Chapter 13 Perspective Projection Geometry

13.1 Introduction

Computer vision problems often involve interpreting information on a two-dimensional (2D) image of a three-dimensional (3D) world in order to determine the placement of the 3D objects portrayed in the image.
To do this requires understanding the perspective transformation governing the geometric way 3D information is projected onto the 2D image.
image formation on the retina, according to Descartes
scrape ox eye, observe from darkened room inverted image of scene
=====Nalwa, *A Guided Tour of Computer Vision*, Fig. 2.1=====

13.2 One-Dimensional Perspective Projection

f: focal length of lens
u: distance between object and lens center
v: distance between image and lens center
thin-lens equation: lens law:

$$\frac{1}{f} = \frac{1}{u} + \frac{1}{v}$$

light passing lens center does not deflect
light parallel to optical axis will pass focal point

=====Nalwa, *A Guided Tour of Computer Vision*, Fig. 2.11=====
perspective projection: object point projected along straight line —caption—
=====Nalwa, *A Guided Tour of Computer Vision*, Fig. 2.3=====
image distortion: room ceiling appears bowed in the image
=====Nalwa, *A Guided Tour of Computer Vision*, Fig. 2.13=====

pinhole camera: infinitesimally small aperture
pinhole camera: approximated by lens with aperture adjusted to the smallest
pinhole camera: simplest device to form image of 3D scene on 2D surface
=====Nalwa, *A Guided Tour of Computer Vision*, Fig. 2.2=====
aperture size decreased: image become sharper
diameter of aperture is 0.06 inch, 0.015 inch, 0.0025 inch
aperture below certain size: diffraction: bending of light rays around edge
=====Nalwa, *A Guided Tour of Computer Vision*, Fig. 2.10=====

lens oriented along the y-axis and image plane parallel to x-axis
perspective projection gives point (r, s) coordinates $(rf/s, f)$ on image
=====Fig. 13.1=====
f: camera constant (different from above equation)
$(r, s, 1)$: homogeneous coordinate system for point (r, s)

first linear transformation: translates $(r, s, 1)$ by distance of f
second linear transformation: takes perspective transformation to image line

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/f & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -f \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r \\ s \\ 1 \end{pmatrix} = \begin{pmatrix} r \\ s/f \end{pmatrix}$$

1D image line coordinate: $x_I = u/v = rf/s$

lens: at origin and looks down y'-axis
image line: distance f in front of lens and parallel to x'-axis
$x' - y'$ axes: the $x - y$ axes rotated anticlockwise by angle θ
=====Fig. 13.2=====
rewriting the relationship in terms of homogeneous coordinate system

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/f & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -f \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r \\ s \\ 1 \end{pmatrix}$$

13.3 The Perspective Projection in 3D

camera lens: along line parallel to z-axis
position of lens: center of perspectivity: (x_0, y_0, z_0)

(u, v): coordinates of perspective projection of (x, y, z) on image plane

$$\begin{pmatrix} x^* \\ y^* \\ t^* \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -f \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & -x_0 \\ 0 & 1 & 0 & -y_0 \\ 0 & 0 & 1 & -z_0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

thus

$$u = \frac{x^*}{t^*} = f\frac{x - x_0}{z - z_0} \qquad v = \frac{y^*}{t^*} = f\frac{y - y_0}{z - z_0}$$

13.3.1 Smaller Appearance of Farther Objects

without loss of generality: take center of perspectivity to be origin

perspective projection: objects appear smaller the farther they are

=====Fig. 13.3=====

foreshortening: line segments in plane parallel to image has maximum size

=====Fig. 13.4=====

13.3.2 Lines to Lines

lines in the 3D world transform to lines in the image plane

parallel lines in 3D with nonzero z slope: meet in a vanishing point

=====Nalwa, *A Guided Tour of Computer Vision*, Fig. 2.4=====

13.3.3 Perspective Projection of Convex Polyhedra are Convex

proofs in textbook, simple but tedious, study as exercise by yourself

13.3.4 Vanishing Point

Perspective projections of parallel 3D lines having nonzero slope along the optic z-axis meet in a vanishing point on the image projection plane.

13.3.5 Vanishing Line

All lines lying in planes parallel to the slanted floor have vanishing points that lie along a vanishing line.

