3D Reconstruction

Part I – All I wanted to do was taking a picture...

Elmar Eisemann

Computer Graphics and Visualization

Delft University of Technology

Creating the “Omniverse” for Machine Learning

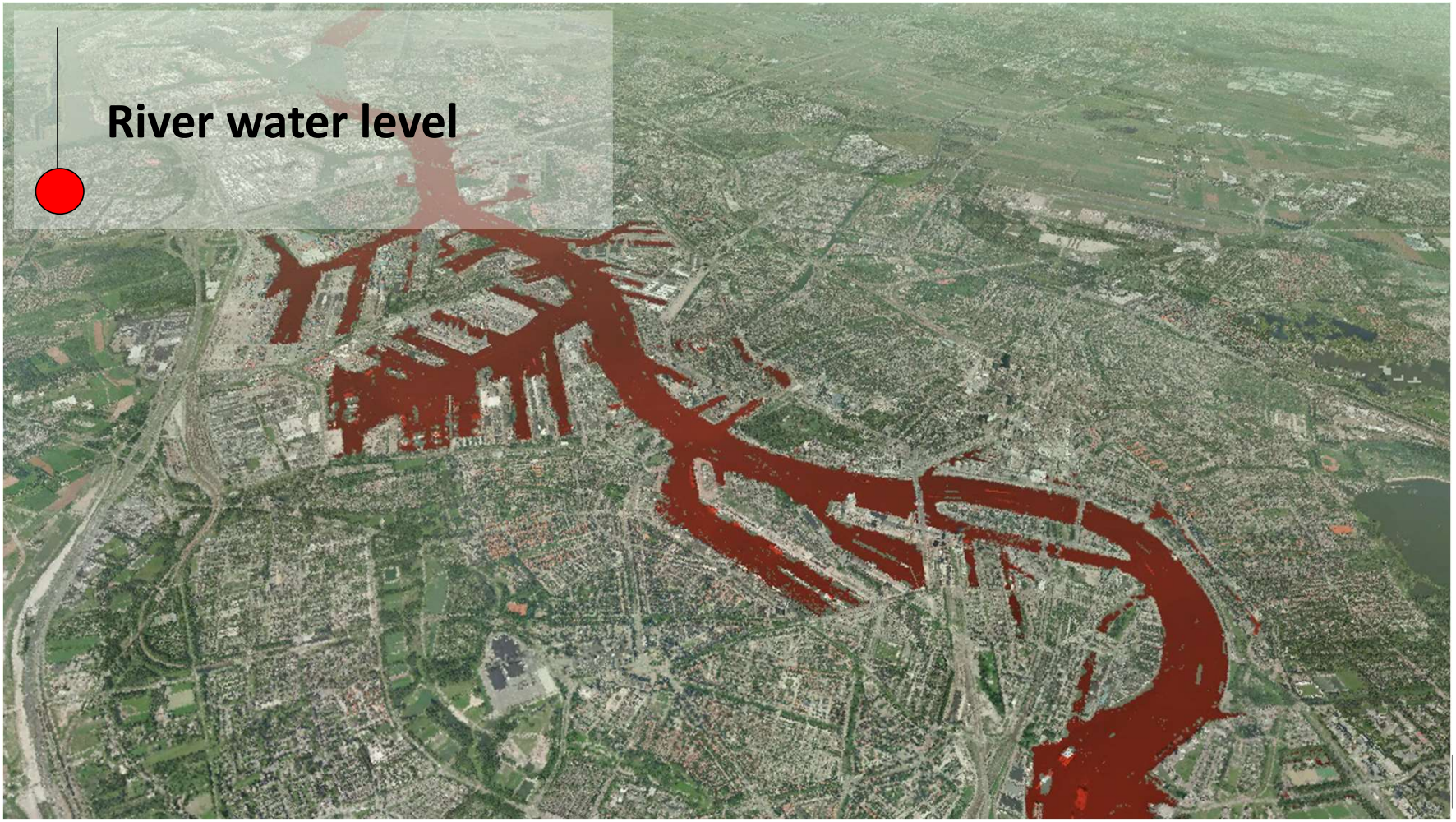
- NVIDIAs vision for future machine learning

D & T FOODS INT'L

D & T FOODS INT'L

Large-Scale Visualization

Models



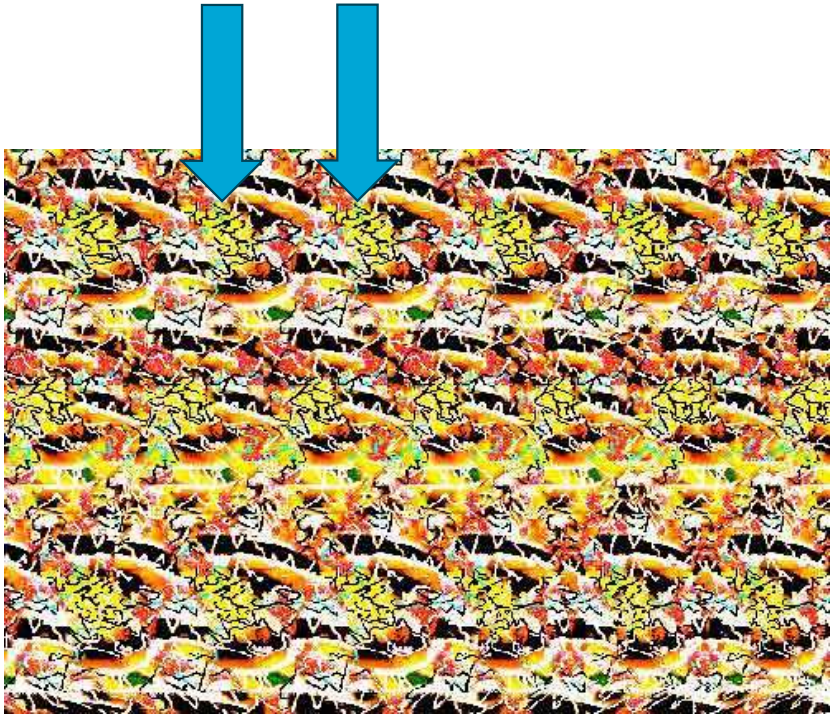
How does 3D Reconstruction work?

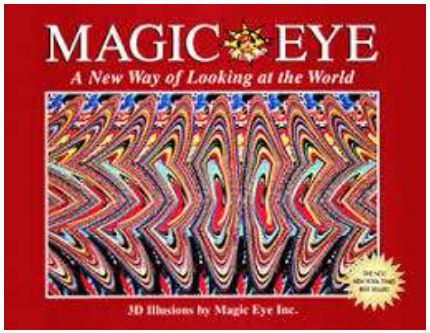
Stereograms

Stereograms

- Stereo pairs reveal 3D information

The Magic Eye

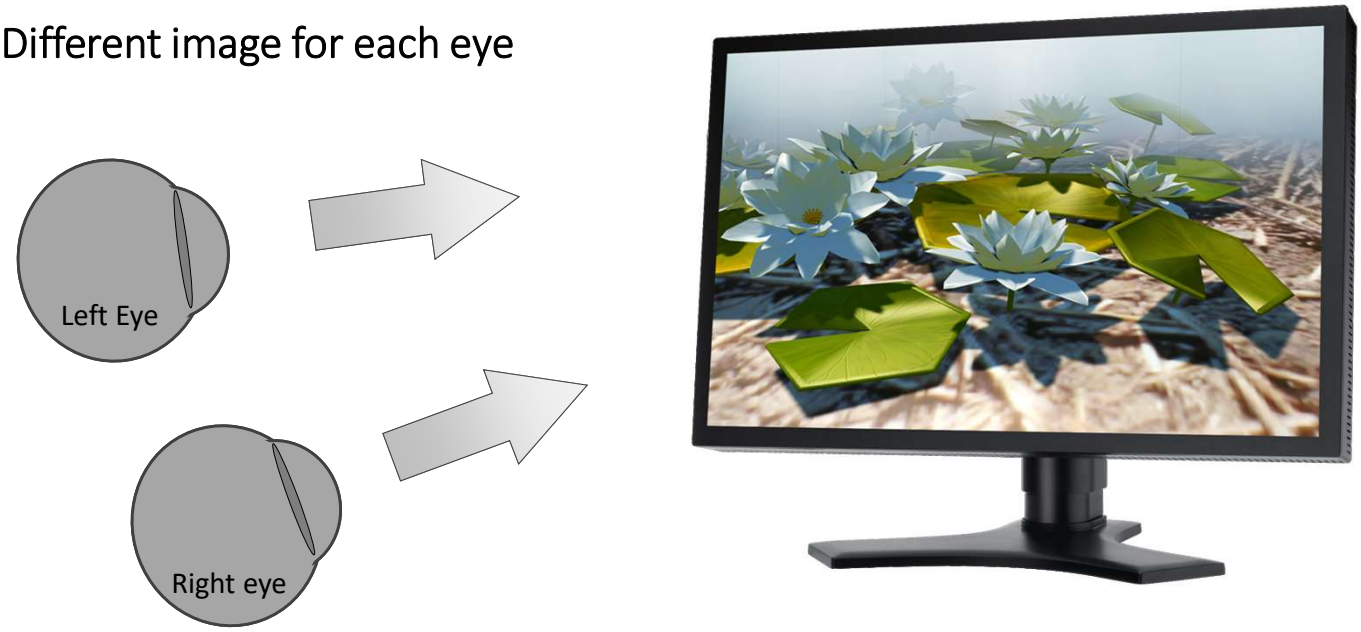




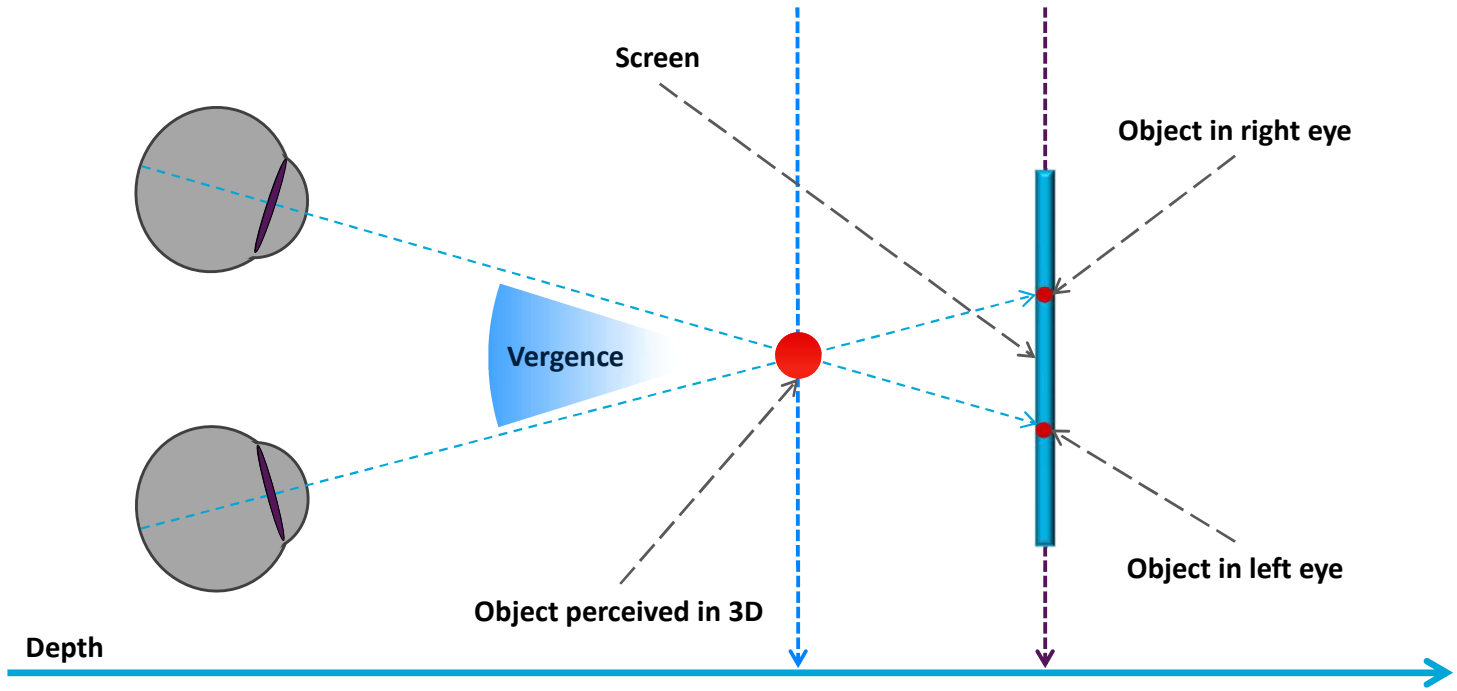
https://www.flickr.com/photos/31690139@N02/2965952081 provided by Magic Eye 6

Stereo Principle

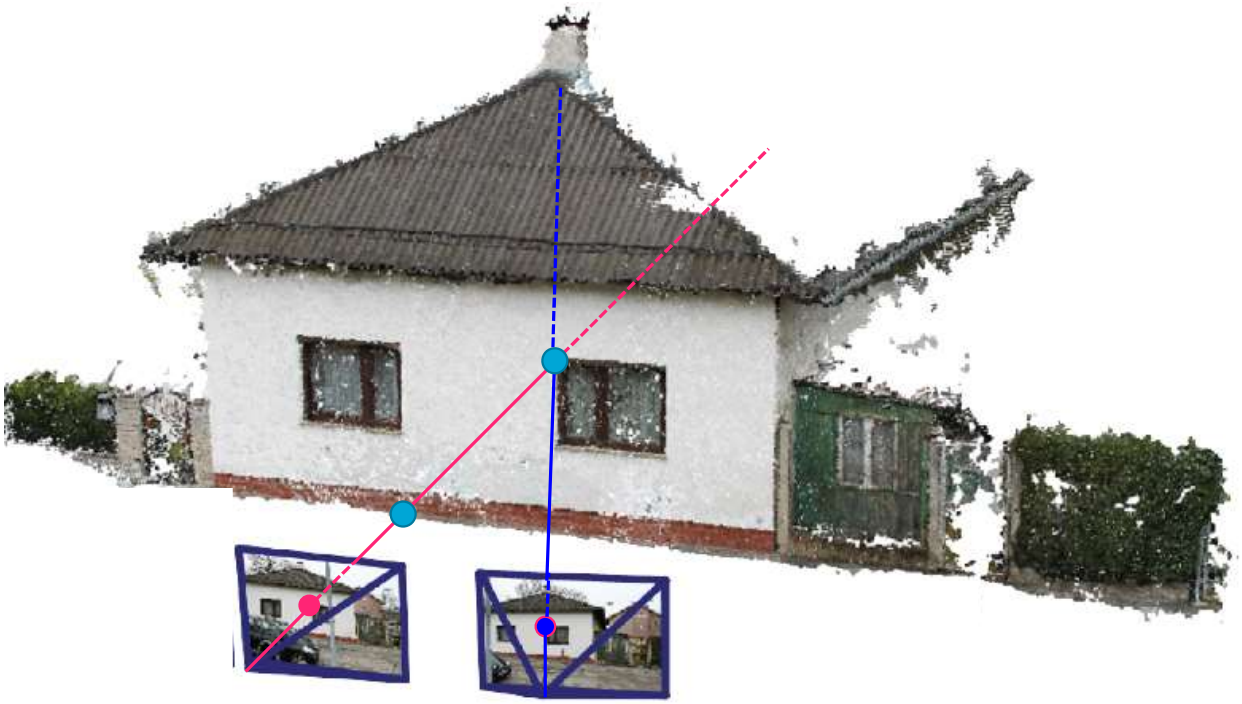
- Different image for each eye



3D Screen

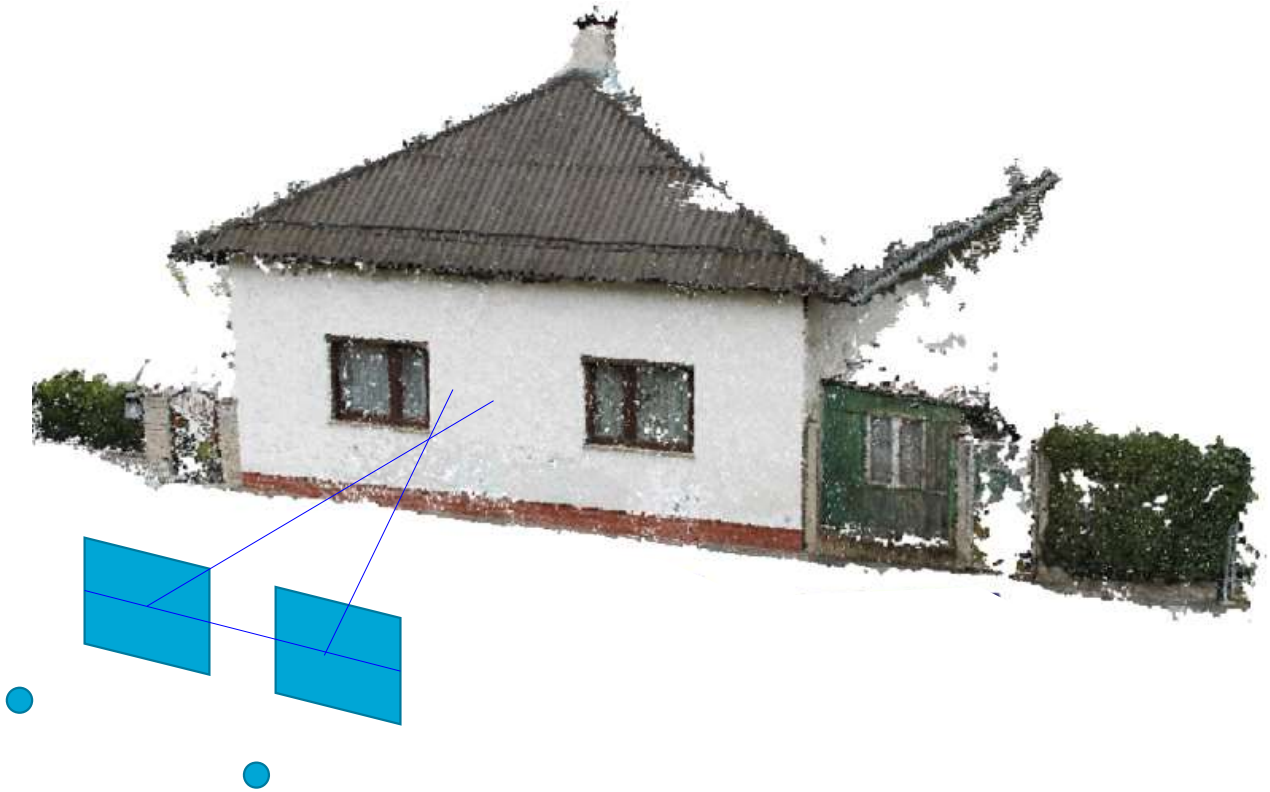

Screen
Object in right eye
Vergence
Object in left eye
Object perceived in 3D
Depth

3D Reconstruction



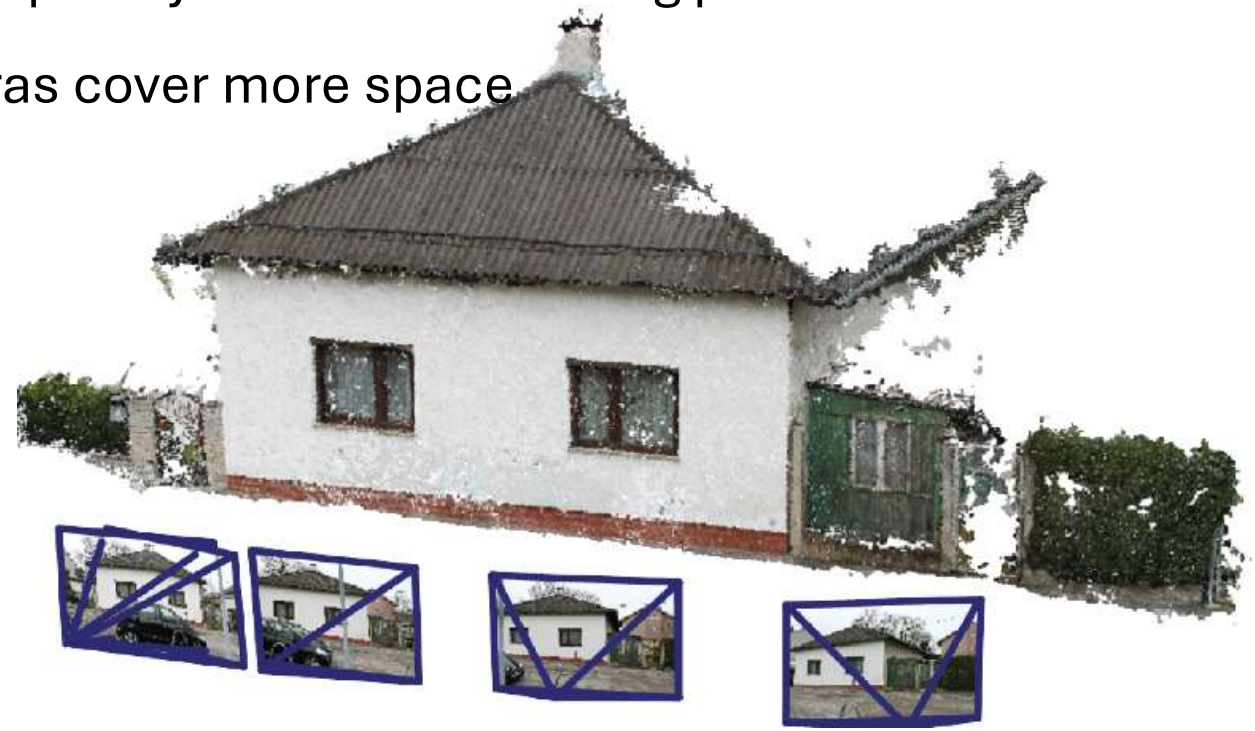
3D Reconstruction

- The easy way… perfect alignment
- Each pixel line = 2D problem



3D Reconstruction

- Usually not perfectly aligned
- For every 3D point you need a matching pair
- More cameras cover more space



3D Reconstruction

- Many cameras:
  Our simple 2D solution breaks down…
- Need better Math

3D Reconstruction

Two Part Lecture!

- How do cameras work?
  - How to model cameras mathematically?
- How to find the pose and calibrate a camera?
- How to reconstruct a scene ?
- How to calibrate from photo collections?

3D Reconstruction

Two Part Lecture!

- How do cameras work?
  - How to model cameras mathematically?
- How to find the pose and calibrate a camera?
- How to reconstruct a scene ?
- How to calibrate from photo collections?

Warning

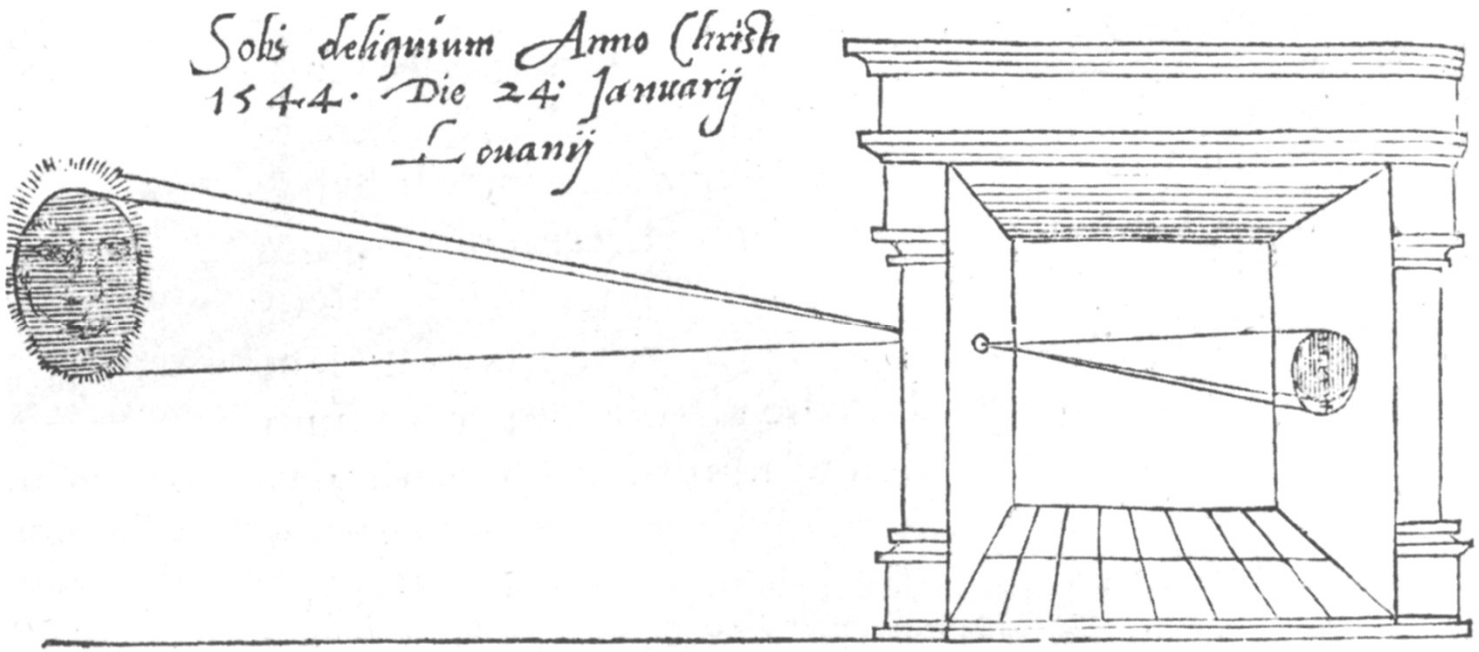
- Focus on the THEORY!
- Presented solutions are NOT OPTIMAL!

