Motto

It hurts but it is definitively healthy

(pediatrician of my daughter)

Introduction to mathematics for computer graphics

...that should be a review of very well known things

Mathematics for CG

Geometry.

Deals with the position of the objects.

So the question is "*where it is?*"

Deals with coordinates.

Topology.

Answers the question how the things are connected

From this viewpoint circle and ellipse are equal

Vector Space

Def. An n-dimensional ***vector space*** consists of a set of *vectors* and two operations: addition and scalar multiplication. The vector space is closed under these two operations. There exists a distinguished member of the set called *zero vector* **O** that has the following properties:

a.**O** = **O** for all scalars a

O+**v**=**v**+**O**=**v** for all vectors **v**

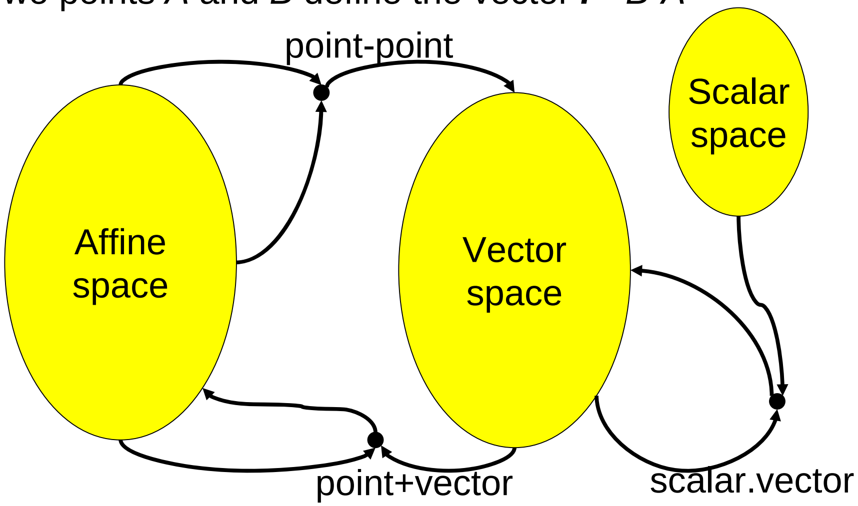
Affine Space

Def. An n-dimensional ***affinne space*** consists of a set of *points*, an associated n-dimensional vector space, and two operations: subtraction of two points in the set and addition of a point and a vector in the associated vector space. The subtraction results in a vector whereas the addition results in a point.

- ✓ *A vector* is expressed by its coordinates ***P***=(x,y,z)
- ✓ *A point* is expressed by its coordinates *A*=[x,y,z]

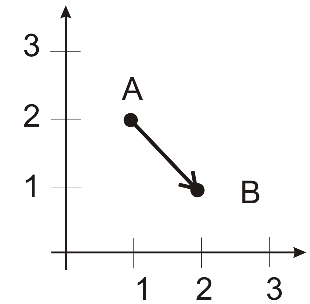
Spaces

Two points *A* and *B* define the vector ***P***=*B*-*A*



Spaces

Let's have two points *A* and *B*
The vector ***P*** is the ***P***=*B*-*A*



<u>*example:*</u>

A=[1,2,0], *B*=[2,1,0]

P= *B*-*A* = [2,1,0] - [1,2,0] = (1,-1,0)

note that the same vector can be obtained

e.g., from *A*=[0,0,0], *B*=[1,-1,0]

and infinite number of different pairs!

Spaces

A vector is a representative of a *set* of oriented lines

In other words a vector can be translated!