=====Fig. 13.5=====

=====Tour Into Picture=====

=====Garfield 17:21=====

13.3.6 3D Lines-2D Perspective Projection Lines

There is a relationship between the parameters of a 3D line and the parameters of the perspective projection of the line.

3D line L:

$$L = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \middle| \text{for some } \lambda, \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + \lambda \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \right\}$$

perspective projection of the line L:

$$M = \left\{ \begin{pmatrix} u \\ v \end{pmatrix} \middle| \text{for some } \eta, \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \eta \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} \right\}$$

take camera lens as origin of coordinate system

$$u = f\frac{x}{z}$$

$$v = f\frac{y}{z}$$

for any given λ, η:

$$c_1 + \eta d_1 = f(a_1 + \lambda b_1)/(a_3 + \lambda b_3)$$

$$c_2 + \eta d_2 = f(a_2 + \lambda b_2)/(a_3 + \lambda b_3)$$

eliminate η:

$$c_1 d_2 - c_2 d_1 = \frac{d_2 f(a_1 + \lambda b_1) - d_1 f(a_2 + \lambda b_2)}{a_3 + \lambda b_3}$$

for any λ:

$$(c_1 d_2 - c_2 d_1) a_3 - (d_2 a_1 - d_1 a_2) f + \lambda[b_3(c_1 d_2 - c_2 d_1) - (d_2 b_1 - d_1 b_2) f] = 0$$

thus

$$d_2 f a_1 - d_1 f a_2 + (c_2 d_1 - c_1 d_2) a_3 = 0$$
$$d_2 f b_1 - d_1 f b_2 + (c_2 d_1 - c_1 d_2) b_3 = 0$$

13.4 2D to 3D Inference Using Perspective Projection

perspective projection on unknown 3D line: provides four of six constraints
additional constraints: 3D-world-model information about points, lines

13.4.1 Inverse Perspective Projection

(u, v): perspective projection of a point
f: image plane distance from camera lens
thus (u, v, f): 3D coordinate of the point in image plane
camera lens: at the origin
line L: inverse perspective projection of the point (u, v)

$$L = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \middle| \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \lambda \begin{pmatrix} u \\ v \\ f \end{pmatrix} \right\}$$

13.4.2 Line Segment with Known Direction Cosines and Known Length

known:

- δ: line segment length
- $\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$: line segment direction cosine
- $\begin{pmatrix} u_1 \\ v_1 \end{pmatrix}, \begin{pmatrix} u_2 \\ v_2 \end{pmatrix}$: perspective projections of endpoints

unknown:

- $p_1 = \eta_1 \begin{pmatrix} u_1 \\ v_1 \\ f \end{pmatrix}, p_2 = \eta_2 \begin{pmatrix} u_2 \\ v_2 \\ f \end{pmatrix}$: 3D coordinates of endpoints

13.4.3 Collinear Points with Known Interpoint Distances

known:

- $\begin{pmatrix} u_n \\ v_n \end{pmatrix}$: perspective projections of nth collinear points, $n = 0, ..., N-1$
- $\delta_1, \delta_2, ..., \delta_{N-1}$: distance between $(n+1)$th point and first point

unknown:

- $b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$: direction cosine of line

- $p = \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix}, p, p + \delta_1 b, ..., p + \delta_{N-1} b$: 3D coordinates of points

13.4.4 N Parallel Lines

known:

- $L_n = \left\{ \begin{pmatrix} u \\ v \end{pmatrix} \middle| \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} c_n \\ d_n \end{pmatrix} + \eta \begin{pmatrix} g_n \\ h_n \end{pmatrix} \right\}, \quad n = 1, ..., N$: perspective projections of nth parallel line

unknown:

- $b = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$: direction cosine of line

13.4.5 N Lines Intersecting at a Point with Known Angles

known:

- $\begin{pmatrix} g \\ h \end{pmatrix}$: perspective projection of intersecting point

- $M_n = \left\{ \begin{pmatrix} u \\ v \end{pmatrix} \middle| \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} g \\ h \end{pmatrix} + \eta \begin{pmatrix} \alpha_n \\ \beta_n \end{pmatrix} \right\}, \quad n = 1, ..., N$: perspective projections of nth intersecting line
- θ_{pq}: known angle between L_p and L_q

unknown:

- $L_n = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \middle| \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} + \lambda \begin{pmatrix} i_n \\ j_n \\ k_n \end{pmatrix} \right\}, \quad n = 1, ..., N$: 3D nth intersecting line

