KAIST

CS 380

Introduction to Computer Graphics

LAB (3)

2018.03.26

VISUAL COMPUTING Lab

- Note that we will not teach the answer of this homework, as the homework must be done by yourself.

- Instead, we will explain the necessary background which is required to finish this HW.

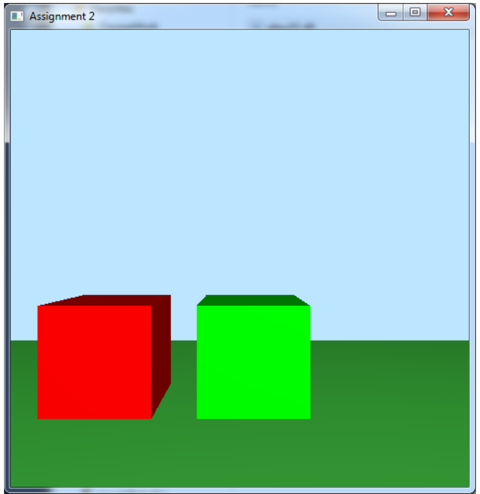
- Some part of this slides are from lecture notes of this course.

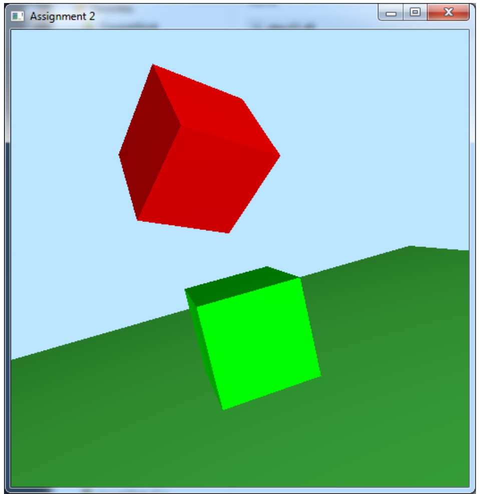
- If you have problem with std::min, add #include <algorithm> in cvec.h

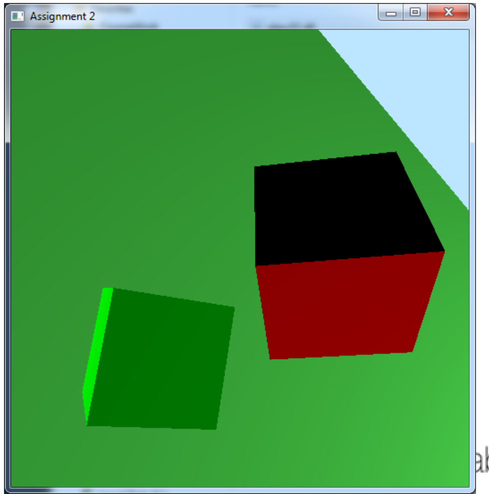
Goals

1. Draw two cubes
2. Create different viewpoints
3. Move the objects freely w.r.t. the current viewpoint frame

- We have four tasks to complete in this hw2.







Affine Transformation

Full affine transformation — Translation — Rotation

$$\begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & h \\ 0 & 0 & 1 & l \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a & b & c & 0 \\ e & f & g & 0 \\ i & j & k & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$\begin{bmatrix} l & t \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} i & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} l & 0 \\ 0 & 1 \end{bmatrix}$$

$$A = TL$$

Transformation Respect

KAIST

- We are transforming a point $\tilde{p}$ in a frame $\vec{\mathbf{f}}^t$

$$\tilde{p} = \vec{\mathbf{f}}^t \mathbf{c}$$

- With a matrix

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

the stretches by factor of two in first axis of $\vec{\mathbf{f}}^t$

- Performing a transform: $\vec{\mathbf{f}}^t \mathbf{c} \Rightarrow \vec{\mathbf{f}}^t S\mathbf{c}$

