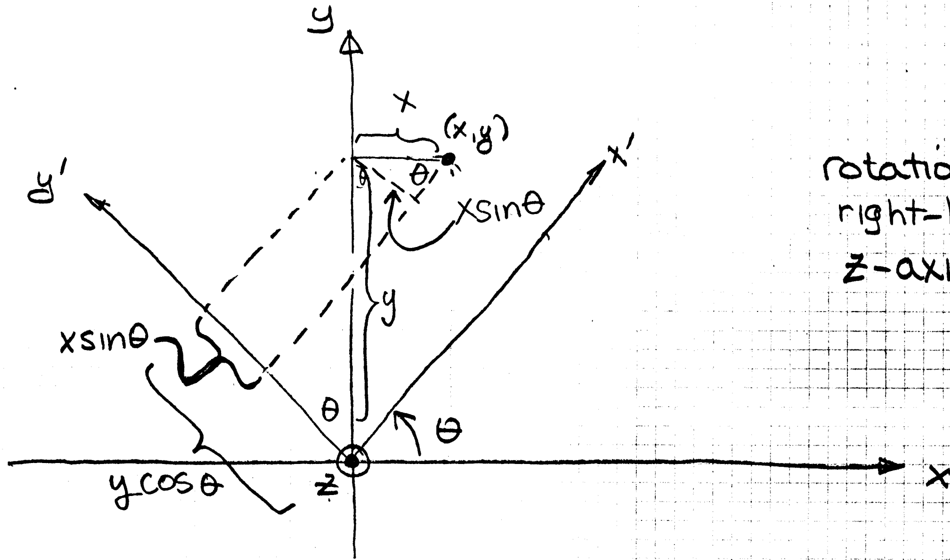
3x3 transform matrix

$$\begin{bmatrix} Q_x \\ Q_y \\ Q_z \end{bmatrix} = [m] \begin{bmatrix} R\cos\phi\cos\lambda - R_s\cos\phi_s\cos\lambda_s \\ R\cos\phi\sin\lambda - R_s\cos\phi_s\sin\lambda_s \\ R\sin\phi - R_s\sin\phi_s \end{bmatrix}.$$

in camera coordinate frame

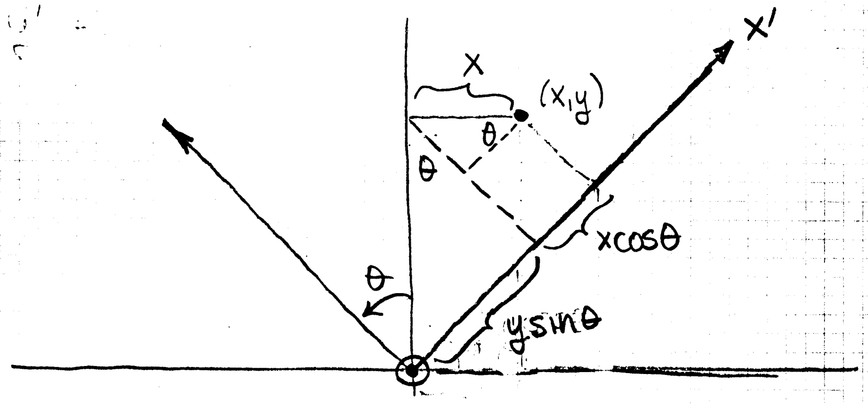
planet centered coordinates to spacecraft coordinates

Derivation of rotational transformations



rotation +θ (given by right-hand rule) about z-axis.

$$y' = y\cos\theta - x\sin\theta$$



$$x' = y\sin\theta + x\cos\theta$$

$$[x \;\; y \;\; z \;\; 1]\begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

as given.

8.3.2 General transformations

show geometric transform of Ranger spacecraft camera.

8.3.3. Specification by control points

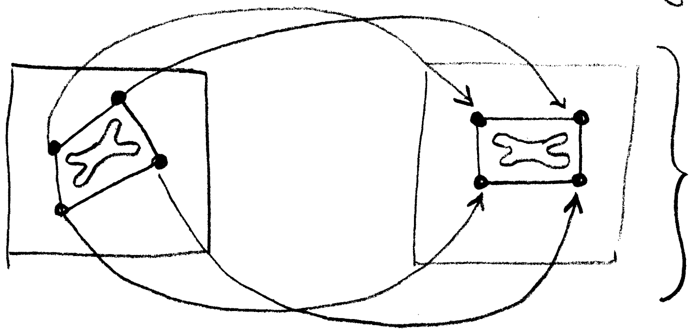
specify the spatial transform as a series of displacement values for selected <u>control points</u> in the image. Displacements of non-control points determined by interpolation

often use polynomials up to 5th degree.