- Provide a basis to understand how more sophisticated solutions work

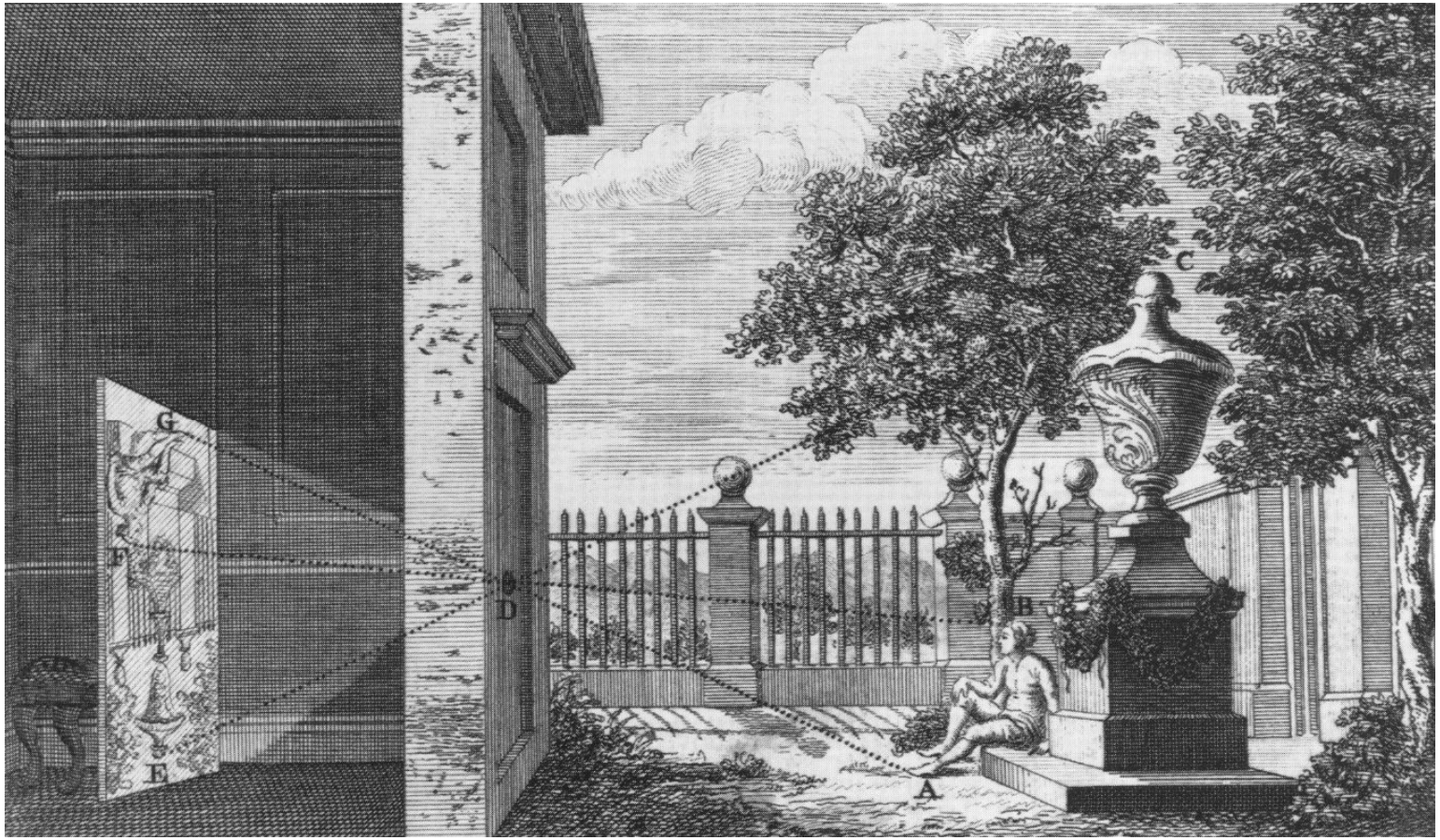
Producing Images in the Real World

Producing Images in the Real World

- R. Gemma Frisius, 1545



Producing Images in the Real World



Camera obscura

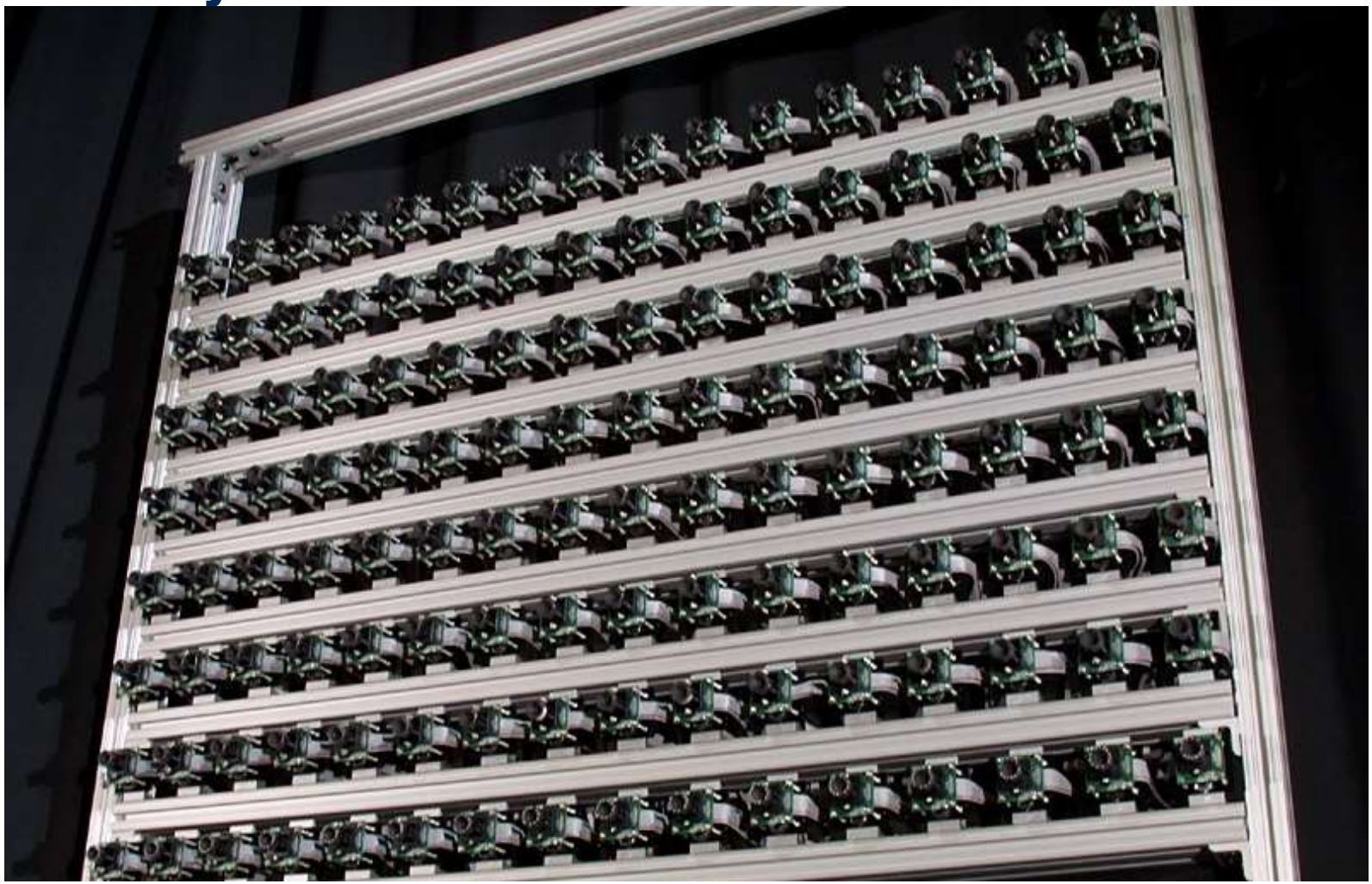
The Great Picture

Gigapixel

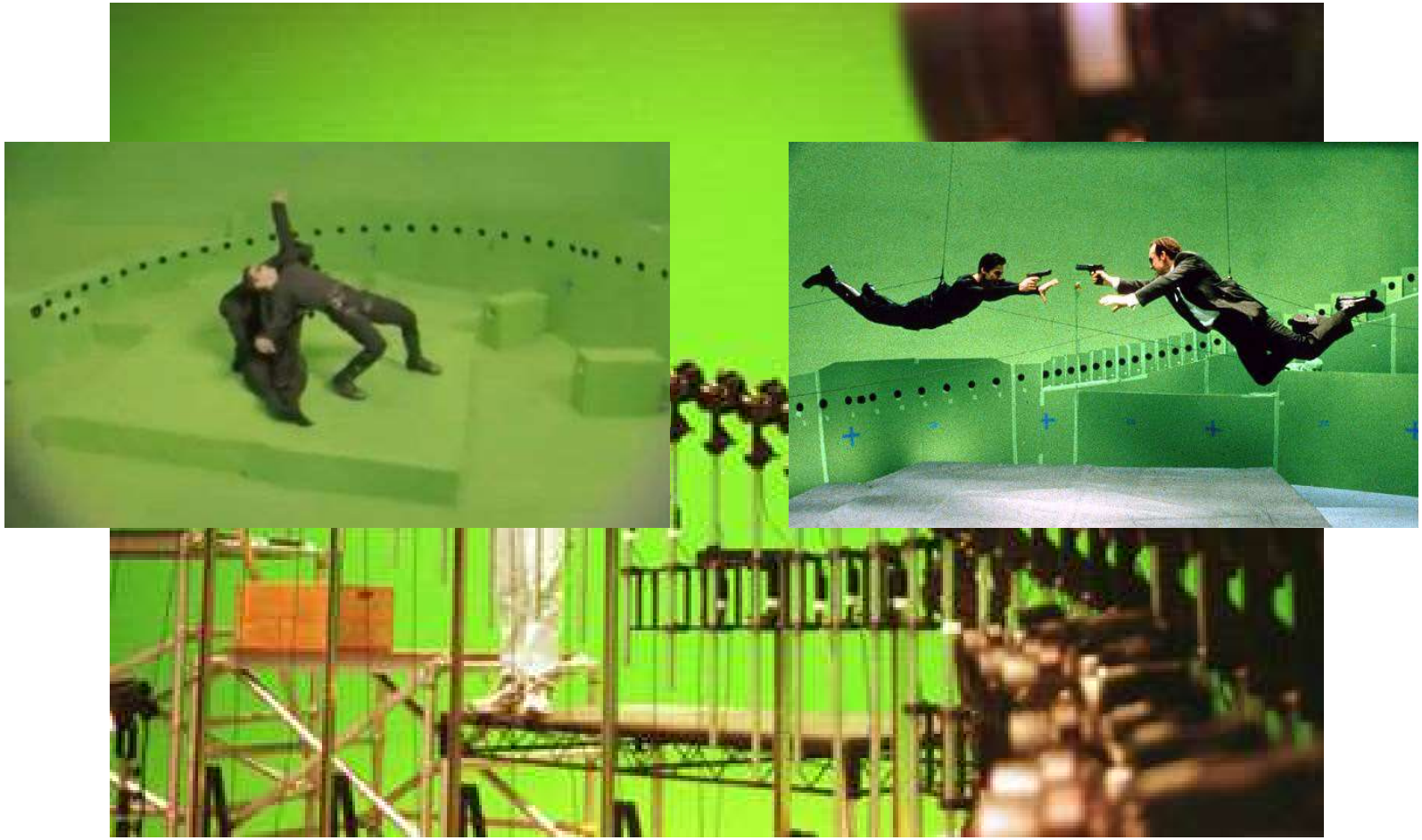
Gigapixel Demo

You already know how to do better than this!
Transition artifacts can be removed …

Camera Arrays

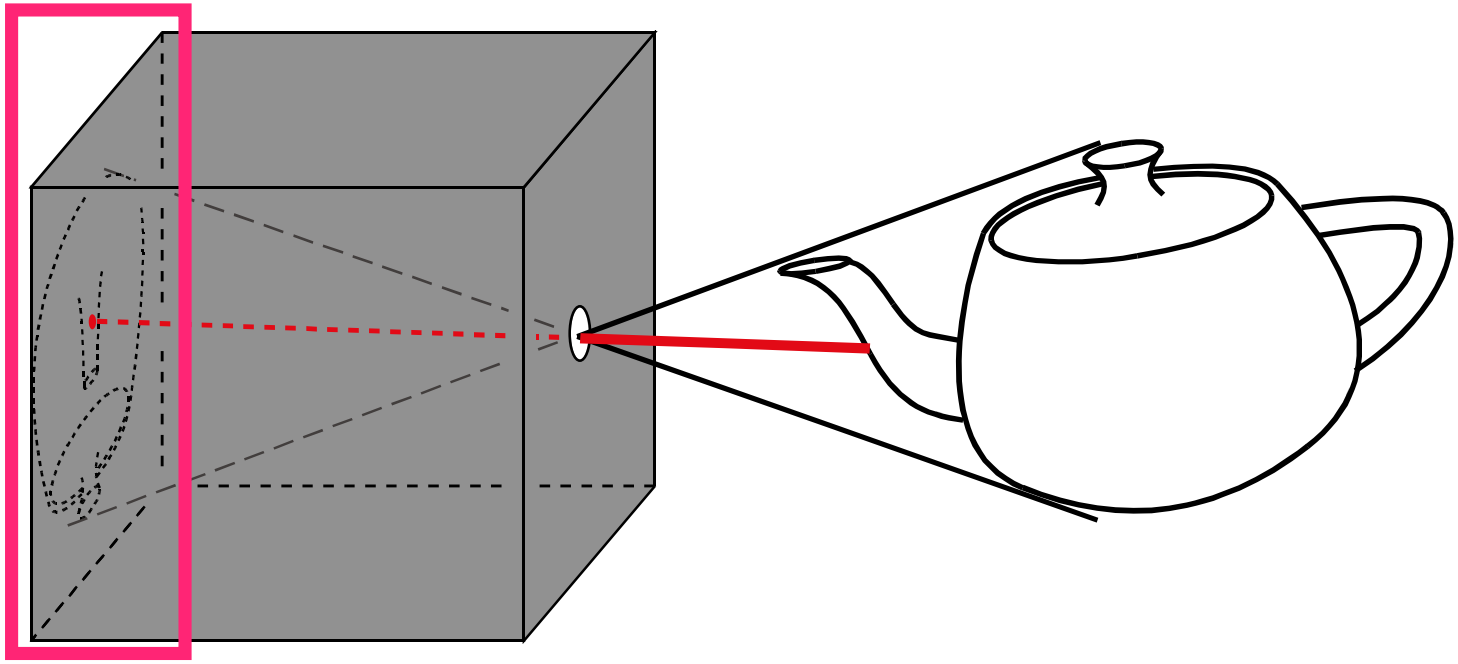


Camera Arrays

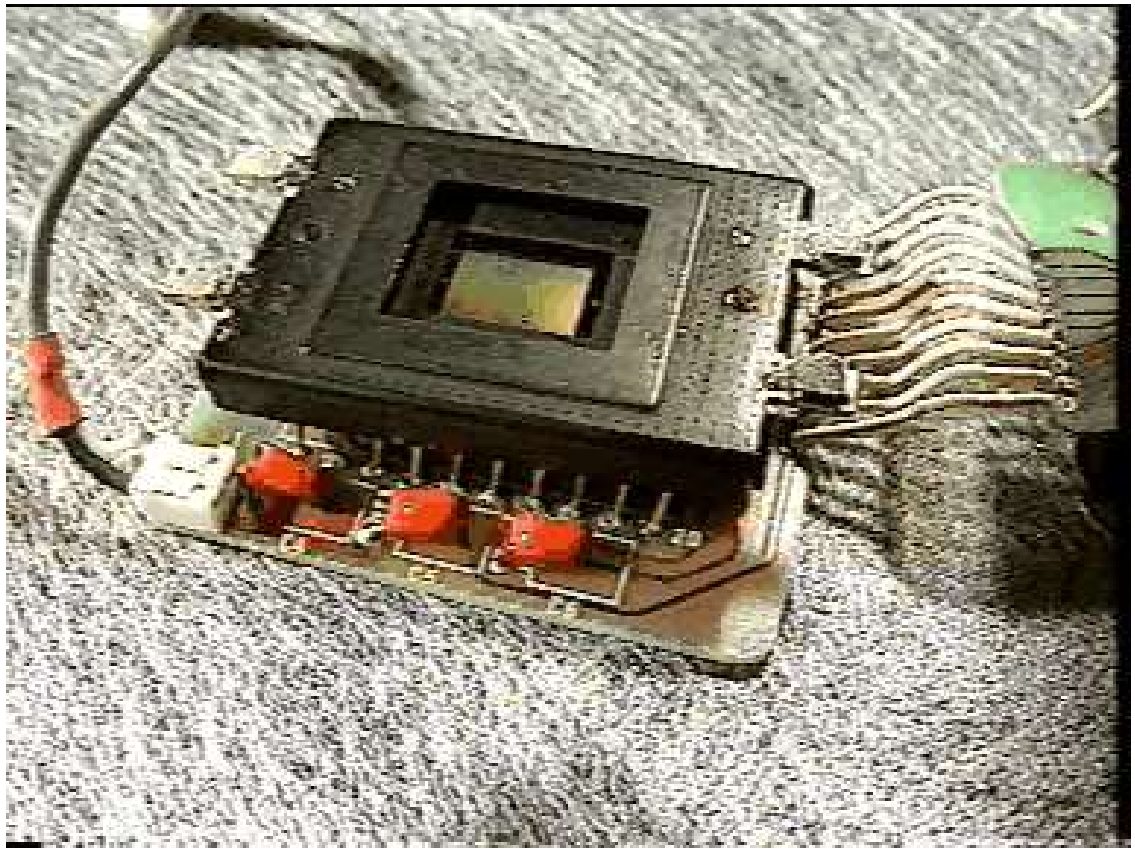


Pinhole camera

- Box with hole
- Perfect image for “point-sized” hole



The CCD camera



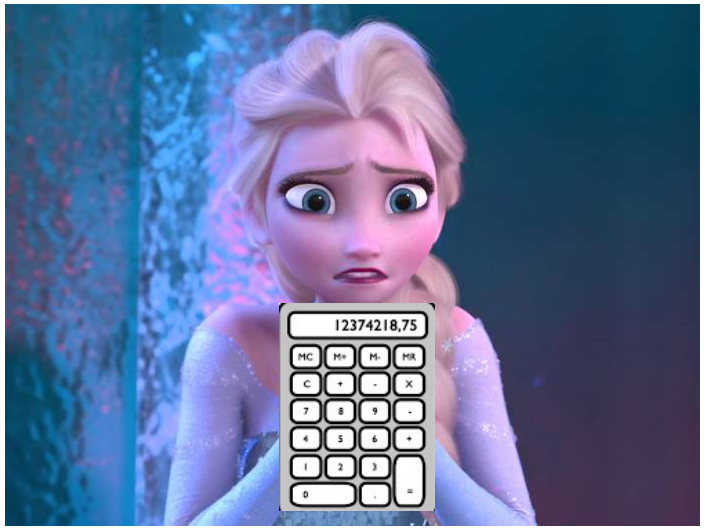
Questions?

Do you wanna build...

~~a snowman?~~

a camera...?

...mathematically? ☺



What does this mean?

- Given 3D point M, we want a function C such that C(M) is the point's projection in the photo.

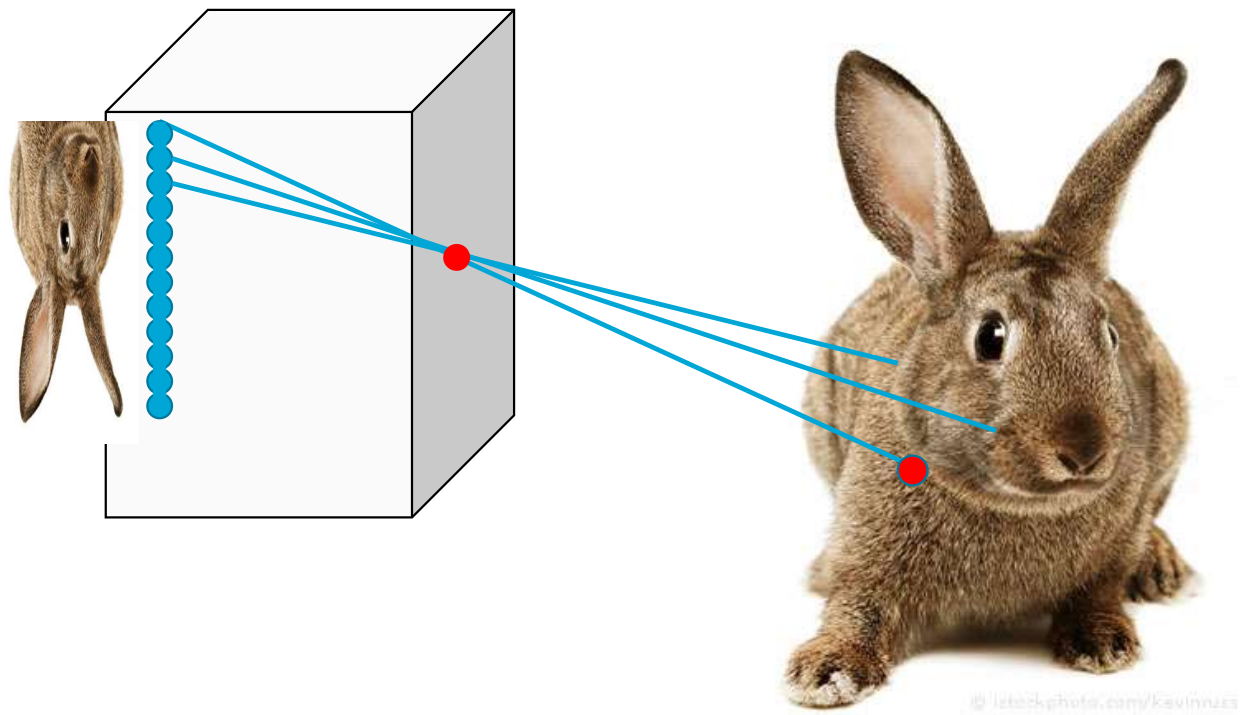
Stereo Reconstruction?

- Assume two different cameras (C_1, C_2).
  Given a 3D point M then $C_1(M)=m_1$, $C_2(M)=m_2$, where m_1,m_2 are the point's projection.