$\cos\theta_{pq} = i_p i_q + j_p j_q + k_p k_q$: known angle between line pair

13.4.6 N Intersecting Lines in a Known Plane

known:

- $\begin{pmatrix} g \\ h \end{pmatrix}$: perspective projection of intersecting point
- $M_n = \left\{ \begin{pmatrix} u \\ v \end{pmatrix} \middle| \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} g \\ h \end{pmatrix} + \eta \begin{pmatrix} \alpha_n \\ \beta_n \end{pmatrix} \right\}, \quad n = 1, ..., N$: perspective projections of nth intersecting line
- $Aa + Bb + Cc + D = 0$: plane equation

unknown:

- $L_n = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \middle| \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} + \lambda \begin{pmatrix} i_n \\ j_n \\ k_n \end{pmatrix} \right\}, \quad n = 1, ..., N$: 3D nth intersecting line

$$Ai_n + Bj_n + Ck_n = 0$$

normal vector: perpedicular to direction cosine
=====joke=====

13.4.7 Three Lines in a Plane with One Perpendicular to the Other Two
known:

- $M_1 = \left\{ \begin{pmatrix} u \\ v \end{pmatrix} \middle| \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} g_1 \\ h_1 \end{pmatrix} + \eta \begin{pmatrix} \alpha_1 \\ \beta_1 \end{pmatrix} \right\}$: perspective projection of line
- $M_2 = \left\{ \begin{pmatrix} u \\ v \end{pmatrix} \middle| \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} g_2 \\ h_2 \end{pmatrix} + \eta \begin{pmatrix} \alpha_2 \\ \beta_2 \end{pmatrix} \right\}$: perspective projection of line
- $M_3 = \left\{ \begin{pmatrix} u \\ v \end{pmatrix} \middle| \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} g_3 \\ h_3 \end{pmatrix} + \eta \begin{pmatrix} \alpha_3 \\ \beta_3 \end{pmatrix} \right\}$: perspective projection of line

unknown: three lines in same plane, one perpendicular to other two

- $L_1 = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \middle| \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + \lambda \begin{pmatrix} k_1 \\ k_2 \\ k_3 \end{pmatrix} \right\}$: 3D line

- $L_2 = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \middle| \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} + \lambda \begin{pmatrix} k_1 \\ k_2 \\ k_3 \end{pmatrix} \right\}$: 3D line

- $L_3 = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \middle| \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} + \lambda \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} \right\}$: 3D line

$k_1 m_1 + k_2 m_2 + k_3 m_3 = 0$: since L_3 perpendicular to L_1, L_2

13.4.8 Point with Given Distance to a Known Point

known:

- $\begin{pmatrix} u \\ v \end{pmatrix}$: perspective projection of unknown point

- $\begin{pmatrix} a \\ b \\ c \end{pmatrix}$: known 3D point

- ρ: distance between the two points

unknown:

- $\begin{pmatrix} i \\ j \\ k \end{pmatrix}$: direction cosine between two points

$$\sqrt{(\eta u - a)^2 + (\eta v - b)^2 + (\eta f - c)^2} = \rho$$

13.4.9 Point in a Known Plane

known:

- $\begin{pmatrix} u \\ v \end{pmatrix}$: perspective projection of unknown point
- $Ax + By + Cz + D = 0$: known plane equation where point lies

unknown:

- $\begin{pmatrix} a \\ b \\ c \end{pmatrix}$: 3D coordinate of the point

solution:

- $\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} -\frac{Du}{Au+Bv+Cf} \\ -\frac{Dv}{Au+Bv+Cf} \\ -\frac{Df}{Au+Bv+Cf} \end{pmatrix}$: 3D coordinate of the point

13.4.10 Line in a Known Plane

known:

- $Ax + By + Cz + D = 0$: known plane equation where line lies
- $M = \left\{ \begin{pmatrix} u \\ v \end{pmatrix} \middle| \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} g \\ h \end{pmatrix} + \eta \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \right\}$: perspective projection of line

unknown:

- $L = \left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \middle| \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \end{pmatrix} + \lambda \begin{pmatrix} i \\ j \\ k \end{pmatrix} \right\}$: 3D line

$Ai + Bj + Ck = 0$: since line lies in plane
$Aa + Bb + Cc + D = 0$: since line lies in plane