sometimes need more complex transformations

break picture into polygons and use

piecewise bilinear mapping functions



control grid interpolation

8.3.4. Polynomial warping

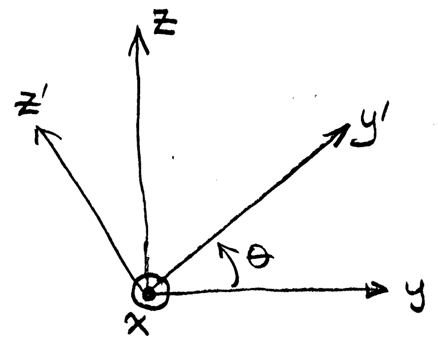
\# of control points = # of terms in polynomial — linear equations

\# of control points > # of terms in polynomial

use fitting techniques

8.3.5 Control grid interpolation

the same derivations hold true for similar rotations

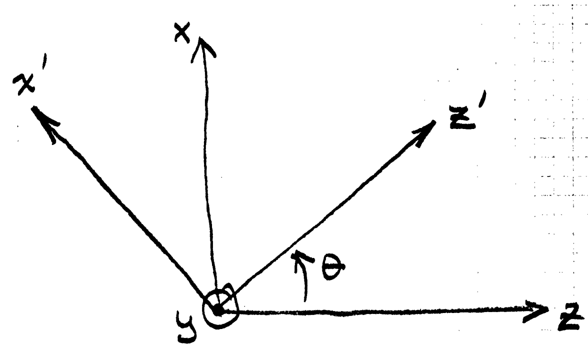


rotation +θ given by right hand rule.

$$y' = z\sin\theta + y\cos\theta$$

$$z' = z\cos\theta - y\sin\theta$$

$$[x \quad y \quad z \quad 1]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



$$z' = x\sin\theta + z\cos\theta$$

$$x' = x\cos\theta - z\sin\theta$$

$$[x \quad y \quad z \quad 1]\begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Ballard & Brown:

use linear transformations to relate object and image coordinates

homogeneous coordinates – we may not want to pivot or rotate about the origin

$(x, y) \longrightarrow (a, b, c)$ where $x = \frac{a}{c}$, $y = \frac{b}{c}$

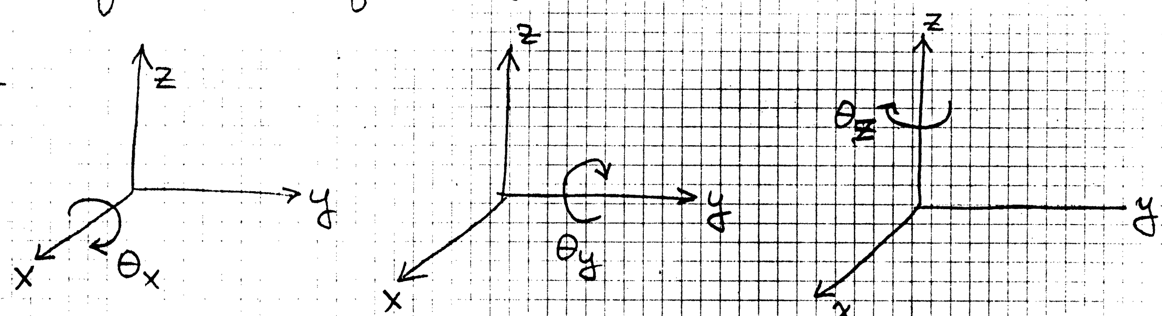
if $c = 1$ these are normalized

in general, homogeneous coordinates are not unique

linear transformations of homogeneous coordinates

Examples:

rotation



use clockwise direction as seen looking into origin along coordinate axis

$$R_{x,\theta} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x & 0 \\ 0 & +\sin\theta_x & \cos\theta_x & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad R_{y,\theta} = \begin{bmatrix} \cos\theta_y & 0 & +\sin\theta_y & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad R_{z,\theta} = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 & 0 \\ +\sin\theta_z & \cos\theta_z & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

scaling: multiplying coordinate by a constant

$$\begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

skewing: representing one coordinate as a linear combination of others.

$$\begin{bmatrix} 1 & k & n & 0 \\ d & 1 & p & 0 \\ e & m & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

rotation = scale + skew.

