

Computer Animation

Lesson 7 - Camera animation

Remi Ronfard, Nov 2019

3 Staging

Staging is the presentation of an idea so it is completely and unmistakably clear; this principle translates directly from 2-D hand drawn animation. An action is staged so that it is understood; a personality is staged so that it is recognizable; an expression so that it can be seen; a mood so that it will affect the audience. [26]

To stage an idea clearly, the audience's eye must be led to exactly where it needs to be at the right moment, so that they will not miss the idea. Staging, anticipation and timing are all integral to directing the eye. A well-timed anticipation will be wasted if it is not staged clearly.



« *Present the idea so that it is unmistakably*

FIGURES 7-8. In *Leo Jr.*, all action was staged to the side for clarity.

3. Staging

Motivation

- ☒ There are 3 elements in computer graphics
 - lights, scene objects, and
 - the camera (focus of this talk)
- ☒ Camera control is hard (7 DOFs)
 - position (3), direction (3), field of view (1)
- ☒ We may learn from cinematography

- Source: Virtual cinematography theory and practice for automatic real-time camera control and directing, by Liwei He, Microsoft Research.

Roles in cinematography

☒ Low-level responsibilities

– Cameraman

- Position and move the camera

– Film editor

- Keep the film to proper length
- Decide shot transitions

☒ High-level responsibilities

– Script writer and director

- Story-telling

Camera control level 1

- ☒ Input: camera position and direction
- ☒ Output: camera transformation
- ☒ Application: low-level graphics library routines (Direct3D)

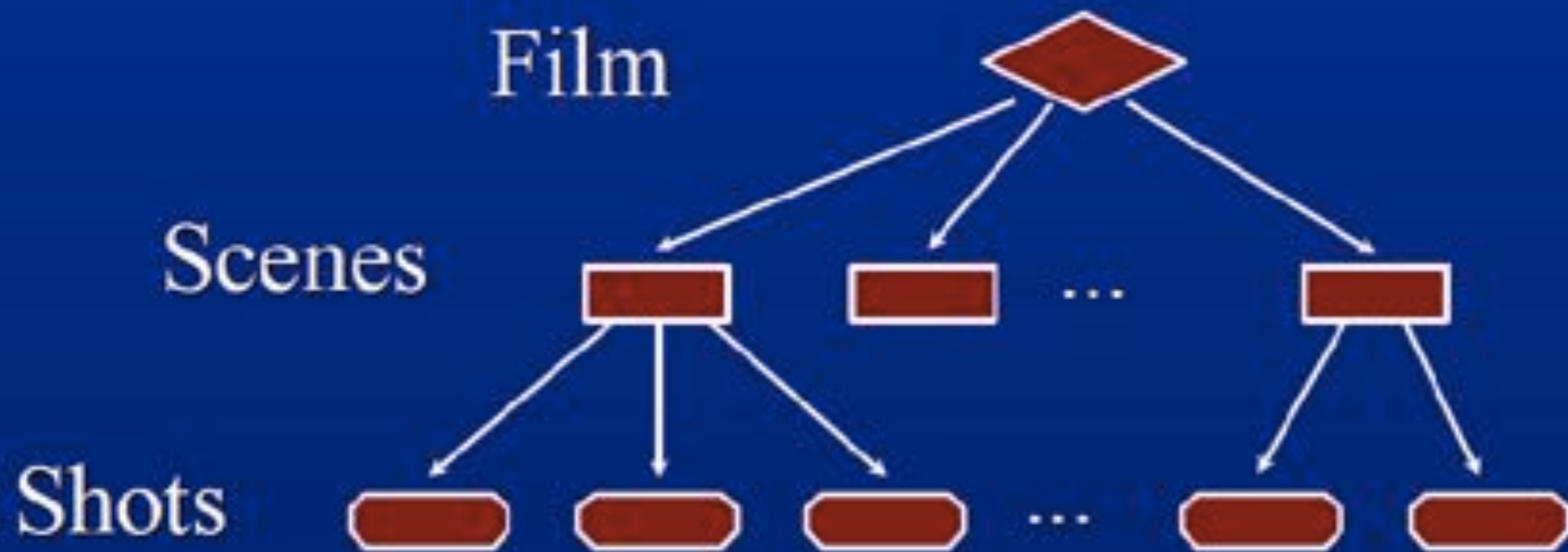
Camera control level 2

- ☒ Input: show both A and B, follow C, etc.
- ☒ Output: camera position and direction
- ☒ Application: 3D game, computer animation package

Camera control level 3

- ☒ Input: high-level user directions
 - show a conversation
 - show a car chase scene
- ☒ Output: a sequence of level 2 camera spec
- ☒ Applications: graphical chat, teleconferencing, Virtual Reality games

