# Virtual Cinematography

## Theory and Practice for Automatic Real-Time Camera Control and Directing

Liwei He

Microsoft Research

http://research.microsoft.com/users/lhe

# 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

# 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

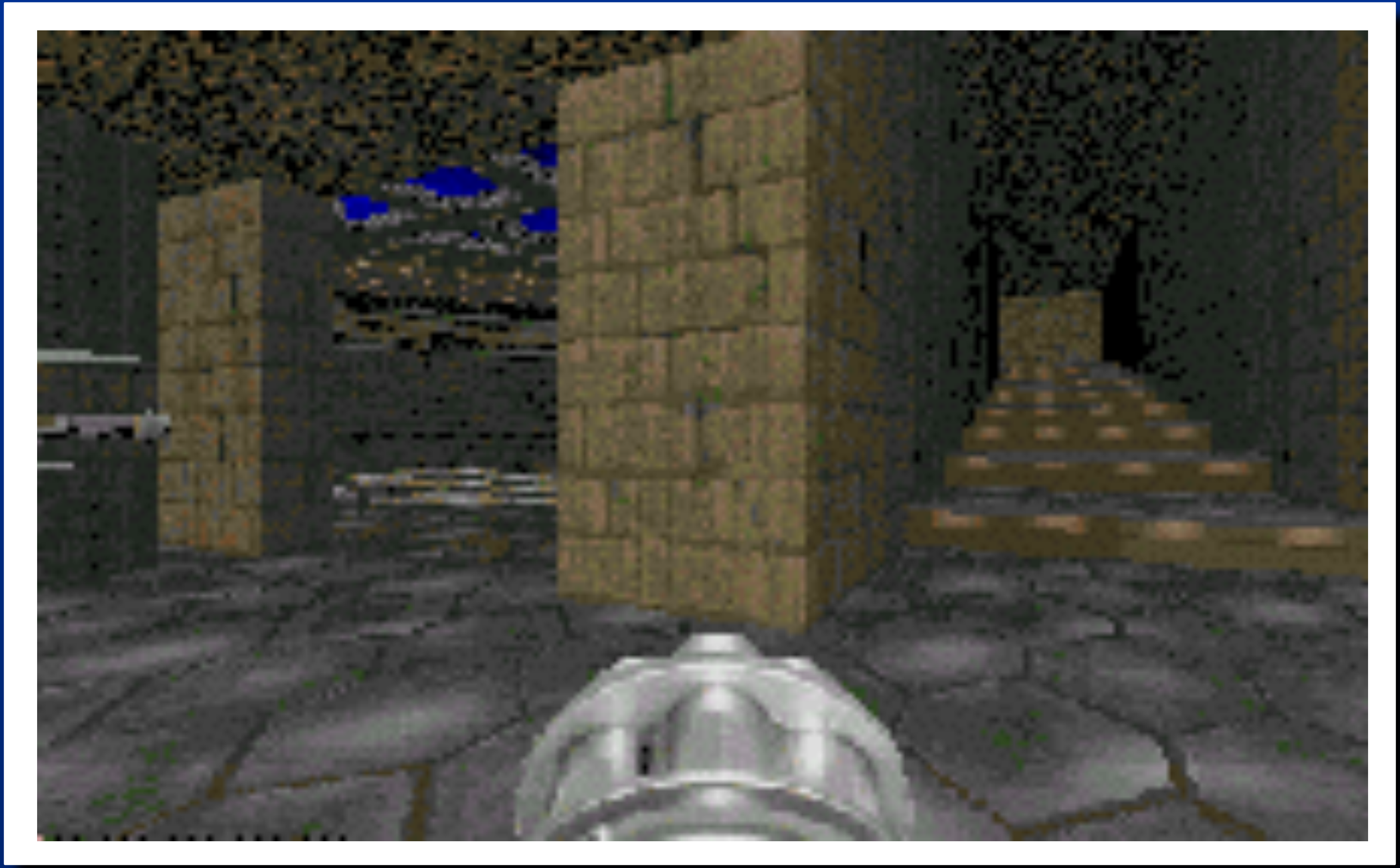
- Input: show both A and B, follow C, etc.
- Output: camera position and direction
- Application: 3D game, computer animation package