translation

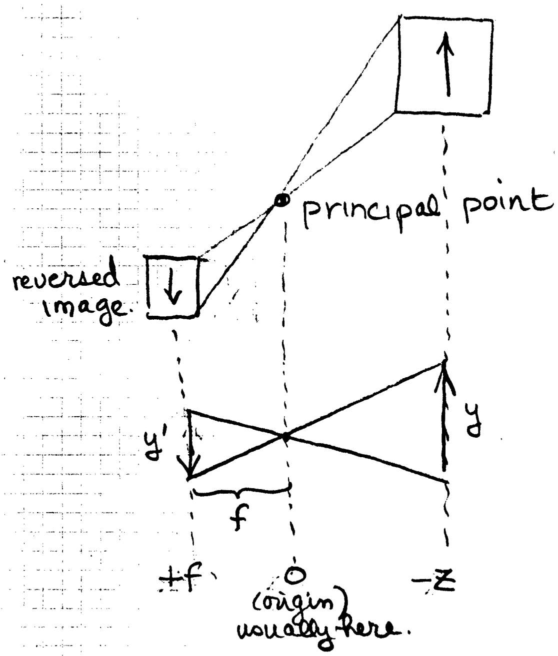
different than robotics (multiply left to right)

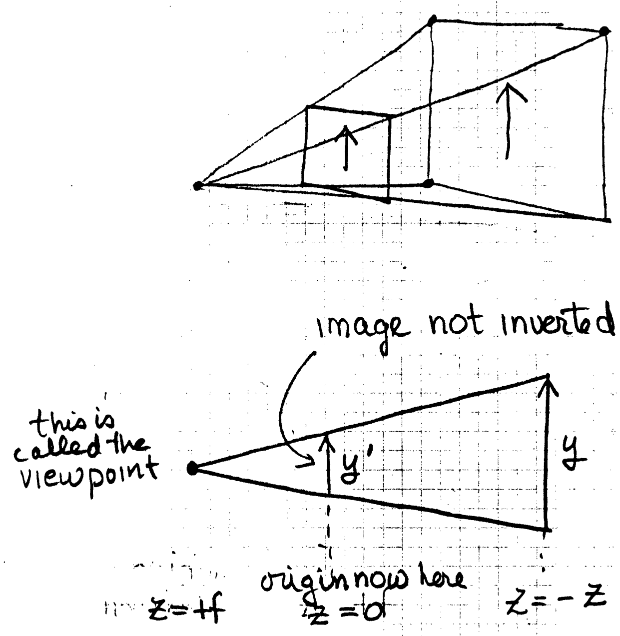
translation matrix:
$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t & u & v & 1 \end{bmatrix}$$

multiplication:
using row matrix on left.

$$[x \quad y \quad z \quad 1]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t & u & v & 1 \end{bmatrix} = \left.\begin{bmatrix} x+t \\ y+u \\ z+v \\ 1 \end{bmatrix}^{T}\right\} \text{translation}$$

perspective





Perspective transformations in x, y

$$\frac{y'}{f} = \frac{y}{f-z} \qquad \frac{x'}{f} = \frac{x}{f-z}$$

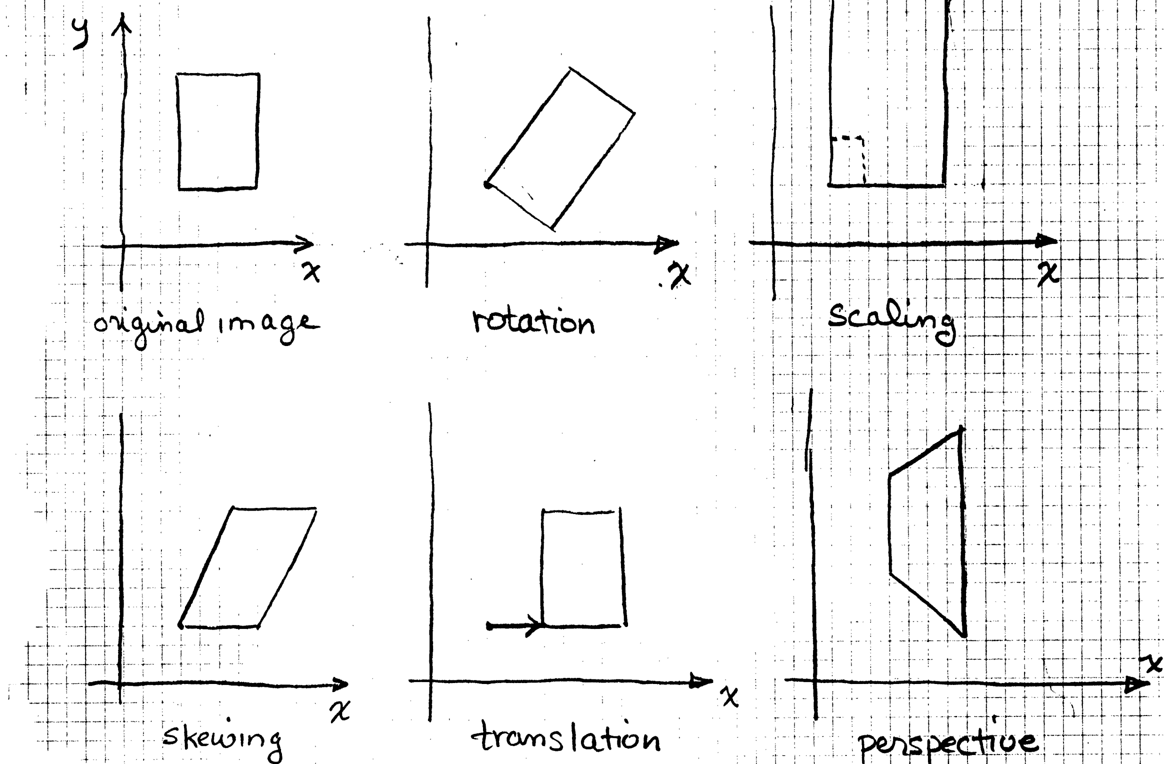
mathematically, we can have a z-perspective transformation, but in practice it cannot occur.