Film structure



Camera distance



Close up



Close shot



Medium shot



Full shot



Long shot

The line of interest



External camera



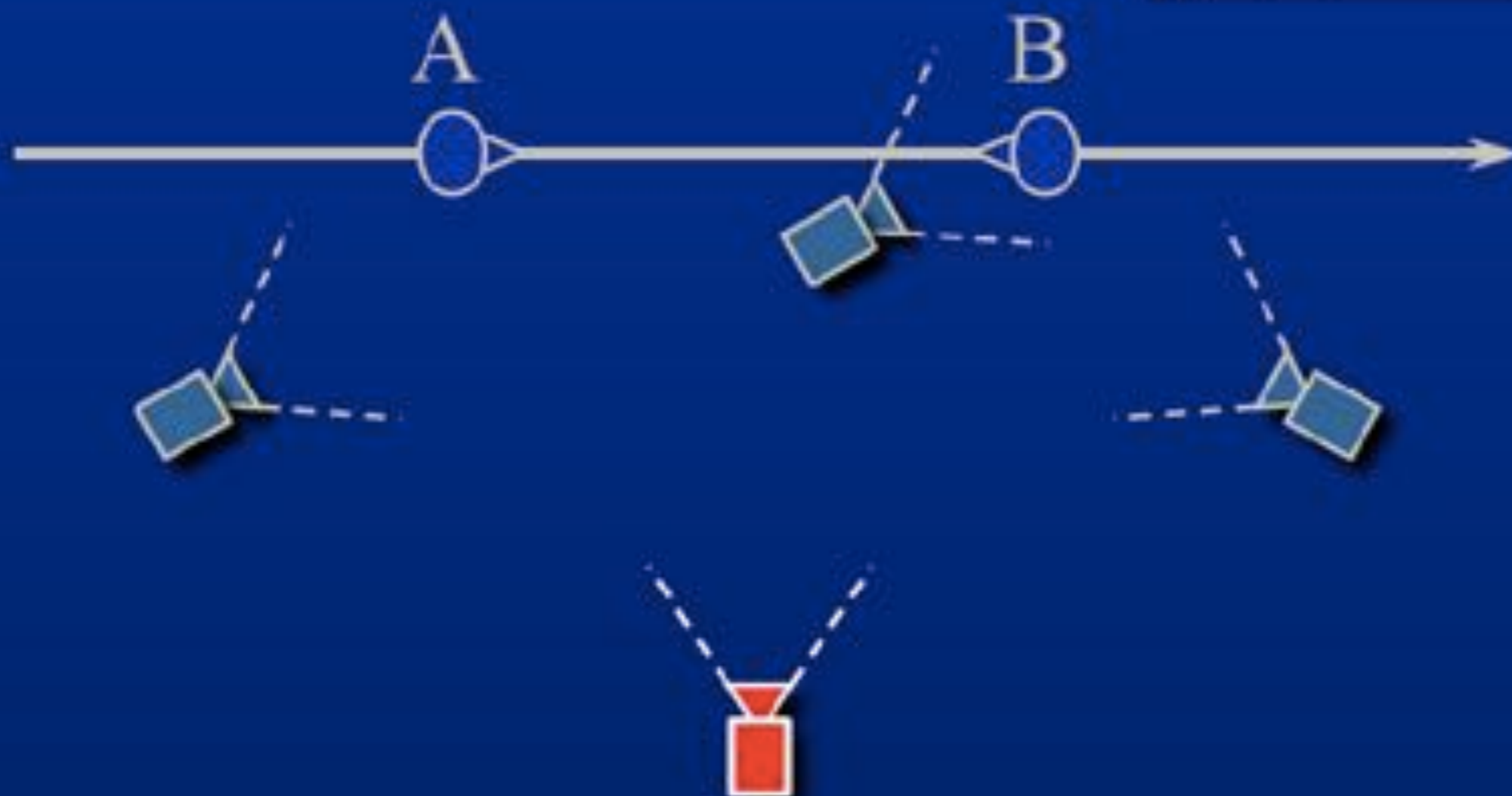
External camera



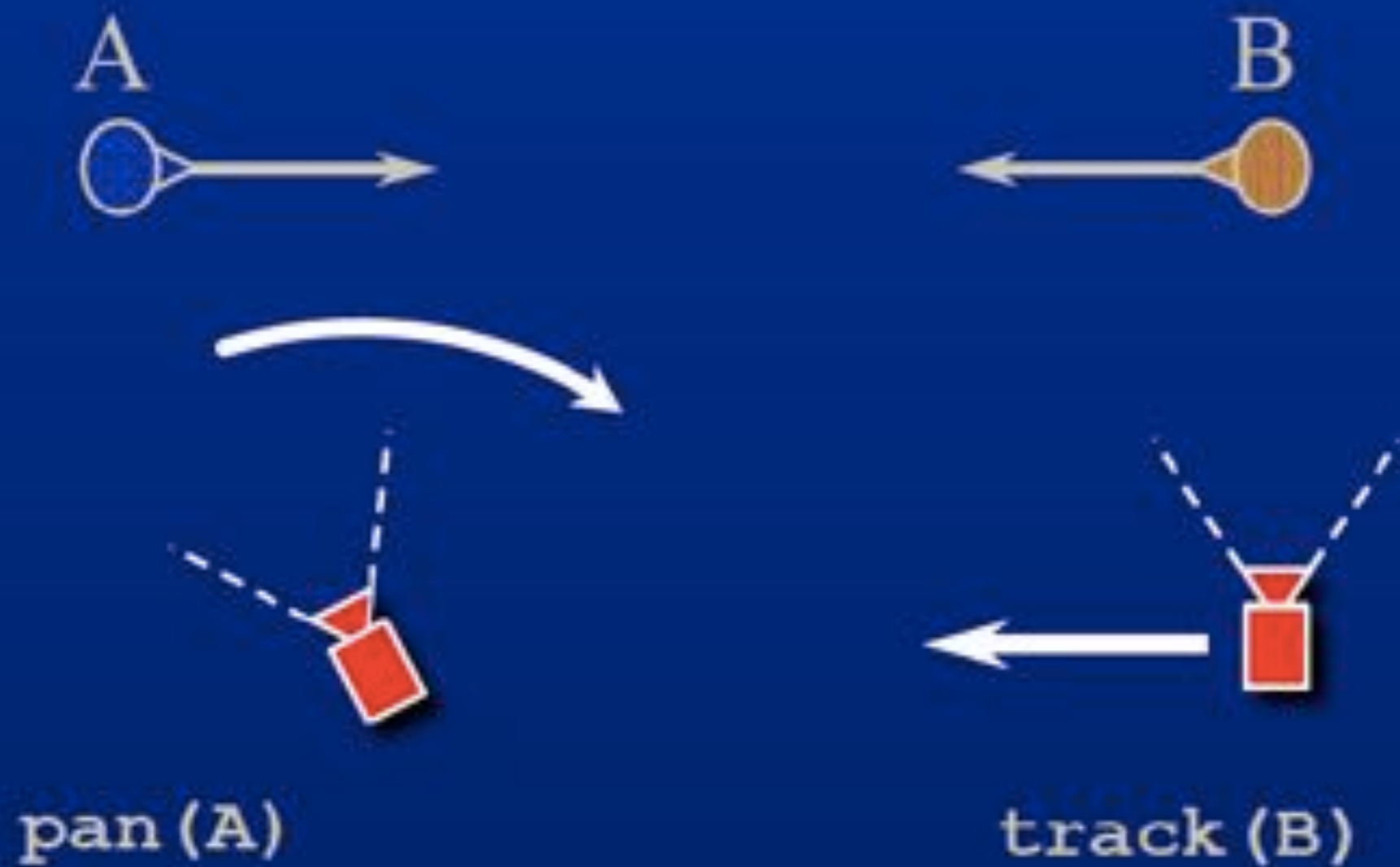
Internal camera



Apex camera



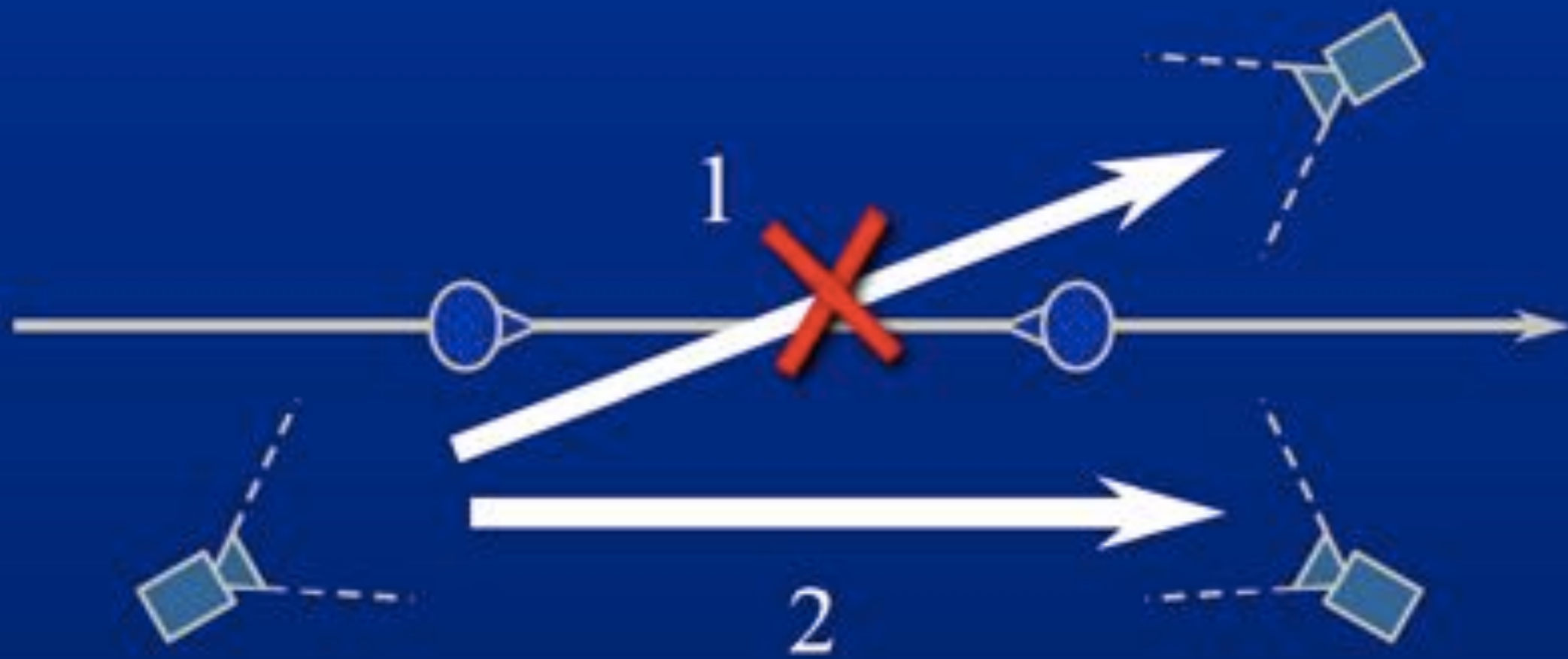
Moving cameras



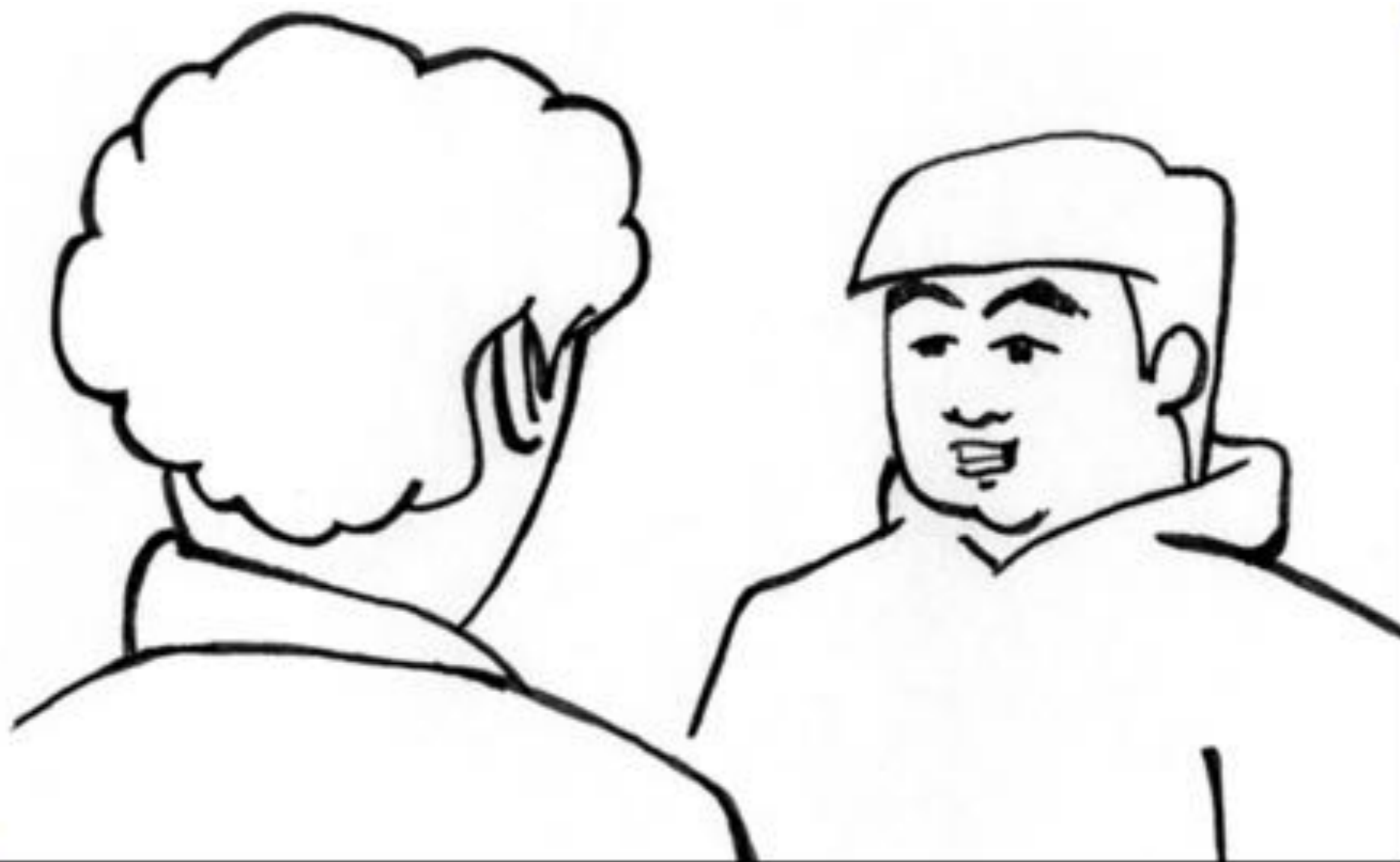
Some rules in film editing

- ☒ Don't cross the line of interest
- ☒ Avoid jump cuts
- ☒ Let the actor lead
- ☒ Break movement

Don't cross the line of interest







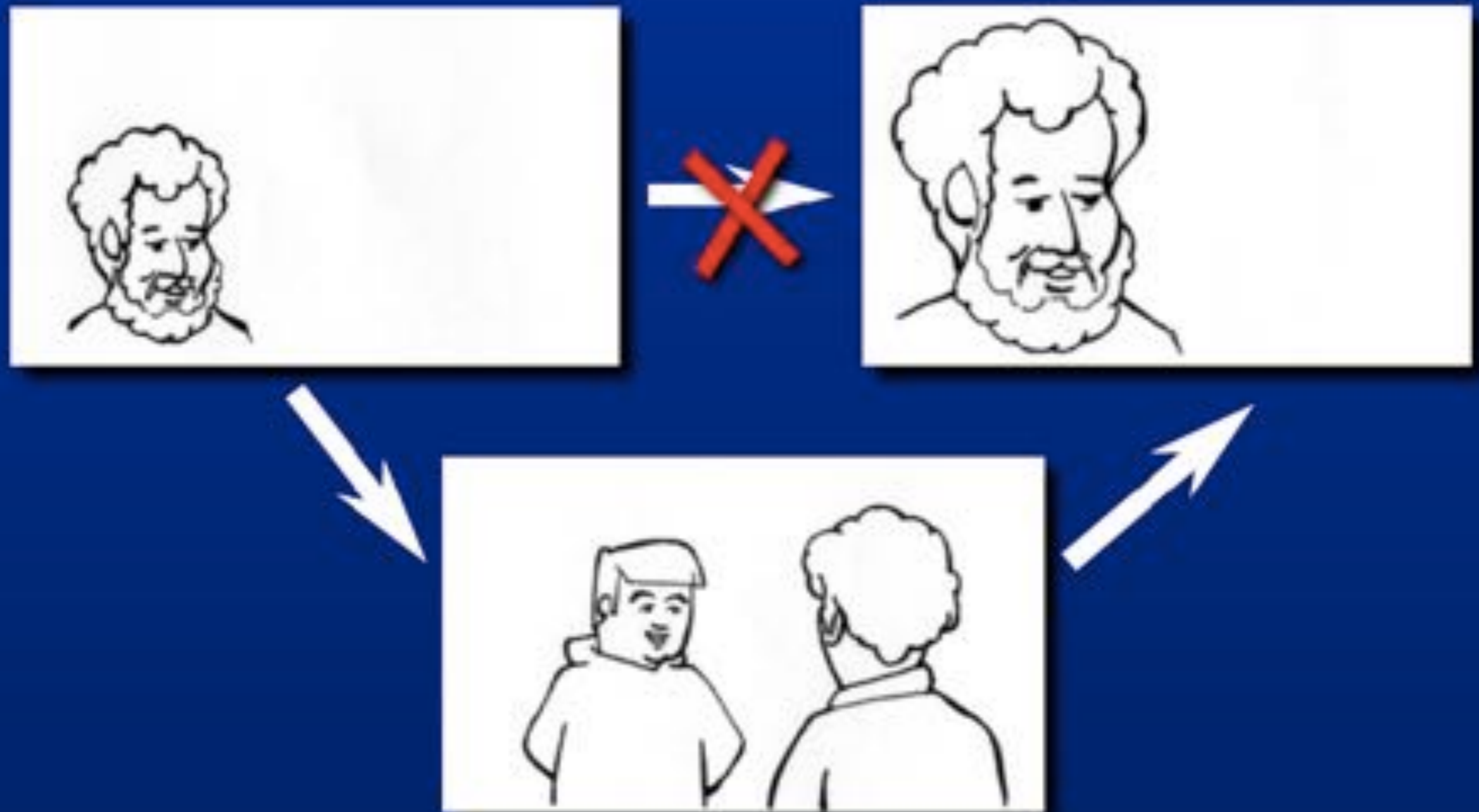




Some rules in film editing

- ☒ Don't cross the line of interest
- ☒ Avoid jump cuts
- ☒ Let the actor lead
- ☒ Break movement