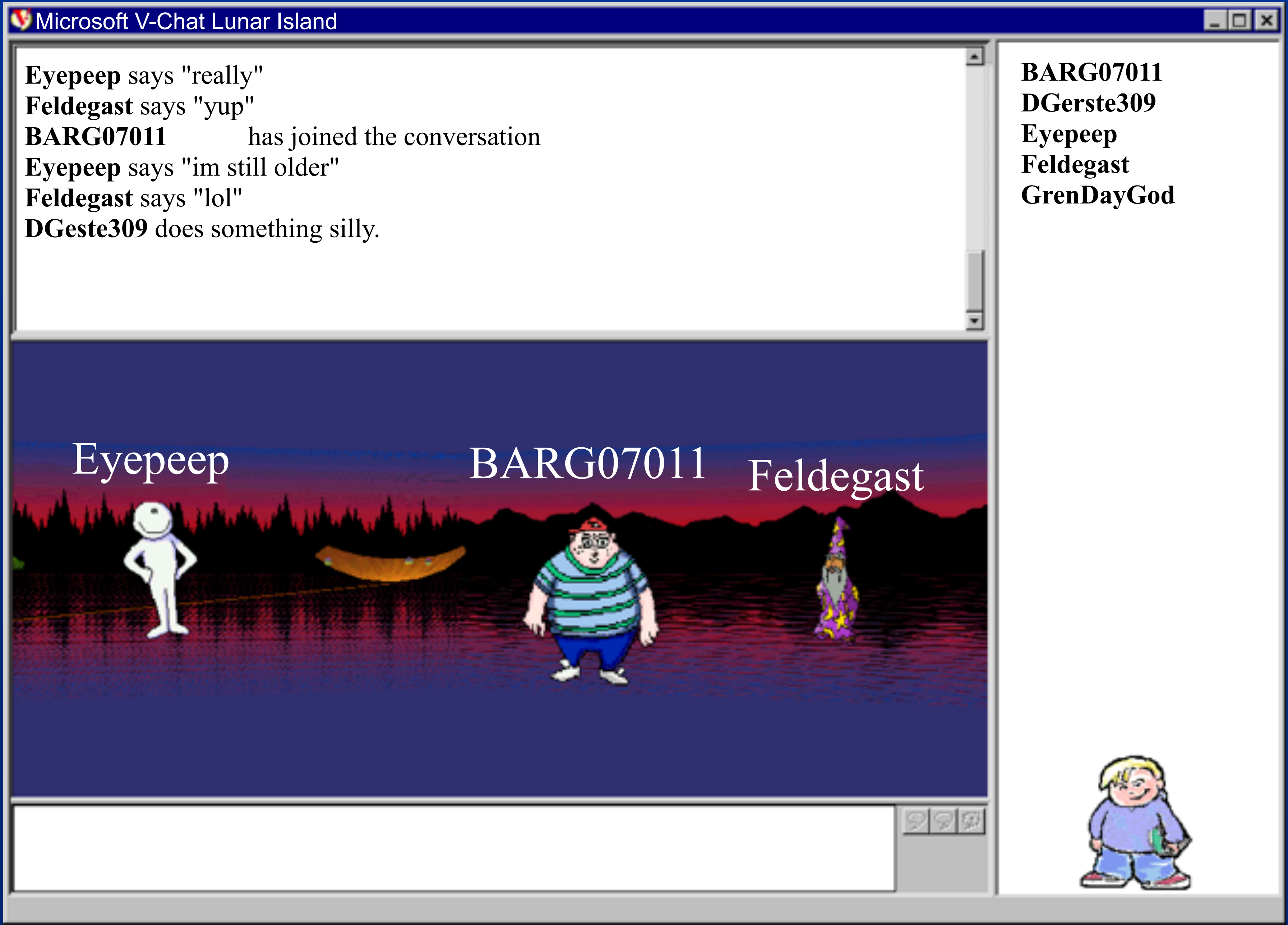
# Camera control level 3

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

# Camera control level 3 (cont.)

- Encodes cinematographic expertise
  - an interesting problem in itself
- Provides an interface that is
  - real-time
  - visually entertaining
  - more informative