A vector added to point yields to a new point

<u>*example:*</u>

A=[1,2,0], ***P***=(1,-1,0)

A+***P***=[1,2,0] + (1,-1,0) = [2,1,0]

Spaces

In general, the operation POINT+POINT is undefined

but there are exceptions:

C=(A+B)/2 is the mid point of a line segment

note: In general the operation $\sum a_i A_i$ is defined iff

$\sum a_i = 1$ or $\sum a_i = 0$

in the first case the result is a point,

in the second one a vector

Spaces

The operation

$\sum a_i A_i$

is called

Affine combination

iff

$\sum a_i = 1$

Affine combination results in a point

Space

Example:

Write a procedure that evaluates mid-point of a line

```
float *MidPoint(float *a, float *b, float *ret)
{
  ret[0]=(a[0]+b[0])/2.f;
  ret[1]=(a[1]+b[1])/2.f;
  ret[2]=(a[2]+b[2])/2.f;
  return ret;
}
```

note: a cycle will be slow!

Spaces

Example:

Linear interpolation

let's have two points *A* and *B*
and the parameter $0 \le t \le 1$

the point *Q* inside is

$Q=(1-t)A+tB$

note: ~ t=0, Q=A
~ t=1, Q=B
~ t=0.5; Q=mid point
~ (1-t)+t=1, Q is a point!

Space

Example:

Write a procedure of linear interpolation

```
float *Interpolate(float *a, float *b, float t, float *ret)
{
  static float t1=1-t;
  ret[0]= t1*a[0]+t*b[0]
  ret[1]= t1*a[1]+t*b[1]
  ret[2]= t1*a[2]+t*b[2]
  return ret;
}
```

note: a cycle will be slow!

Mathematics for CG

Def:
An affine space with an additional concept of metrics is called Euclidean space

$|\mathbf{u}| = \sqrt{\mathbf{u}.\mathbf{u}}$

Example:
procedure that returns distance of two points

```
float Dist(float *a, float *b)
{return sqrt(a[0]-b[0])*(a[0]-b[0])+
             (a[1]-b[1])*(a[1]-b[1])+
             (a[2]-b[2])*(a[2]-b[2]));}
```

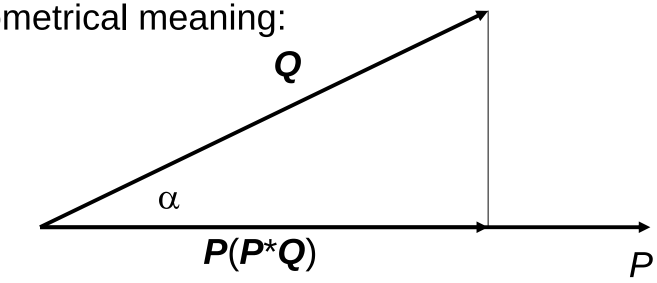
note:very bad…
many things repeat,
macro or inline is better

The Dot Product

- *The dot product* of two vectors is a number

$\boldsymbol{P}^*\boldsymbol{Q}=|\boldsymbol{P}|\ |\boldsymbol{Q}| \cos\alpha$,
where the α is angle between the ***P*** and ***Q***

Geometrical meaning:



P(***P********Q***) is the perpendicular projection of the vector ***Q*** onto the vector ***P***

The Dot Product

having two vectors

P and ***Q***

the dot product is calculated as

$\boldsymbol{P}.\boldsymbol{Q}=(P_xQ_x+ P_yQ_y +P_zQ_z)$

or:

dot=P[0]Q[0]+P[1]Q[1]+P[2]Q[2];

The Dot Product

- Why is it so important?

Example:

Having three points
A=[1,0,0], *B*=[1,1,1], and *C*=[-1,1,-1].
Are the lines *AB* and *AC* perpendicular?

Yes, if the vectors ***AB*** and ***AC*** are perpendicular.

AB=*B*-*A*=[1,1,1] -[1,0,0]=(0,1,1)
AC=*C*-*A*=[-1,1,-1] -[1,0,0]=(-1,1,-1)

AB.AC=0(-1)+1.1+1(-1)=0

yes they are….