Note as $f \to \infty$ perspective transform $\to$ orthographic transform (projection)

$f \to 0$ lots of distortion (not useful)

References: Ballard & Brown, Ch. 2, p. 17-22 (optics)
p. 467 homogeneous coordinates
p. 477 geometric transformations

different types of effects:



A perspective transformation is <u>exactly</u> an imaging transformation.

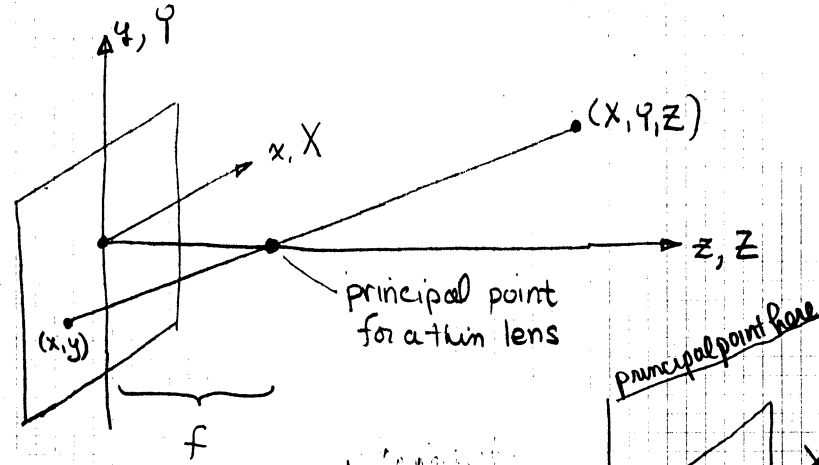
Transform which shows perspective in z-only.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -\frac{1}{f} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Actual coordinate transformation

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -\frac{1}{f} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} x & y & z & 1-\frac{z}{f} \end{bmatrix}$$

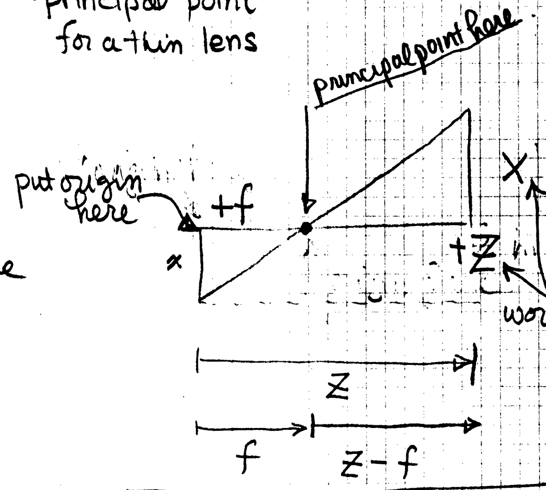
all elements scaled by $\frac{f-z}{f}$. In general, we don't see a z-perspecti transform with a came although we regard it as occurring.