- Suppose another frame: $\vec{\mathbf{a}}^t = \vec{\mathbf{f}}^t A$

Transformation Respect

KAIST

- We could express the point with a new coordinate vector

$$\tilde{p} = \vec{\mathbf{f}}^t \mathbf{c} = \vec{\mathbf{a}}^t \mathbf{d}$$

$$\vec{\mathbf{f}}^t \mathbf{c} = \vec{\mathbf{f}}^t A \mathbf{d}$$

$$\mathbf{d} = A^{-1} \mathbf{c}$$

$$\vec{\mathbf{a}}^t = \vec{\mathbf{f}}^t A$$

$$\vec{\mathbf{f}}^t = \vec{\mathbf{a}}^t A^{-1}$$

- Now S transforms the point $\tilde{p}$ with respect to $\vec{\mathbf{a}}^t$

$$\vec{\mathbf{a}}^t \mathbf{d} \Rightarrow \vec{\mathbf{a}}^t S \mathbf{d}$$

Transformation Respect

KAIST

- Point is transformed with respect to the the frame that appears immediately to the left of the transformation matrix in the expression.
- We read

$$\vec{\mathbf{f}}^t \Rightarrow \vec{\mathbf{f}}^t S$$

$\vec{\mathbf{f}}^t$ is transformed by S with respect to $\vec{\mathbf{f}}^t$

- We read

$$\vec{\mathbf{f}}^t = \vec{\mathbf{a}}^t A^{-1} \Rightarrow \vec{\mathbf{a}}^t S A^{-1}$$

$\vec{\mathbf{f}}^t$ is transformed by S with respect to $\vec{\mathbf{a}}^t$

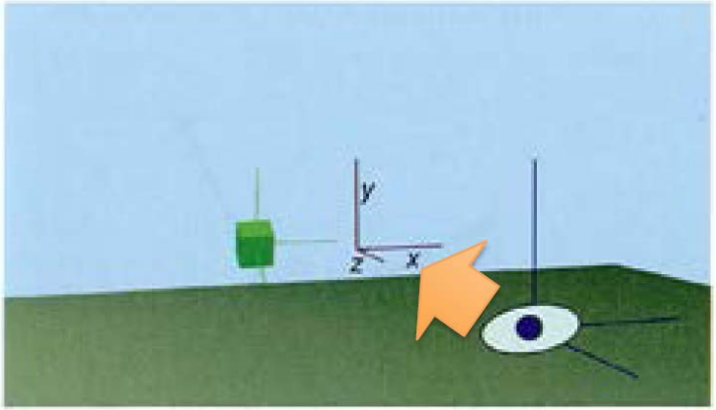
Frame

- World frame $\vec{w}^t$
  - an absolute frame in 3D space
  - All other frames are represented by this frame
- Object frame $\vec{o}^t$
  - All objects should have own frame

$$\vec{o}^t = \vec{w}^t O$$

- Eye frame $\vec{e}^t$

$$\vec{e}^t = \vec{w}^t E$$



(a) The frames

Eye Coordinate

KAIST

- we explicitly store the matrix E $\qquad \vec{\mathbf{e}}^t = \vec{\mathbf{w}}^t E$

$$\tilde{p} = \vec{\mathbf{o}}^t \mathbf{c} = \vec{\mathbf{w}}^t O\mathbf{c} = \vec{\mathbf{e}}^t E^{-1} O\mathbf{c}$$

  - Object coordinates: $\mathbf{c}$
  - World coordinates: $O\mathbf{c}$
  - Eye coordinates: $E^{-1}O\mathbf{c}$ MVM = inv(eyeRbt) * objRbt;
    (Model View Matrix)
  - Calculating the eye coordinates of every vertexes:

$$\begin{bmatrix} x_e \\ y_e \\ z_e \\ 1 \end{bmatrix} = E^{-1} O \begin{bmatrix} x_o \\ y_o \\ z_o \\ 1 \end{bmatrix}$$

Task1. Draw Two Cubes

- A frame needs to be defined to the each object.
- A frame of a red cube

World frame

$$\vec{o}_R^{\,t} = \vec{w}^{\,t} O_R$$

Transformed frame

Affine transformation

- A frame of a green cube

$$\vec{o}_L^{\,t} = \vec{w}^{\,t} O_L$$

Task2. Matrix Operations

KAIST

- linFact
  - Rotation matrix: 4 x 4
- transFact
  - Translation matrix: 4 x 4
- M = transFact(M) * linFact(M)
- By using these two function, you can obtain the full affine transformation matrix

Task3. Change Viewpoint

- Now, we have a pre-defined view point, which is called eye-view.
- The rendered image is generated in the frame of the eye-view.
- Our job is to change the eye-view according to the input of the user **by modifying the eye-view matrix**. (sky, cube1, cube2)