Other useful operations

Size of a vector

$|\boldsymbol{P}| = \sqrt{(P_x^2 + P_y^2 + P_z^2)}$

☯ this is exactly distance of two points!

$|A,B| = \sqrt{((B_x - A_x)^2 + (B_y - A_y)^2 + (B_z - A_z)^2)}$

Normalized vector is a vector of the unit size

this can be easily achieved

$\boldsymbol{P}' = \boldsymbol{P}/|\boldsymbol{P}|$

do not confuse it with *the normal vector*!

The Dot Product

☯ How can we get the angle between two lines?

1) Get the vectors ***P,Q***
2) Normalize them, ***P'***, ***Q'***
3) Get the dot product $\boldsymbol{P'}*\boldsymbol{Q'}=|1||1|\cos\alpha$
4) arccos (***P'***.***Q'***) is the angle

The Dot Product

Example:

take vector ***P***=(10,0,0) and ***Q***=(0,-5,0)
normalize them and calculate the dot product

P' = (1,0,0), ***Q'*** = (0,-1,0)

$\boldsymbol{P'} \cdot \boldsymbol{Q'} = 1\ 0 + (-1)\ 0 + 0\ 0 = 0$

why?

The Vector Product

the vector product

the vector product of two vectors P and Q is a vector

$\boldsymbol{R} = \boldsymbol{P} \times \boldsymbol{Q} = (P_yQ_z-P_zQ_y, -(P_xQ_z-P_zQ_x), P_xQ_y-P_yQ_x)$

this gives vector perpendicular to ***P*** and ***Q***,
or zero vector if ***P*** is parallel to ***Q*** or
zero vector if ***P***=***o*** or ***Q***=***o***

PQR form a basis of the coordinate system!
(The vector space)

The Vector Product

the vector product

easy way to remember the formula

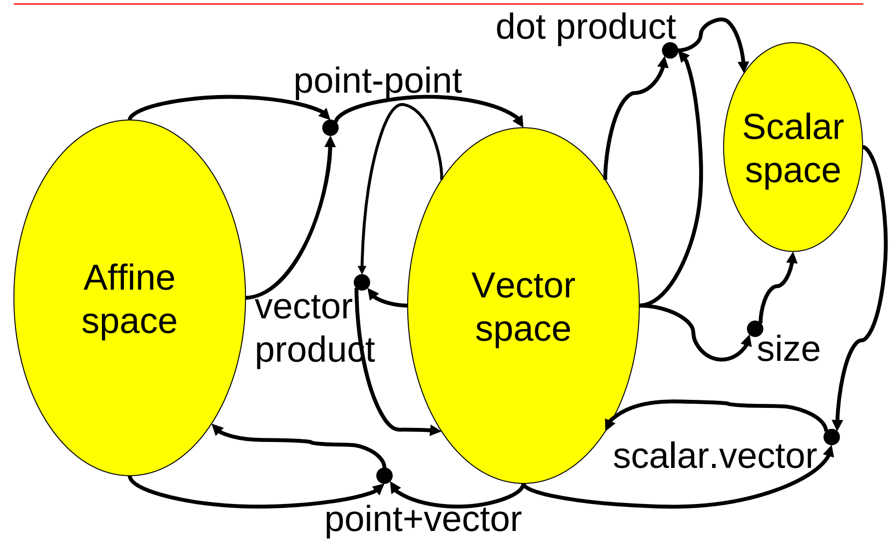
$$\boldsymbol{R} = \boldsymbol{P} \times \boldsymbol{Q} = \begin{vmatrix} R_x & R_y & R_z \\ P_x & P_y & P_z \\ Q_x & Q_y & Q_z \end{vmatrix}$$

Note: it is NOT commutative, i.e., ***P***x***Q*** != ***Q***x***P***

in fact

$-\boldsymbol{R} = \boldsymbol{Q} \times \boldsymbol{P}$

Mathematics for CG



Line

Parametric equation of a line

Defined by two points *A* and *B*
its equation is

$P(t) = [x(t), y(t), z(t)] = A + t(B-A); \quad -\infty < t < \infty$

there are special cases:

$P(0) = A \quad P(1) = B$

Note also:

Point+tVector -> point

Line (contd.)

example:

A=[1,2,3], *B*=[0,1,6]

P(t)=[x(t), y(t), z(t)] =
= [1,2,3] + t ([0,1,6] - [1,2,3])=
= [1-t, 2-t, 3+3t]

try

t=0 *P*(t) = [1,2,3]

t=1 *P*(t) = [1-1,2-1,3+3] = [0,1,6]

Line (contd.)

Can be also thought of as a blending function

x(t) = (1-t)*X1 + t*X2

y(t) = (1-t)*Y1 + t*Y2

z(t) = (1-t)*Z1 + t*Z2

this *linear blending* is frequently used in CG:

$P(t) = A + t(B-A)$

$P(t) = (1-t)\,A + t\,B$

Note: $(1-t)+t=1$

Line (contd.)

Why the explicit form is not good for computer graphics?

Line is defined

y = mx + b

☯ cannot represent vertical lines

☯ cannot represent line segments (only infinite lines)

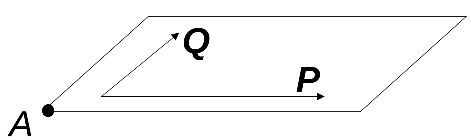
☯ cannot represent lines in 3D (!)

Plane

Parametric equation of a plane

Given by:
1) Two vectors ***P,Q*** inside the plane and a point *A*

$R(u,v)=[x(u,v),\ y(u,v),\ z(u,v)] = A+u\mathbf{P}+v\mathbf{Q}\quad -\infty<u,v<\infty$



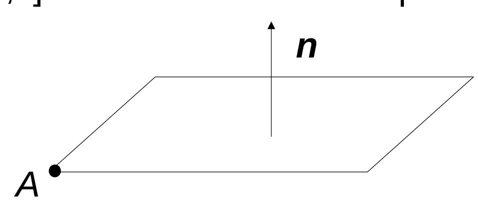
Plane (contd.)

Parametric equation of a plane

Given by:
2) The normal vector ***n*** and a point *A*

any point $X=[x,y,z]$ is the member of this plane iff

$(X-A)*\mathbf{n}=0$



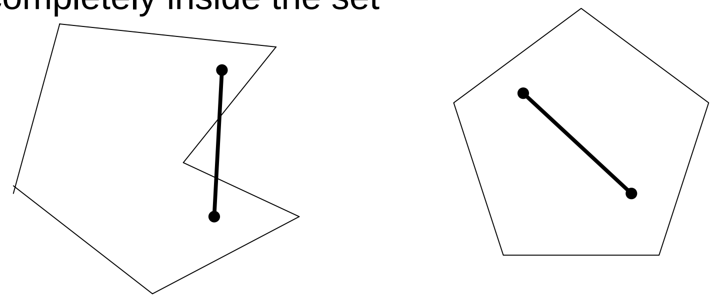
why?
X-A is a vector, A is inside the plane,
if n.(*X-A*)=0, they are orthogonal

Convex vs. Concave

Convex vs Concave

convex set:

any line connecting two points of the set lies completely inside the set

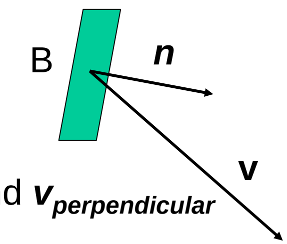


concave convex

Applications

Fast mirroring

Plane is defined by a point *B* and normalized normal vector ***n***



We want to mirror vector ***v***

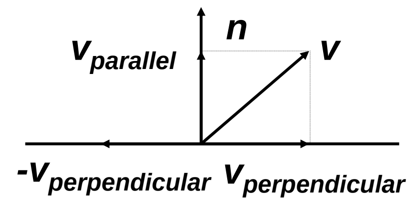
v can be expressed as $\boldsymbol{v}_{parallel}$ and $\boldsymbol{v}_{perpendicular}$