13.4.11 Angle

known:

- $M = \left\{ \begin{pmatrix} u \\ v \end{pmatrix} \middle| \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} g \\ h \end{pmatrix} + \eta \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \right\}$: perspective projection of the unknown line
- $\begin{pmatrix} i_0 \\ j_0 \\ k_0 \end{pmatrix}$: direction cosine for the known line

- θ: angle between the 3D lines

unknown:

- $\begin{pmatrix} i_1 \\ j_1 \\ k_1 \end{pmatrix}$: direction cosine for the unknown line

$\cos\theta = i_0 i_1 + j_0 j_1 + k_0 k_1$: since θ is the angle between 3D lines

13.4.12 Parallelogram

known:

- perspective projection of four corner points of a parallelogram

unknown:

- $\begin{pmatrix} A \\ B \\ C \end{pmatrix}$: normal to the plane where the parallelogram lies

$\begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix}, \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix}$: direction cosines of two sides of parallelogram

$$\begin{pmatrix} A \\ B \\ C \end{pmatrix} = \frac{\begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} \times \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix}}{\left\| \begin{pmatrix} m_1 \\ m_2 \\ m_3 \end{pmatrix} \times \begin{pmatrix} n_1 \\ n_2 \\ n_3 \end{pmatrix} \right\|}$$

13.4.13 Triangle with One Vertex Known

known:

- $\begin{pmatrix} u_1 \\ v_1 \end{pmatrix}, \begin{pmatrix} u_2 \\ v_2 \end{pmatrix}, \begin{pmatrix} u_3 \\ v_3 \end{pmatrix}$: perspective projection of three vertices
- $\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}$: one known 3D vertex of the three vertices
- S_{12}, S_{13}, S_{23}: known length of the triangle in 3D

unknown:

- $\begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix}, \begin{pmatrix} x_3 \\ y_3 \\ z_3 \end{pmatrix}$: two unknown 3D vertices of the three vertices

$$(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 = S_{12}^2$$
$$(x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2 = S_{13}^2$$
$$(x_2 - x_3)^2 + (y_2 - y_3)^2 + (z_2 - z_3)^2 = S_{23}^2$$
$$x_2 = \frac{u_2 z_2}{f}$$
$$y_2 = \frac{v_2 z_2}{f}$$
$$x_3 = \frac{u_3 z_3}{f}$$
$$y_3 = \frac{v_3 z_3}{f}$$

13.4.14 Triangle with Orientation of One Leg Known

known:

- $\begin{pmatrix} u_1 \\ v_1 \end{pmatrix}, \begin{pmatrix} u_2 \\ v_2 \end{pmatrix}, \begin{pmatrix} u_3 \\ v_3 \end{pmatrix}$: perspective projection of three vertices

- S_{12}, S_{13}, S_{23}: known length of the triangle in 3D
- $\begin{pmatrix} i_1 \\ j_1 \\ k_1 \end{pmatrix}$: known direction cosines between the first two vertices

unknown:

- $\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}, \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix}, \begin{pmatrix} x_3 \\ y_3 \\ z_3 \end{pmatrix}$: three unknown 3D vertices

$$
\begin{gathered}
(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 = S_{12}^2 \\
(x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2 = S_{13}^2 \\
(x_2 - x_3)^2 + (y_2 - y_3)^2 + (z_2 - z_3)^2 = S_{23}^2 \\
\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix} + S_{12} \begin{pmatrix} i_1 \\ j_1 \\ k_1 \end{pmatrix}
\end{gathered}
$$

13.4.15 Triangle: three-point spatial resection problem in photogrammetry
known:

- $\begin{pmatrix} u_1 \\ v_1 \end{pmatrix}, \begin{pmatrix} u_2 \\ v_2 \end{pmatrix}, \begin{pmatrix} u_3 \\ v_3 \end{pmatrix}$: perspective projection of three vertices
- S_{12}, S_{13}, S_{23}: known length of the triangle in 3D

unknown:

- $\begin{pmatrix} x_1 \\ y_1 \\ z_1 \end{pmatrix}, \begin{pmatrix} x_2 \\ y_2 \\ z_2 \end{pmatrix}, \begin{pmatrix} x_3 \\ y_3 \\ z_3 \end{pmatrix}$: three unknown 3D vertices

four solutions

$$(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2 = S_{12}^2$$
$$(x_3 - x_1)^2 + (y_3 - y_1)^2 + (z_3 - z_1)^2 = S_{13}^2$$
$$(x_2 - x_3)^2 + (y_2 - y_3)^2 + (z_2 - z_3)^2 = S_{23}^2$$
$$u_n = \frac{f x_n}{z_n} \quad n = 1, 2, 3$$
$$v_n = \frac{f y_n}{z_n} \quad n = 1, 2, 3$$

13.4.16 Determining the Principal Point by Using Parallel Lines

principal point: point through which the optic axis passes

principal point: so far assumes origin of image reference frame

known:

- $M_n = \left\{ \begin{pmatrix} u \\ v \end{pmatrix} \middle| \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} g \\ h \end{pmatrix} + \eta \begin{pmatrix} \alpha_n \\ \beta_n \end{pmatrix} \right\}, \quad n = 1, ..., N$: perspective projection of nth parallel line

unknown:

- $\begin{pmatrix} \Delta u \\ \Delta v \end{pmatrix}$: coordinates of the principal point

13.5 Circles

known:

- perspective projection of a circle having known radius d_0

unknown:

- plane on which the circle lies
- the 3D center of the circle

=====Oldie 33:70=====

13.6 Range from Structured Light

structured light: active visual sensing technique upon perspective geometry
structured light: controlled light source with regular pattern onto scene

regular pattern: stripes, grid, ...
light striping and a typical arrangement
=====Ballard and Brown, *Computer Vision*, Fig. 2.25=====
intensity and range images
=====Ballard and Brown, *Computer Vision*, Fig. 2.26=====
Two light sources with cylindrical lenses produce sheets of light that intersect in a line lying on the surface of a conveyor belt.
A camera above the belt is aimed so that this line is imaged on a linear array of photosensors.
When there is no object present, all the sensor cells are brightly illuminated.
When part of an object interrupts the incident light, the corresponding region on the linear array is darkened.
The motion of the belt scans the object past the sensor, generating the second image dimension.
=====Horn, *Robot Vision*, Fig. 5.4=====

13.7 Cross-Ratio

cross-ratio: of perspective projection of 4 collinear points, takes same value

13.7.1 Cross-Ratio Definitions and Invariance

four collinear points: $p+\lambda_1 b, p+\lambda_2 b, p+\lambda_3 b, p+\lambda_4 b$
q, r: centers of perspectivity for two projection images
=====Fig. 13.7=====
Let $p = \begin{pmatrix} p_1 \\ p_2 \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$. by perspective projection equations

$$u_n = f\frac{p_1+\lambda_n b_1}{p_2+\lambda_n b_2} \qquad n = 1, 2, 3, 4$$

$$u_i - u_j = f\frac{p_1+\lambda_i b_1}{p_2+\lambda_i b_2} - f\frac{p_1+\lambda_j b_1}{p_2+\lambda_j b_2} = \frac{f(p_2 b_1 - p_1 b_2)(\lambda_i - \lambda_j)}{(p_2+\lambda_i b_2)(p_2+\lambda_j b_2)}$$

cross-ratio:

$$\gamma(u_1, u_2, u_3, u_4) = \frac{(u_1-u_2)(u_3-u_4)}{(u_1-u_3)(u_2-u_4)} = \frac{(\lambda_1-\lambda_2)(\lambda_3-\lambda_4)}{(\lambda_1-\lambda_3)(\lambda_2-\lambda_4)}$$

cross-ratio: independent of reference frame, point p, direction cosine b
cross-ratio: depends only on directed distance of collinear points

13.7.2 Only One Cross-Ratio

each of 4! cross-ratios is a function of cross-ratio $\gamma(u_1, u_2, u_3, u_4)$

$$\gamma(u_1, u_2, u_3, u_4) = \gamma(u_2, u_1, u_4, u_3) = \gamma(u_3, u_4, u_1, u_2) = \gamma(u_4, u_3, u_2, u_1)$$
$$\gamma(u_1, u_2, u_4, u_3) = \gamma(u_2, u_1, u_3, u_4) = \gamma(u_4, u_3, u_1, u_2) = \gamma(u_3, u_4, u_2, u_1)$$