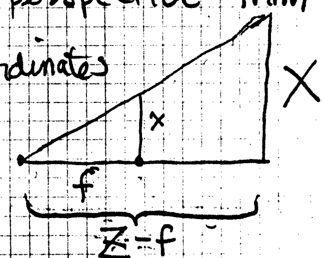
(x, y, z) camera coordinates

(X, Y, Z) world coordinates

Examining figure



when we go to perspective from world coordinates



$$\therefore \quad \frac{X}{Z-f} = \frac{-x}{f}$$ ($\because$ x is negative because its inverted)

$$\therefore \quad \frac{x}{f} = -\frac{X}{Z-f} = \frac{X}{f-Z}$$

similarily

$$\frac{y}{f} = -\frac{Y}{Z-f} = \frac{Y}{f-Z}$$

The focal plane coordinates are then

$$x = \frac{f}{f-Z}\, X$$

$$y = \frac{f}{f-Z}\, Y$$

This is exactly the perspective transformation, i.e.

$$x' = \frac{x}{1-\frac{z}{f}} = \frac{f}{f-z}\, x$$

Note: mapping a 3-D scene onto the image plane is a many-to-one transformation.

we actually have the parametric equation of The line given the image plane coordinates (x_0, y_0)

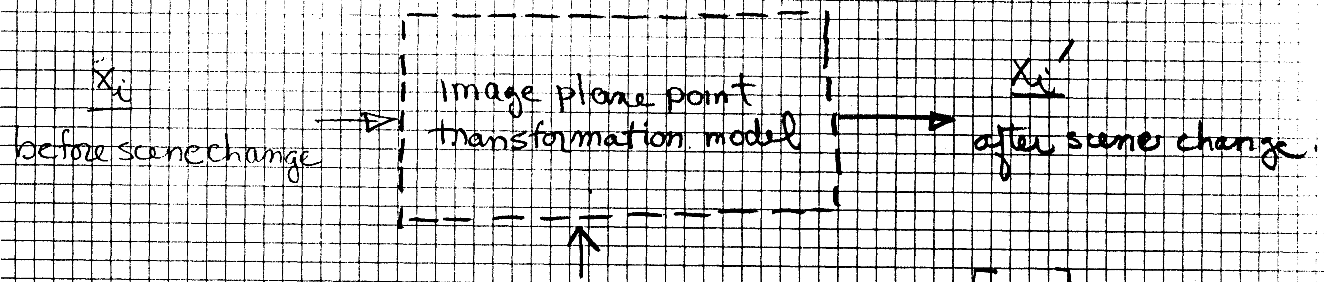
$$x_0 = \frac{f}{f-Z} X$$

$$X = \frac{f x_0 - x_0 Z}{f} = x_0 - \frac{x_0}{f} Z$$

$$Y = y_0 - \frac{y_0}{f} Z$$

Note that given any (x_0, y_0) There are infinitely many points on the line given by $(X(z), Y(z), z)$. This means that you <u>CANNOT</u> recover 3-D point information by means of the inverse perspective transform <u>unless</u> you know at least one of the world coordinates of the point.

image plane — image plane (geometric) transformations



$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} \qquad\qquad \underline{x}_i' = \begin{bmatrix} x_i' \\ y_i' \end{bmatrix}$$

$$\underset{\substack{\uparrow \\ \text{geometrically} \\ \text{perturbed image}}}{f'(\underline{x}_i)} = f(\underline{x}_i') = f\left[\underbrace{g(\underline{x}_i, \underline{a})}_{\text{planar transform}}\right]$$

models changes in intensity only through mapping of coordinate intensities

Application #1:

modeling image plane – image plane transformations due to a change in viewing conditions

Application #2:

modeling the movement or "motion" of 3-D object points relating their positions in the image plane (using the p-p transform) before and after the motion.

model procedure

1. compute using the p-p transform the mapping of object points in the image plane.

2. Compute the same, for the corresponding object point locations that results from the change.

3. Relate the results from these two computations.

p-p transform

$$[wx_i \ \ wy_i \ \ w] = [x_0 \ \ y_0 \ \ z_0 \ \ 1]\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{f} \\ 0 & 0 & 1 \end{bmatrix}$$

ignoring z

consider Application #1, a change in focal length f,

Example: autofocus inspection camera which "autonomously" adjusts field of view to fit size of part.

$$[w'x_i' \ \ w'y_i' \ \ w'] = [x_0 \ \ y_0 \ \ z_0 \ \ 1]\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{f'} \\ 0 & 0 & 1 \end{bmatrix}$$

since z_0 does <u>not</u> change.

$$w = -\frac{z_0}{f} + 1 \qquad\qquad w' = -\frac{z_0}{f'} + 1$$

$$wf - f = -z_0 \qquad\qquad w'f' - f' = -z_0$$

$$\therefore \quad f(w-1) \approx f'(w'-1)$$

for large magnifications and reasonably small focal length changes $|w| \gg 1$ giving

$$fw \approx f'w'$$

relationship of image points before and after transformation is

$$\begin{bmatrix} w'x_i' \\ w'y_i' \\ w' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{f}{f'} \end{bmatrix} \begin{bmatrix} wx_i \\ wy_i \\ w \end{bmatrix}$$