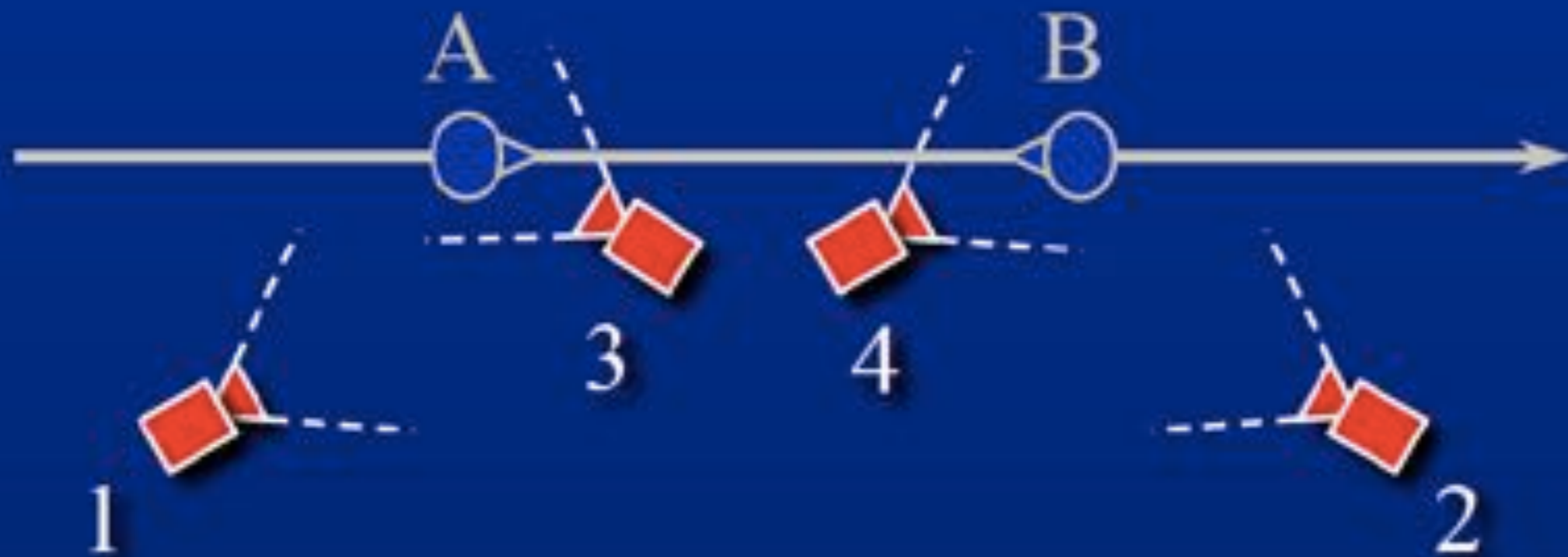
Avoid jump cut

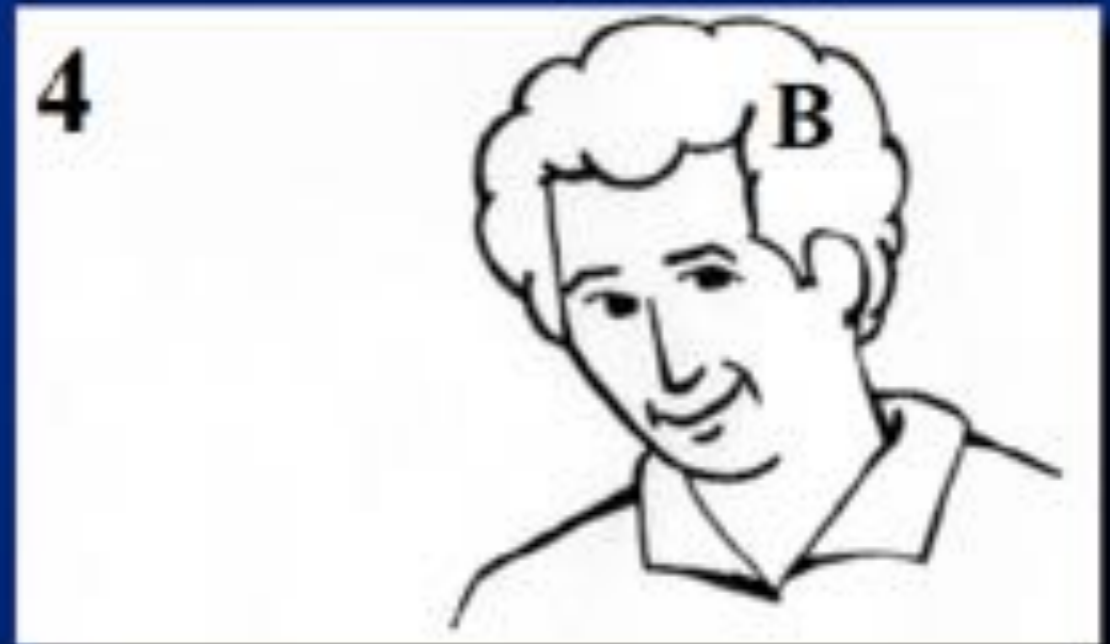
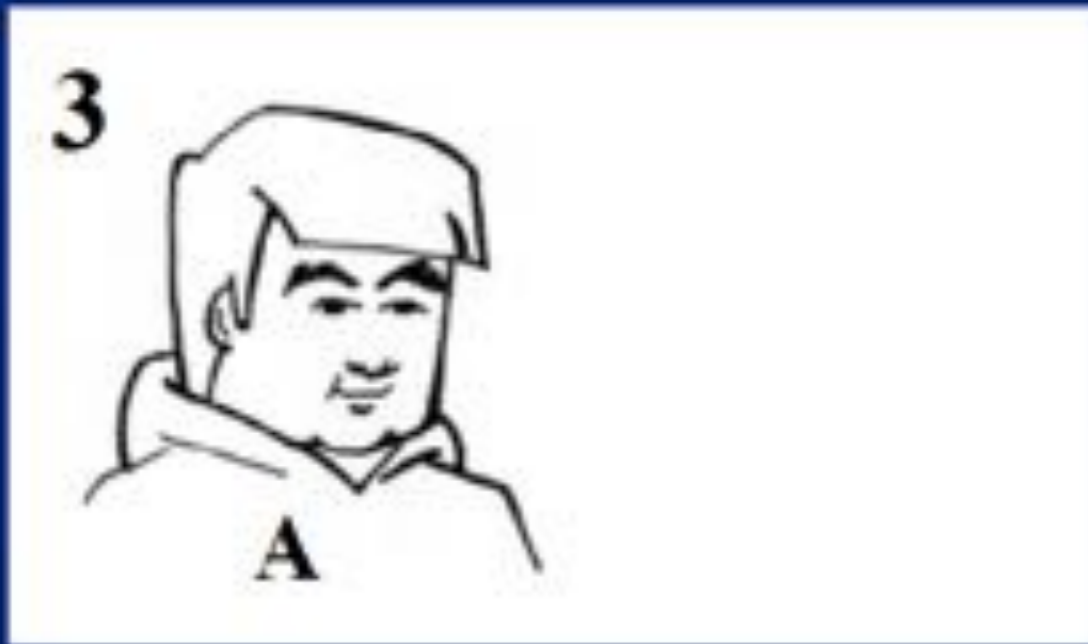
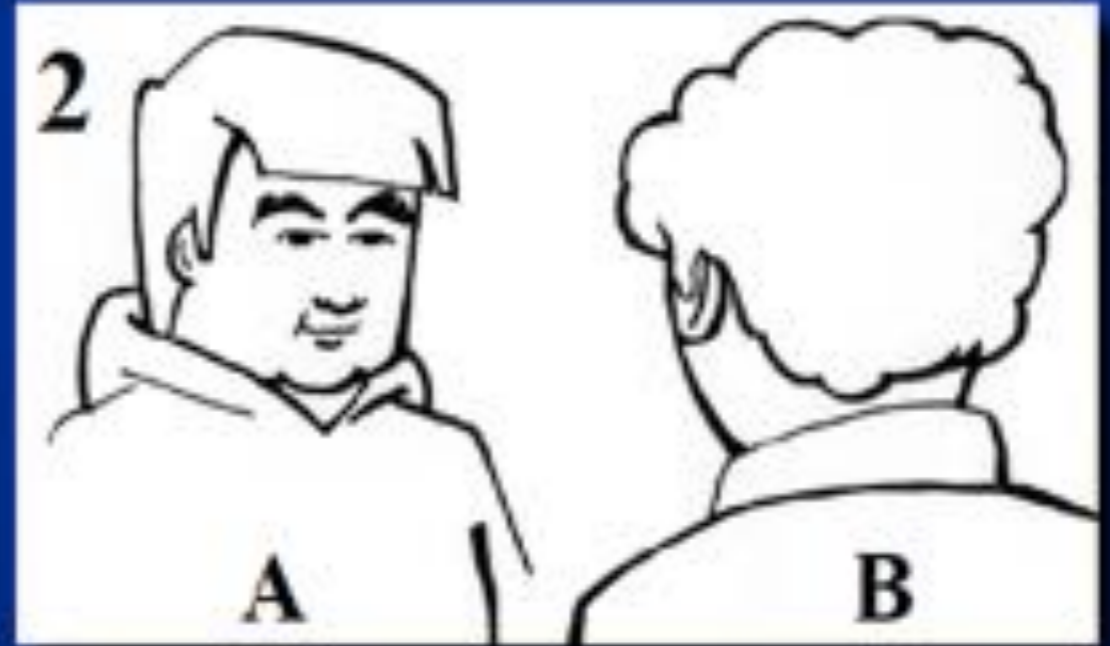
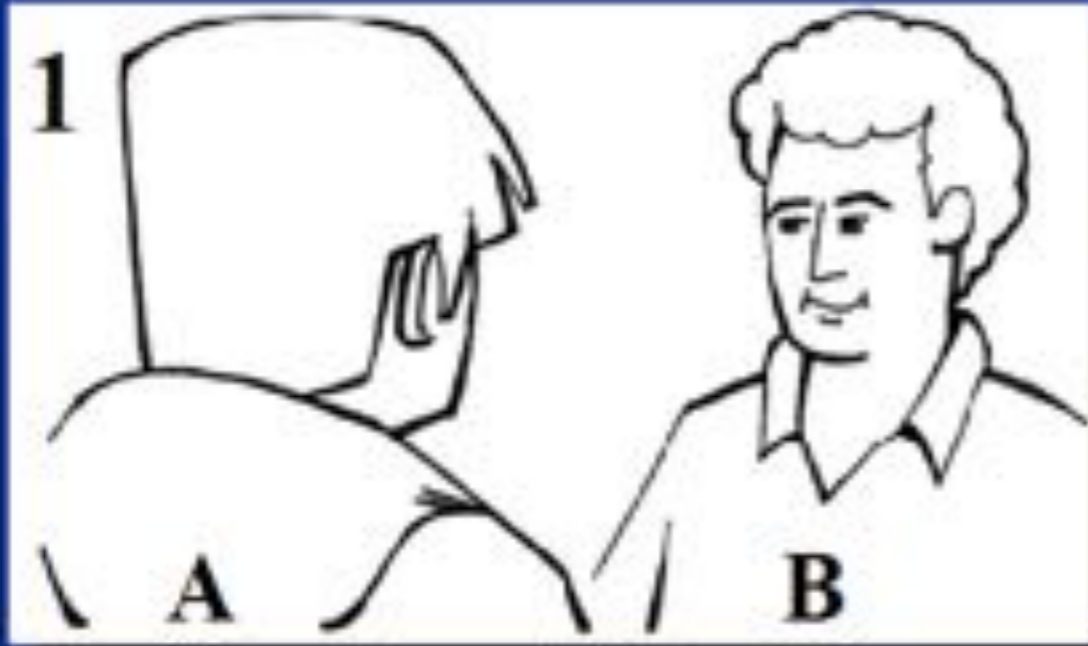


Some rules in film editing

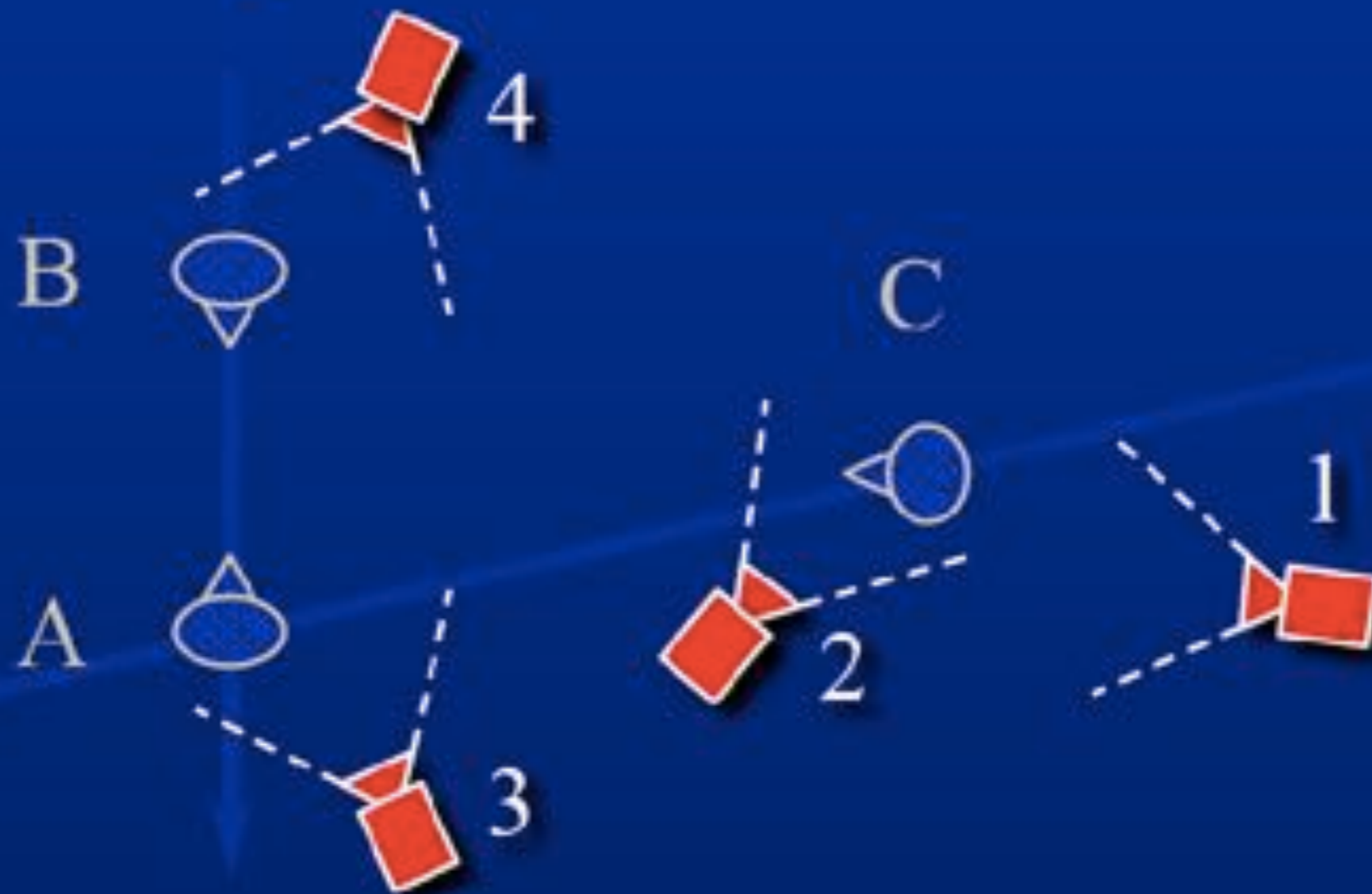
- ☒ Don't cross the line of interest
- ☒ Avoid jump cuts
- ☒ Let the actor lead
- ☒ Break movement