- We will see:

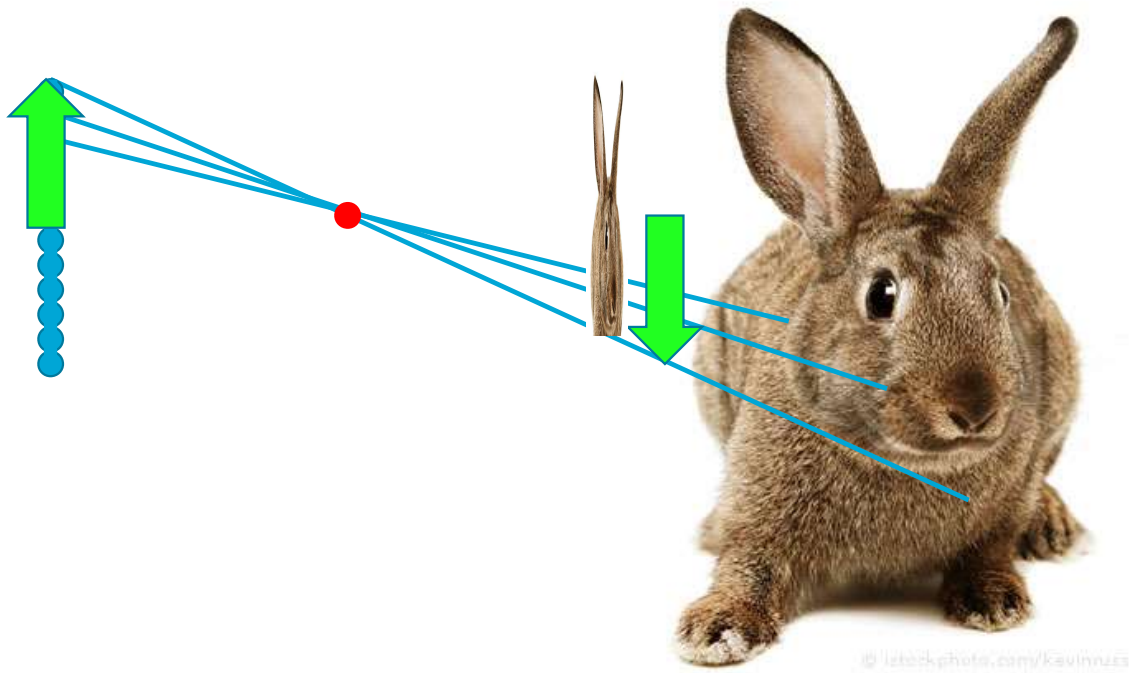
- With the equations $C_1(M)=m_1$, $C_2(M)=m_2$,
  we can calculate M – as it turns out C_1 and C_2 are simply matrices!

Virtual Camera

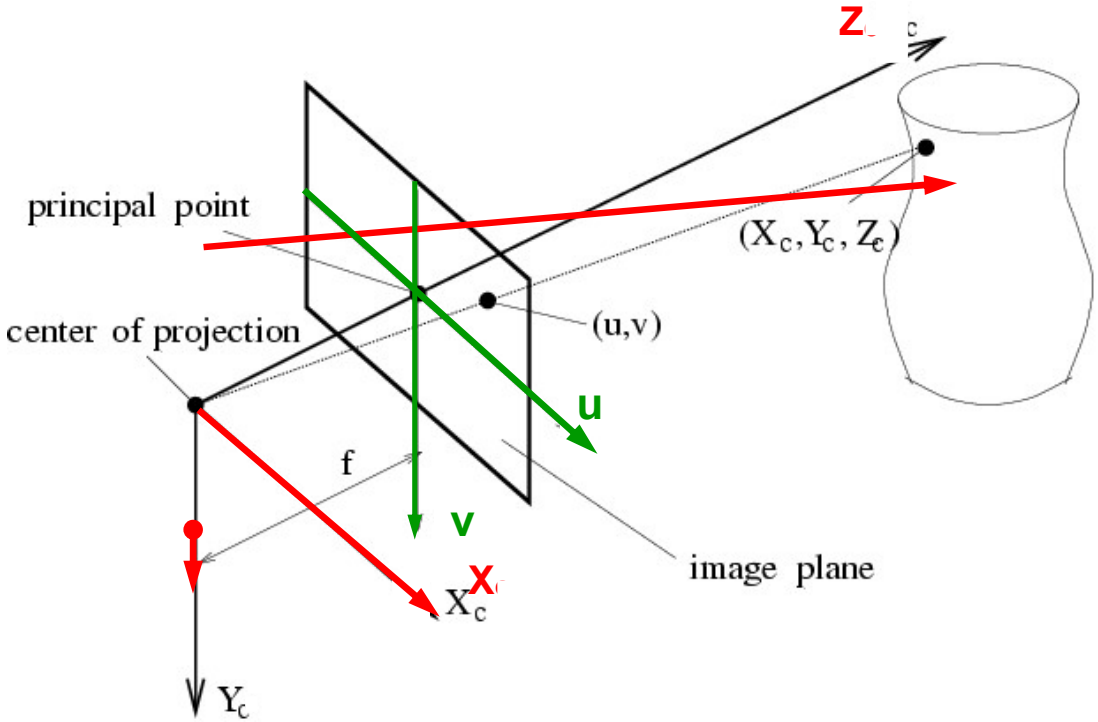


Virtual Camera

- Camera Plane in front of the eye

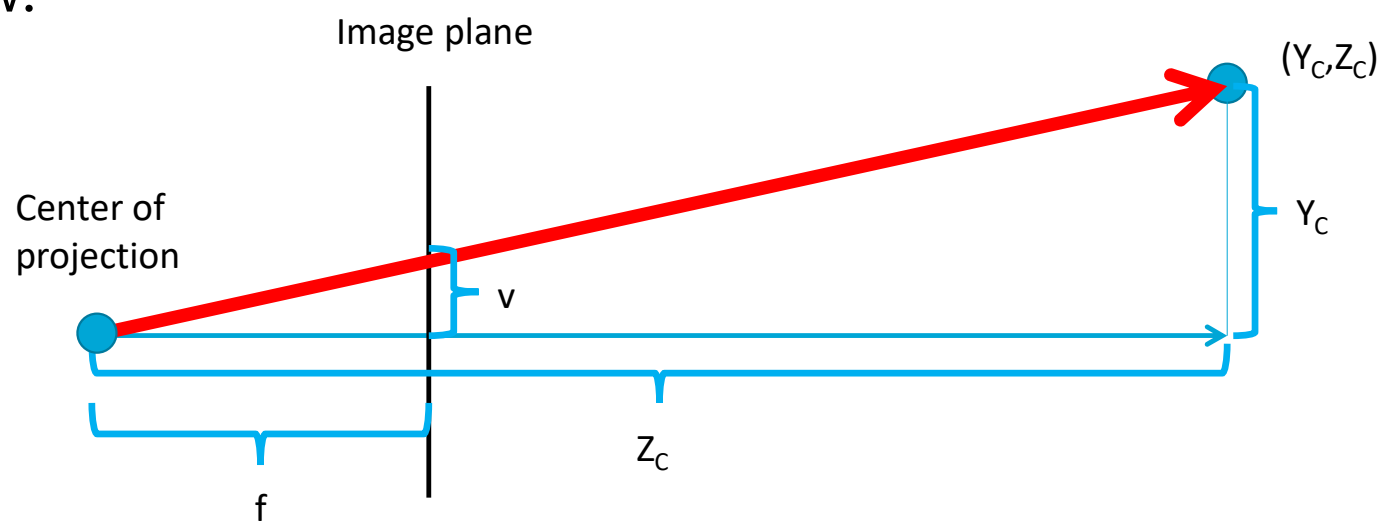


Perspective Projection



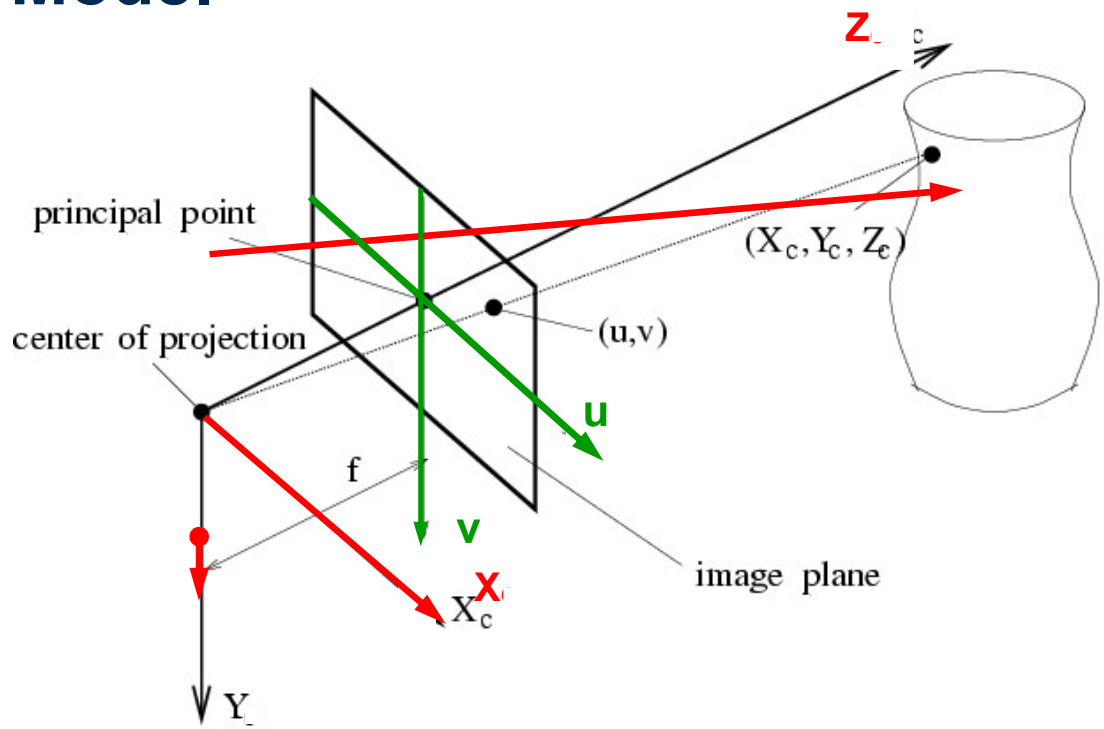
Perspective Projection

- sideview:



Similar triangles: $v / f = Y_C / Z_C$

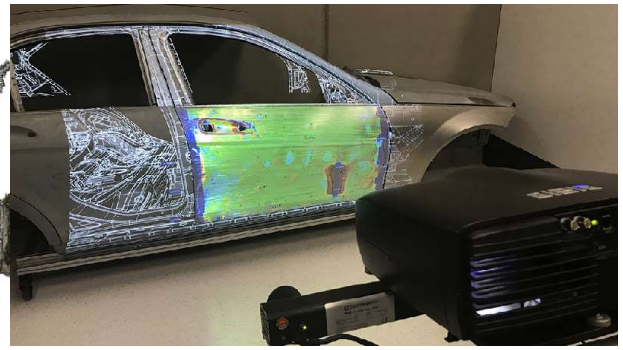
Virtual Camera Model



- Assuming that the camera center is at the origin and x,y axis aligned with the image
- Projection: $$(X, Y, Z)^T \mapsto (fX/Z, fY/Z)^T$$

Virtual Camera Model

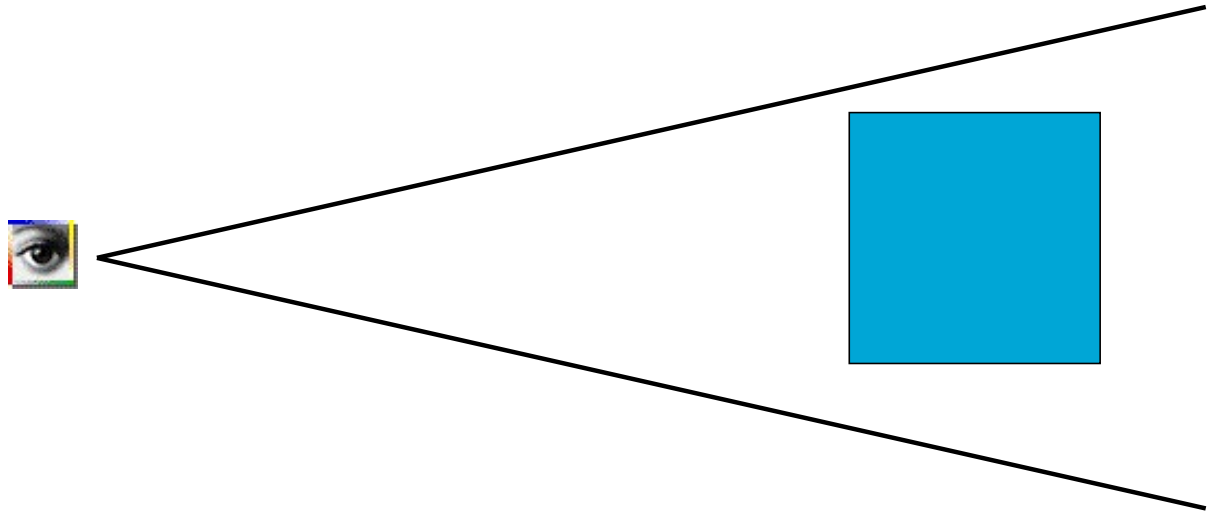
- If we have several camera relations, not all can be at the "origin and align with XYZ"



- Similarly, stitching images to a panorama
- and many more applications…
- Need general cameras and related math

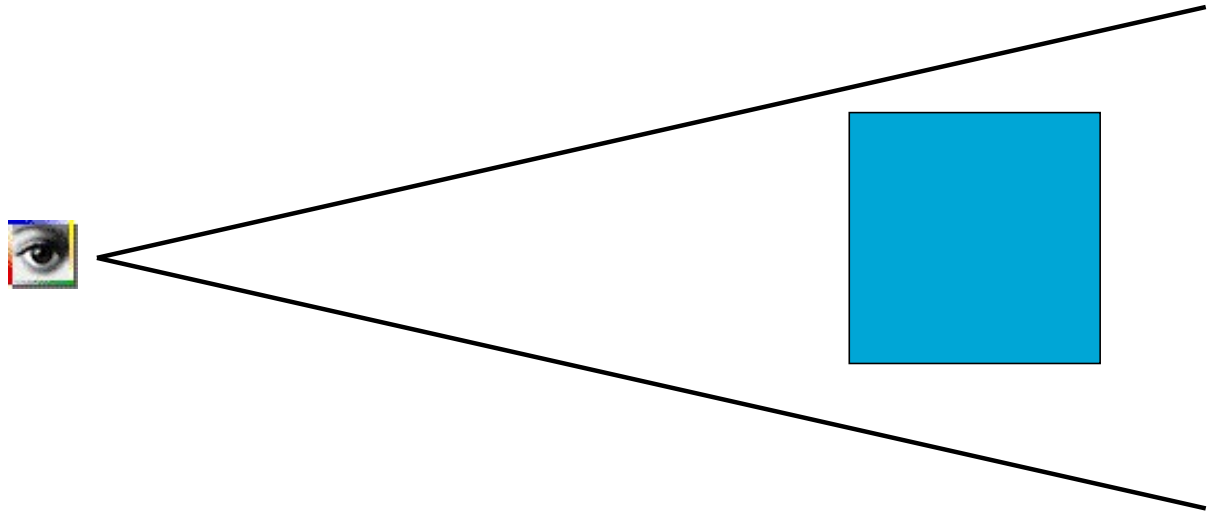
Virtual Camera Model

- Camera movement?



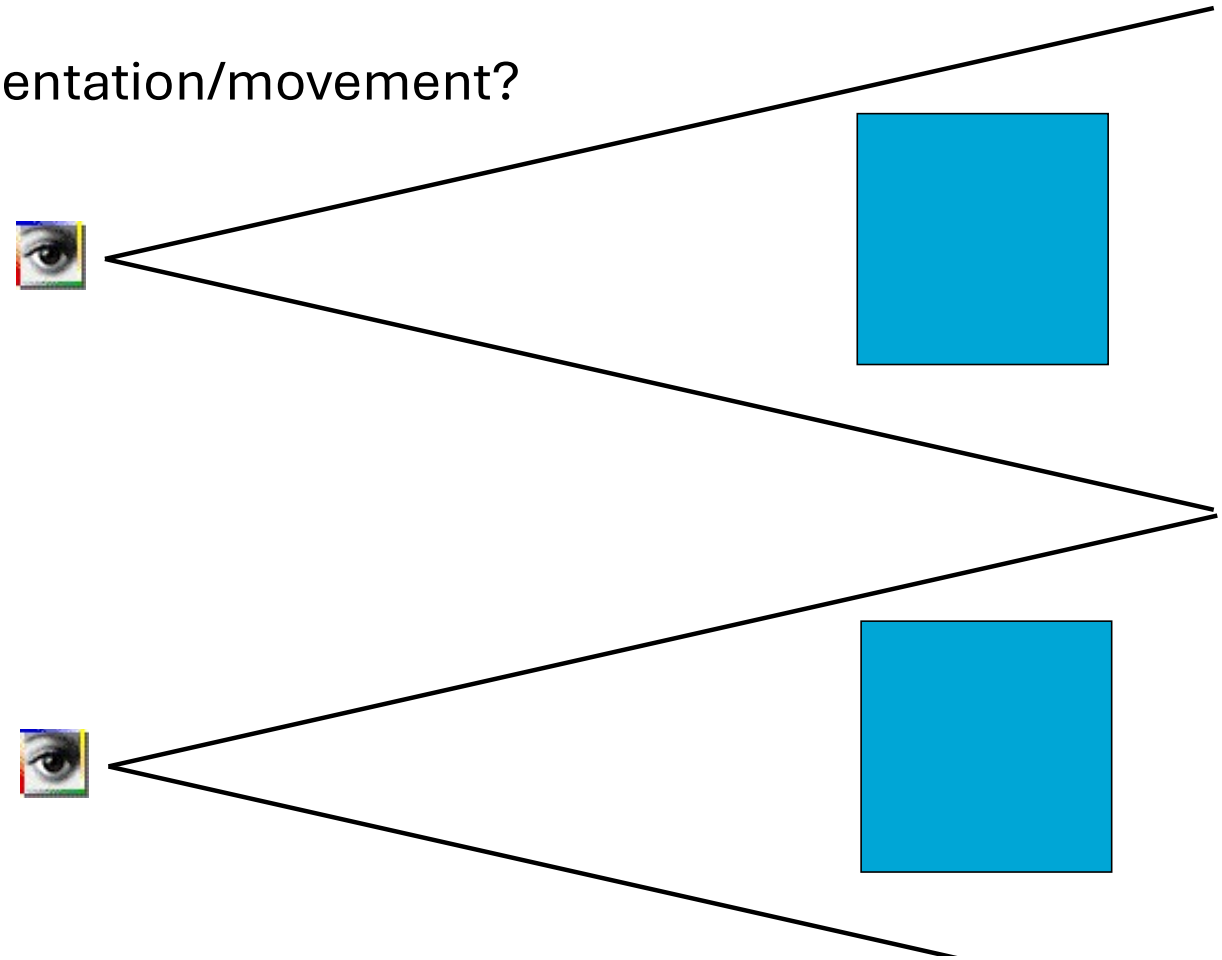
Virtual Camera Model

- Camera movement?



Virtual Camera Model

- Camera orientation/movement?

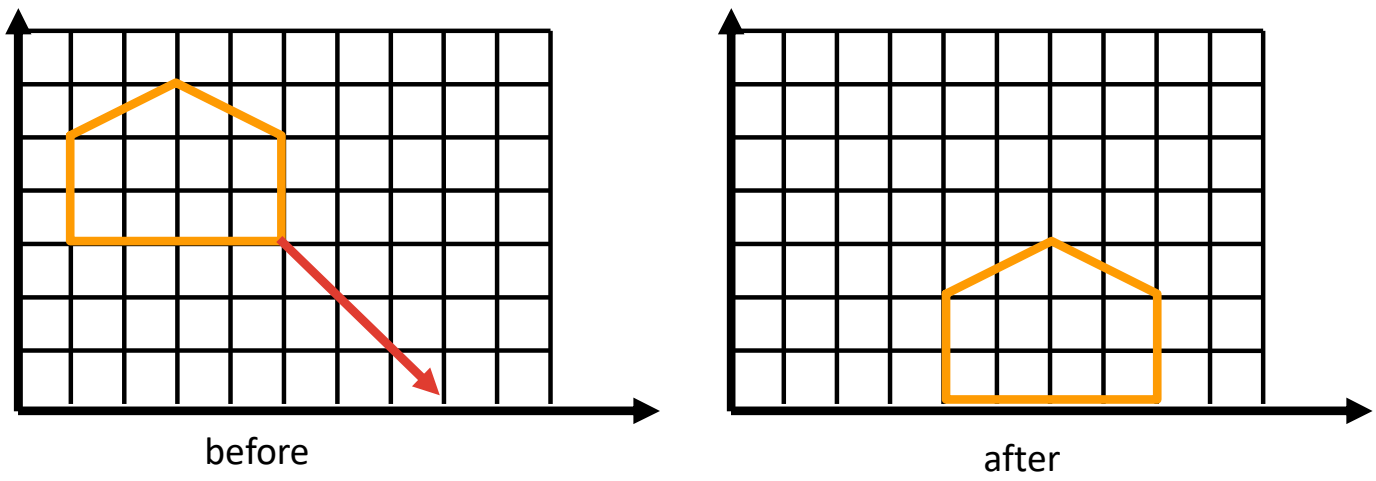


Movement and Orientation in 2D

- Starting in 2D
  - Simpler to represent

Translations

- Simple Modification :
  - $x' = x + t_x$
  - $y' = y + t_y$



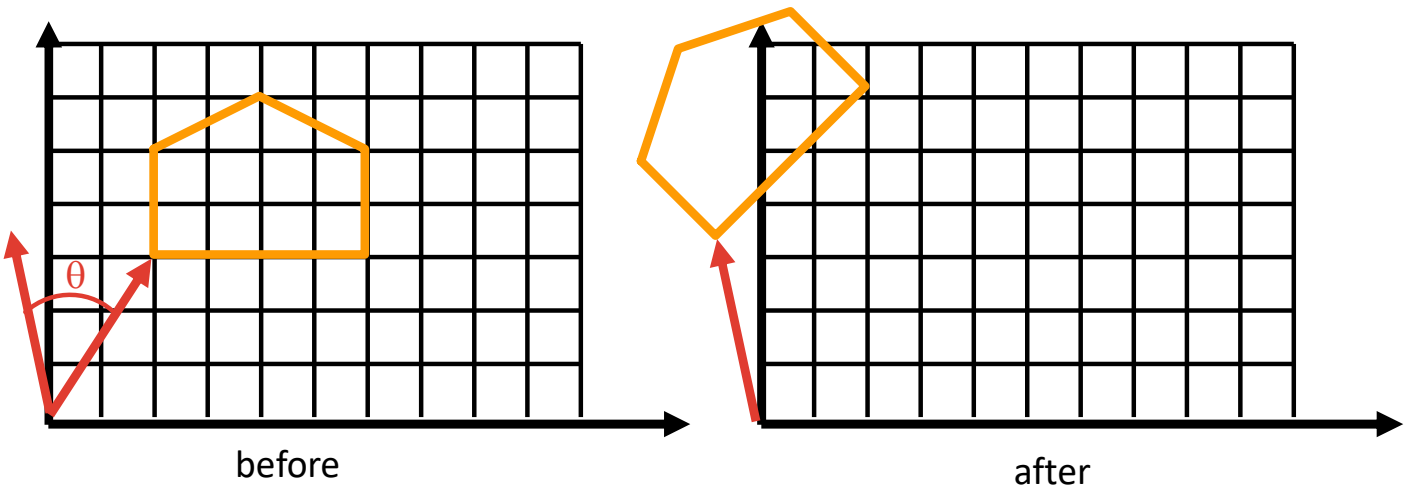
Vector Notation

- A point is a vector : $\begin{bmatrix} x \\ y \end{bmatrix}$

- Translation is a sum of vectors: P’ = P + T

Rotation

- Rotation in 2D :
  - $x' = \cos\theta\, x - \sin\theta\, y$
  - $y' = \sin\theta\, x + \cos\theta\, y$



Rotation

- Is a matrix multiplication:

$$P' = \mathbf{R}P$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Virtual Camera Model

Projecting a scene point with the camera:

- Apply camera position (adding an offset)
- Apply rotation (matrix multiplication)
- Apply projection (non-linear scaling)

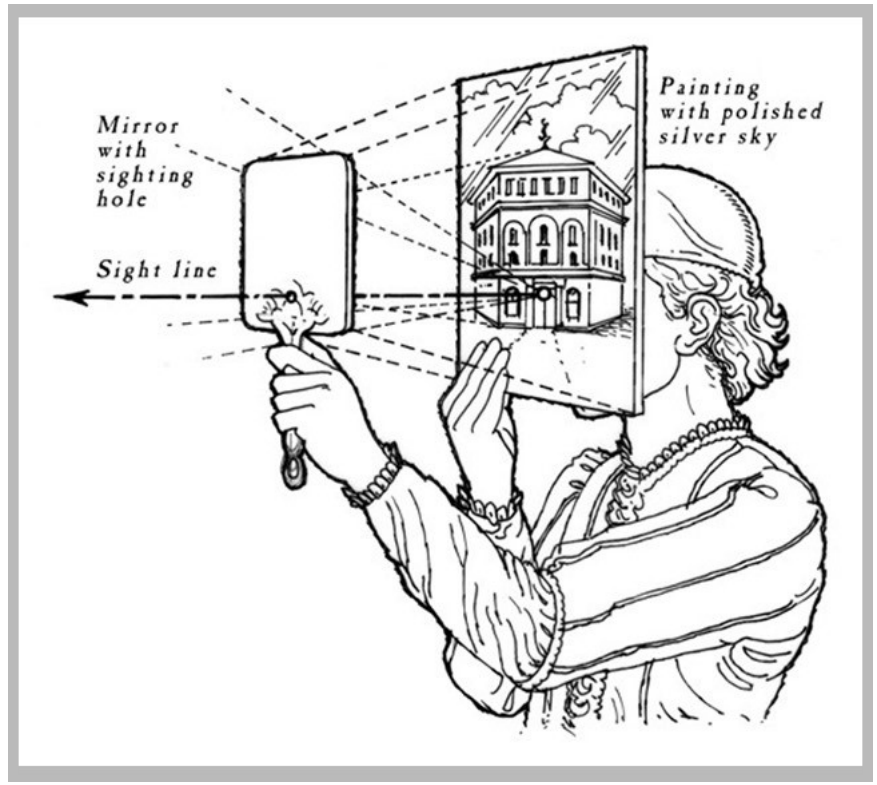
Our camera starts to become very complicated, especially if we want to calculate with it...