1) $\boldsymbol{v}_{parallel} = \boldsymbol{n}\ (\boldsymbol{n} * \boldsymbol{v})$

2) $\boldsymbol{v}_{perpendicular} = \boldsymbol{v} - \boldsymbol{v}_{parallel}$

then the mirror reflection is

3) $\boldsymbol{v}_{mirror} = \boldsymbol{v}_{parllel} - \boldsymbol{v}_{perpendicular}$

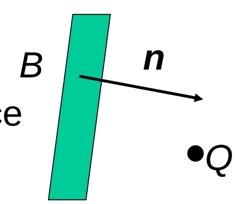


Applications

Distance of a point from a plane

Plane defined by the point *B* and normalized normal vector ***n***
We want to compute distance of the point Q

1) Express vector $\boldsymbol{v} = (Q\text{-}B)$
2) Evaluate projection of ***v*** onto ***n***
3) Size of this vector is the distance
dist = $\boldsymbol{n}.(Q\text{-}B)$why?



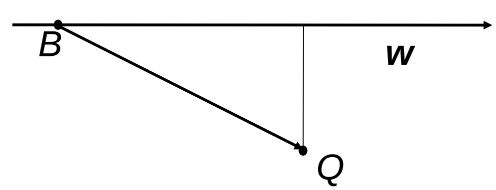
$\mathbf{n}.\mathbf{v} = \cos\alpha \, . \, |\mathbf{n}||\mathbf{v}| = \cos\alpha \, |\mathbf{v}|$
i.e., projection of the **v** onto **n**

Mathematics for CG (contd.)

Distance of a point from a line

Line is defined by point *B* and normalized vector *w*
We want to compute distance of the point *Q*

1) Express vector $\boldsymbol{v} = (Q\text{-}B)$
2) Decompose ***v*** to $\boldsymbol{v}_{parallel}$ and $\boldsymbol{v}_{perpendicular}$
3) Size of the vector $\boldsymbol{v}_{perpendicular}$ is the distance



Geometric transformations

Why?

✓ Instancing:

an object is represented only once and its copies are expressed as its transforms this saves space!

✓ Hierarchical modeling:

we can group a collection of geometry under a transform node and manipulate it all at once

Geometric transformations

<u>Properties of transform:</u>

✓ compact representation (matrices)

✓ fast implementation (HW support)

✓ easy to invert (inverse matrix)

✓ easy to compose (matrix multiplication)

Coordinates

Transformations of a single point
Transform of an object is a transforms of all its points.

Point is expressed by its coordinates P=[x,y,z] but in CG we usually use *homogenous coordinates*

the homogenous coordinates of a point P are
P=[X,Y,Z,W],
where
x=X/W, y=Y/W, z=Z/W

we usually put W=1

Geometric transformations (contd.)

<u>there are several advantages of this:</u>

- we can express affine transforms (see bellow) as one matrix
- we can express projections as one matrix as well
- transforms can be composed by matrix multiplication
- compact representation of points and vectors

points $W \neq 0$
vectors $W=0$

Geometric transformations (contd.)

example:
point P has homogenous coordinates [2,4,12,2].
What are its Cartesian coordinates?
They are $P=[2/2, 4/2, 12/2]=[1,2,6]$

example:
Point has Cartesian coordinates [1,2,3]

What are its homogenous coordinates?

$[1w,2w,3w,w]$ and $w \neq 0$

for example [1,2,3,1], [2,4,6,2],etc.

Geometric transformations (contd.)