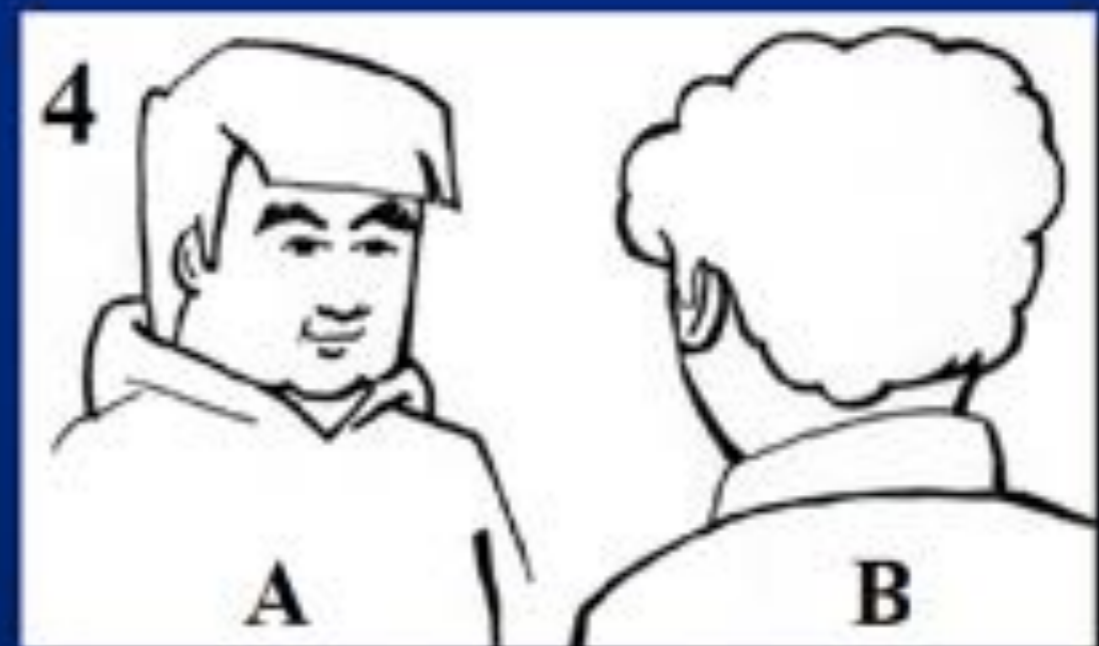
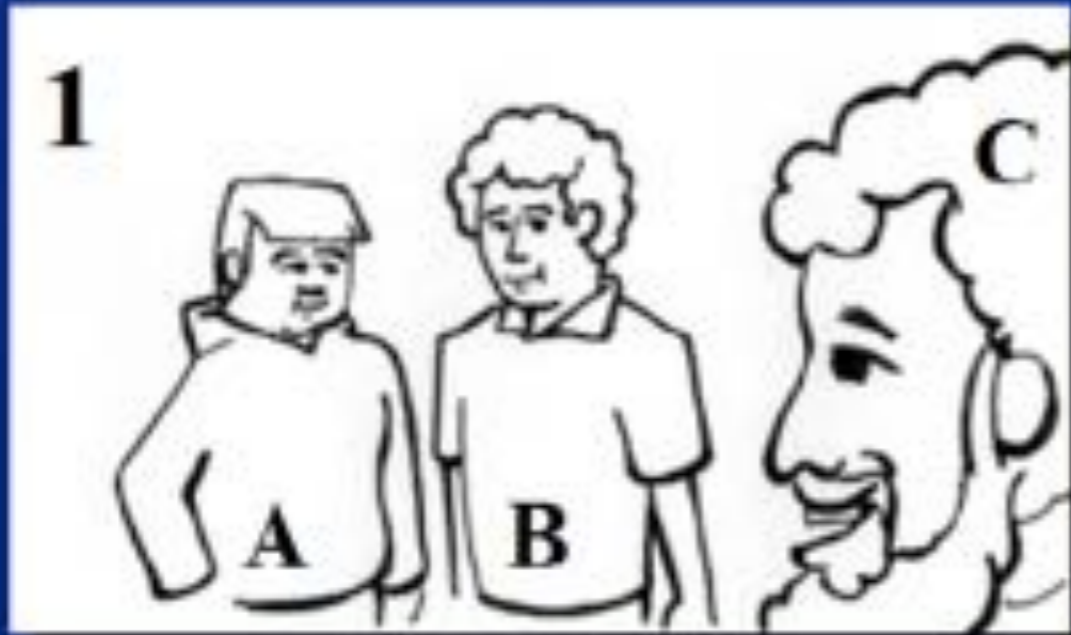
Two-person conversation





Three-person conversation





Bad

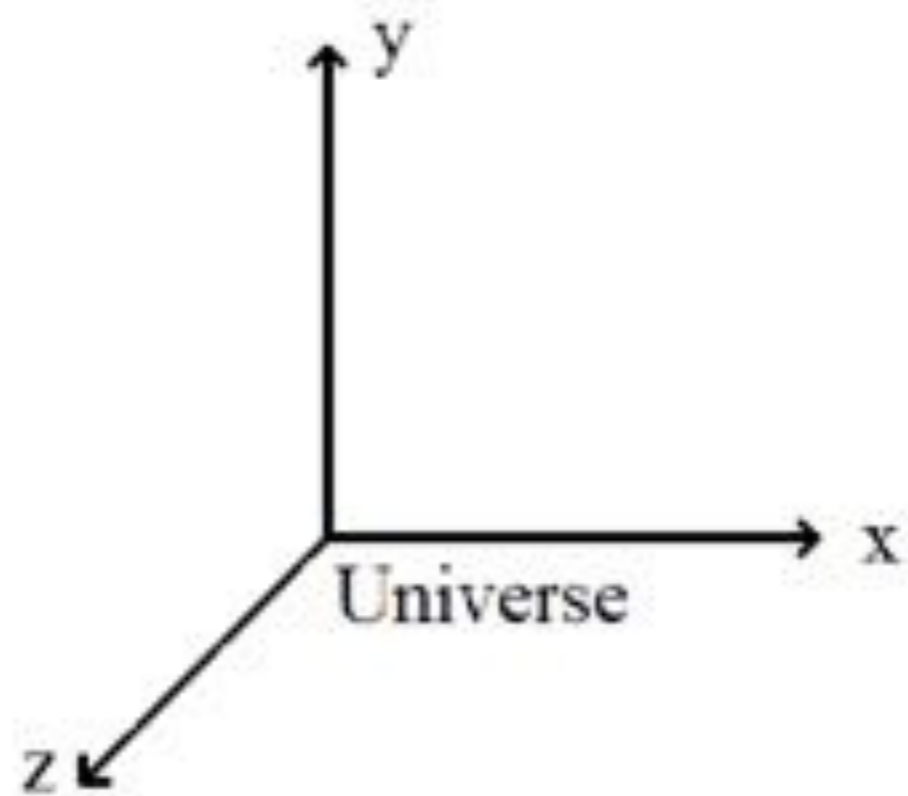


Good



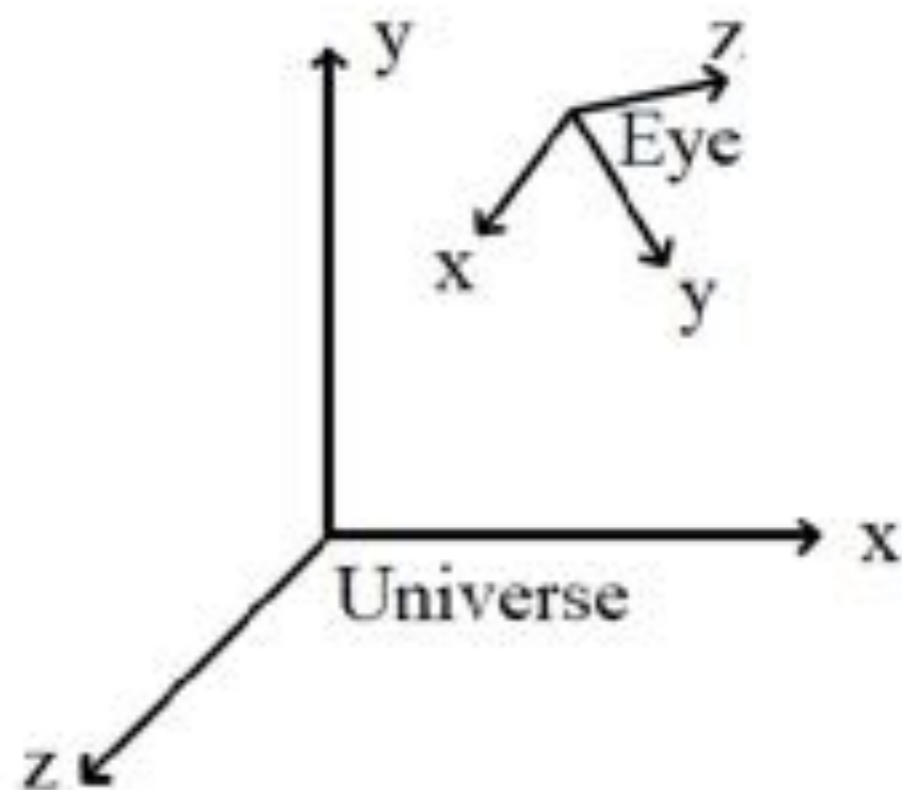
Three spaces

The universe space



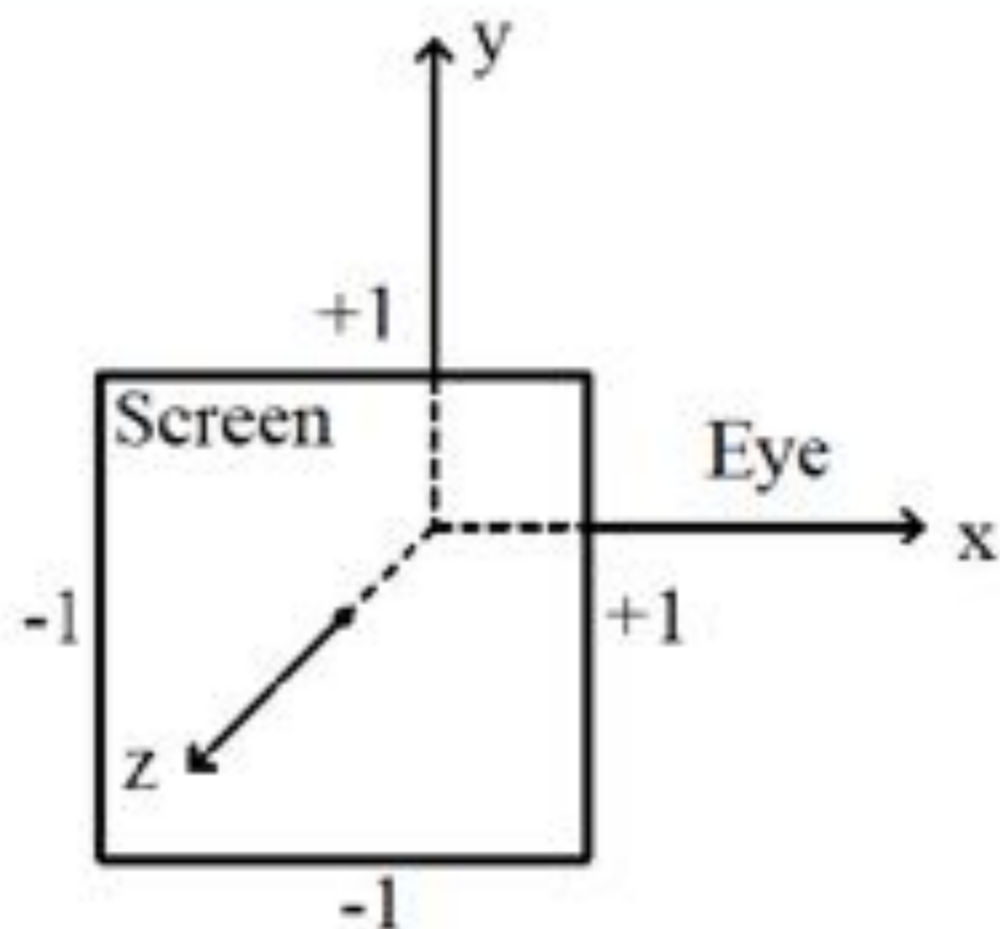
Three spaces

The eye space (z-axis is the look at direction)