There has to be a better way...

Linear Perspective

$$\left\{\emptyset \colon \bar{F} - -1 < \int_{-1}^{\infty} \bigcup_{A \in A} \cos^{-1}\left(\|A\|\right) \, d\Xi''\right\}$$
$$= \prod_{Z_{\mathscr{I}}=1}^{0} \mathfrak{t}\left(- - \infty, \dots, \frac{1}{1}\right) - \dots \pm 2X$$
$$\neq \bigcap_{\mathscr{F}=1}^{2} \mathbf{x}\left(d, \dots, 1 \times e\right) \vee S\left(r_{i,\mathscr{X}}(\Phi'), \dots, L(\mathscr{W}_{G,\Psi}) \cap -1\right)$$
$$\geq \frac{\overline{\varphi^{(\epsilon)}}}{-O} - \log\left(-c_Z(\mathfrak{s}')\right).$$

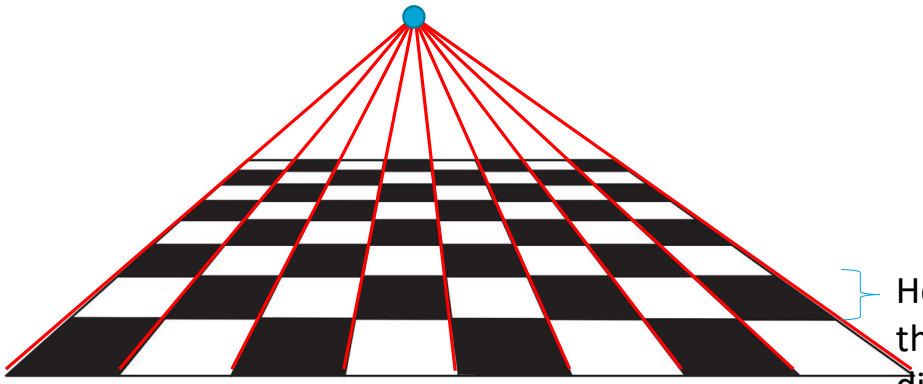
Linear Perspective



Filippo Brunelleschi – 1377–1446

Linear Perspective

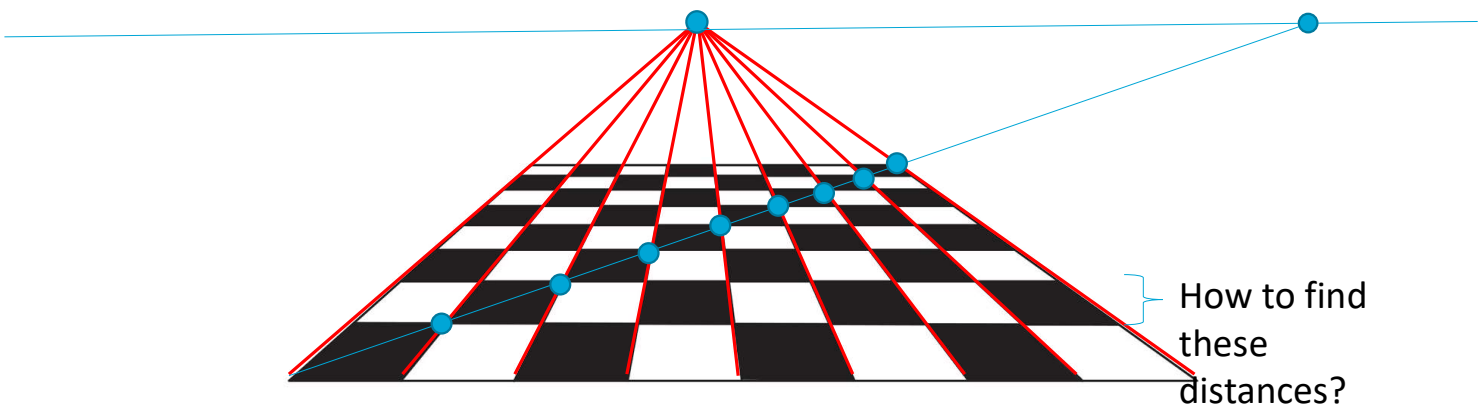
- Central Perspective



How to find these distances?

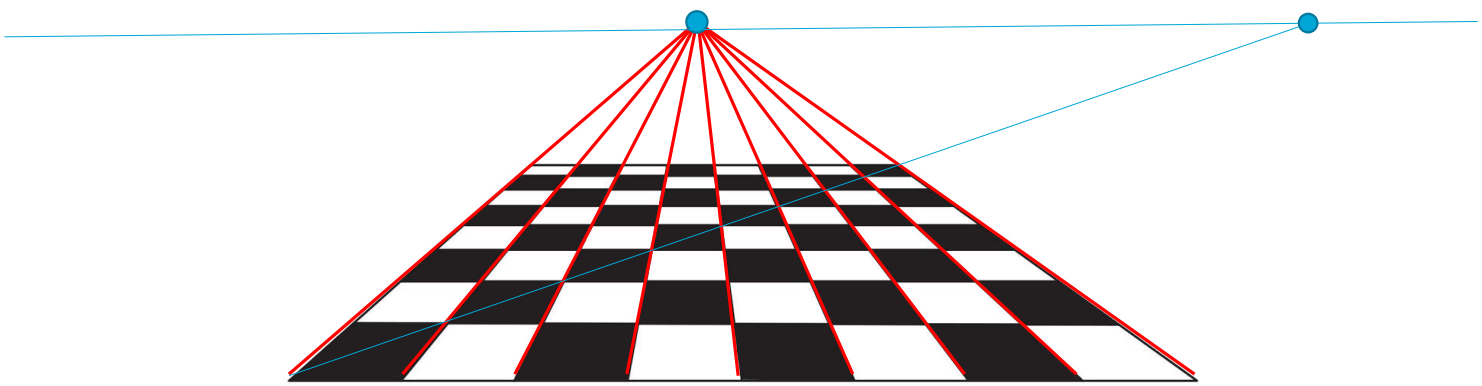
Linear Perspective

- Central Perspective



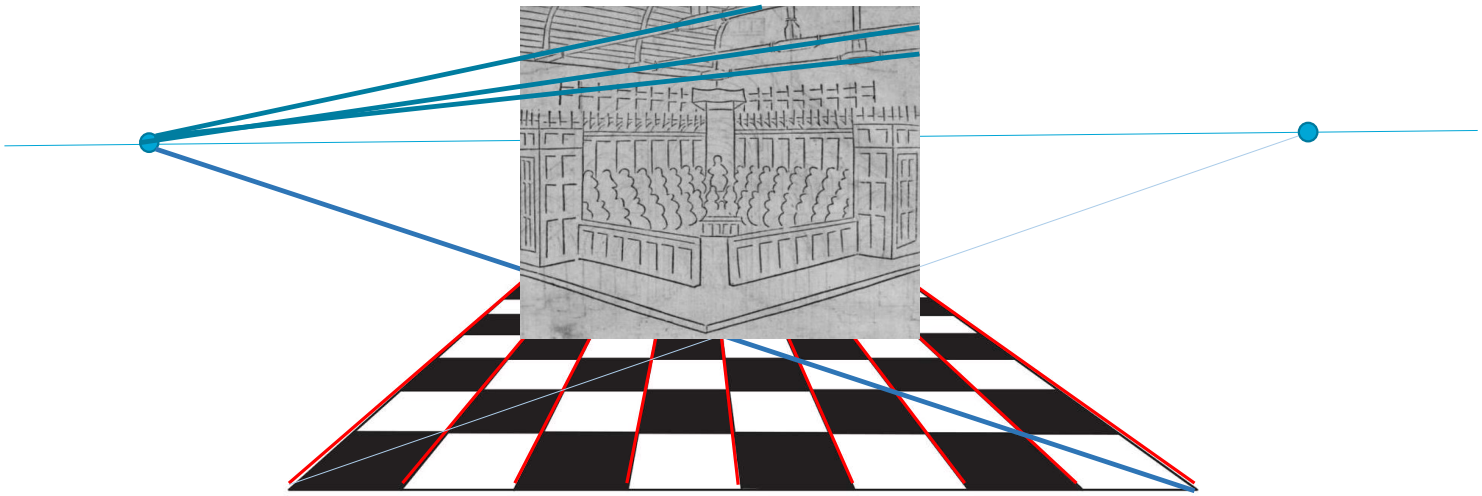
Linear Perspective

- Central Perspective



Linear Perspective

- Two Point Perspective



Vermeer and the linear perspective

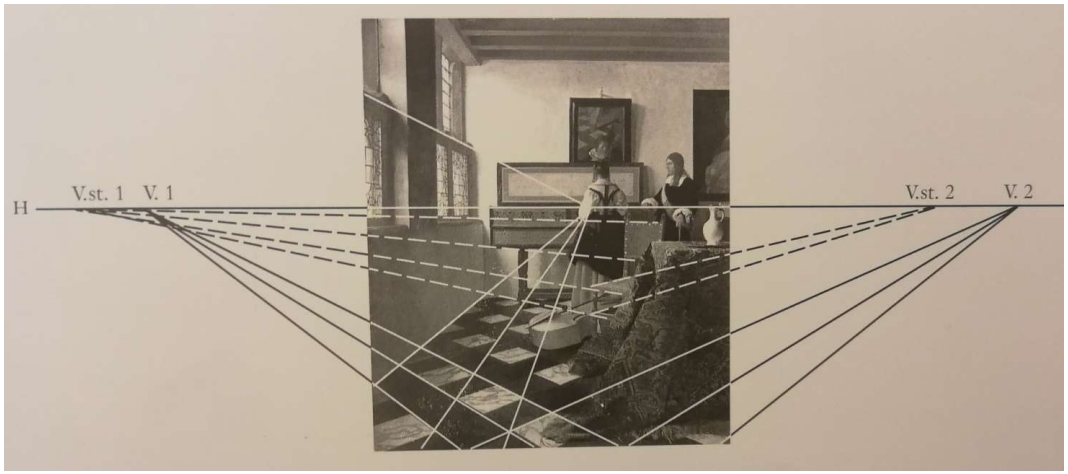
Johannes Vermeer, *The Milkmaid,* Rijksmuseum

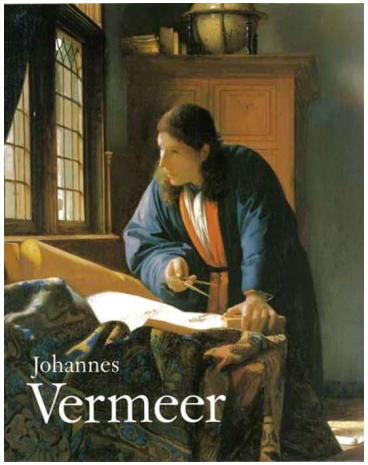
Detail of an x-ray of *The Milkmaid*

(courtesy Rijksmuseum)

Vermeer and the linear perspective

Johannes Vermeer, *The Music Lesson*, ca. 1662-1665, Royal Collection Trust





How to convince in a modern age?

THE SWEET SMELL
OF EPIC SUCCESS
memegenerator.net

Linear Algebra
A gentle introduction

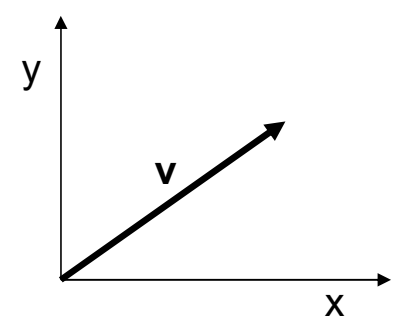
Linear Algebra has become as basic and as applicable as calculus, and fortunately it is easier.
--Gilbert Strang, MIT

What is a Vector ?

- Think of a vector as a *directed line segment in N-dimensions*! (has "length" and "direction")

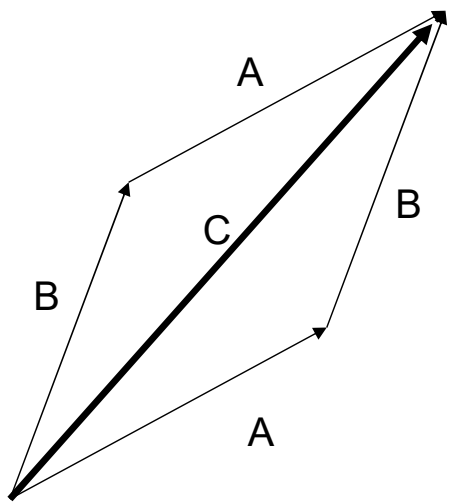
$$\vec{v} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

- Basic idea: convert geometry in higher dimensions into algebra!
  - Once you define a "nice" *basis* along each dimension: x-, y-, z-axis ...
  - Vector becomes a 1 x N matrix!
  - **v** = $[a \;\; b \;\; c]^T$
  - Geometry starts to become linear algebra on vectors like **v**!



Vector Addition: A+B

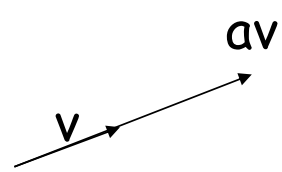
$$\mathbf{A+B} = (x_1, x_2) + (y_1, y_2) = (x_1 + y_1, x_2 + y_2)$$



A+B = C
(use the head-to-tail method to combine vectors)

Scalar multiplication: $a\mathbf{v}$

$$a\mathbf{v} = a(x_1, x_2) = (ax_1, ax_2)$$



Change only the length ("scaling"), but keep <u>*direction fixed*</u>.

<u>**Sneak peek:**</u> matrix operation (**Av**) can change *length, direction and also dimensionality*!

Vectors: Dot Product

$$A \cdot B = A^T B = \begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} d \\ e \\ f \end{bmatrix} = ad + be + cf$$

Think of the dot product as a matrix multiplication

$$\|A\|^2 = A^T A = aa + bb + cc$$

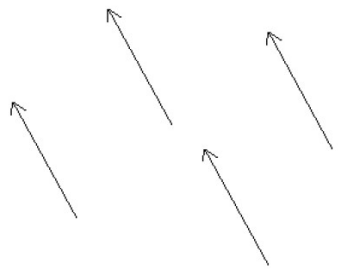
The dot product of a vector with itself is its squared length

$$A \cdot B = \|A\| \, \|B\| \cos(\theta)$$

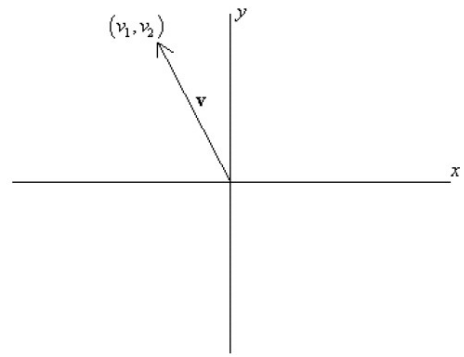
The dot product is also related to the angle between the two vectors

Bases & Orthonormal Bases

- Basis (or axes): frame of reference



vs



Basis: a space is totally defined by a set of vectors – any point is a *linear combination* of the basis

Ortho-Normal: orthogonal + normal

[**<u>Sneak peek</u>**:
<u>Orthogonal</u>: dot product is zero
<u>Normal</u>: magnitude is one]

$$x = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \qquad x \cdot y = 0$$

$$y = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T \qquad x \cdot z = 0$$

$$z = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \qquad y \cdot z = 0$$

What is a Matrix?

- A matrix is a set of elements, organized into rows and columns

rows

columns

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

Matrix Times Matrix

$$\mathbf{L} = \mathbf{M} \cdot \mathbf{N}$$

$$\begin{bmatrix} l_{11} & l_{12} & l_{13} \\ l_{21} & l_{22} & l_{23} \\ l_{31} & l_{32} & l_{33} \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \cdot \begin{bmatrix} n_{11} & n_{12} & n_{13} \\ n_{21} & n_{22} & n_{23} \\ n_{31} & n_{32} & n_{33} \end{bmatrix}$$

$$l_{12} = m_{11}n_{12} + m_{12}n_{22} + m_{13}n_{32}$$

Multiplication

- Is AB = BA? Maybe, but **maybe not**!

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & \dots \\ \dots & \dots \end{bmatrix} \quad \begin{bmatrix} e & f \\ g & h \end{bmatrix}\begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} ea+fc & \dots \\ \dots & \dots \end{bmatrix}$$

- Matrix multiplication AB: apply transformation B first, and then again transform using A!
- Heads up: multiplication is NOT commutative!

Vector Dot Matrix

$$\mathbf{v}' = \mathbf{M} \cdot \mathbf{v} = \begin{array}{c} \begin{array}{cccc} \mathbf{a} & \mathbf{b} & \mathbf{c} & \mathbf{d} \\ \Downarrow & \Downarrow & \Downarrow & \Downarrow \end{array} \\ \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \end{bmatrix} \end{array} \cdot \begin{bmatrix} v_x \\ v_y \\ v_z \\ 1 \end{bmatrix}$$

$$v'_x = v_x a_x + v_y b_x + v_z c_x + d_x$$
$$v'_y = v_x a_y + v_y b_y + v_z c_y + d_y$$
$$v'_z = v_x a_z + v_y b_z + v_z c_z + d_z$$

$$v_x \mathbf{a} + v_y \mathbf{b} + v_z \mathbf{c} + \mathbf{d}$$

Vector Dot Matrix

$$\mathbf{v}' = \mathbf{M} \cdot \mathbf{v} = \begin{bmatrix} a_x & b_x & c_x & d_x \\ a_y & b_y & c_y & d_y \\ a_z & b_z & c_z & d_z \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \\ v_z \\ 1 \end{bmatrix}$$

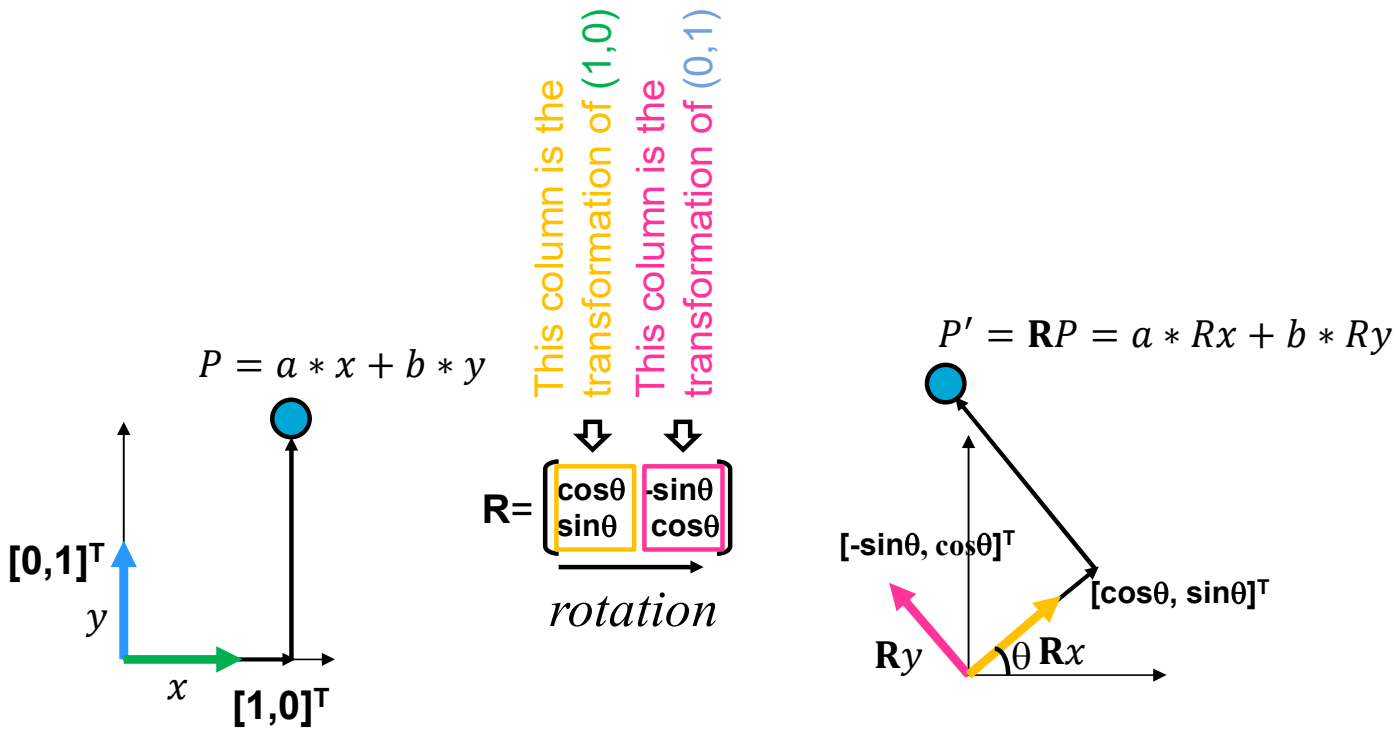
$$v'_x = \boxed{v_x a_x + v_y b_x + v_z c_x + d_x}$$

$$v'_y = \boxed{v_x a_y + v_y b_y + v_z c_y + d_y}$$

$$v'_z = \boxed{v_x a_z + v_y b_z + v_z c_z + d_z}$$

$$v_x \mathbf{a} + v_y \mathbf{b} + v_z \mathbf{c} + \mathbf{d}$$

Basis transformation



Identity Matrix and Inverse of a Matrix

- Identity matrix:
  AI = A

- Inverse exists only for <u>square matrices</u> that are <u>non-singular</u>
  - Maps N-d space to another N-d space bijectively

- Some matrices have an inverse, such that:
  $\mathbf{AA^{-1} = I}$

- Inversion is tricky:
  $\mathbf{(ABC)^{-1} = C^{-1}B^{-1}A^{-1}}$

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Why are matrices cool?

Let M, N be matrices and P and Q points,

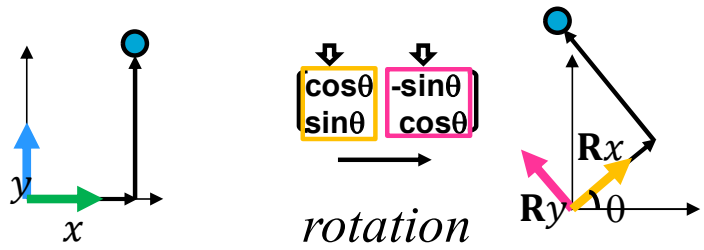
then we have

MN P = M (NP) = (MN) P

(instead of multiplying with 2 matrices, you can multiply them up, i.e., the operations can be concatenated by matrix multiplication)

And matrices are linear applications

M(αP+Q)= α MP+MQ, with α being a scalar



End of Linear Algebra Excursion

What we want

- Simple, concise math

- Unification
  - Translation, rotation, projection

- And if I am allowed to dream:

  Make all linear, i.e., only matrix multiplies…

Is that really possible?