Ok, but how does this relate to a transformation of object coordinates

Suppose $\underline{\hat{X}_0} = [x_0 \; y_0 \; z_0]$ moves along $\begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix}$ to a new position

$$\hat{X}_0' = [x_0 + d_x \quad y_0 + d_y \quad z_0 + d_z]$$

Then $\quad \underline{\hat{X}_0'} = \underline{\underline{T}} \; \underline{\hat{X}_0}$

or $$[x_0+d_x \quad y_0+d_y \quad z_0+d_z \quad 1] = [x_0 \quad y_0 \quad z_0 \quad 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ d_x & d_y & d_z & 1 \end{bmatrix}$$

If your model is $\quad \underline{\hat{X}_i} = P \underline{\hat{X}_0}$

and $\quad \underline{\hat{X}_i'} = PT \underline{\hat{X}_0}$

yields a non-linear relationship between image points

super homogeneous coordinates

$$[w'x_i' \quad w'y_i' \quad w' \quad 1] = [wx_i \quad wy_i \quad w \quad 1] \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ d_x & d_y & \frac{d_z}{f} & 1 \end{bmatrix}$$

makes the problem linear.

Notes:

① The only thing that requires superhomogeneous coordinates is w' and that is accounted for using the term

$$-\frac{d_z}{f}$$

② If $d_x = d_y = 0$ (i.e. only motion along the optical axis) you get exactly the case of a scale change.

③ The result implicitly involves the object - image plane distance through the magnification ratio in w.

Potential application but too long to do: camera pan & tilt

1. order of pan & tilt is important mechanically and mathematicall
2. good example of transformation matrices
3. reasonable engineering approximations
4. framework for example in camera control.

(Dzialo & Schalkoff, 1986)
Control Considerations in tracking moving objects using time-varying perspective-projective imagery, IEEE Trans. Indust. Electronics