Three spaces

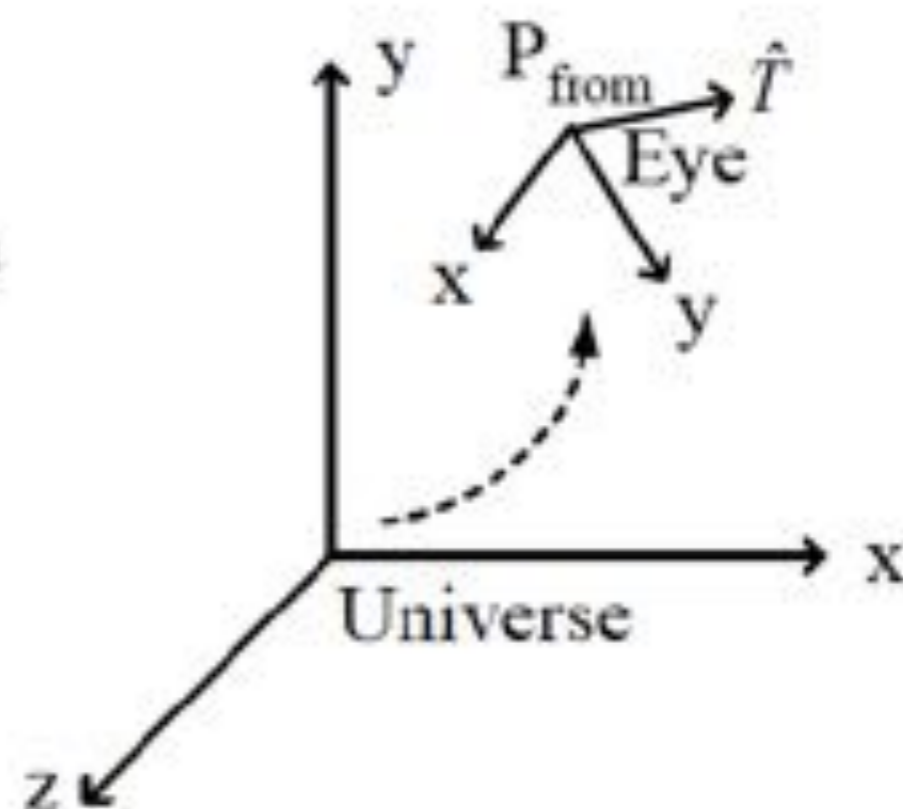
The screen space (screen is $c \tan(\text{fov}/2)$ away from eye point)



Standard look-at transformation

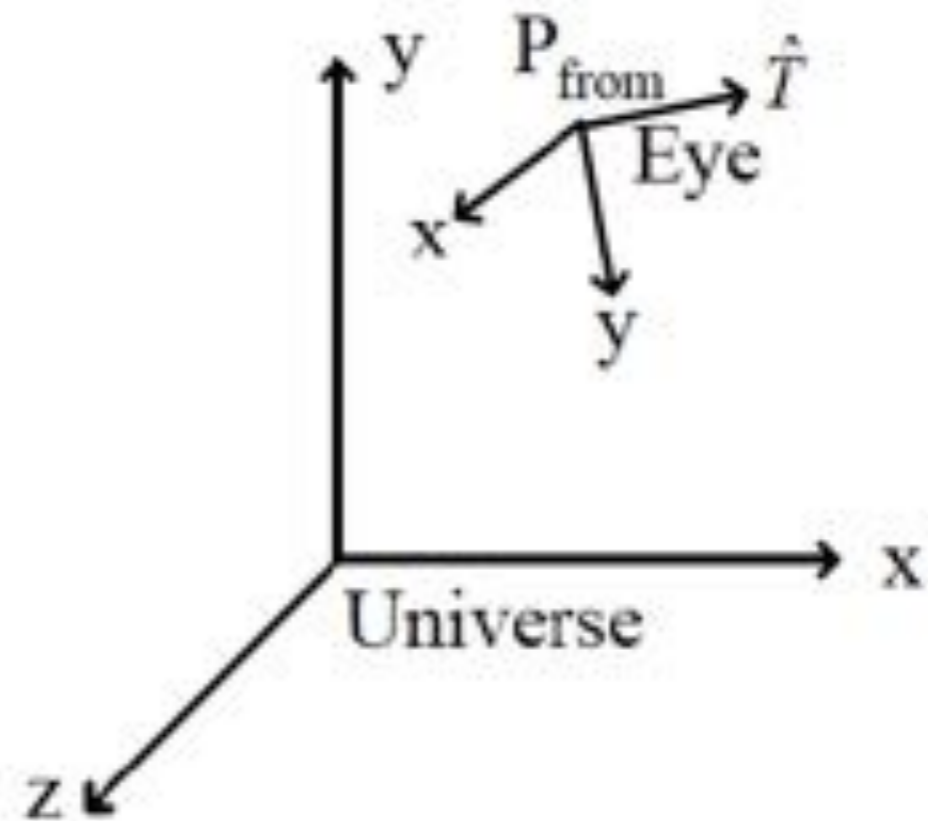
Given eye position P_{from} , a
look at direction \hat{T}

Want a rotation matrix R and a
translation to transform a
vector from universe space to
eye space



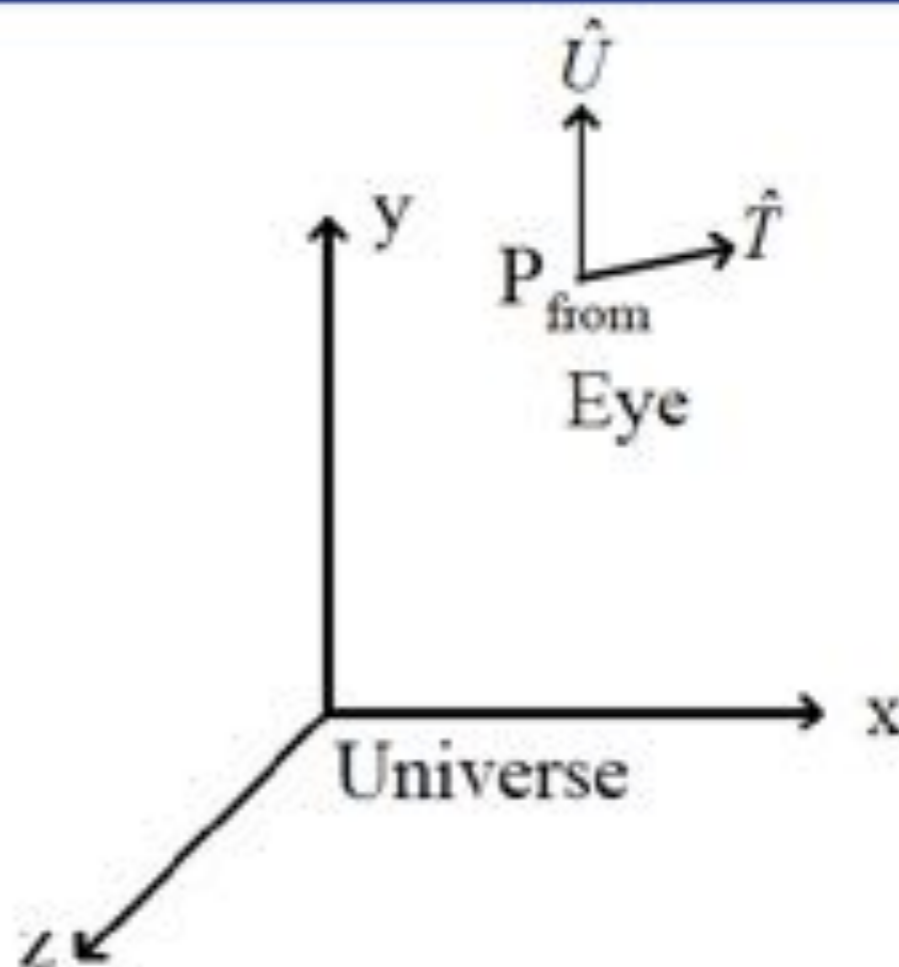
Standard look-at transformation

But this is not unique



Standard look-at transformation

We will need to specify an up vector \hat{U} , usually $[0,1,0]$



Standard look-at transformation

In eye space

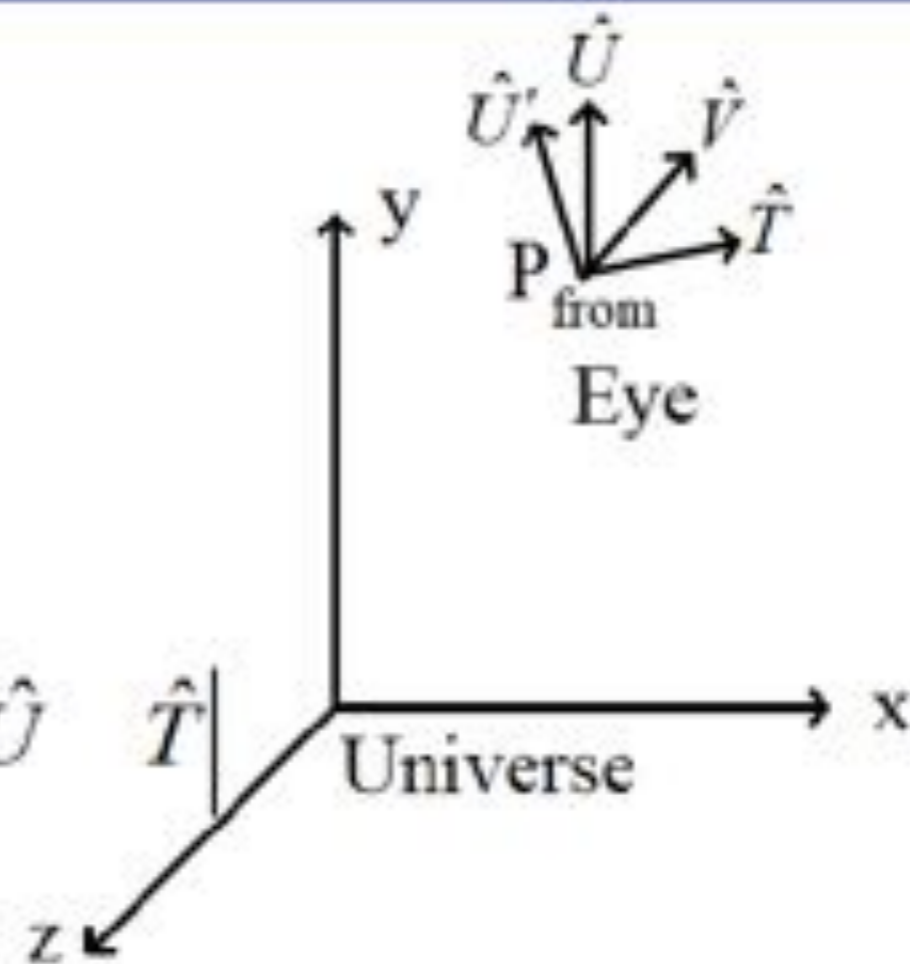
Z-axis is \hat{T}

X-axis is $\hat{V} = \hat{U} \times \hat{Z}$

Y-axis is $\hat{U}' = \hat{T} \times \hat{V}$

Rotation matrix $R_{3 \times 3} = \begin{vmatrix} \hat{V} & \hat{U} & \hat{T} \end{vmatrix}$