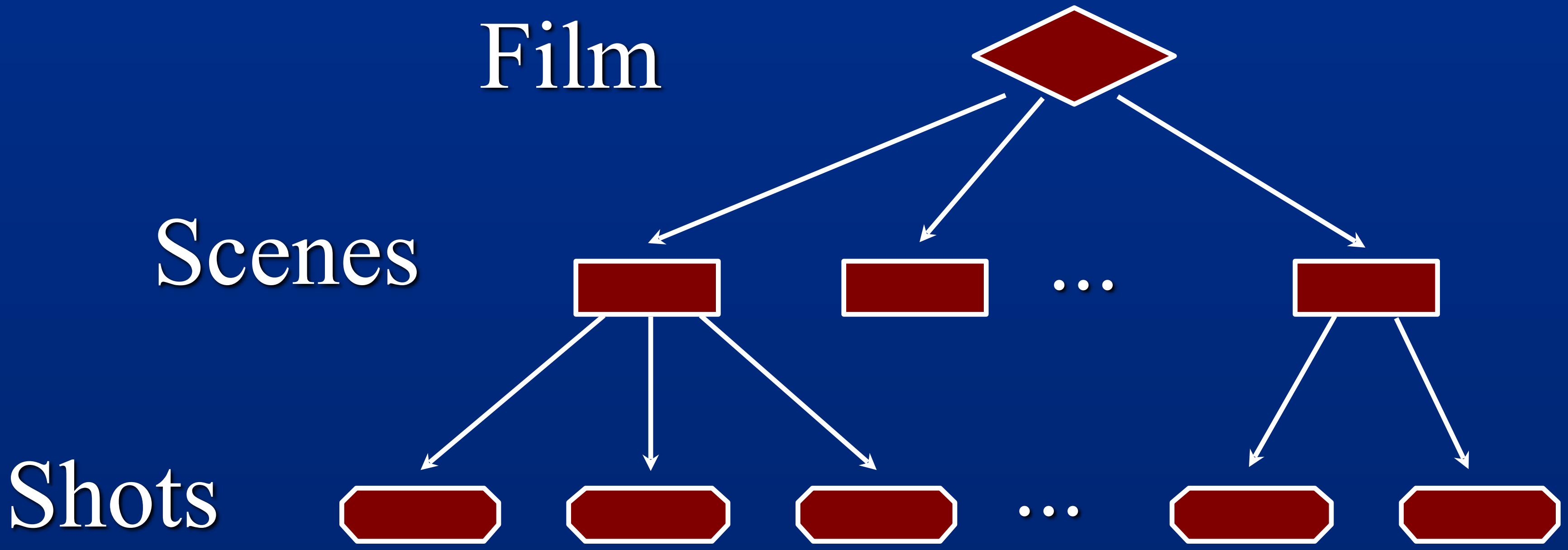


**Eyepeep** says "really"
**Feldegast** says "yup"
**BARG07011** has joined the conversation
**Eyepeep** says "im still older"
**Feldegast** says "lol"
**DGeste309** does something silly.