Transformations:
linear (translation, scale, rotation, shear)
non-linear

Linear are expressed as a matrix 3x3, affine by 4x4

$$\mathbf{M} = \begin{bmatrix} m11 & m12 & m13 & m14 \\ m21 & m22 & m23 & m24 \\ m31 & m32 & m33 & m34 \\ m41 & m42 & m43 & m44 \end{bmatrix}$$

P is transformed to P' by multiplication by the matrix

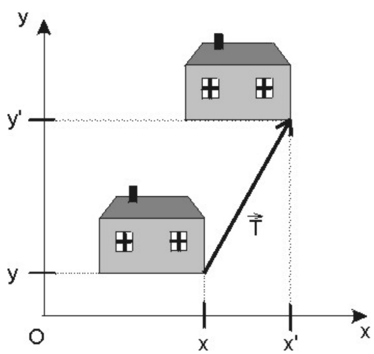
$P'^T=\mathbf{M}\,P^T$ i.e.,

$$P'=[X',Y',Z',W']=[m_{11}X+m_{12}Y+m_{13}Z+m_{14}W,\ m_{21}X+m_{22}Y+m_{23}Z+m_{24}W,\ m_{31}X+m_{32}Y+m_{33}Z+m_{34}W,\ m_{41}X+m_{42}Y+m_{43}Z+m_{44}W]$$

Geometric transformations - Translation

Translation:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & X_T \\ 0 & 1 & 0 & Y_T \\ 0 & 0 & 1 & Z_T \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



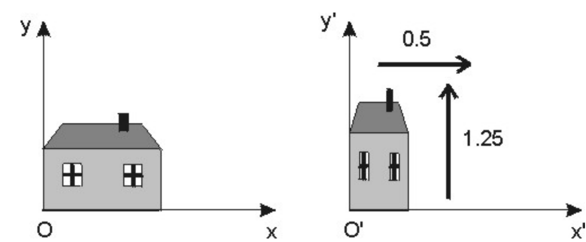
what is in fact:
$X'=X+X_T$
$Y'=Y+Y_T$
$Z'=Z+Z_T$
$W'=1$

The point is moved in the direction of the vector ***T***
(Remember point+vector yields to point)

Geometric transformations - Scale

Scale:

$$\mathbf{M} = \begin{bmatrix} S_X & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

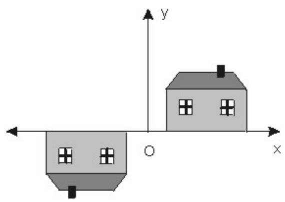


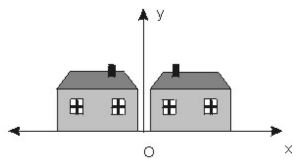
what is in fact:
$X'=S_X X$
$Y'= S_Y Y$
$Z'= S_Z Z$
$W'=1$

Geometric transformations - Symmetry

Symmetry:
is a special case of scale for
some of the coefficients

$S_X = -1$
$S_Y = -1$
$S_Z = -1$





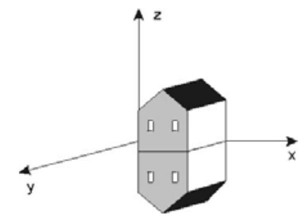
Axis symmetry:
$S_X = -1\ X$
$S_Y = -1\ Y$
$S_Z = -1\ Z$

Geometric transformations - Symmetry

Plane symmetry:
$S_X = -1$ and $S_Y = -1$ symmetry according to the plane *xy*
$S_X = -1$ and $S_Z = -1$ symmetry according to the plane *xy*
$S_Y = -1$ and $S_Z = -1$ symmetry according to the plane *yz*

Symmetry according to the origin:

$S_X = S_Y = S_Z = -1$



Geometric transformations - Shear

Shear in direction

YZ

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ SH_Y & 1 & 0 & 0 \\ SH_Z & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