Translation is $-\mathbf{P}_{\text{from}}$



Internal close-up of actor A



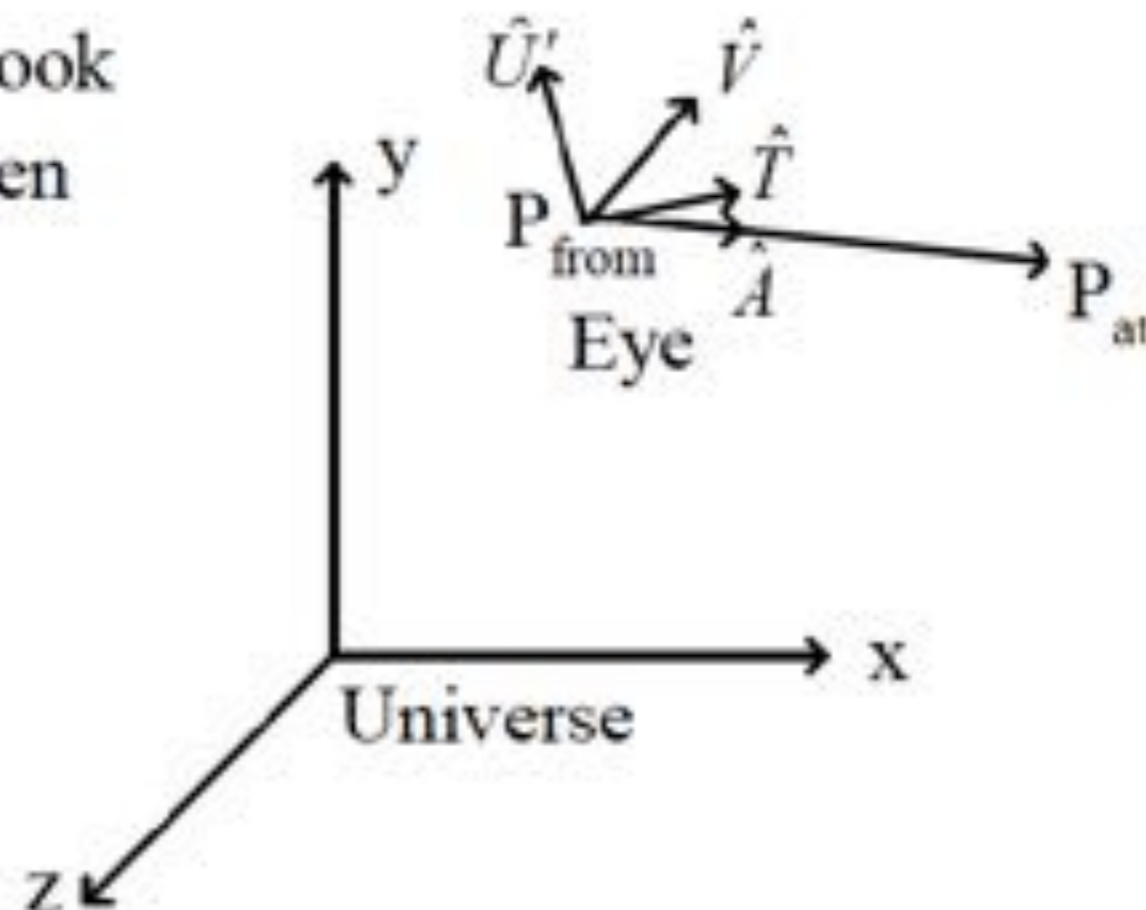
Look at $[x_{at}, y_{at}]$ transformation

Given eye position P_{from} , look at point P_{at} , and P_{at} in screen space $[x_{at}, y_{at}]$

Want \hat{T}

In universe space

$$\hat{A} = \text{unit}(P_{at} - P_{from})$$



Look at $[x_{at}, y_{at}]$ transformation

In screen space:

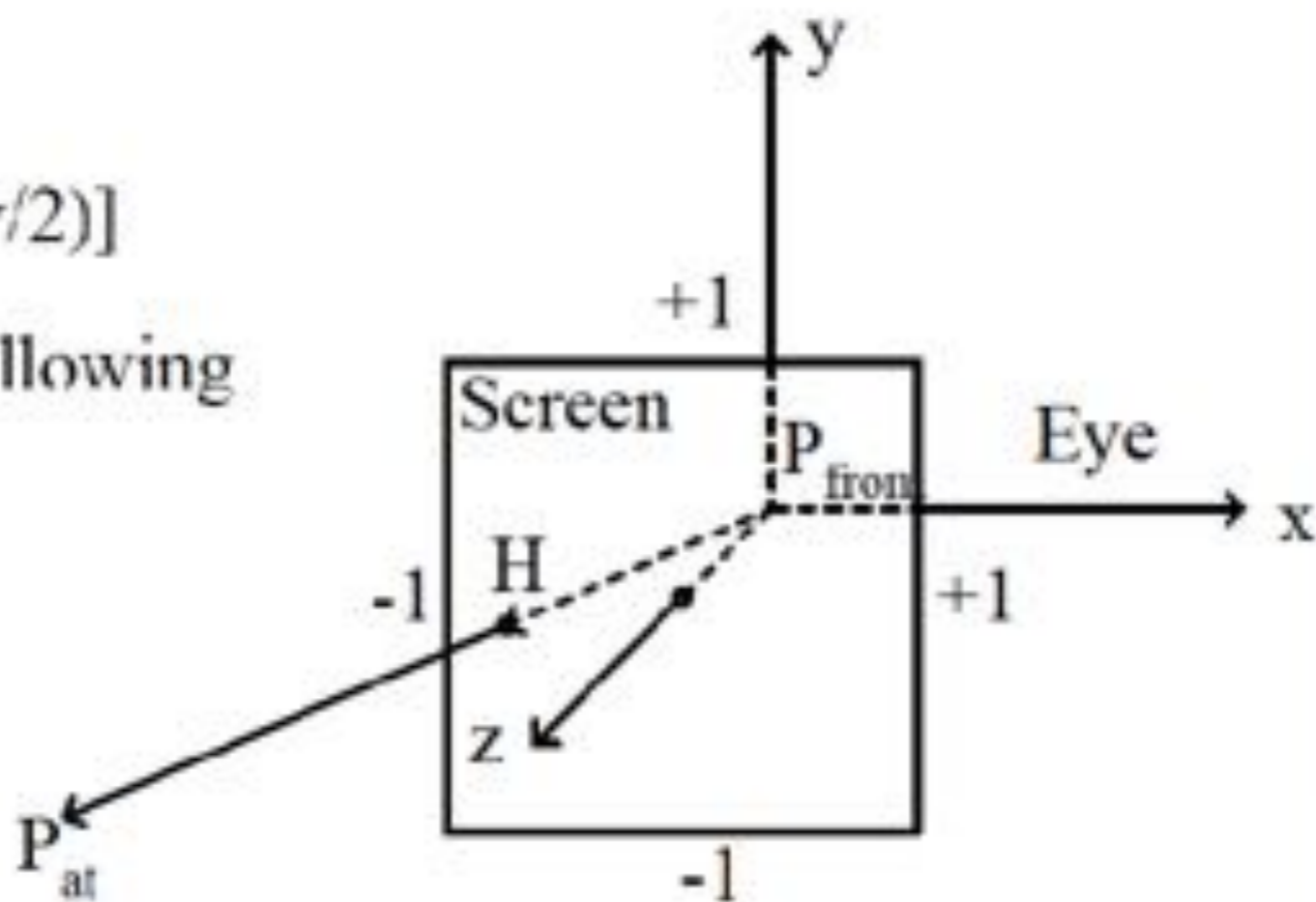
$$H = [x_{at}, y_{at}, c \tan(\text{fov}/2)]$$

Solve \hat{T} from the following equations:

$$\hat{H} \cdot [0, 0, 1] = \hat{A} \cdot \hat{T}$$

$$\hat{H} \cdot [0, 1, 0] = \hat{A} \cdot \hat{U}'$$

$$\hat{T} \cdot \hat{U}' = 1$$



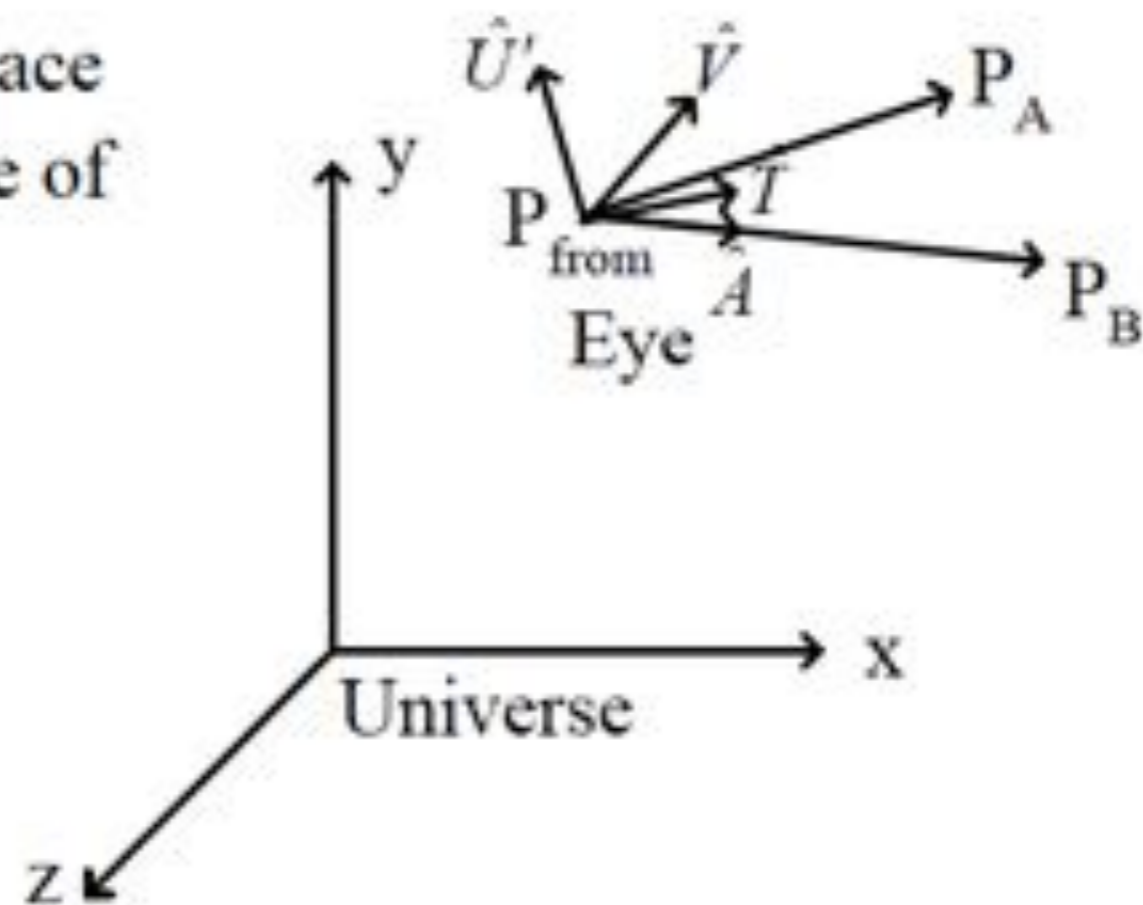
External of actor A and actor B



Look at $[x_A, y_A]$ and $[x_B, y_B]$

Given P_A, P_B in universe space
and eye space, and distance of
eye position P_{from} and P_A .