- Imagine a point X and a matrix T that describes a translation…

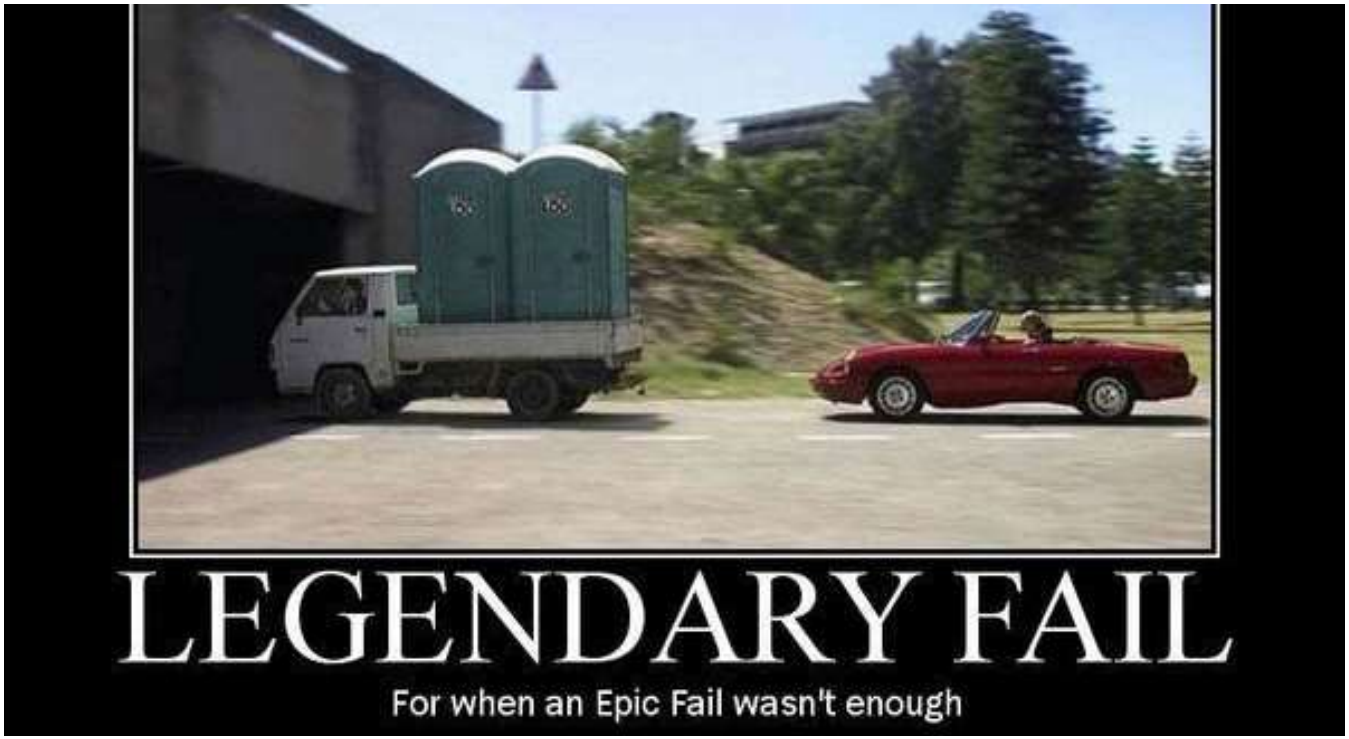
- T (X) = X+t
- T(X+0)

  =T(X)+T(0)

  =X+t+0+t = X+2t

Fail...

- Translations are not linear nor are projections, thus, cannot be represented by a matrix ...



- **Teacher: “Easy** because with **projective geometry** everything is **linear** using **homogeneous coordinates”**

Me when I was young

Homogenous Coordinates - Definition

- N-D *projective space* P^n is represented by N+1 coordinates, has no null vector, but a special equivalence relation:

Two points p, q are **equal**

iff (if and only if)

exists a!=0 such that p*a = q

Examples in a 2D projective space P^2:
(2,2,2) = (3,3,3) = (4,4,4)=(π, π, π)
(2,2,2) $\neq$ (3,1,3)
(0,1,0) = (0,2,0)
(0,0,0) not part of the space

Homogeneous Coordinates

To embed a standard vector space R^n in an n-D projective space P^n, we can map:

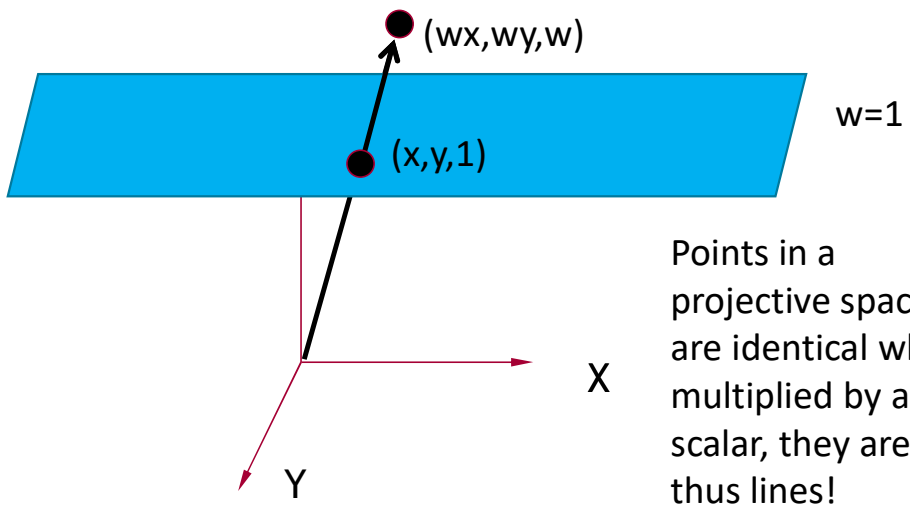
$(x_0,x_1,\ldots x_{n-1})$ in R^n to $(x_0,x_1,\ldots x_{n-1}, 1)$ in P^n

Typically, the last coordinate in a projective space is denoted with w.

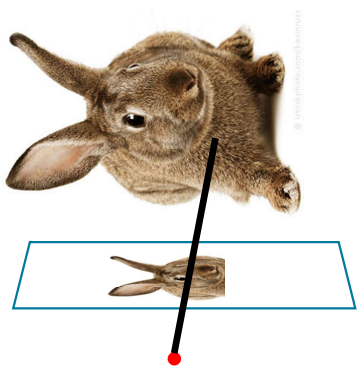
Homogeneous Coordinates

- A point (x,y) in R^2 embedded in a projective space corresponds to (x,y,1). All points (x,y,1) form a plane (referred to as *affine plane*)

The points in this plane correspond to points in our standard vector space R^2

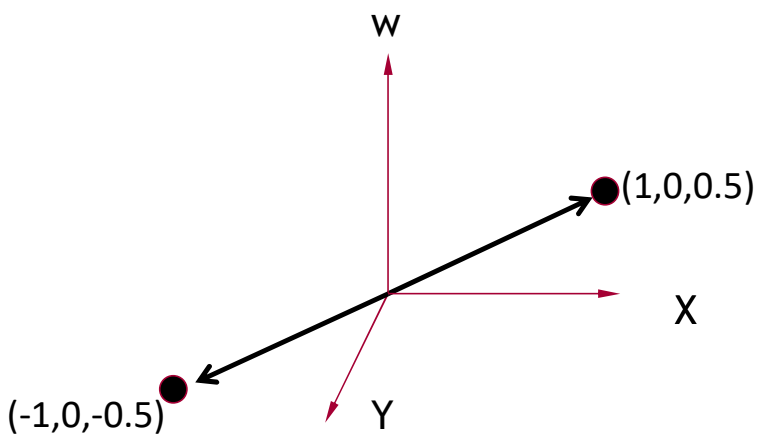


Points in a projective space are identical when multiplied by any scalar, they are thus lines!



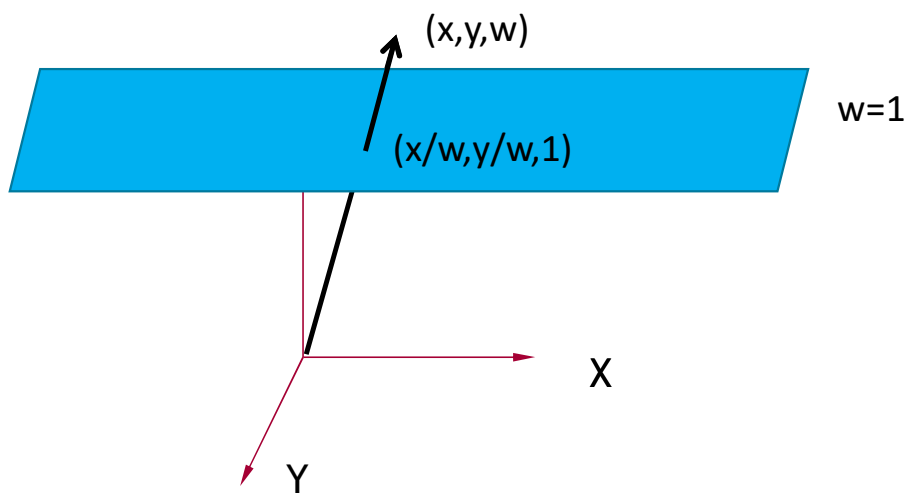
Homogeneous Coordinates

- Example:



Homogeneous Coordinates

- Thus, we can go back to R^2 by dividing the coordinates by the last entry.
  (x,y,w) in P^2 corresponds to (x/w,y/w) in R^2.



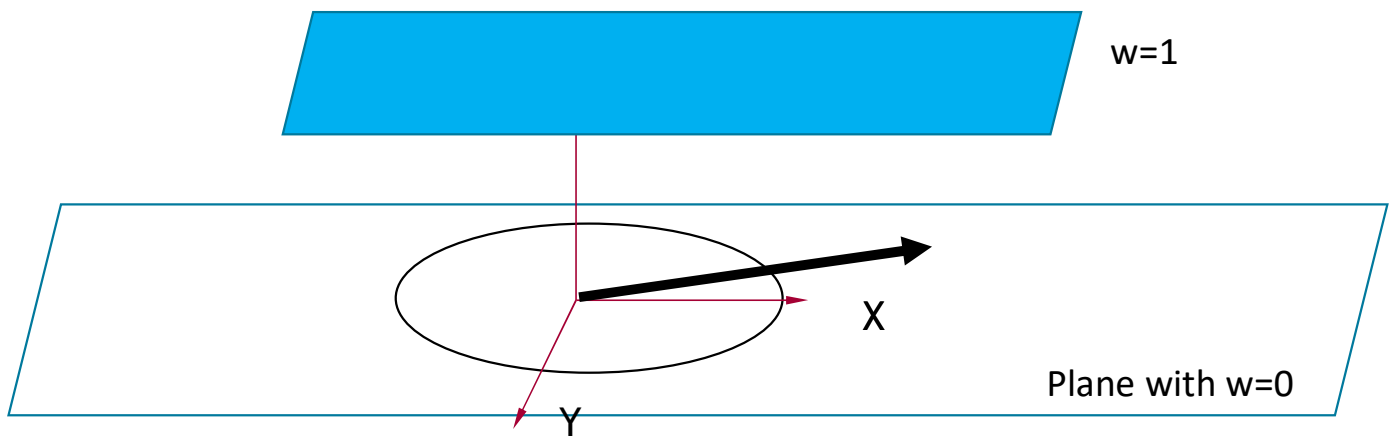
From projective to standard vector space:

- For $(x_0,x_1,\ldots x_{n-1},w)$ with w!=0, the corresponding point in R^n is $(x_0/w,x_1/w,\ldots x_{n-1}/w)$

- $(x_0,x_1,\ldots x_{n-1},0)$ has no correspondence in R^n!

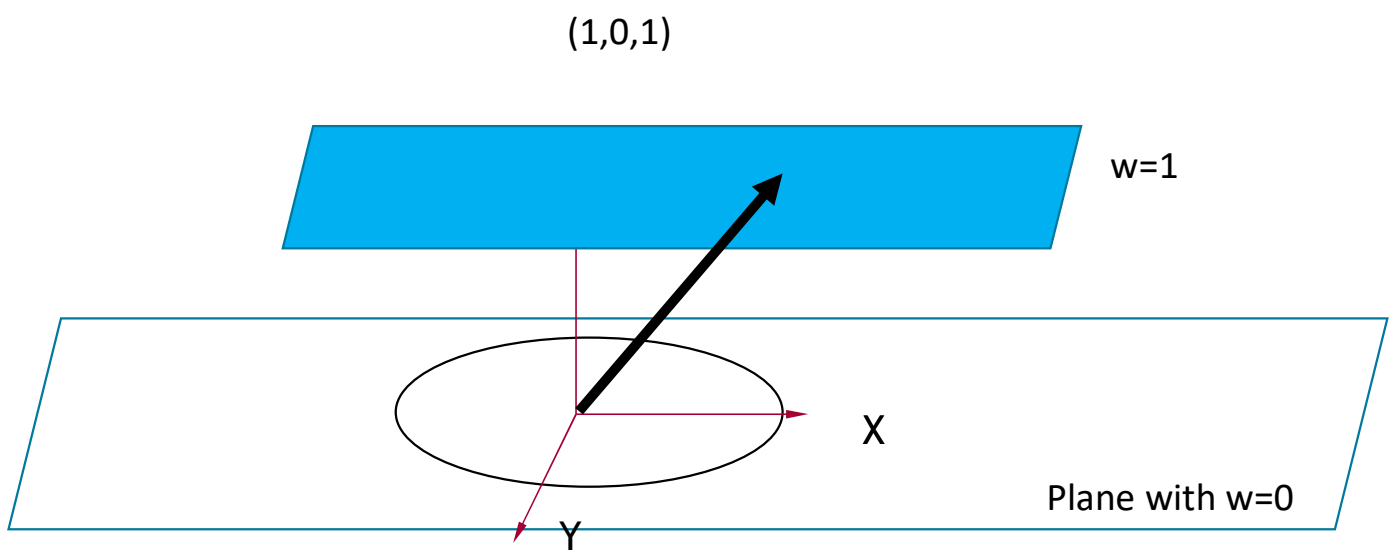
Homogeneous Coordinates

- What about the points with w=0?



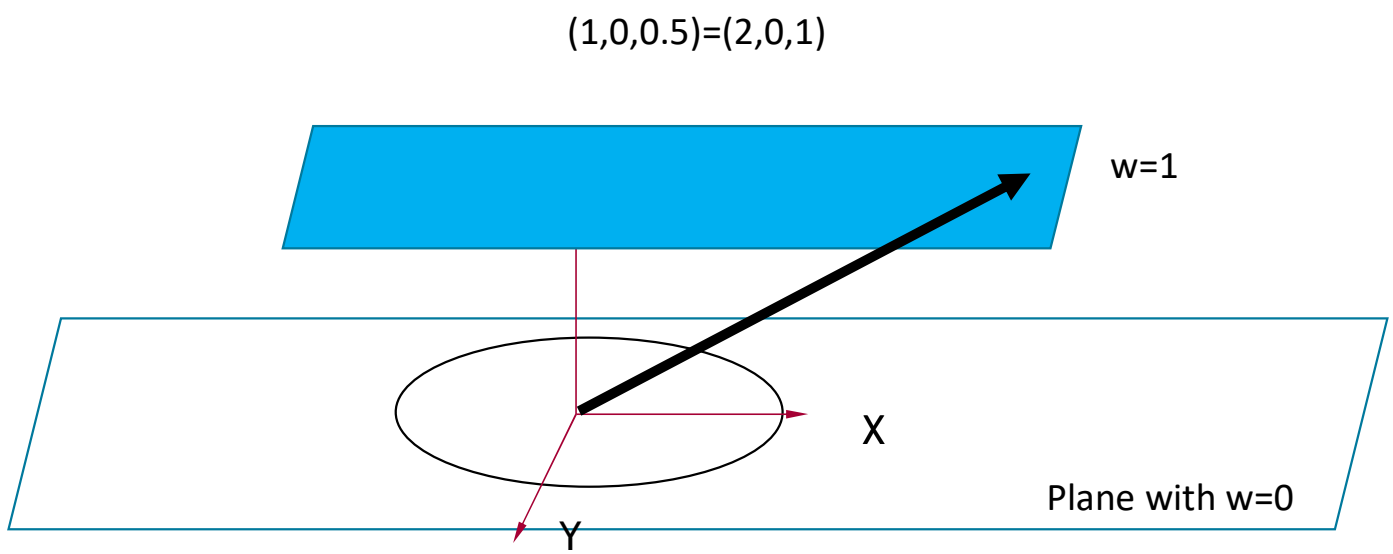
Homogeneous Coordinates

- What about the points with w=0?
- Let's try it out with a point (1,0,w) and we decrease w…



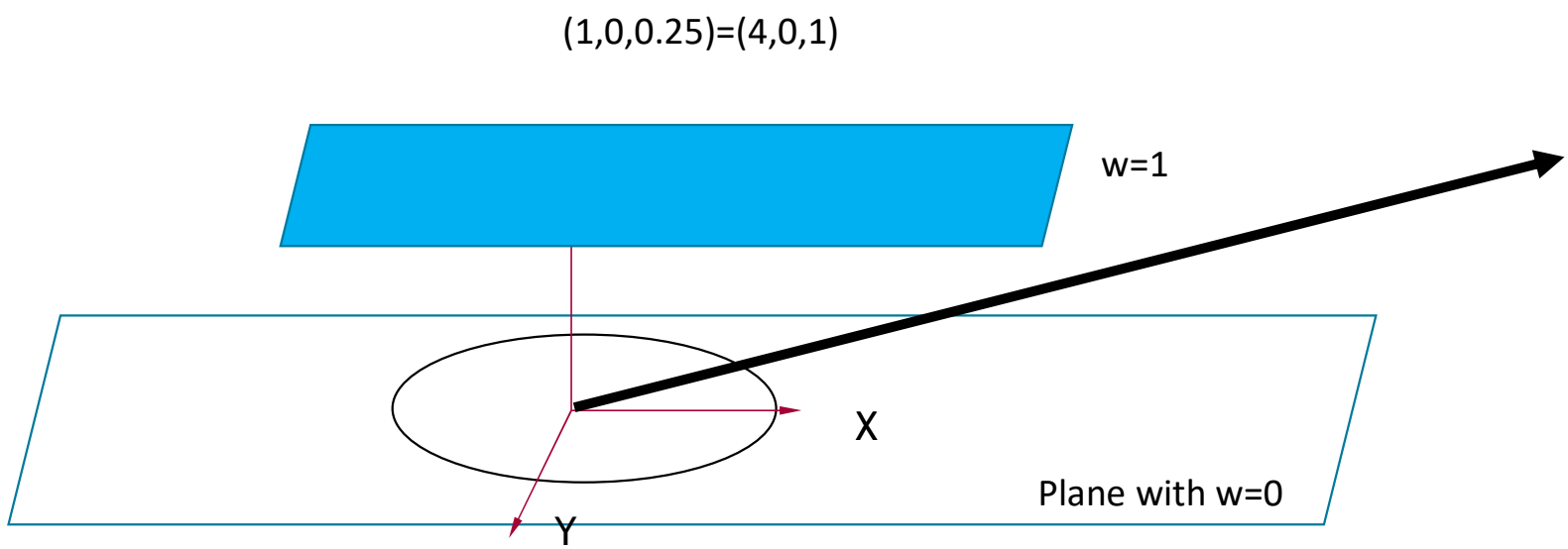
Homogeneous Coordinates

- What about the points with w=0?
- Let's try it out with a point (1,0,w) and we decrease w…



Homogeneous Coordinates

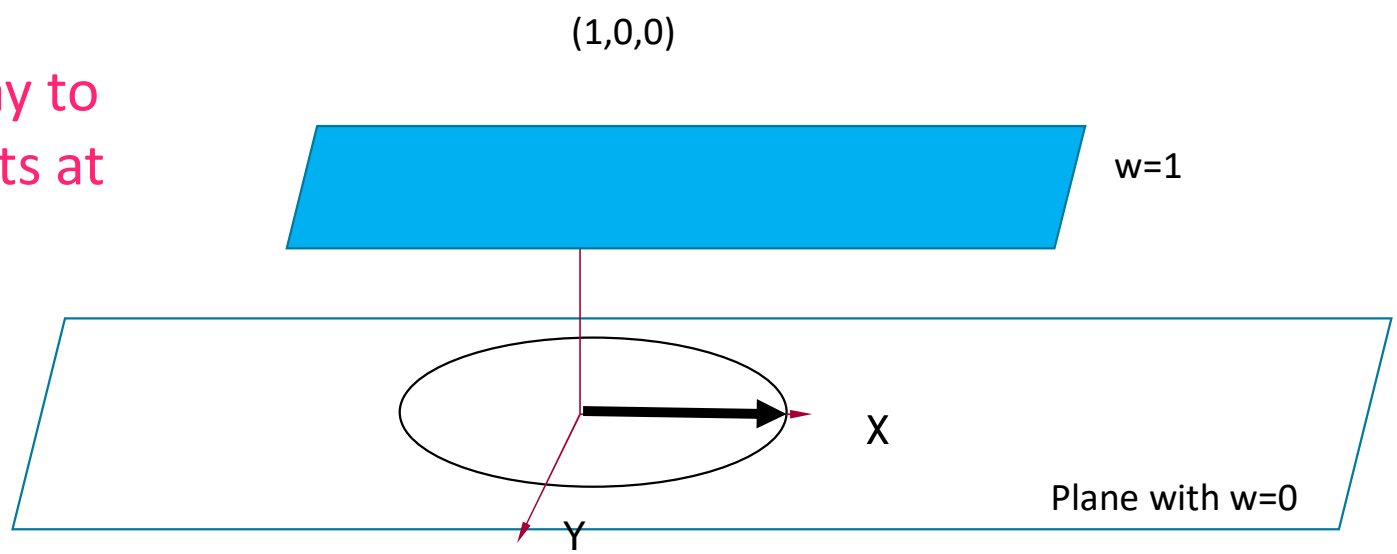
- What about the points with w=0?
- Let's try it out with a point (1,0,w) and we decrease w…



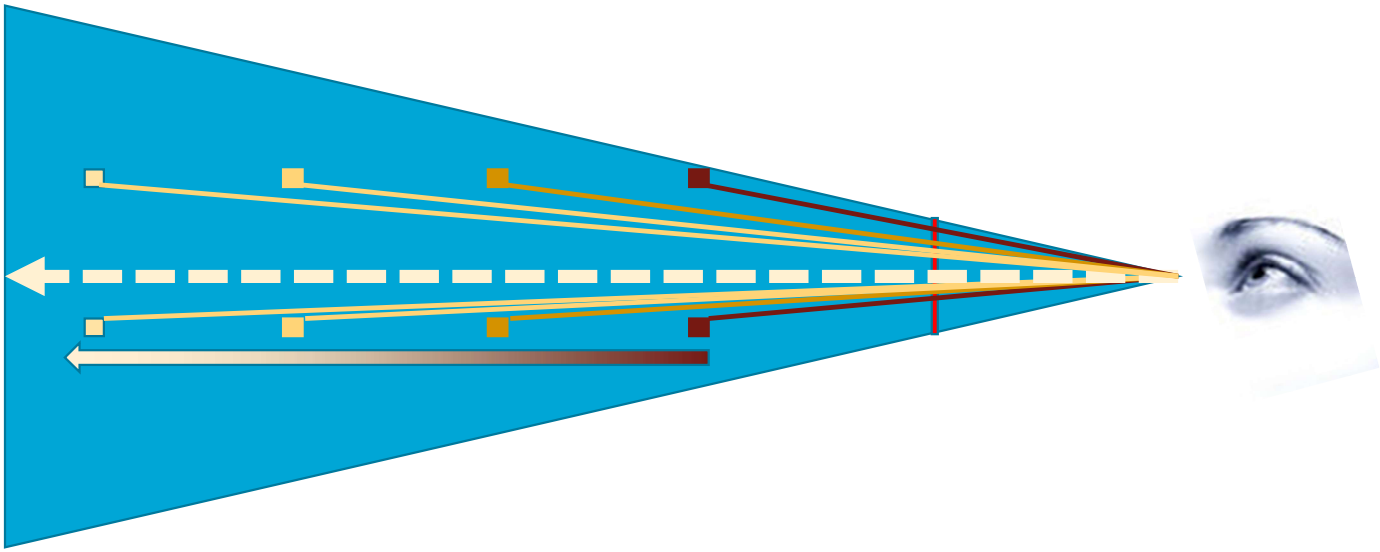
Homogeneous Coordinates

- What about the points with w=0?
- Let's try it out with a point (1,0,w) and we decrease w…

We have a way to describe points at INFINITY!