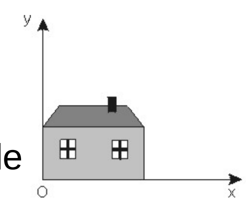
XZ

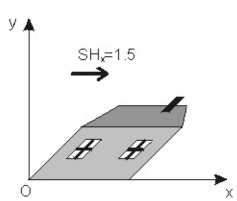
$$\mathbf{M} = \begin{bmatrix} 1 & SH_X & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & SH_Z & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

XY

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & SH_X & 0 \\ 0 & 1 & SH_Y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

two dimensional example





shear in the direction of the *X* axis

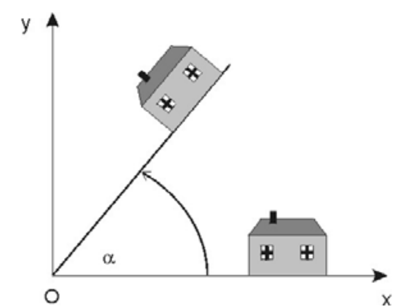
Geometric transformations - Rotation

Around axis *X* of angle α

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha & 0 \\ 0 & \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Around axis *Y* of angle α

$$\mathbf{M} = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha & 0 \\ 0 & 0 & 0 & 0 \\ \sin\alpha & 0 & \cos\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



Around axis *Z* of angle α

$$\mathbf{M} = \begin{bmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

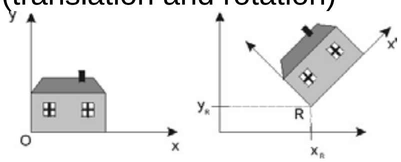
note that $\sin\alpha$ and $\cos\alpha$
can be precomputed!!

Geometric transformations - Example

Define matrix for rotation around line Q=[10, 20, t] and α

It is rotation about axis *Z* but moved to the point [10, 20, 0]
Its is composition of two matrices (translation and rotation)

1) Move point to the origin
2) Rotate about axis *Z* and α
3) Move back



Let **T** be the matrix for translation to the origin
Let **R** be the matrix for rotation around the *Z*
and let $\mathbf{T}^{-1}$ be the matrix for inverse translation

$\mathbf{M} = \mathbf{T}\,\mathbf{R}\,\mathbf{T}^{-1}$

We are multiplying from the right,
so the order of the matrices is exactly the same
as the order of the performed operations

Projections

Projections transform point from m-dimensional space into n-dimensional space and m>n

We are interested in m=3 and n=2 i.e.,
i.e., from 3D space to the plane

we need to define two things:

center of projection (COP)

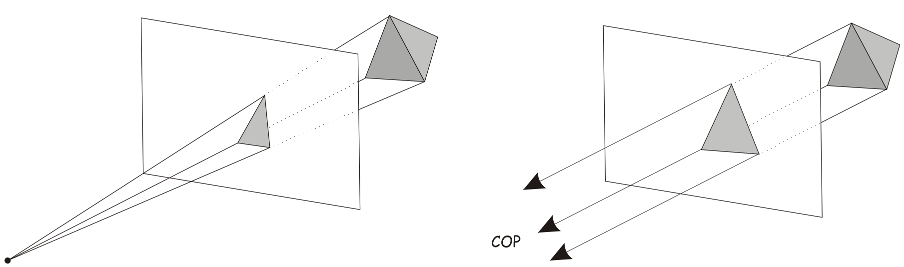
plane of projection (PP)

Projections

There are two basic types of projections

✓ **Perspective** projection:
COP is in finite distance from PP

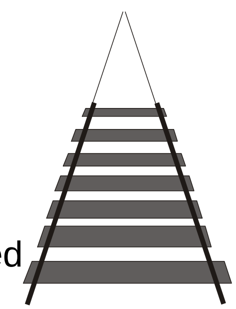
✓ **Parallel** projection:
distance between COP and PP is infinite



Projections (contd.)