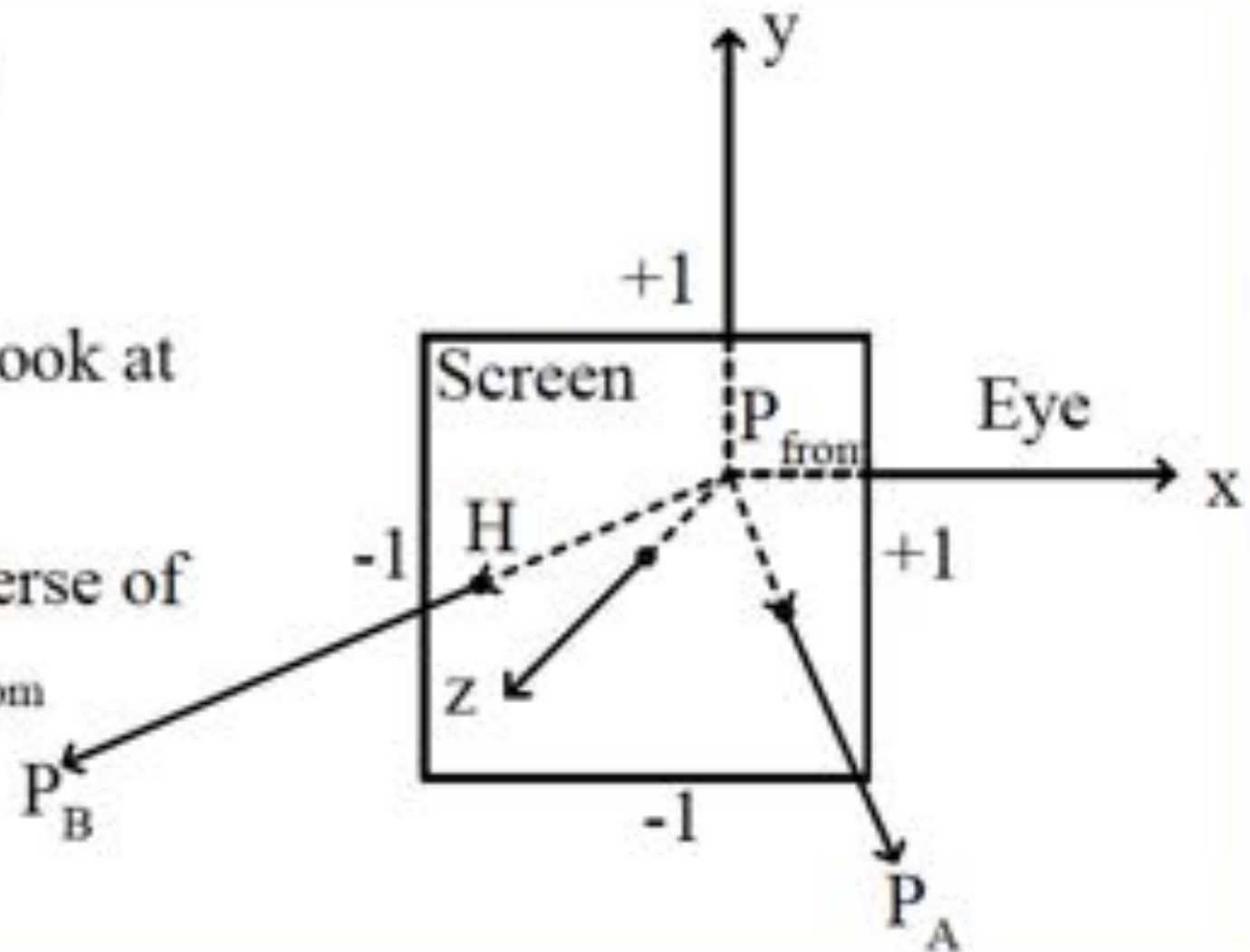
Want P_{from} and \hat{T}



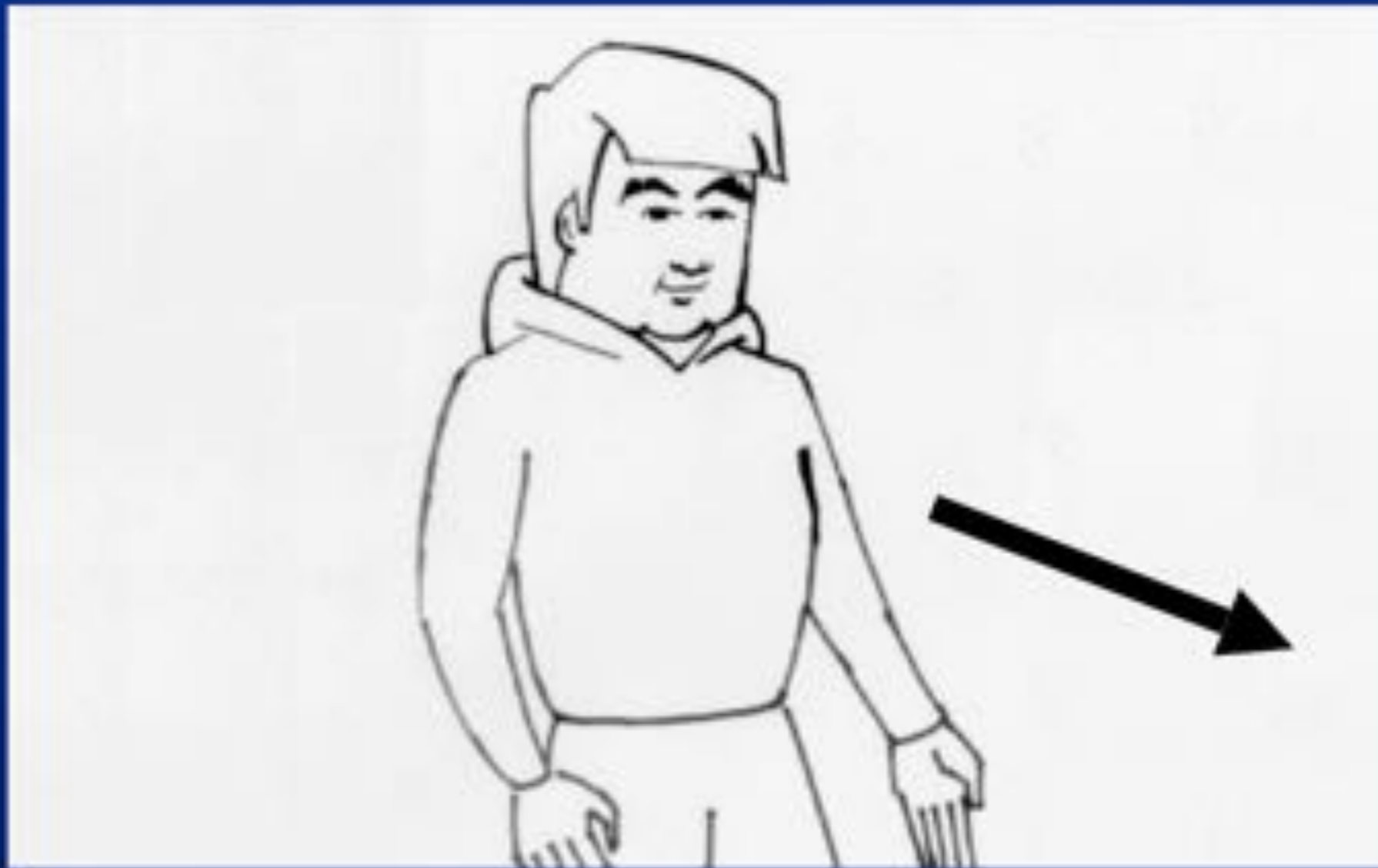
Look at $[x_A, y_A]$ and $[x_B, y_B]$

Use numerical method:

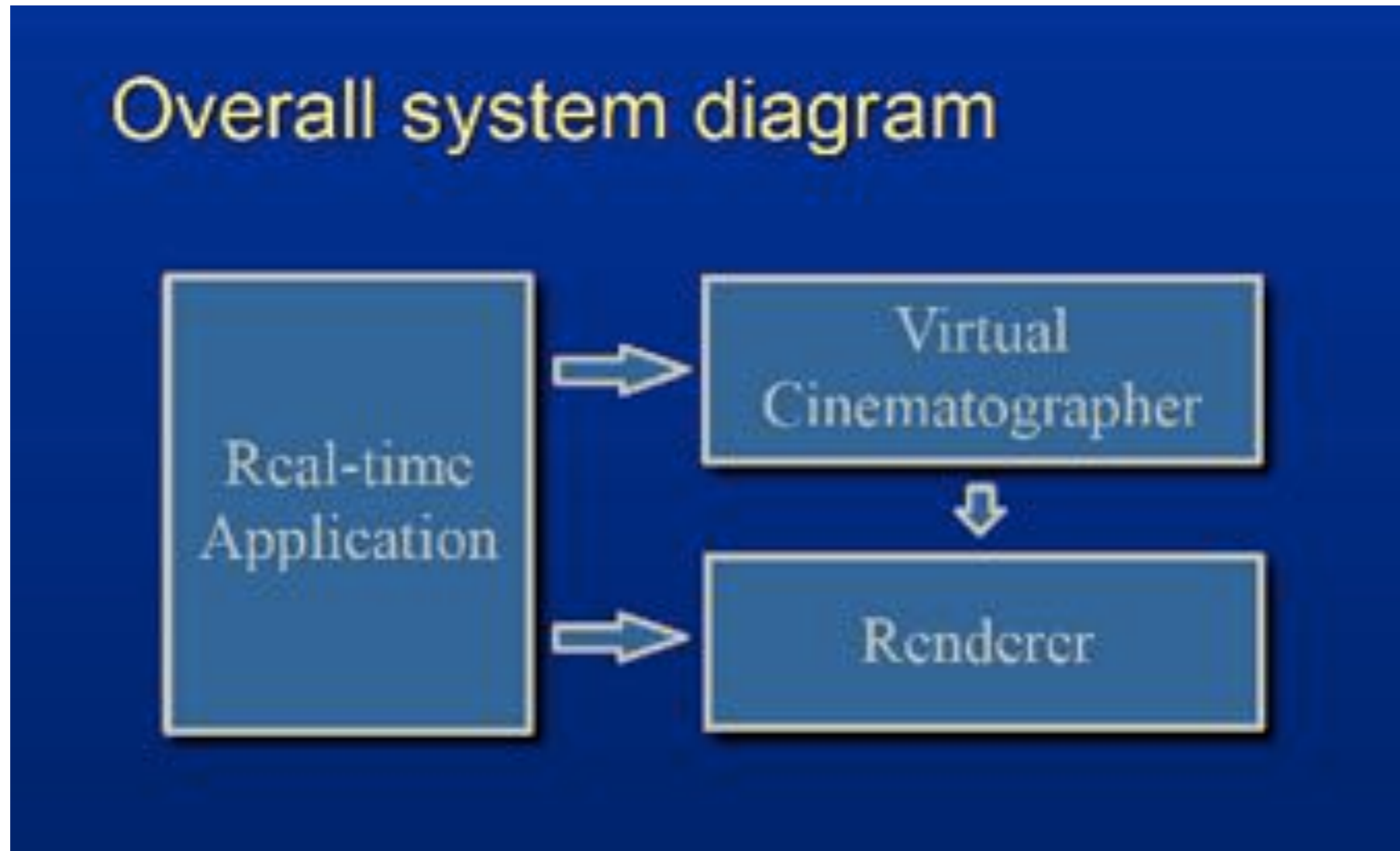
1. Set P_{from} to 0
2. Solve R using the Look at $[x_{\text{at}}, y_{\text{at}}]$ method
3. Transform E by inverse of R, getting a new P_{from} approximation
4. Goto 2



Medium panning shot of actor A



Paper 7 - The virtual cinematographer



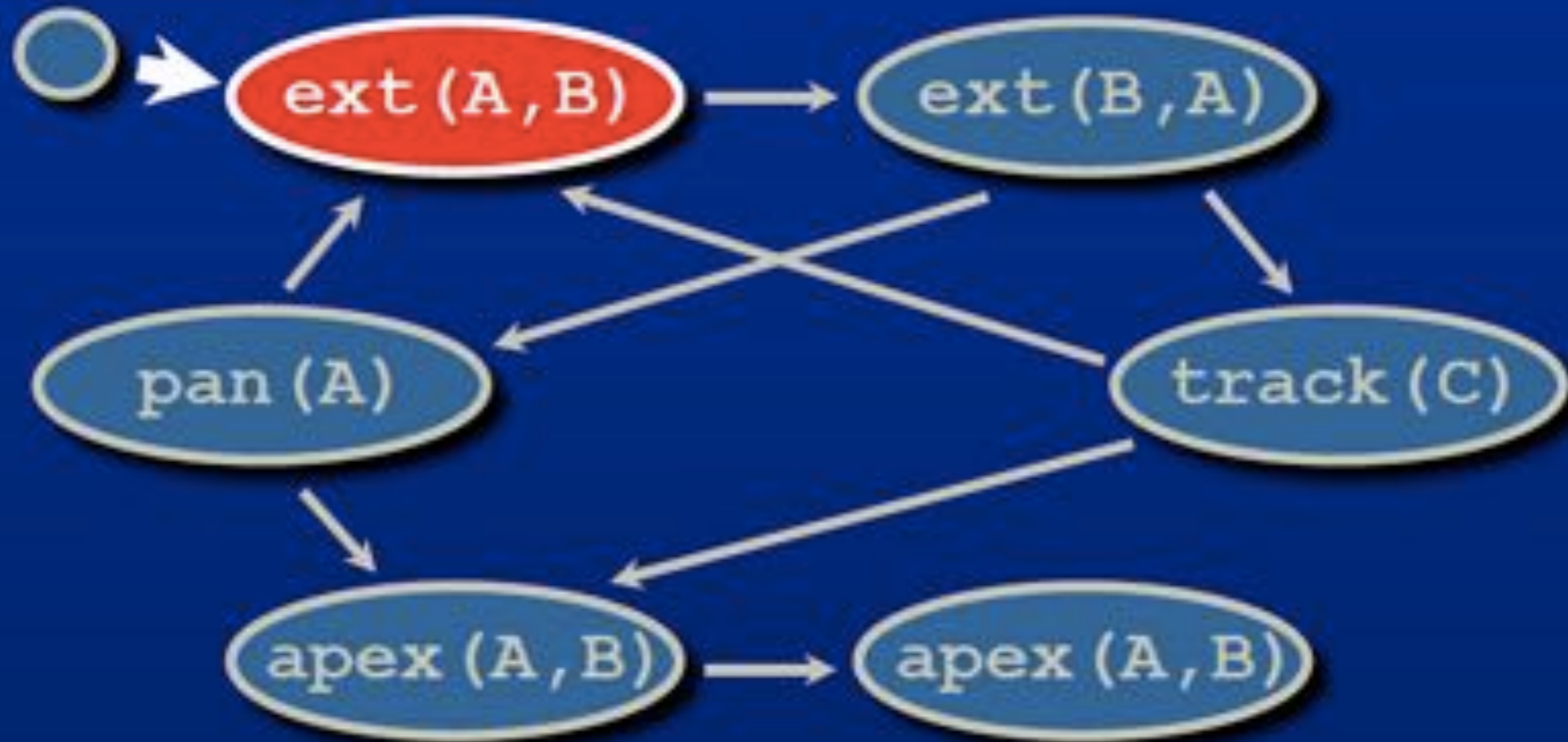
- Li-wei He, Michael F. Cohen, and David H. Salesin. The virtual cinematographer: a paradigm for automatic real-time camera control and directing. Siggraph 1996.