Linear Perspective



Linear Perspective

Homogeneous Coordinates

We wanted:

Easy transformations = matrices

Translations, rotations,
scaling, projection should all be matrices…

Concatenating transformations
means simply matrix multiplications!

Translations in R^2

- To translate vector (x,y),
  we add (t_x,t_y) to obtain: $(x+t_x,y+t_y)$

In a projective space,
we would like to have a matrix M, such that

$M(x,y,1) = (x+t_x,y+t_y,1)$

Translations in homog. coordinates

- R^2

$$\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} = \begin{bmatrix} x + tx \\ y + ty \end{bmatrix}$$

- P^2

Translations in homog. coordinates

- R^2

$$\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} = \begin{bmatrix} x + tx \\ y + ty \end{bmatrix}$$

- P^2

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \begin{bmatrix} x + tx \\ y + ty \\ 1 \end{bmatrix}$$

Translations in homog. coordinates

- R^2

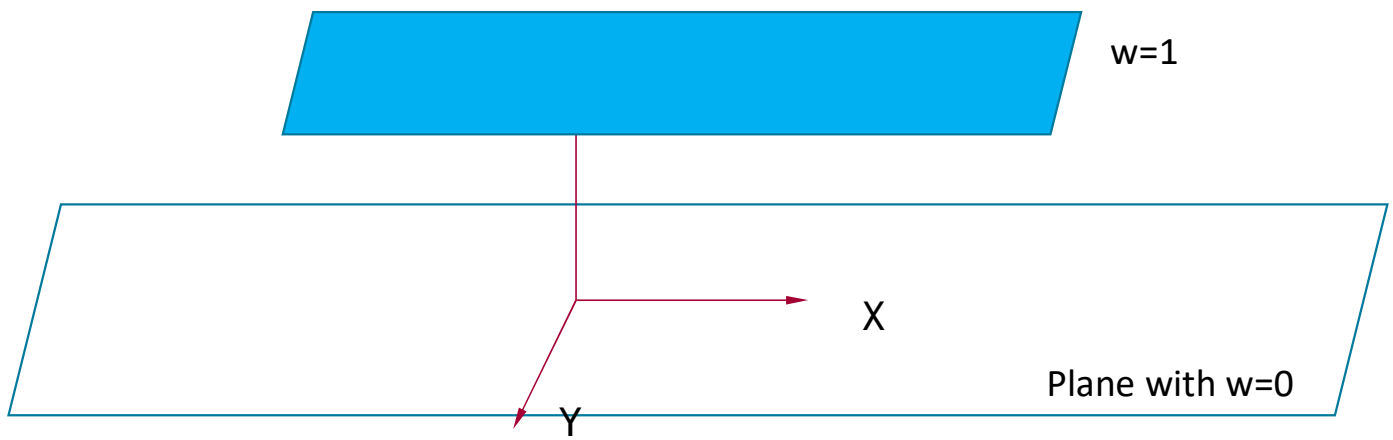
$$\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} = \begin{bmatrix} x + tx \\ y + ty \end{bmatrix}$$

- P^2

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + tx \\ y + ty \\ 1 \end{bmatrix}$$

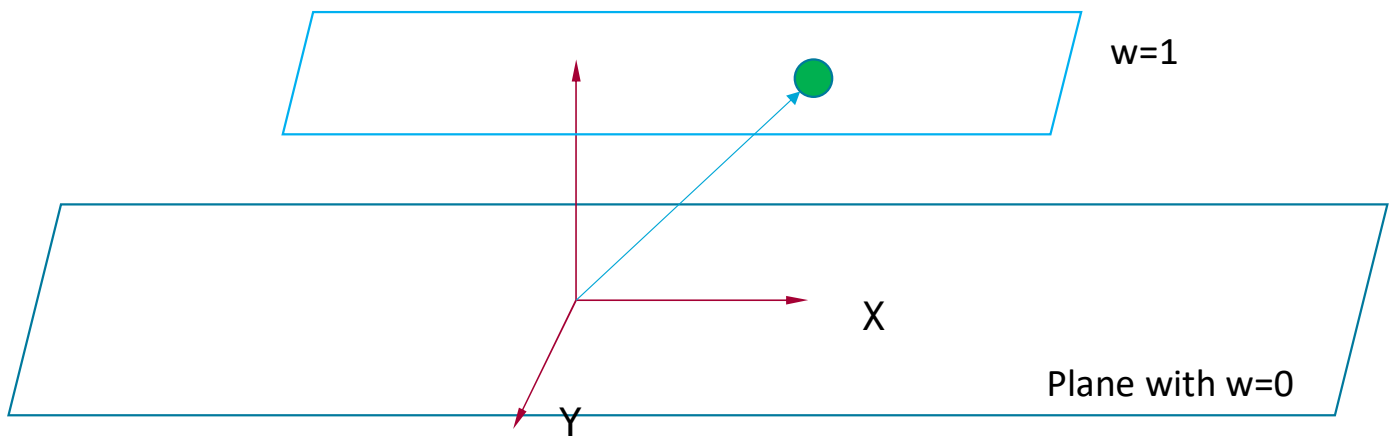
How does this work?

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + tx \\ y + ty \\ 1 \end{bmatrix}$$



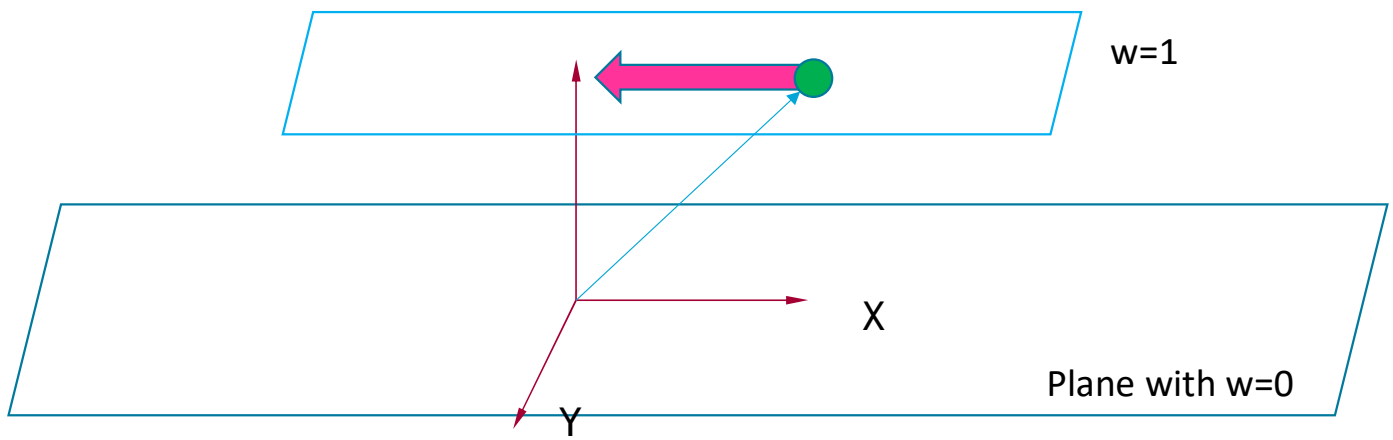
How does this work?

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



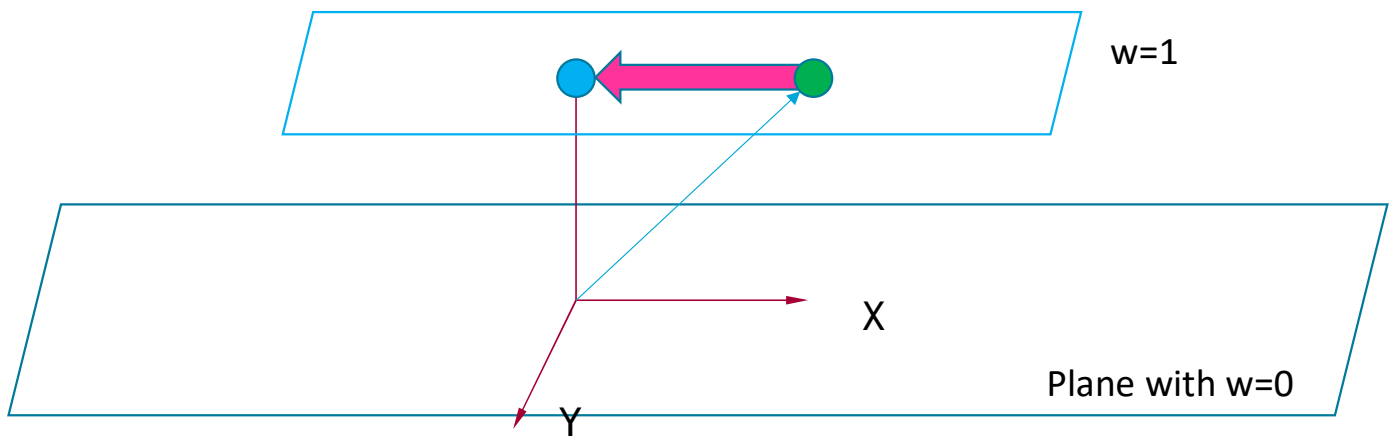
How does this work?

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



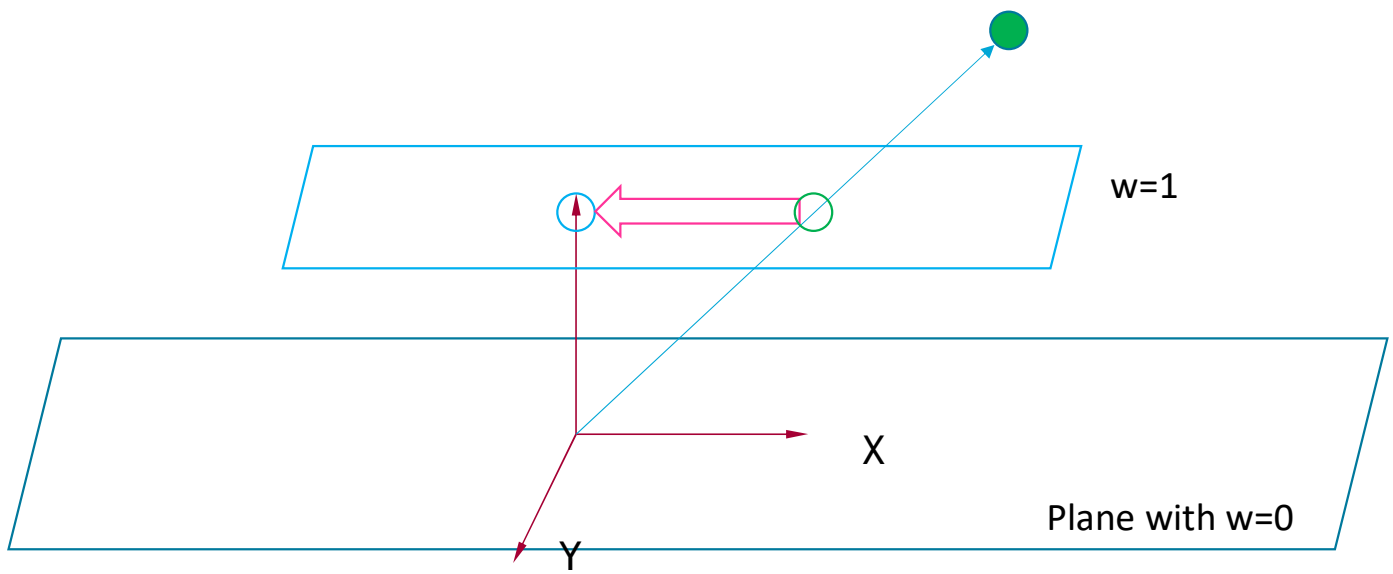
How does this work?

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1-1 \\ 0+0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$



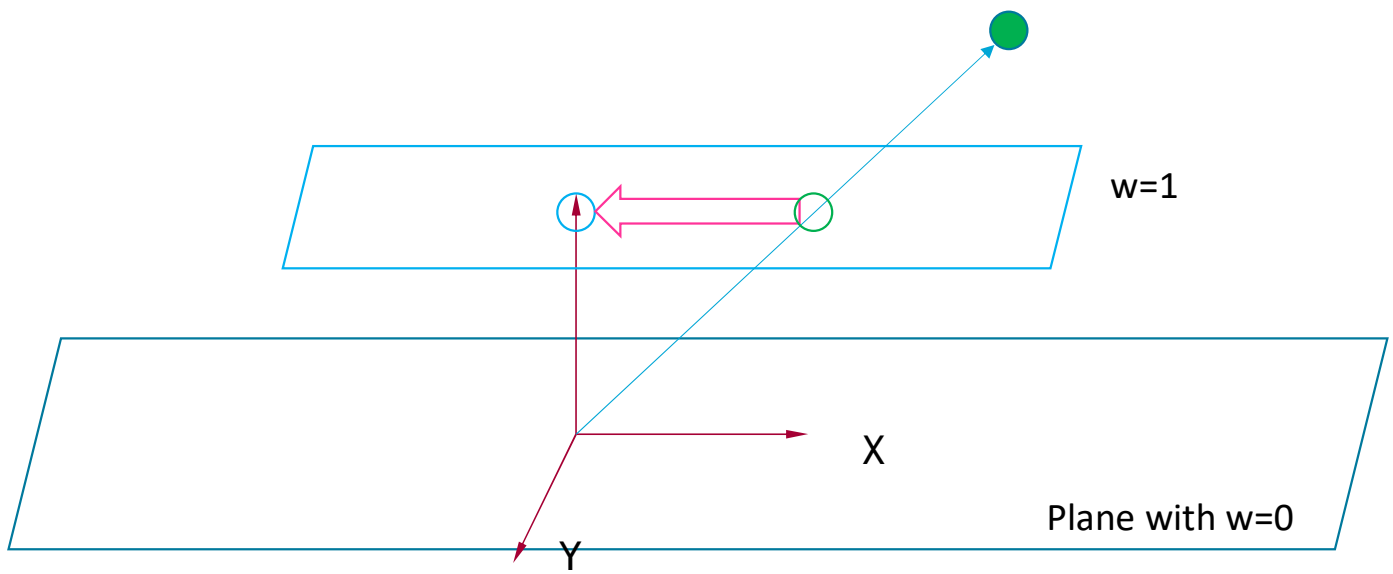
How does this work?

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix}$$



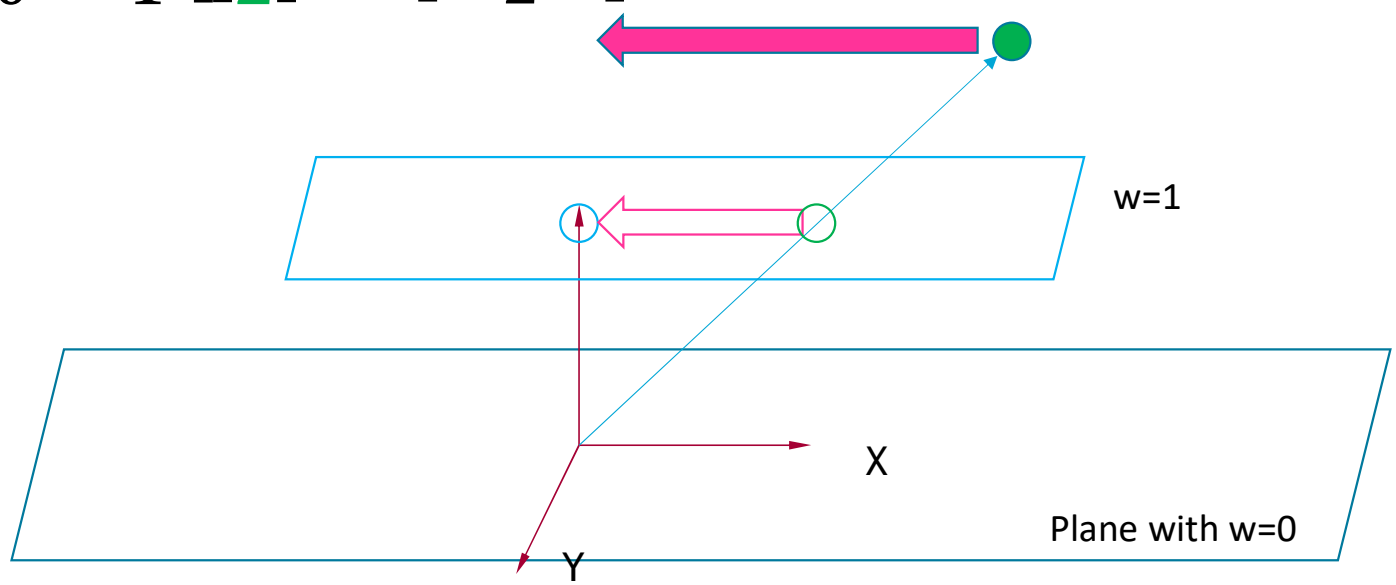
How does this work?

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ \mathbf{2} \end{bmatrix}$$



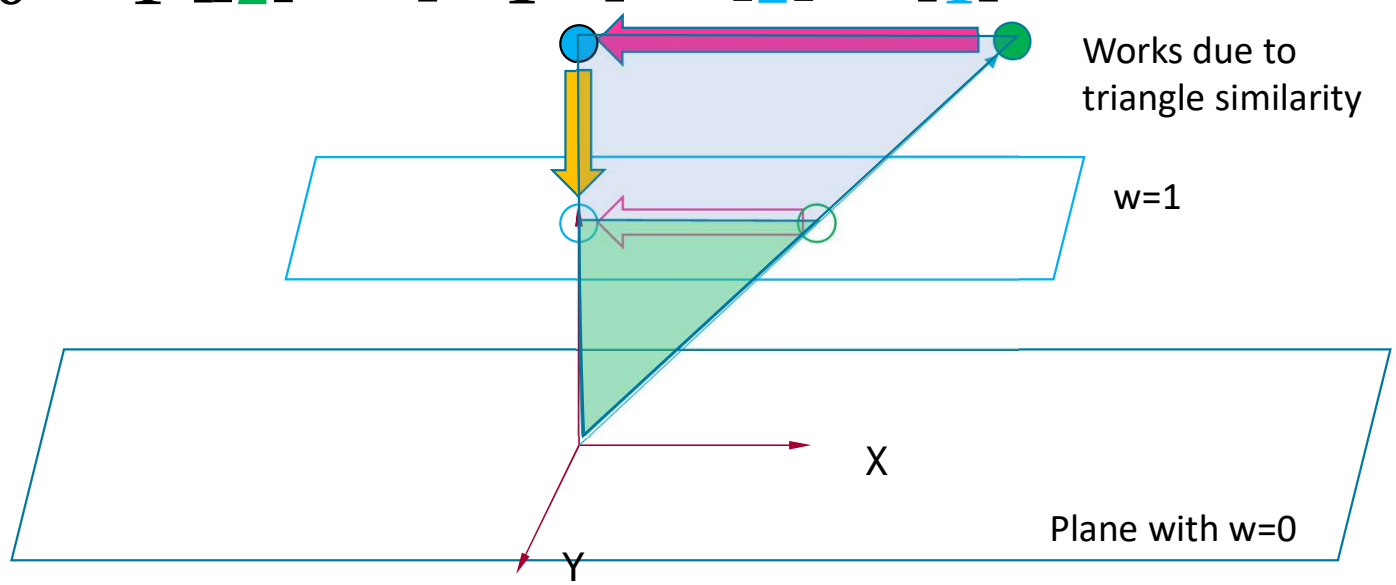
How does this work?

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 - 2 \\ 0 + 0 \\ 2 \end{bmatrix}$$



How does this work?

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 - 2 \\ 0 + 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$



Translations in homog. coordinates

- R^2

$$\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} = \begin{bmatrix} x + tx \\ y + ty \end{bmatrix}$$

- P^2

$$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + tx \\ y + ty \\ 1 \end{bmatrix}$$

Rotation in R^2

- Is a matrix multiplication:

$$P' = \mathbf{R}P$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Rotation in homog. coordinates

- R^2

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\theta)\,x - \sin(\theta)\,y \\ \sin(\theta)\,x + \cos(\theta)\,y \end{bmatrix}$$

- P^2

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

Is there any issue when choosing a different representative?

In a projective space (x,y,1) = (wx,wy,w).
Does it really give the same result if I use that vector?

Yes! This is valid for any matrix!

M(wx,wy,w)

=M(w(x,y,1)) //use vector space

=wM(x,y,1) //use linear transformation

=M(x,y,1) //use projective space

Rotation in homog. coordinates

- R^2

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos(\theta)\, x - \sin(\theta)\, y \\ \sin(\theta)\, x + \cos(\theta)\, y \end{bmatrix}$$

- P^2

$$\begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} wx \\ wy \\ w \end{bmatrix} = \begin{bmatrix} \cos(\theta)\, wx - \sin(\theta)\, wy \\ \sin(\theta)\, wx + \cos(\theta) wy \\ w \end{bmatrix}$$

But…

- So far, we only made things more complicated!
- Is there any benefit to all of this?
- Yes, let’s look at one example!

Transformation Compositing

- All rigid transformations become linear!

  (and many others too…)

- It is possible to produce linear applications that translate and rotate

$$\mathbf{M} = \mathbf{(Rot)(Trans)}$$

Rotation around point Q

- Rotation around point Q:
  - Translate Q to origin($\mathbf{T}_Q$),
  - Rotate around origin ($\mathbf{R}_\Theta$)
  - Translate back to Q ($\mathbf{T}_{-Q}$).

$$\longrightarrow \quad P' = (\mathbf{T}_{-Q})\mathbf{R}_\Theta \mathbf{T}_Q \, P$$

And in 3 dimensions ?

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

- The same!
- Add a fourth coordoninate, *w*

- All transformations are 4x4 matrices

Translations in 3D

$$T(t_x, t_y, t_z) = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{cases} x' = x + wt_x \\ y' = y + wt_y \\ z' = z + wt_z \\ w' = w \end{cases}$$

Rotations in 3D

- Rotation = axis and angle

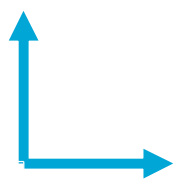
- Rotations around x,y,z axis are simple matrices
  - All axes can be achieved by combining these
    (it is a bit cumbersome though…)

Rotation about *z*-axis

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Axis z not changing

Quick verification : rotation of π/2
Should change *x* in *y*, and *y* in -*x*



$$R_z(\tfrac{\pi}{2}) = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about *x*-axis

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Axis x not changing

Quick verification : rotation of π/2
Should change *y* in *z*, and *z* in -*y*

$$R_x(\tfrac{\pi}{2}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation about *y*-axis

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

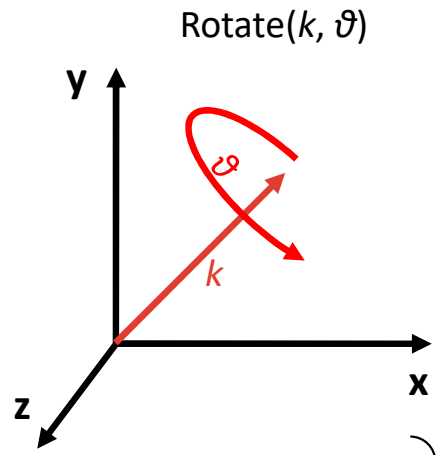
Axis y not changing

Quick verification : rotation of π/2
Should change *z* in *x*, and *x* in -*z*

$$R_y(\tfrac{\pi}{2}) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Rotation around (k_x, k_y, k_z)

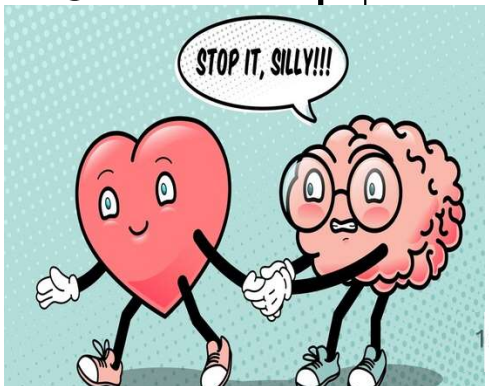
- Rodrigues Formula



$$\begin{pmatrix} k_xk_x(1-c)+c & k_zk_x(1-c)-k_zs & k_xk_z(1-c)+k_ys & 0 \\ k_yk_x(1-c)+k_zs & k_zk_x(1-c)+c & k_yk_z(1-c)-k_xs & 0 \\ k_zk_x(1-c)-k_ys & k_zk_x(1-c)-k_xs & k_zk_z(1-c)+c & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

DO NOT LEARN THIS BY HEART!

where $c = \cos\vartheta$ & $s = \sin\vartheta$

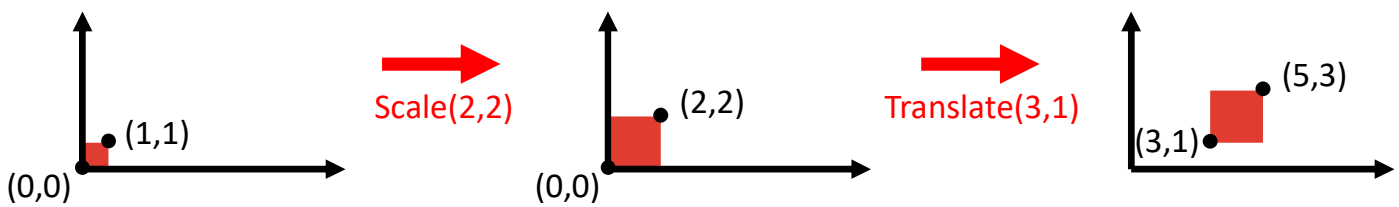


Attention !