$$
\begin{aligned}
&= \frac{-\gamma(u_1,u_2,u_3,u_4)}{1-\gamma(u_1,u_2,u_3,u_4)} \\
\gamma(u_1,u_4,u_3,u_2) = \gamma(u_4,u_1,u_2,u_3) = \gamma(u_2,u_3,u_4,u_1) = \gamma(u_3,u_2,u_1,u_4) \\
&= 1-\gamma(u_1,u_2,u_3,u_4) \\
\gamma(u_1,u_3,u_2,u_4) = \gamma(u_3,u_1,u_4,u_2) = \gamma(u_4,u_2,u_3,u_1) = \gamma(u_2,u_4,u_1,u_3) \\
&= \frac{1}{\gamma(u_1,u_2,u_3,u_4)} \\
\gamma(u_1,u_3,u_4,u_2) = \gamma(u_3,u_1,u_2,u_4) = \gamma(u_2,u_4,u_3,u_1) = \gamma(u_4,u_2,u_1,u_3) \\
&= \frac{1}{1-\gamma(u_1,u_2,u_3,u_4)} \\
\gamma(u_1,u_4,u_2,u_3) = \gamma(u_4,u_1,u_3,u_2) = \gamma(u_2,u_3,u_1,u_4) = \gamma(u_3,u_2,u_4,u_1) \\
&= \frac{1-\gamma(u_1,u_2,u_3,u_4)}{-\gamma(u_1,u_2,u_3,u_4)}
\end{aligned}
$$

13.7.3 Cross-Ratio in Three Dimensions

The cross-ratio derived from 1D perspective projections in 2D world can be generalized to 2D perspective projections in 3D world.

five co-planar points

γ_{23}: cross-ratio for the line segment $p_2 p_3$ and $0 < \lambda_1 < \lambda_2 < L_1$

$$\gamma_{23} = \frac{\lambda_1(L_1 - \lambda_2)}{\lambda_2(L_1 - \lambda_1)}$$

γ_{25}: cross-ratio for the line segment $p_2 p_5$ and $0 < \eta_1 < \eta_2 < L_2$

$$\gamma_{25} = \frac{\eta_1(L_2 - \eta_2)}{\eta_2(L_2 - \eta_1)}$$

=====Fig. 13.8=====

13.7.4 Using Cross-Ratios

cross-ratios: to aid in establishing correspondences

=====joke=====

tentative term project problems, (30%) total grade:

submit one page in English explaining method, steps, expected results

submit report in a month; report progress every other week

all right to be the same problem with Master's thesis

objective: a working prototype with new, original, novel ideas

objective: not just literature survey

objective: not just straightforward implementation of existing algorithms

objective: all right to modify existing algorithms

1. Neural Network, Gonzalez, Sec. 9.3.3, p. 595

=====Gonzalez, *Digital Image Processing*, Fig. 9.14=====

2. Fuzzy Logic
3. Image Compression: JPEG (Joint Photographic Experts Group), MPEG (Moving Picture Experts Group), JPEG 2000, Gonzalez, Sec. 6.6, p. 389

=====Gonzalez, *Digital Image Processing*, Plate X=====

4. wavelet transform
5. segmentation based on texture, Gonzalez, p. 594

=====Gonzalez, *Digital Image Processing*, Fig. 9.13=====

6. optical character reading: a, b, c, d, ... 0, 1, 2, ...

=====fonts=====

7. stereo vision, Nalwa, Chapter 7

=====Nalwa, *A Guided Tour of Computer Vision*, Fig. 7.9=====

=====Nalwa, *A Guided Tour of Computer Vision*, Fig. 7.10=====

8. handwriting recognition: zip code; on-line, off-line Chinese characters ...

=====zip=====

=====Chinese=====

9. histogram specification: Gonzalez, p. 182, Sec. 4.2.2

=====Gonzalez, *Digital Image Processing*, Fig. 4.14=====

10. Homomorphic filtering: Gonzalez, Sec. 4.4.3

=====Gonzalez, *Digital Image Processing*, Fig. 4.41=====

=====Gonzalez, *Digital Image Processing*, Fig. 4.42=====

11. real-time counting number of cars and their sizes.
12. calculating the sizes of stones, cells, cell nucleus.