**BARG07011**
**DGerste309**
**Eyepeep**
**Feldegast**
**GrenDayGod**

# Principles of Cinematography

# Film structure

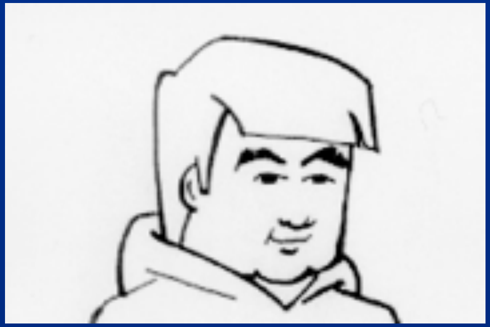


Film

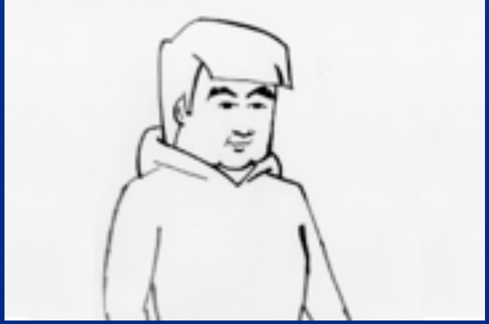
Scenes …

Shots …

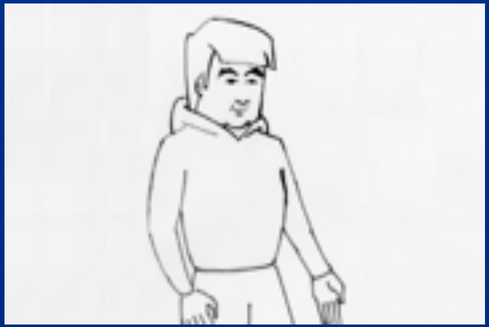
# Camera distance



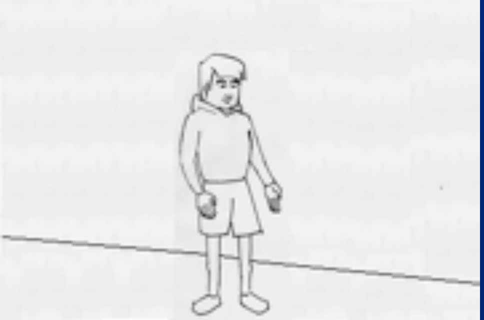