- Matrix Multiplication is not commutative
- The order of transformations is important
  - Rotation then translation != transl. then rotation

Example

Scaling + Translation

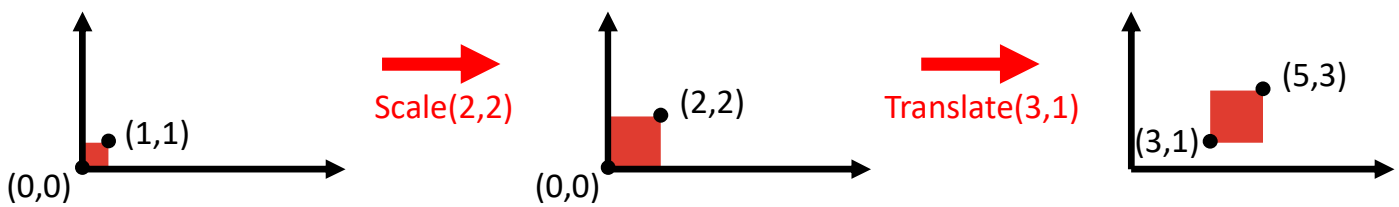


Matrix combination: $p' = T(Sp) = TSp$

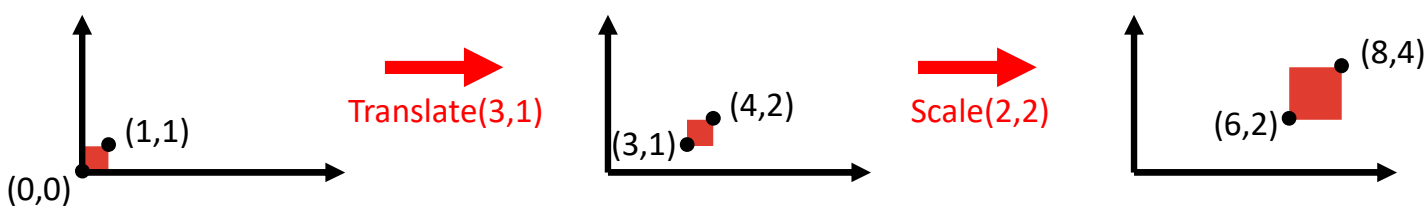
$$TS = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

NOT commutative

Scaling + translation: $p' = T(Sp) = TSp$



Translation + scaling: $p' = S(Tp) = STp$



NOT commutative

$$TS = \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 3 \\ 0 & 2 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

$$ST = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 0 & 6 \\ 0 & 2 & 2 \\ 0 & 0 & 1 \end{pmatrix}$$

Questions?

Camera Model

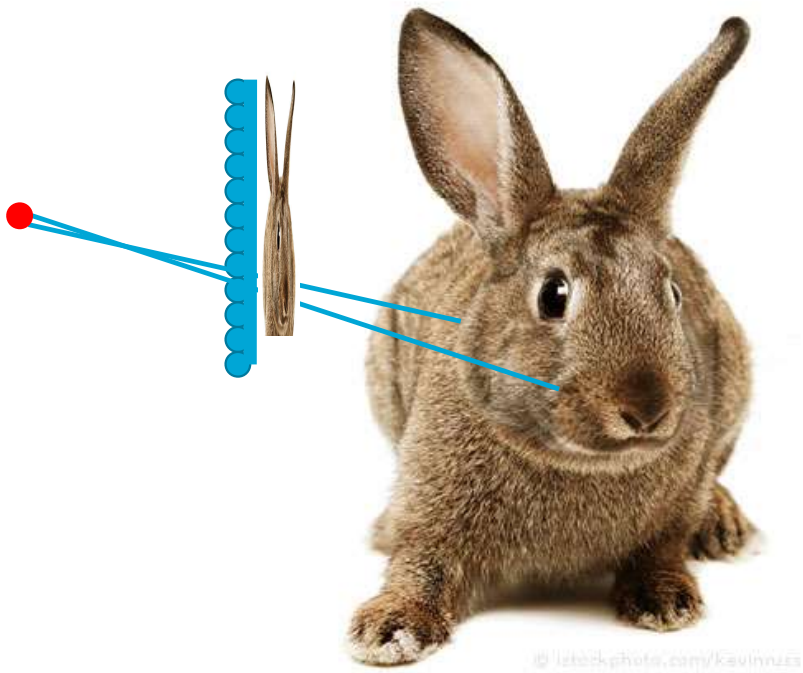
- Pinhole Perspective Projection

  using matrices

= ?

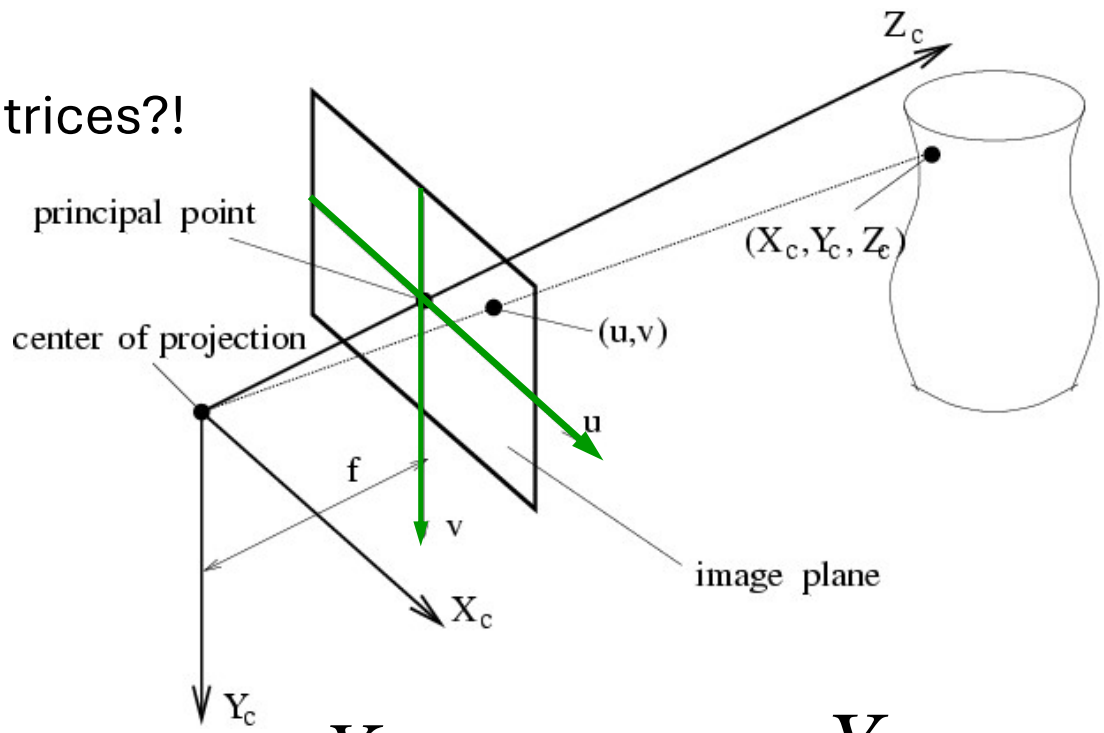
Virtual Camera

- Pinhole where real cameras have lens
- Virtual Camera Plane



Perspective Projection

- Represent with matrices?!



$$u = f\frac{X}{Z} \qquad v = f\frac{Y}{Z}$$

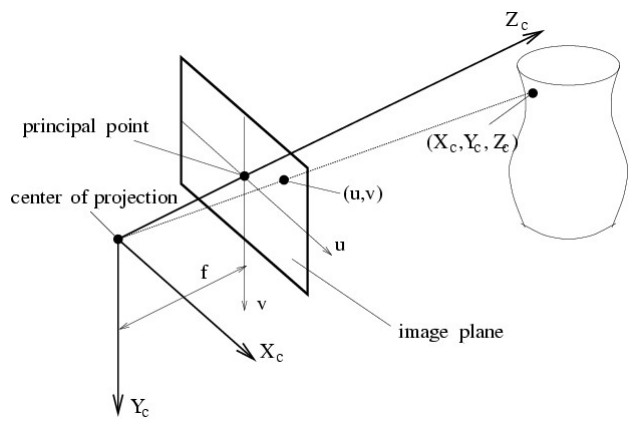
Perspective Projection

- "Homogenize" the points

$$P = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad \Rightarrow \quad \tilde{P} = \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

- Look for M such that:

$$M\tilde{P} = \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} fX/Z \\ fY/Z \\ 1 \end{pmatrix}$$



$$u = f\frac{X}{Z} \qquad v = f\frac{Y}{Z}$$

What is the problem?

$$M\widetilde{P} = \begin{bmatrix} a & b & c & d \\ e & q & g & h \\ i & j & m & n \\ k & l & o & p \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix}$$

What is the problem?

$$M\widetilde{P} = \begin{bmatrix} a & b & c & d \\ e & q & g & h \\ i & j & m & n \\ k & l & o & p \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} ax + by + cz + dw \\ \dots \\ \\ \end{bmatrix}$$

What is the problem?

$$M\widetilde{P} = \begin{bmatrix} a & b & c & d \\ e & q & g & h \\ i & j & m & n \\ k & l & o & p \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} = \begin{bmatrix} ax + by + cz + dw \\ \dots \\ \\ \end{bmatrix} \Rightarrow \frac{fx}{z} \; ?$$

$$u = f\,\frac{X_C}{Z_C}$$

Perspective Projection

- Hint: Think projective!

$$M\widetilde{P} = \begin{pmatrix} fX/Z \\ fY/Z \\ 1 \end{pmatrix} \Rightarrow M\widetilde{P} = \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix}$$

Perspective Projection

- Solution:

$$M = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

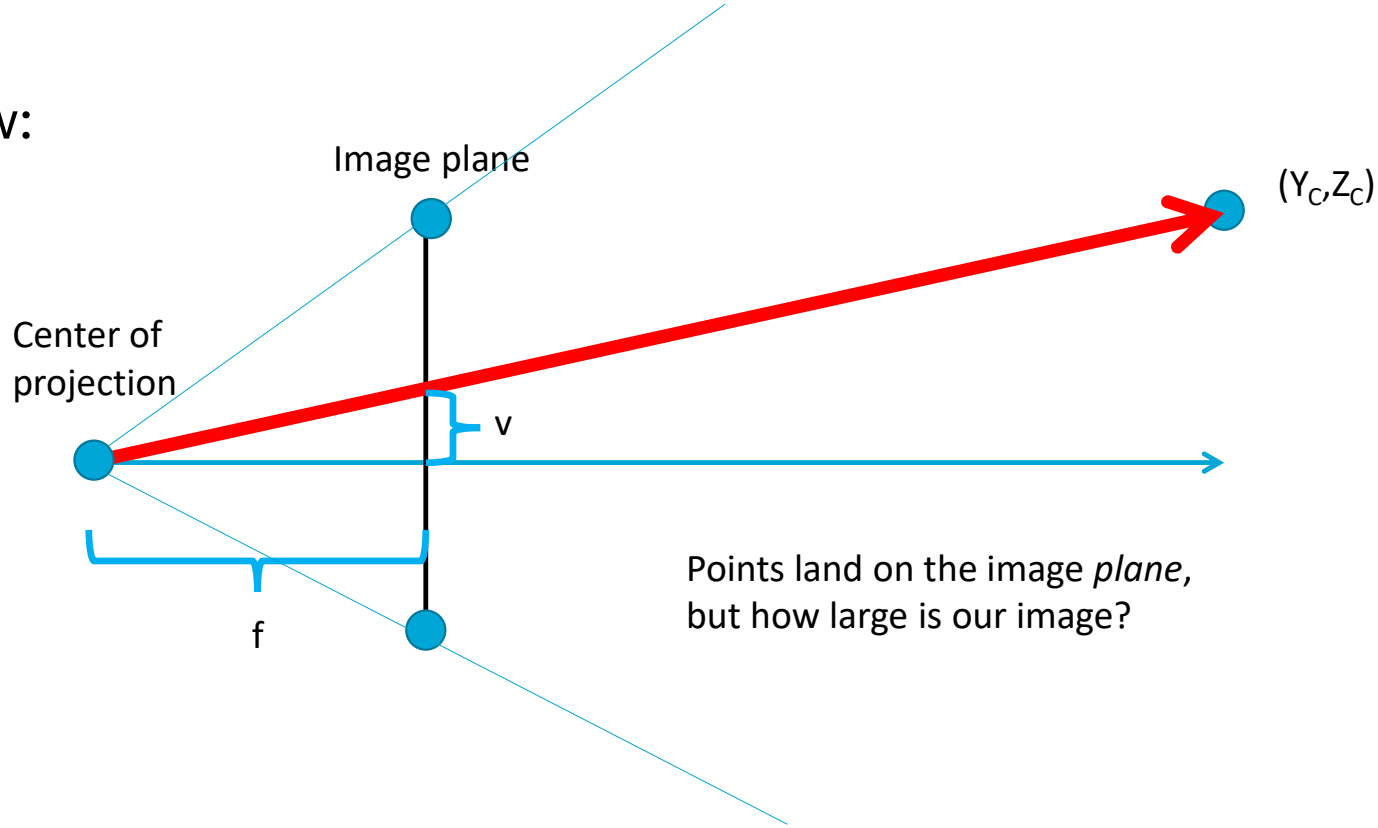
Putting things together: Camera Model

- Projection + Movement + Rotation

$$M = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{00} & r_{01} & r_{02} & t_0 \\ r_{10} & r_{11} & r_{12} & t_1 \\ r_{20} & r_{21} & r_{22} & t_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Virtual Camera Model

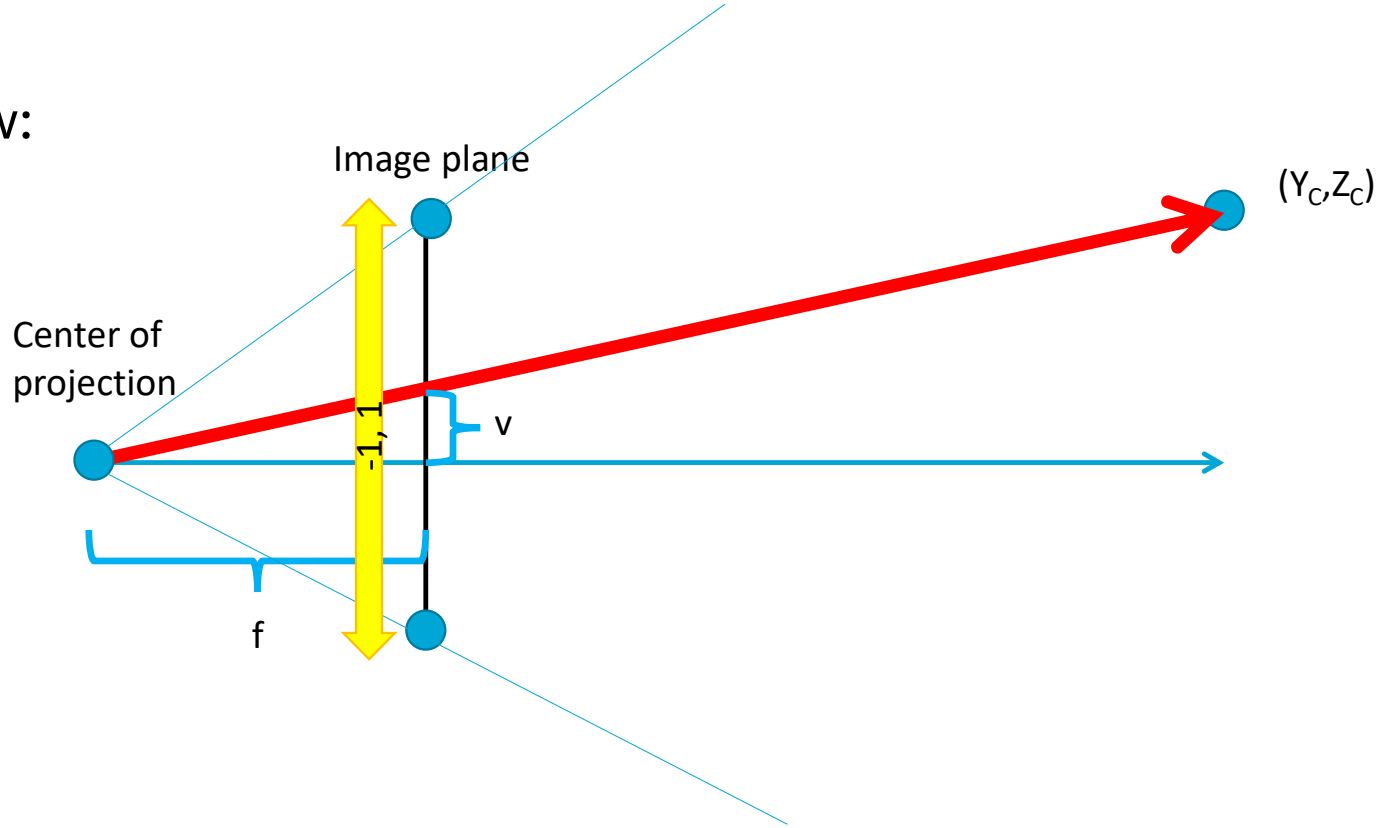
- sideview:



Points land on the image *plane*, but how large is our image?

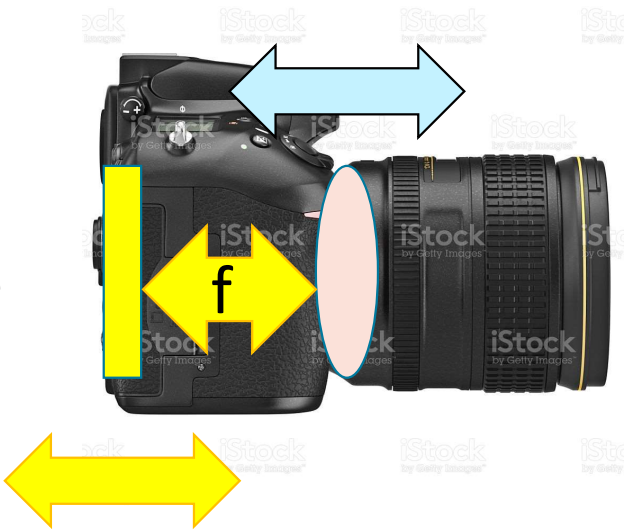
Virtual Camera Model

- sideview:



Real camera

Sensor has a fixed size
“[-1,1]”

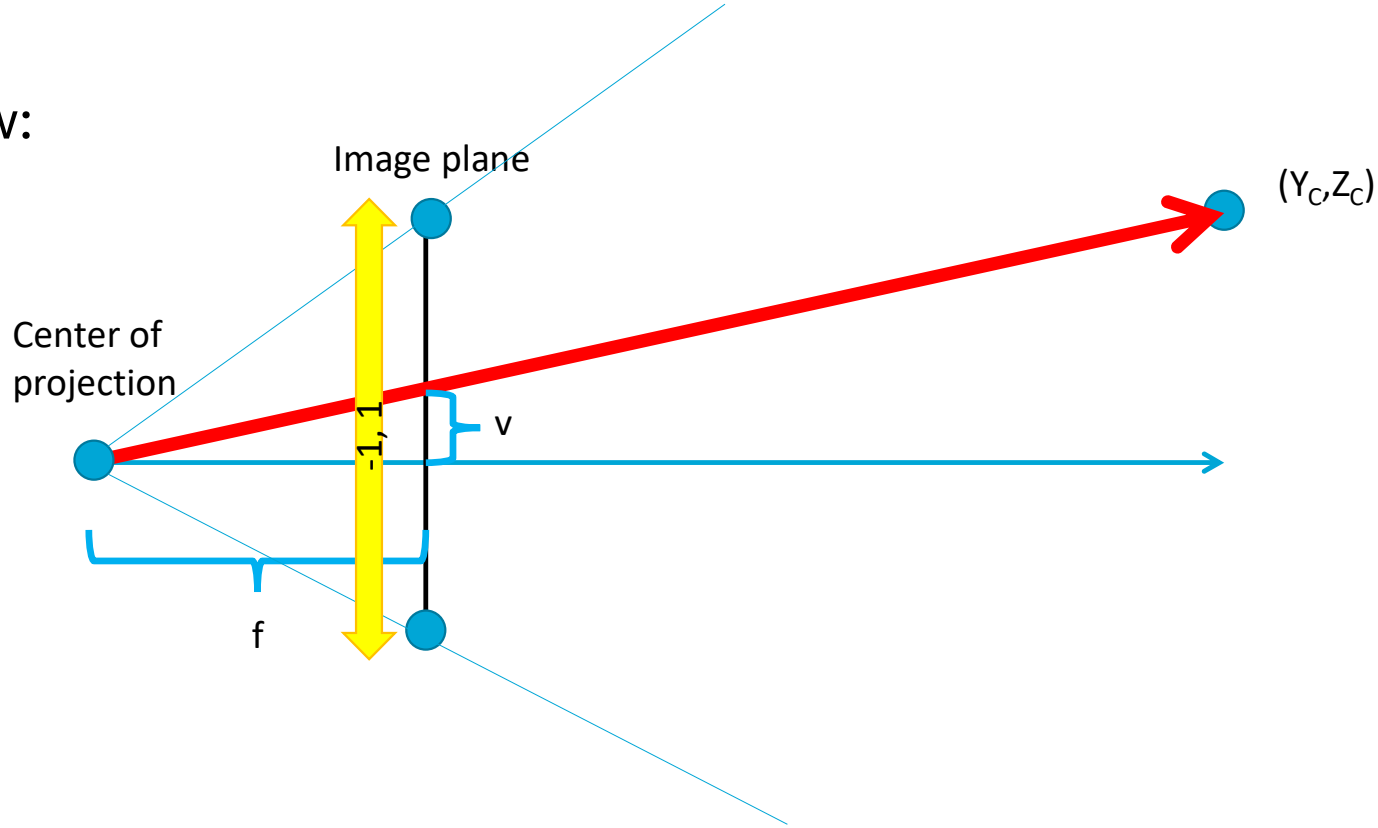


Virtual Camera Model

- The world literally revolves around the camera!
- Just deform the scene in the “right” way and we can always assume that
  - Camera centre (eye) is at the origin
  - Camera is oriented along the z-axis
  - Image aligned with x-y and has unit size $[-1,1]^2$
  - Only keep parameter f – focal length