=====t_pebbles.im=====

13. trademark resemblance, semi-automatic similarity classification

=====trademark.im=====

14. car plate recognition.
15. structured light 3-D reconstruction. Horn, p. 95

=====Horn, *Robot Vision*, Fig. 5.4=====

=====Ballard and Brown, *Computer Vision*, Fig. 2.25=====

=====Ballard and Brown, *Computer Vision*, Fig. 2.26=====

16. object classification with moments invariant to rotation, scaling, translation, Gonzalez, p. 514, Sec. 8.3.4

=====Gonzalez, *Digital Image Processing*, Eq. 8.3-14=====

=====Gonzalez, *Digital Image Processing*, Fig. 8.24=====

17. photometric stereo, Sec. 12.3, p. 16, Eq. 12.14
18. shape from focus, defocus, Sec. 12.4.1, p. 21
19. shape from polarization, Sec. 12.5, p. 22
20. shape from shading, Horn, p. 226

=====Horn, *Robot Vision*, Fig. 10.18=====

=====Horn, *Robot Vision*, Fig. 10.19=====

21. shape from texture, Nalwa, p. 199
=====Nalwa, *A Guided Tour of Computer Vision*, Fig. 6.1=====
=====Nalwa, *A Guided Tour of Computer Vision*, Fig. 6.6=====
22. solving correspondence problem or optic flow field
=====truck.im=====
23. motion and shape parameter recovery
=====truck.im=====
24. segmentation of newspaper, documents into title, figure, caption,....
=====magazine.im=====
25. optical distortion correction, Nalwa, p. 46
=====Nalwa, *A Guided Tour of Computer Vision*, Fig. 2.13=====
26. line labeling of 2D line drawing of 3D objects, Sec. 17.5, p. 404
=====Fig. 17.13=====
27. Computer Tomography: 3D image reconstruction from 2D projections
=====Cho *et al.*, *Foundations of Medical Imaging*, Fig. 6.6=====
=====Wicke, *Atlas of Radiologic Anatomy*, Fig. 159=====
=====Wicke, *Atlas of Radiologic Anatomy*, Fig. 160=====
28. road surface inspection: cracks, holes, ...
29. traffic counting: number of trucks, cars, motorcycles, ...
30. finger-print validation
=====fingerprint.im=====

31. face recognition
=====face=====
32. use robot to pick stones out of a bin
=====stone.im=====
33. digital morphing
=====morphing.im=====
34. model-based diagnosis
=====thorax.im=====
35. automatic identification and classification of video scenes for indexing
36. automatic break detection or video partitioning or scene change detection
37. automatic full-video search for objects of interest
e.g. find all frames which contain frogs
38. creating video or image database with functions
such as efficient querying, indexing, retrieval, browsing
39. recognition of video content invariant to viewing conditions
e.g. find all other shots of this scene
40. printed music sheet recognition and translation into MIDI format file
=====music.im=====
41. 360 degree image from several pictures (alignment, color interpolation, panorama)
42. wafer defect inspection
=====wafer.defect=====

43. wafer critical dimension measurement

=====Elliott, *Integrated Circuit Fabrication Technology*, Fig. 8.54=====

44. IC pin inspection: coplanarity of surface mount device
    colinearity of dual-in-line package

=====IC.pin=====

45. IC mark printing inspection: smear, contract, scratch, ...

=====IC.mark=====

46. electrical contact point inspection

=====electrical.contact=====

47. digital watermarking: viewing but no printing due to copyright
48. auto focus in digital camera (edge, high frequency, active, passive)
49. auto exposure in digital camera (shutter speed, aperture diameter, AGC: Auto Gain Control)
50. auto white balance in digital camera (candle light, fluorescent light)
51. color management: sRGB: standard Red, Green, Blue
52. 640X480 $\Rightarrow$ 1280X960 from single image
53. super resolution: 640X480 $\Rightarrow$ 1280X960 from multiple images
54. video stabilization for digital camcorder
55. Sony DSC F828 Digital Camera Contest (RGBE: Red, Green, Blue, Emerald)

Project due April 12:

camera calibration i.e compute #pixels/mm object displacement
calculate field of view in angles.
calculate and compare with theoretical values.
use lens of focal length: 16mm, 25mm, 50mm
object displacement of: 1mm, 5mm, 10mm, 20mm
object distance of: 0.5m, 1m, 2m
camera parameters: 8mm$\times$ 6mm $\Rightarrow$ 512$\times$485?
Are pixels square or rectangular?