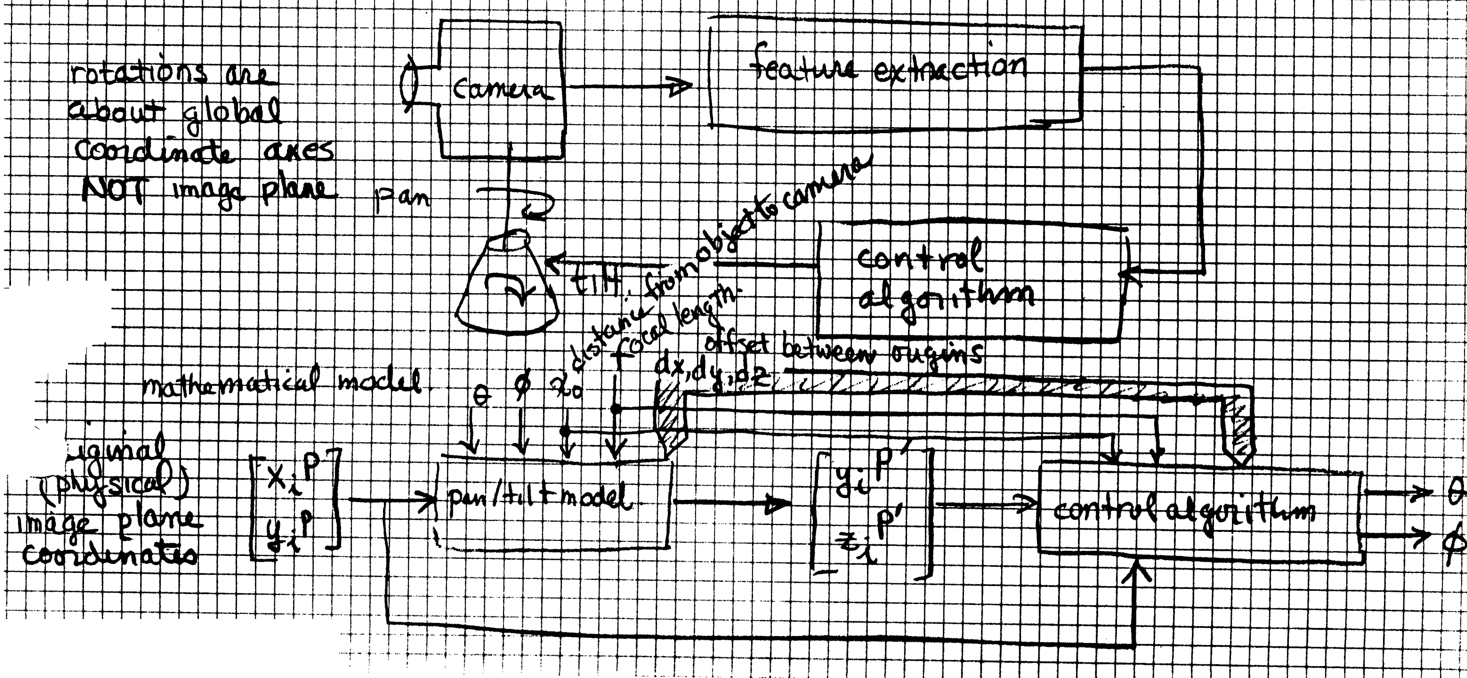


Image plane analysis

① compute the original (unperturbed) image plane point locations as a function of object point locations. The perspective-projection transform.

② Translate the image plane coordinate system to the global coordinate system origin (about which the pan-tilt transformations occur). Note that both coordinate systems here are related by a simple translation; in practice other transformations may also be required.

③ Perform the pan-tilt transformations in the global coordinate system.

④ Transform back to the image plane coordinate system. (This is the inverse of ②).

⑤ Compute the new image plane locations as a function of the object point locations.

⑥ Relate the image plane coordinates in ⑤ to those in ①. This step gives an image-plane to image-plane transformation of the form:

$$[x_i' \quad y_i' \quad w_i' \quad 1] = [x_i \quad y_i \quad w_i \quad 1] \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & 1 \end{bmatrix}$$

a_{ij} is a function of sensor geometry and object/sensor motion.

Not going to derive the transformation, but.

$$\begin{bmatrix} x_i' & y_i' & w_i' & 1 \end{bmatrix} = \begin{bmatrix} x_i & y_i & w_i & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & \sin\theta\sin\phi & \frac{\cos\theta\sin\phi}{f} \\ 0 & \cos\theta & \frac{-\sin\theta}{f} \\ -f\sin\phi & f\sin\theta\cos\phi & \cos\theta\cos\phi \\ A & E & F \end{bmatrix}$$

where
$$A = (f+d_x)\sin\phi + d_y\cos\phi - d_y$$
$$E = -(f+d_x)\cos\phi\sin\theta + d_y\sin\phi\sin\theta + d_z\cos\theta - d_z$$
$$F = \frac{-(f+d_x)\cos\phi\cos\theta + d_y\sin\phi\cos\theta - d_z\sin\theta + d_x + f}{f}$$

The physical coordinates of new image points are then given by:

$$x_i^{p'} = \frac{(x_i\cos\phi - w_i f\sin\phi + A)}{G}$$

$$y_i^{p'} = \frac{x_i\sin\theta\sin\phi + y_i\cos\theta + f w_i\sin\theta\cos\phi + E}{G}$$

where
$$G = \frac{x_i\cos\theta\sin\phi - y_i\sin\theta + f w_i\cos\theta\cos\phi + fF}{f}$$

Expressing x_i and y_i in physical (image plane) coordinates

$$x_i^{p'} = \frac{x_i^p\cos\phi - f\sin\phi + \frac{A}{w_i}}{H}$$

$$y_i^{p'} = \frac{x_i^p\sin\theta\sin\phi + y_i^p\cos\theta + f\sin\theta\cos\phi + \frac{E}{w_i}}{H}$$

where
$$H = \frac{x_i^p\cos\theta\sin\phi - y_i^p\sin\theta + f\cos\theta\cos\phi + \frac{fF}{w_i}}{f}$$

this is lin
in super
homogeneou
coordinate
non-linear
homogeneou
coordinates

For a system with a large magnification ratio : $\frac{x_0}{f}$.

Suppose $x_0 = 300$ feet (100 meters)

$f = 100$ mm

$$\frac{x_0}{f} = 1000$$

$$w_i = -\frac{x_0}{f} + 1 \quad \text{so} \quad w_i \approx -\frac{x_0}{f}$$

and $\frac{1}{w_i}$ can be neglected in these equations.

Good, because hard to measure offset vector $[d_x \; d_y \; d_z]$ since it's in the camera case.