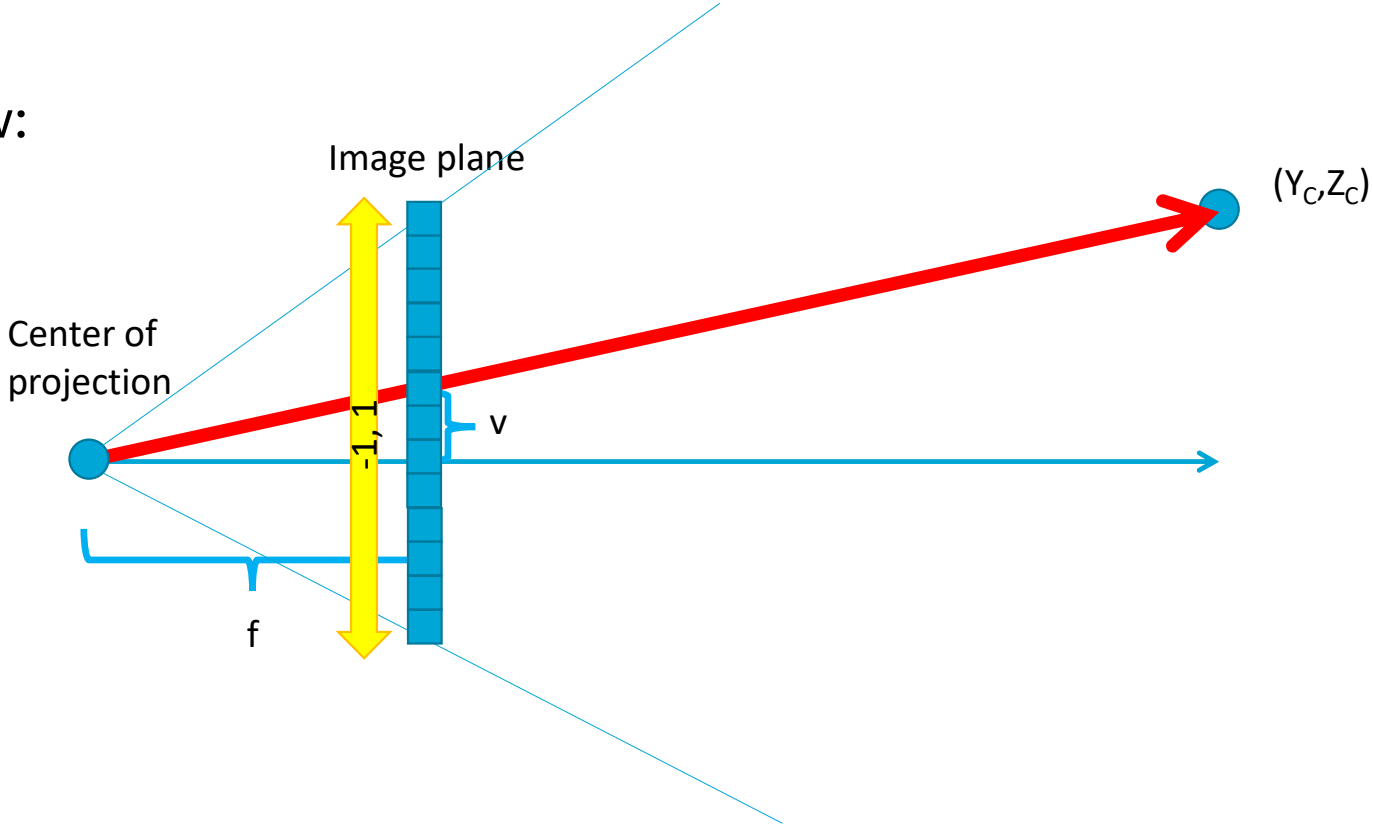
Virtual Camera Model

- sideview:



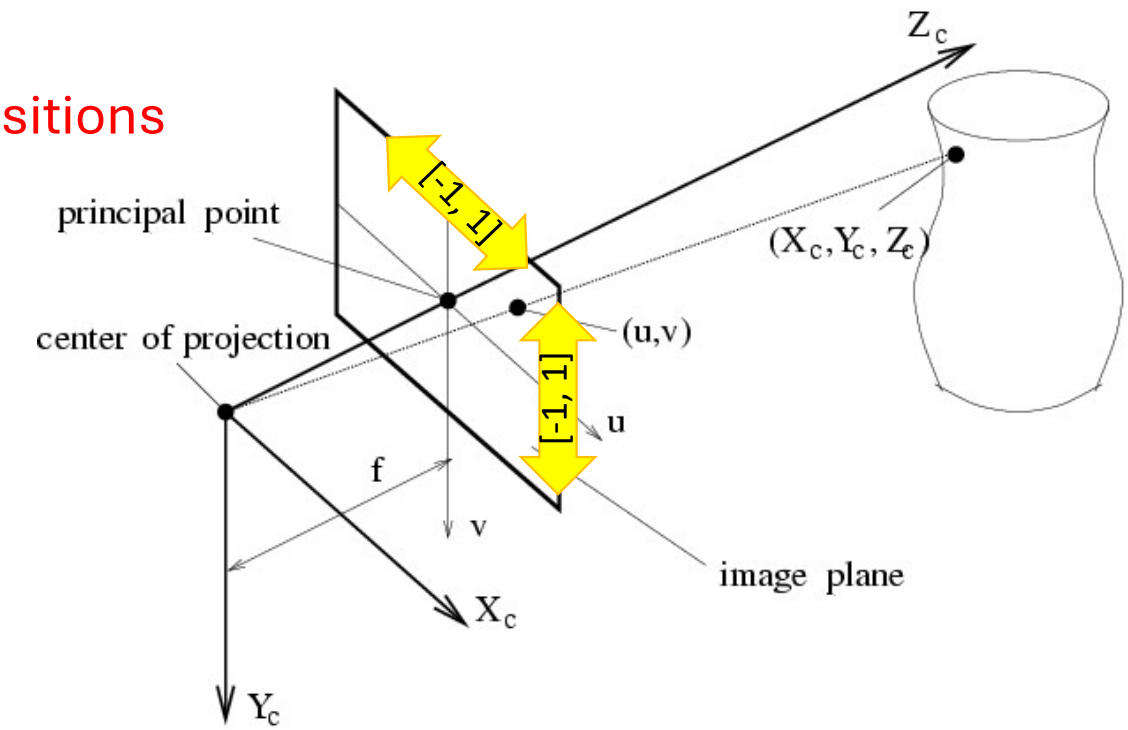
Virtual Camera Model

- sideview:



Adding Image Space to Camera Model

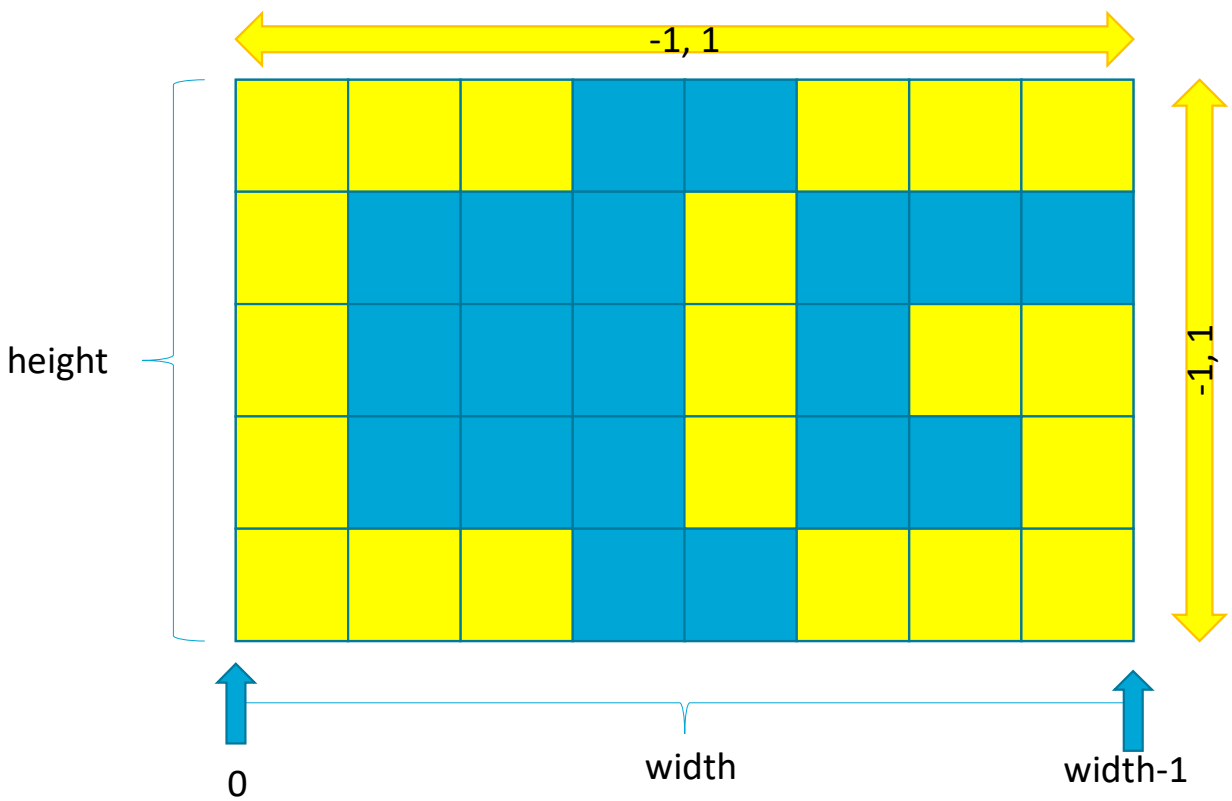
- Make u, v pixel positions



$$u = f\frac{X}{Z} \qquad v = f\frac{Y}{Z}$$

Viewport Transformation

- (-1,1) x (-1,1) ➡ [0, width-1]x[0,height-1]



Homework:
Find a matrix
to do this
mapping

Hint:
it has the form

$$\begin{pmatrix} k_x & 0 & x_0 \\ 0 & k_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}$$

Complete Camera Model

- Finally:

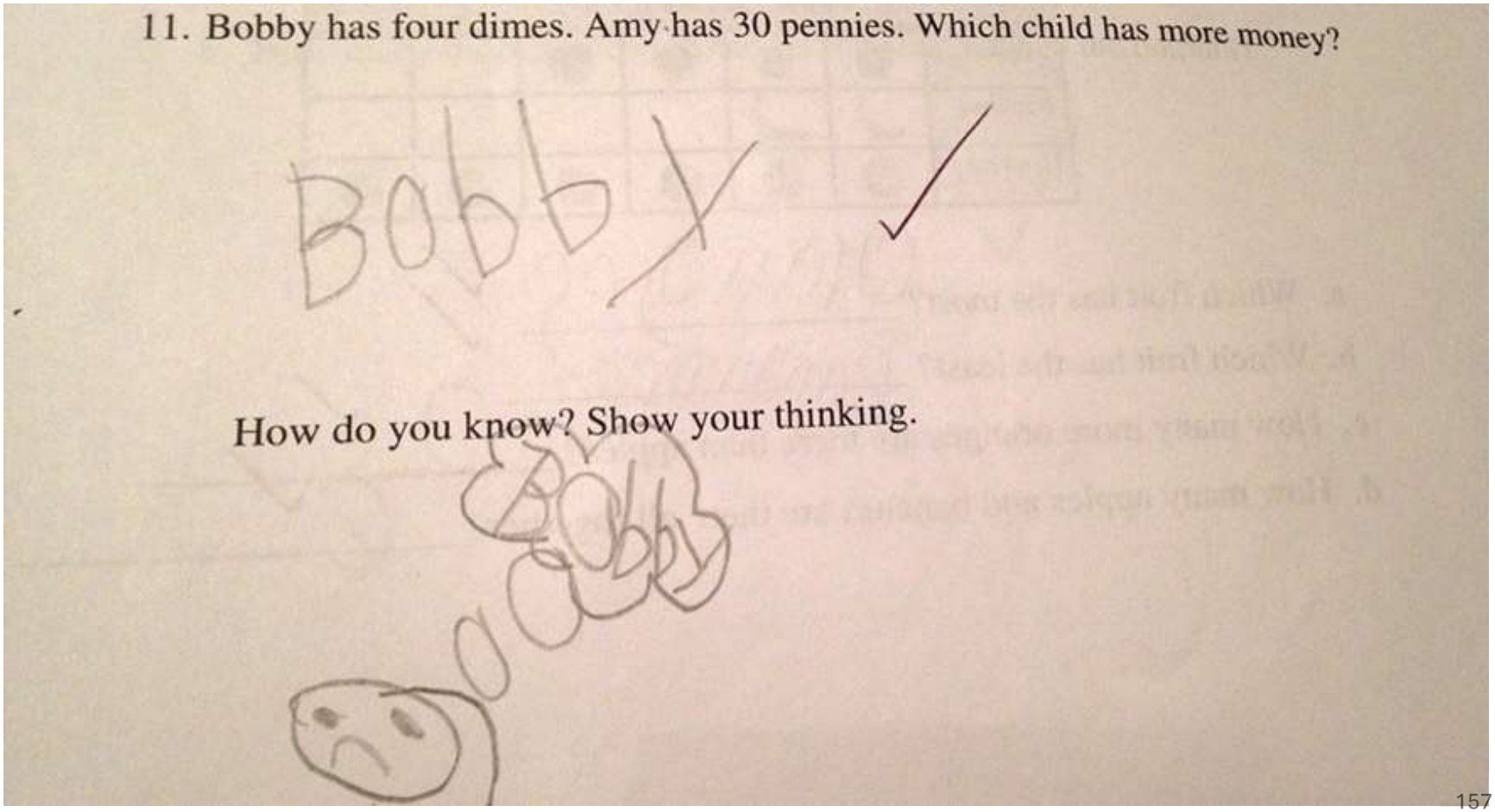
$$\underbrace{\begin{pmatrix} k_x & 0 & x_0 \\ 0 & k_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{Viewport}} \underbrace{\begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\text{Projection}} \underbrace{\begin{pmatrix} r_{00} & r_{01} & r_{02} & t_0 \\ r_{10} & r_{11} & r_{12} & t_1 \\ r_{20} & r_{21} & r_{22} & t_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\text{Camera Matrix}}$$

Pixel mapping

"standard camera" projection

Deforms scene so that a "standard camera" can be used

Questions?


11. Bobby has four dimes. Amy has 30 pennies. Which child has more money?
Bobby
How do you know? Show your thinking.

Complete Camera Model

- Finally:

$$P = \underbrace{\overbrace{\begin{pmatrix} a_x & 0 & x_0 \\ 0 & a_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}}^{\text{image}} \overbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}^{\text{projection}}} \overbrace{\begin{pmatrix} r_{00} & r_{01} & r_{02} & t_0 \\ r_{10} & r_{11} & r_{12} & t_1 \\ r_{20} & r_{21} & r_{22} & t_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}}^{\text{orientation/location}}$$

$$\begin{pmatrix} a_x & 0 & x_0 \\ 0 & a_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

Complete Camera Model

- Finally:

$$P = \begin{pmatrix} a_x & 0 & x_0 \\ 0 & a_y & y_0 \\ 0 & 0 & 1 \end{pmatrix} \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} r_{00} & r_{01} & r_{02} & t_0 \\ r_{10} & r_{11} & r_{12} & t_1 \\ r_{20} & r_{21} & r_{22} & t_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{\begin{pmatrix} r_{00} & r_{01} & r_{02} & t_0 \\ r_{10} & r_{11} & r_{12} & t_1 \\ r_{20} & r_{21} & r_{22} & t_2 \end{pmatrix}}$$

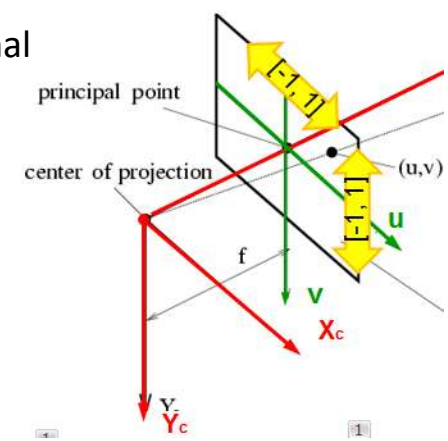
Complete Camera Model

- Finally (notation often used in vision literature):

$$P = \underbrace{\begin{pmatrix} a_x & 0 & x_0 \\ 0 & a_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\text{intrinsic/ internal parameters}} \underbrace{\begin{pmatrix} r_{00} & r_{01} & r_{02} & t_0 \\ r_{10} & r_{11} & r_{12} & t_1 \\ r_{20} & r_{21} & r_{22} & t_2 \end{pmatrix}}_{\text{extrinsic / external parameters}}$$

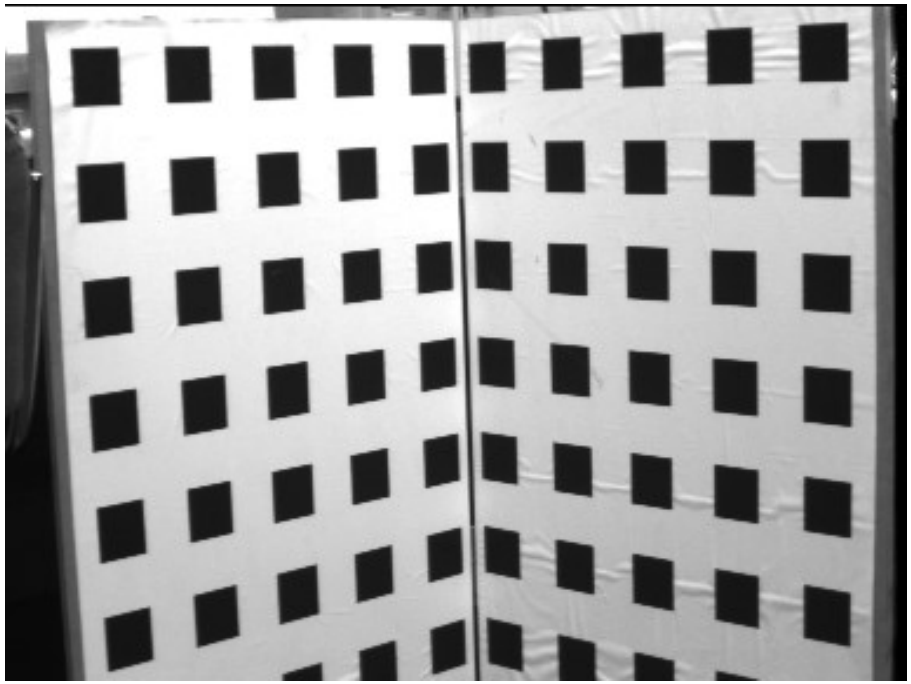
Change settings

Move Camera



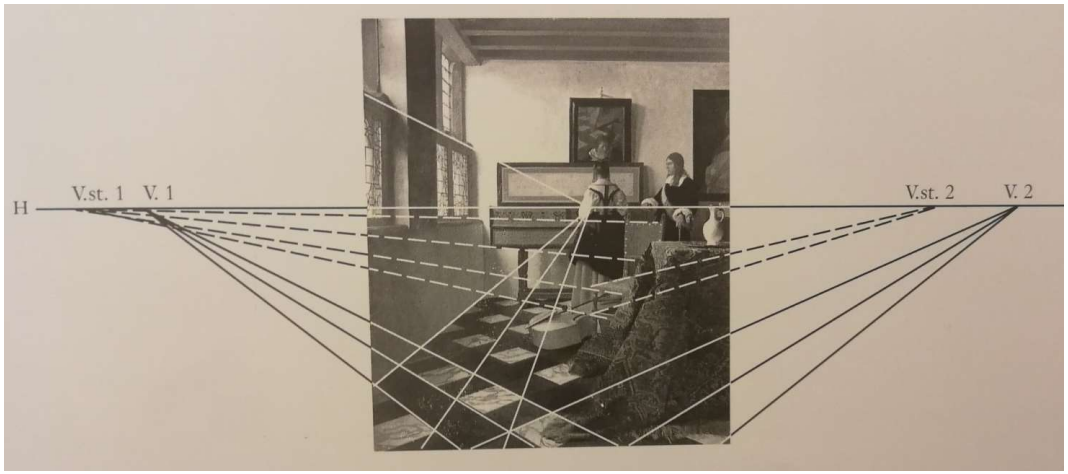
Next time!

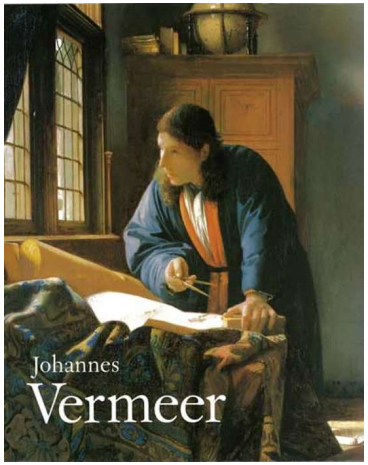
- We can determine these matrices by taking a photo of something we know!



Why is a single camera useful?

Johannes Vermeer, *The Music Lesson*, ca. 1662-1665, Royal Collection Trust





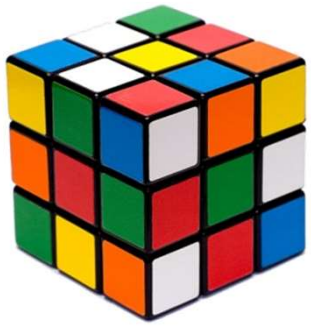
Why is a single camera useful?

- Example 1:

  Augmented reality

- Example 2:

  Fancy control device



Summary

Virtual Camera Model

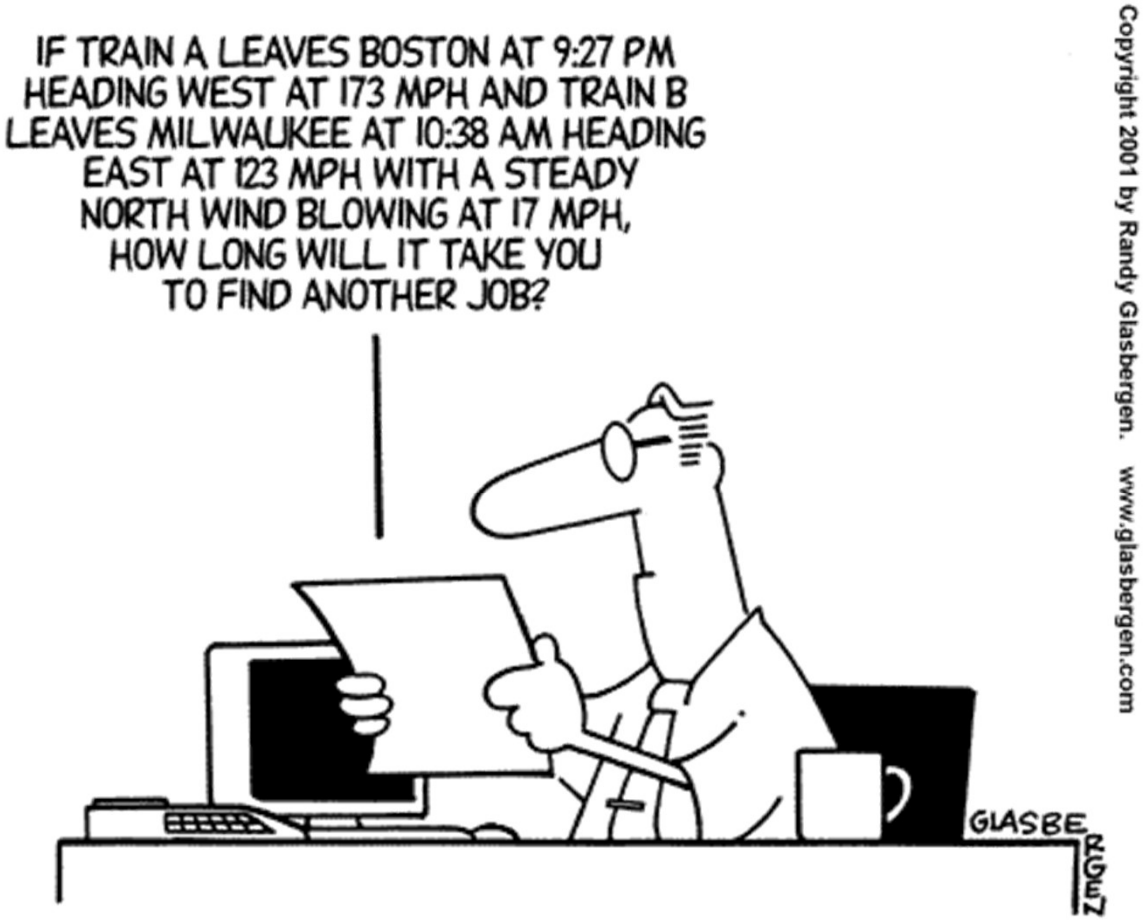
- Homogeneous Coordinates
- Transformations
- Camera Projection
- Camera in matrix representation

Next time:

For a real camera, how to find the matrices?

Given the matrices, how to reconstruct 3D?

Questions?



Another layoff at the textbook publishing company.

Thank you very much for your attention!