Task 4: Manipulation Mode

- If ‘o’ key is pressed, the object that we can manipulate should be changed (sky, cube1, cube2)

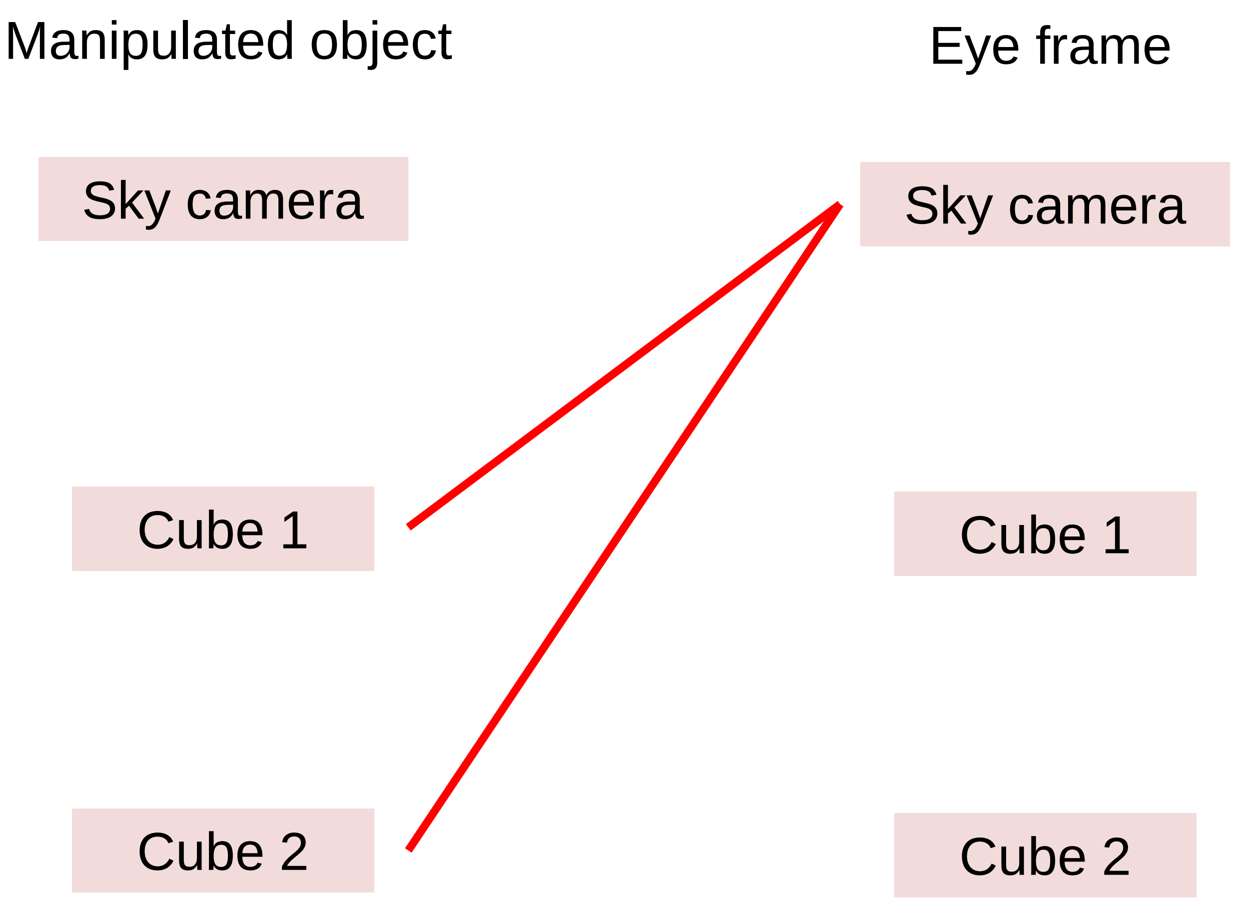
- You should choose different frames carefully depending on object and eye mode

  (8 combinations)