With this assumption

$$x_i^{P'} = \frac{x_i^P \cos\phi - f\sin\phi + A'}{H'}$$

$$y_i^{P'} = \frac{x_i^P \sin\theta \sin\phi + y_i^P \cos\theta + f\sin\theta\cos\phi + E'}{H'}$$

where $A' = -\left[(f+d_x)\sin\phi + d_y(\cos\phi - 1)\right]\frac{f}{x_0}$

$$E' = -\left[-(f+d_x)\cos\phi\sin\theta + d_y\sin\phi\sin\theta + d_z(\cos\theta - 1)\right]\frac{f}{x_0}$$

$$H' = \frac{x_i^P\cos\theta\sin\phi - y_i^P\sin\theta + f\cos\theta\cos\phi + fF'}{f}$$

$$F' = -\frac{\left[-(f+d_x)\cos\theta\cos\phi + d_y\sin\phi\cos\theta - d_z\sin\theta + d_x + f\right]}{x_0}$$

For small-angle approximations

$$x_i^{P'} = x_i^P - f\tan\phi$$

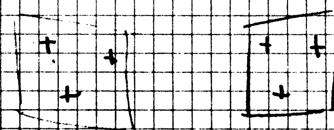
$$y_i^{P'} = x_i^P\tan\theta\tan\phi + y_i^P + f\tan\theta$$

The inverse equations for object centering are then

$$\phi = \mathrm{Tan}^{-1}\left(-\frac{x_i^{P'} - x_i^P}{f}\right)$$

$$\theta = \mathrm{Tan}^{-1}\left(\frac{y_i^{P'} - y_i^P}{x_i^P\tan\phi + f}\right)$$

Geometric correction and registration



View 1   View 2 (distorted version of View 1)

Perspective correction of imagery:

Given two images of the same scene taken by sensors with different or time-varying orientations, correct one of the images to the viewpoint of the other.

(lots of industrial & tactical applications using geometrically distributed sensors, satellites with varying viewing angles, etc)

Polynomial warp model

- mathematical model of the distortion
- a set of corresponding image points of the form $(\underline{x}, \underline{w})$

in image #1 $\underline{x}$ is the vector location of the undistorted image plane points

in image #2 $\underline{w}$ is the vector location of the distorted (or imaged) poin

Now, warp $\underline{w}$ into $\underline{x}$

points in this set are often called control points

useful for multiple sensor images (IR and visible)

image 1	$f_1(\underline{x})$	x_1, x_2 coordinates
image 2	$f_2(\underline{w})$	w_1, w_2 coordinates

$$\left.\begin{aligned} x_1 &= g_1(w_1, w_2) \\ x_2 &= g_2(w_1, w_2) \end{aligned}\right\} \text{ i.e. what is the transform from } \underline{w} \rightarrow \underline{x} \text{ so that we can warp Image 2}$$

approximate by N^{th} order 2-D polynomials

$$x_1 = \sum_{i=0}^{N} \sum_{j=0}^{N} k_{ij}^{(1)} w_1^i w_2^j$$

$$x_2 = \sum_{i=0}^{N} \sum_{j=0}^{N} k_{ij}^{(2)} w_1^i w_2^j$$

Given a set of m corresponding control points in each coordinate system

$$(x_{1i}, x_{2i}, w_{1i}, w_{2i}) \qquad i = 1, 2, 3, \ldots, m$$

For $N=2$ (2nd order warp)

example for $i = k$

image #1 image #2

$$x_{1k} = k_{00}^{(1)} + k_{10}^{(1)} w_{1k} + k_{01}^{(1)} w_{2k} + k_{11}^{(1)} w_{1k} w_{2k} + k_{20}^{(1)} (w_{1k})^2$$

$$+ k_{02}^{(1)} (w_{2k})^2 + k_{21}^{(1)} (w_{1k})^2 w_{2k} + k_{12}^{(1)} w_{1k} (w_{2k})^2$$

$$+ k_{22}^{(1)} (w_{1k})^2 (w_{2k})^2$$

up to second order in each variable

x_{2k} looks identical

$$= k_{00}^{(2)} + k_{10}^{(2)} w_{1k} + k_{01}^{(2)} w_{2k} + k_{11}^{(2)} w_{1k} w_{2k} + \ldots$$

for $N=2$ 18 coefficients (9 per equation) need to be estimated

$\Rightarrow$ at least 9 control points are needed.

in general for N (order of warp) need $2(N+1)^2$ points

Notes:

1. The estimation equations are linear in $k_{ij}^{(\,)}$ (the variables, the w's are known)
2. The estimation process is separable since $k_{ij}^{(1)}$ and $k_{ij}^{(2)}$ only appear in one of the equations, i.e. $k_{ij}^{(1)}$ and $k_{ij}^{(2)}$ can be found separately.
3. The $k_{ij}^{(\,)}$ coefficients are the respective powers of w_1 and w_2 which are identical and only need to be computed once.