Hierarchical idiom structure

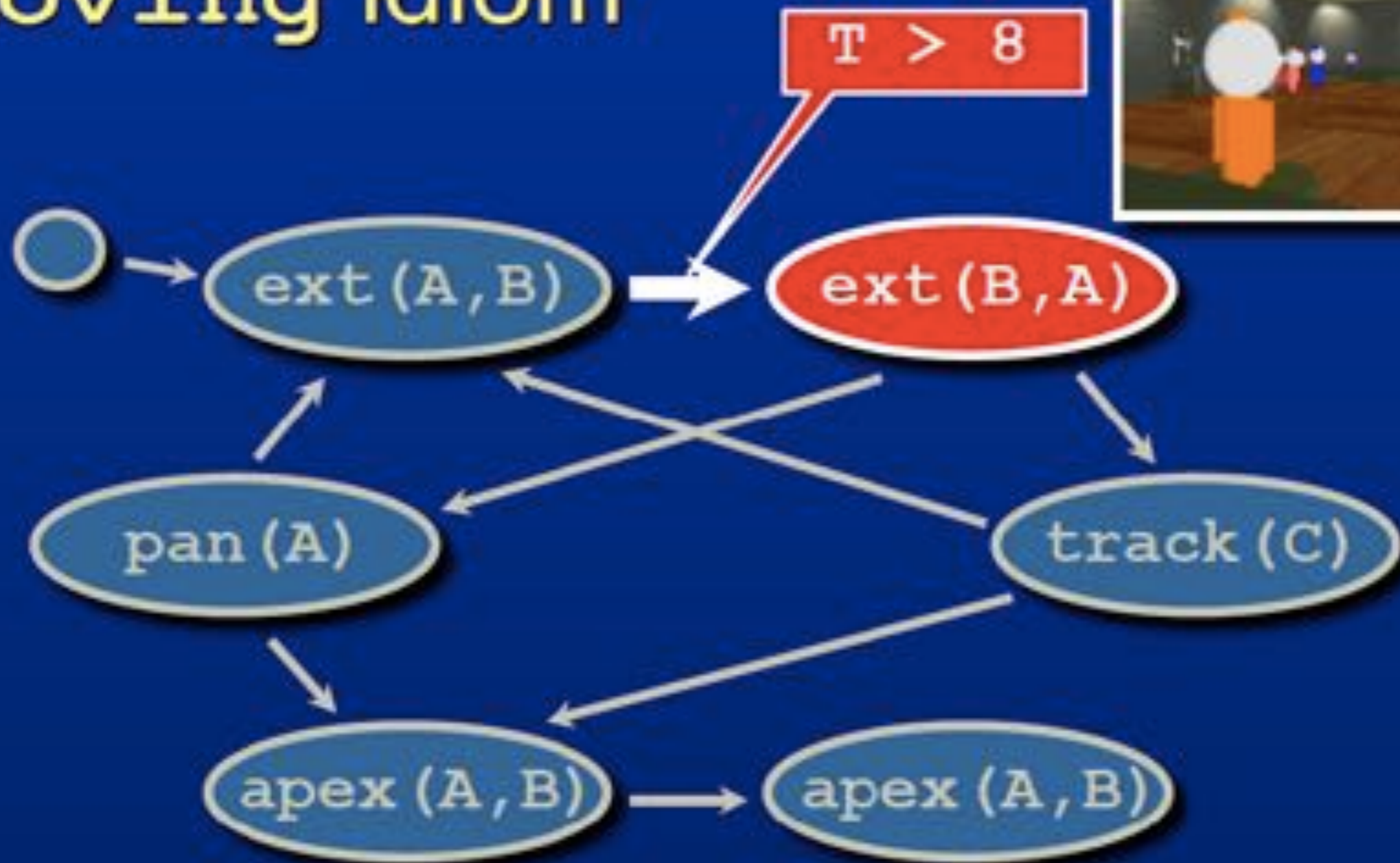




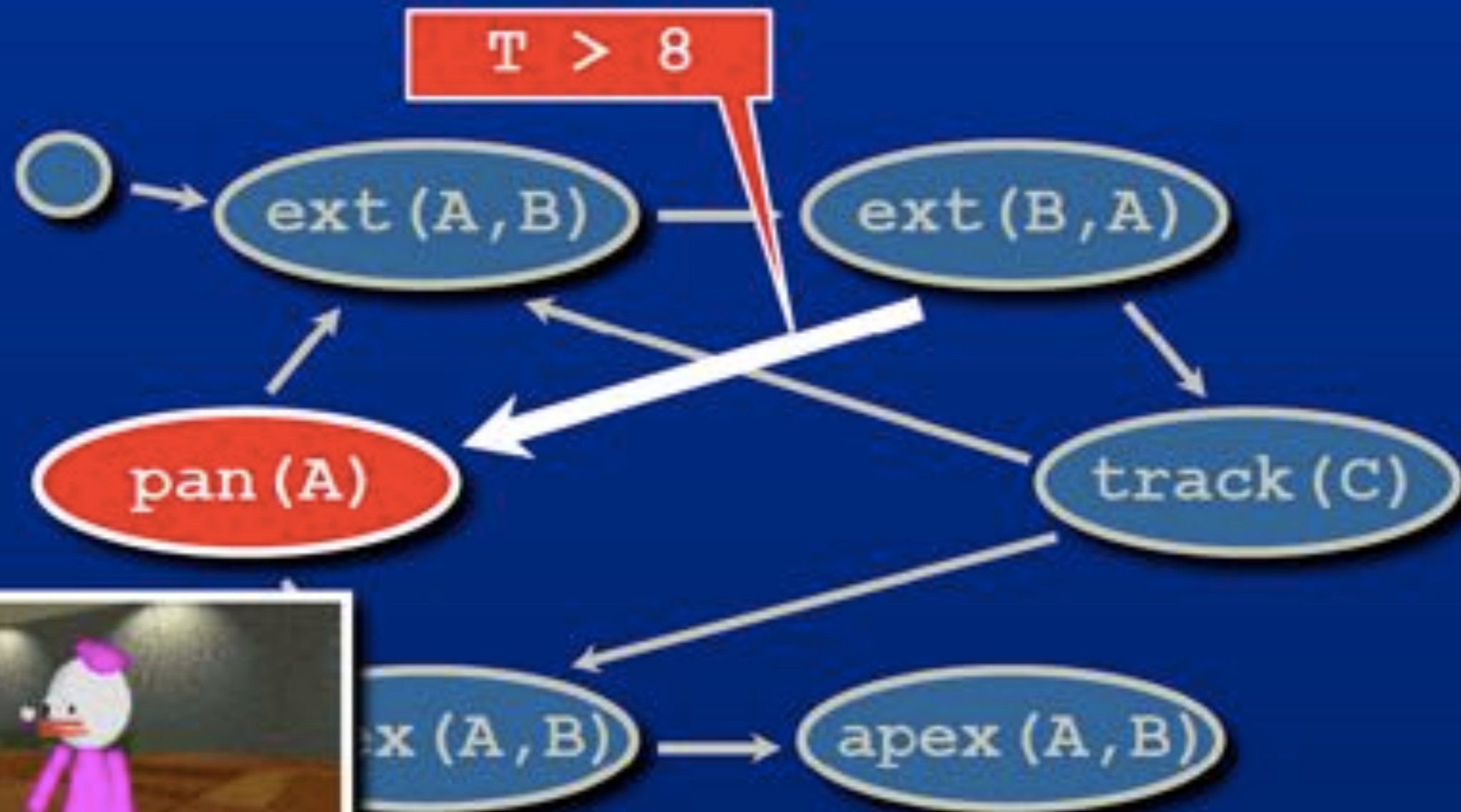
Moving idiom



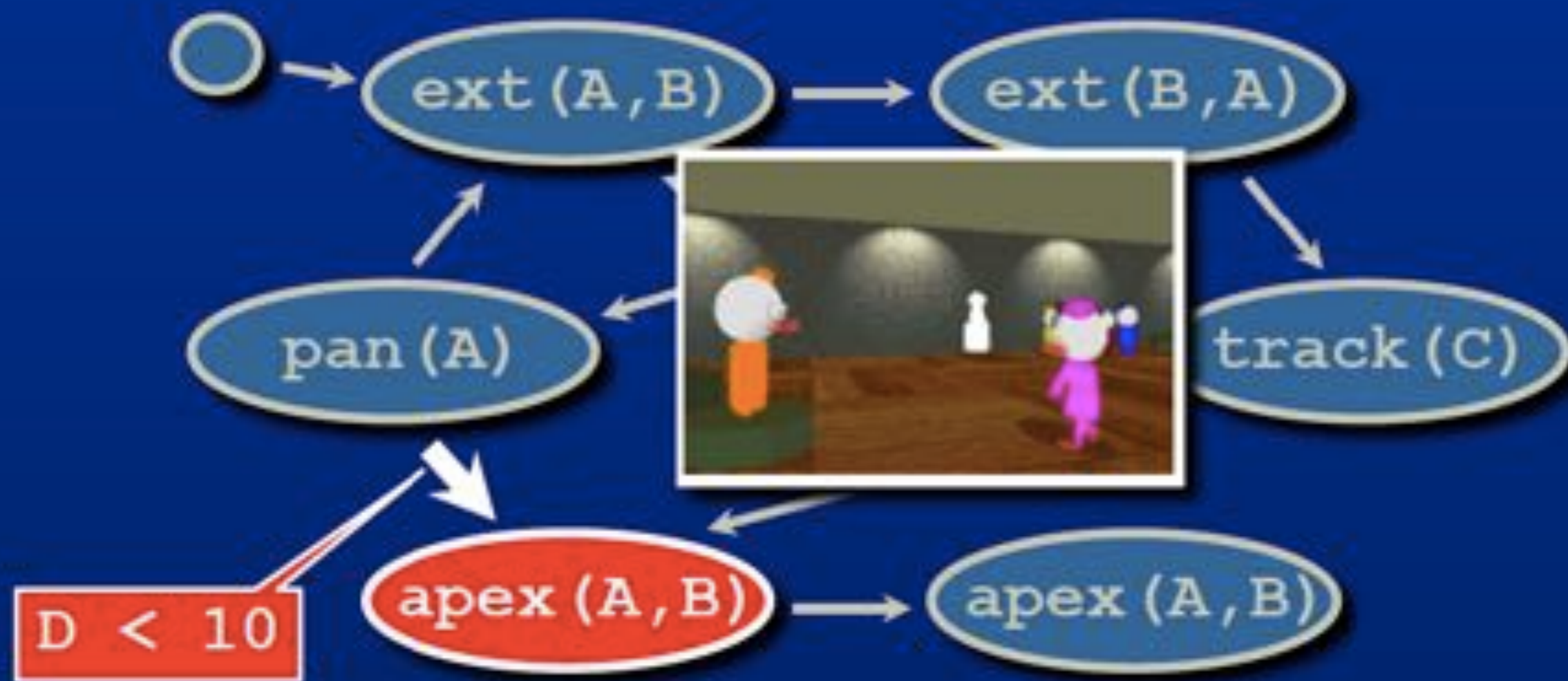
Moving idiom



Moving idiom



Moving idiom



Moving idiom