- In order to complete task4, you should utilize a transformation w.r.t. a frame

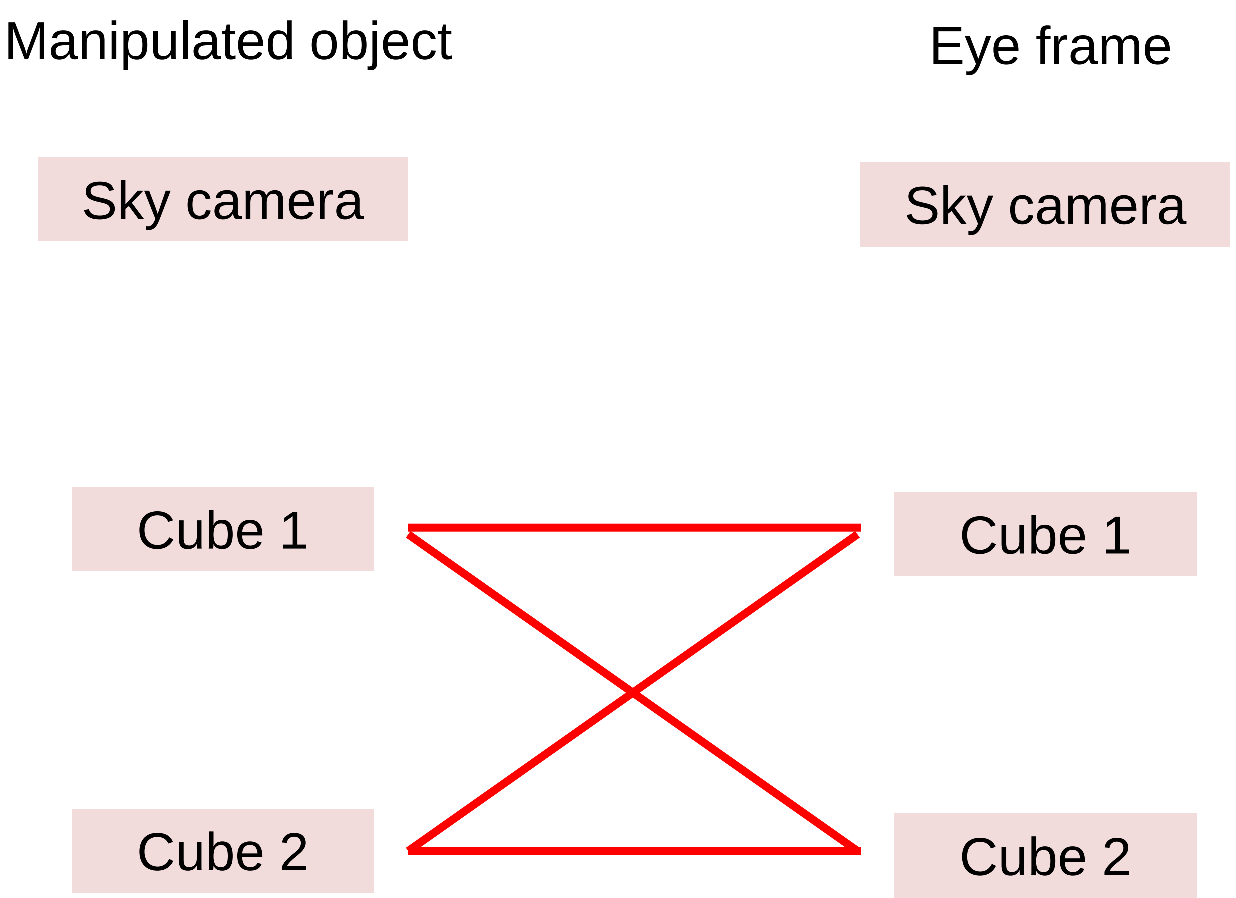
Task 4: Manipulation Mode

$\vec{a}^{t}$ should be the cube-sky frame



Task 4: Manipulation Mode

$\vec{a}^t$ should be the cube-cube frame



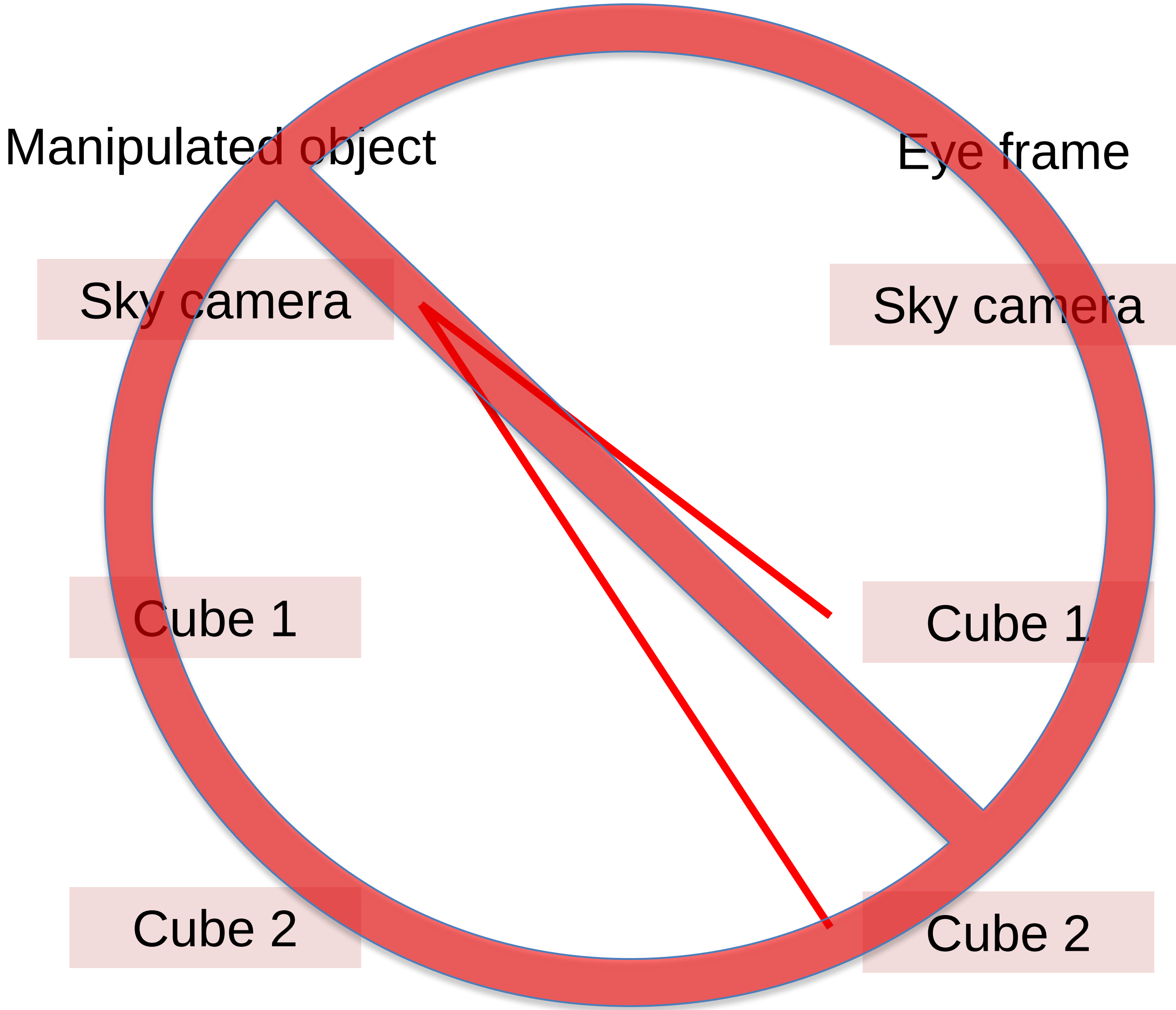