Properties:

Perspective projection:
~ Size of the projected objects varies with distance from PP
~ It looks realistic
~ Distances and angles are not preserved
~ Parallel lines do not remain parallel
~ Lines that are not parallel to PP meet at **vanishing point**
~ Center of a line is not projected as a center of a projected line
~ Used in photorealistic rendering



(for me)
Do not forget the joke

Projections (contd.)

Parallel projection:

~ Size of the projected object
  does not depend on distance from PP
~ Good for exact measurements
~ Parallel lines remain parallel after projection
~ Angles are not preserved
~ Center of a line is projected as
  a center of a projected line
~ Used in CAGD and CAD

Projections (contd.)

Perspective projection example:

Projections (contd.)

Derivation of the equations:

Let *A*=[x,y,z] be the projected point and
PP be the plane z=d

We denote the projected point *A*'=[x', y', d]

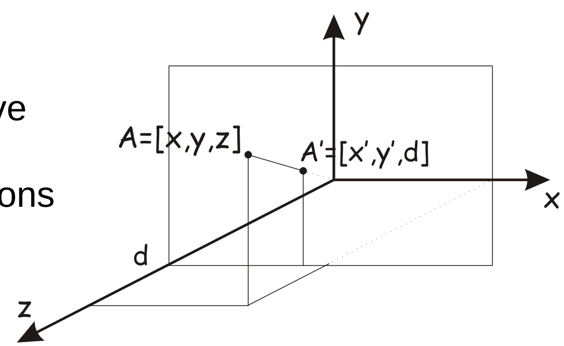
Parallel projection:

A'=[x,y,0] i.e.,
x' = x
y' = y
we simply "forget" the *z* coordinate

Projections (contd.)

Perspective projection:

assume *COP*=[0,0,0]
thus line connecting *A* and *COP* has equation
P(t) = COP + t(*A* - *COP*)
i.e.,
P(t) = [t x, t y, t z]
for plane *xy* we have
d= t z i.e., t= d/z
this leads to equations
x' = x d/z
y' = y d/z



Projections (contd.)

In CG we use many different spaces:
1) *Object space*
used for describing objects
The objects have position and orientation.
Sometimes called the World Space

2) *Camera space*
Corresponds to the camera (or eye).
The camera is usually placed in [0,0,0] and corresponds to COP.
Sometimes called the eye space

Projections (contd.)

3) *Screen space*
result of the projection.
It is usually parallel to the plane *xy* in eye space.
Sometimes called
Normalized Device Coordinate Space (NCD)

4) *Image space*
corresponds to final rasterized image.
It has discrete coordinates - the image resolution

Projections (contd.)

Camera space $\rightarrow$ screen space transformation
camera is usually placed at the origin [0,0,0]
corresponds to COP
screen is placed at distance *d* and is parallel to *xy*
corresponds to PP
the transformation can be expressed as a matrix:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{bmatrix}$$

Projections (contd.)

Camera space$\rightarrow$screen space transformation

if we write the general point transform as
$A' = \mathbf{M}\,A^T$

thus we have:

$A' = [x', y', z', 1] = \mathbf{M}\,[x,y,z,1]^T = [x, y, z, z/d]$

and while it is transformed to
Cartesian coordinates
from homogenous ones we get:

$A' = [x', y', z'] = [x\,d/z, y\,d/z, d]$

Projections (contd.)

Camera space $\rightarrow$ screen space transformation

is there any matrix for parallel projection?

Yes, this one:

$$\mathbf{M} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & d \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

after multiplication

$A' = [x', y', z', 1] = \mathbf{M}\,[x,y,z,1]^T = [x, y, d, 1]$

Mathematics for CG

Reading

- Foley, Van dame, Feiner, Hughes, Philips, *Introduction to Computer Graphics,* Addison-Wesley, 1993
- Rogers, Adams, *Mathematical Elements for Computer Graphics,* McGraw-Hill, 1989
- *Graphics Gems 1-5,* Academic Press