Task 4: Manipulation Mode

$\vec{a}^t$ should be the world-sky frame or sky-sky frame

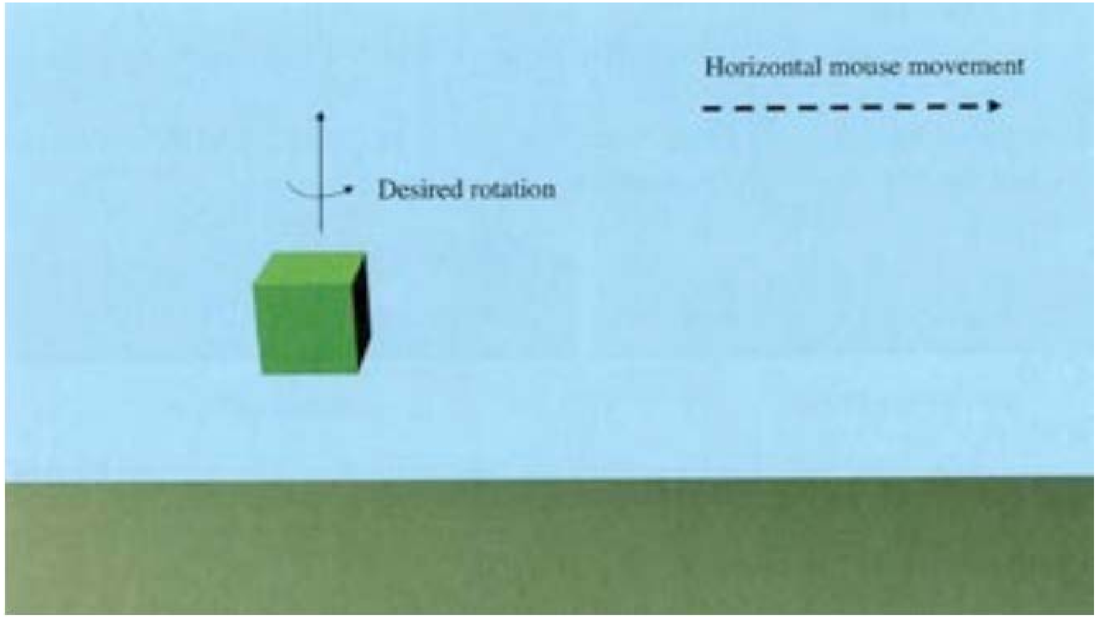
change mode by 'm'

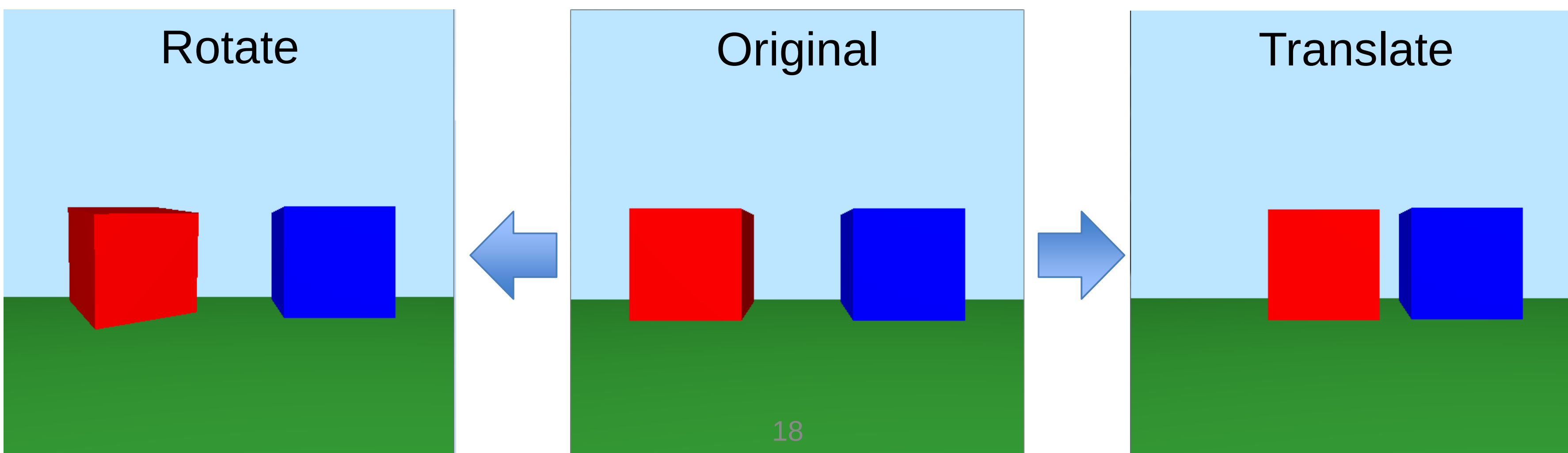
Manipulated object	Eye frame
Sky camera	Sky camera
Cube 1	Cube 1
Cube 2	Cube 2

Task 4: Manipulation Mode



Task 4: Move Objects





Task 4: How to choose the frame?

- Translation : direction
- Rotation : position (origin) + direction (axis)

- Recalling the Affine transform.: $A = TR$
- The object's Affine transform.: $O = (O)_T (O)_R$
- The eye's Affine transform.: $E = (E)_T (E)_R$

$$\vec{\mathbf{a}}^t = \vec{\mathbf{w}}^t (O)_T (E)_R$$