Close up



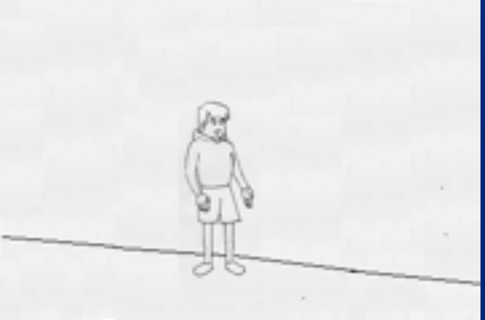
Close shot



Medium shot

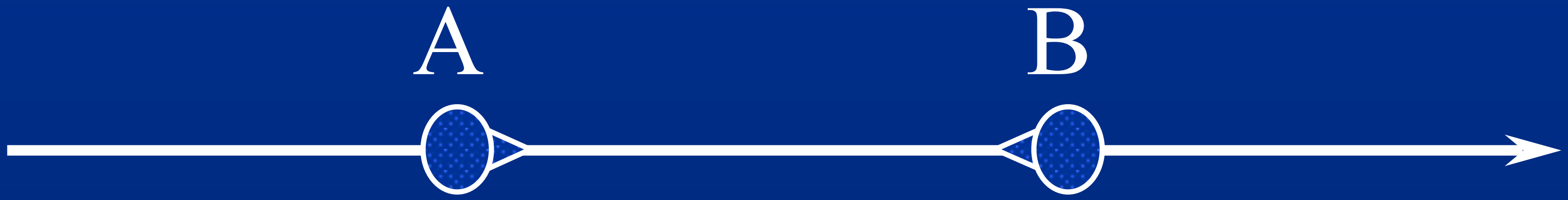


Full shot

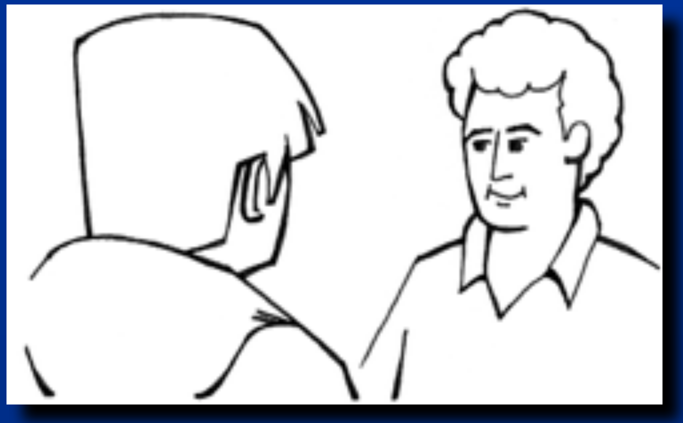


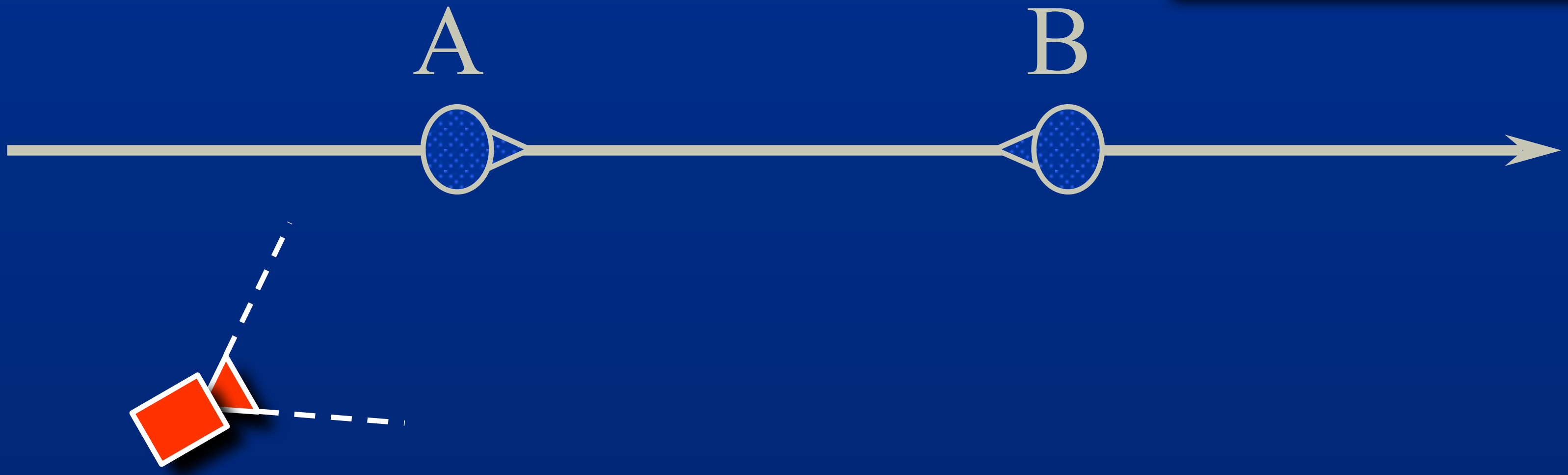
Long shot

# The line of interest

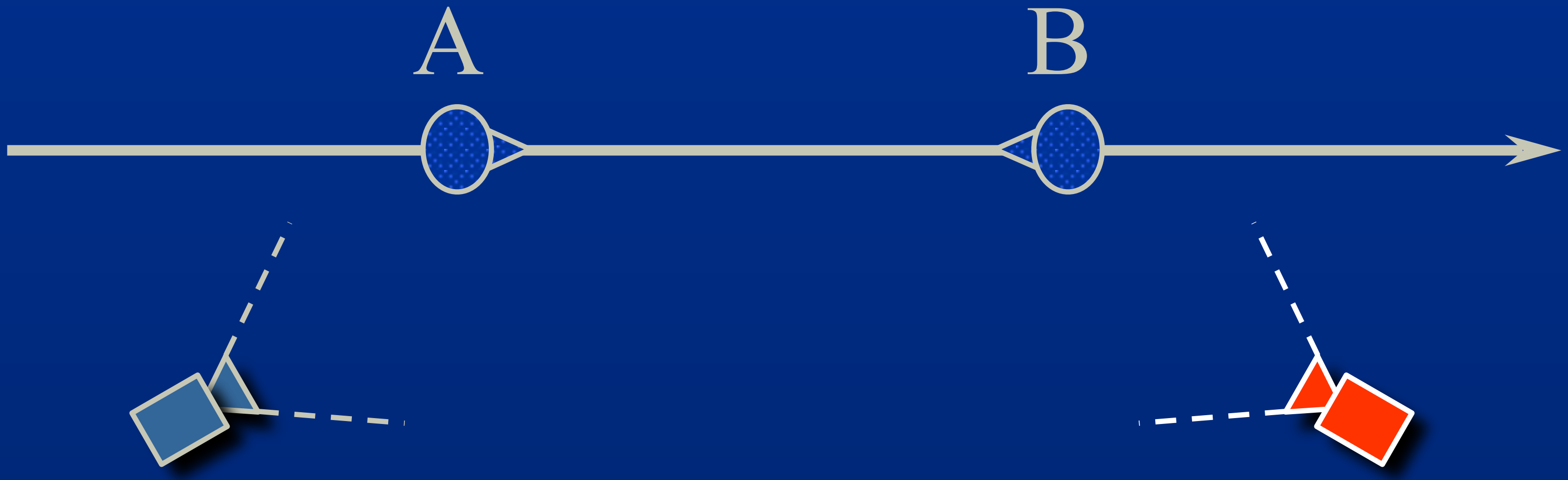


# External camera

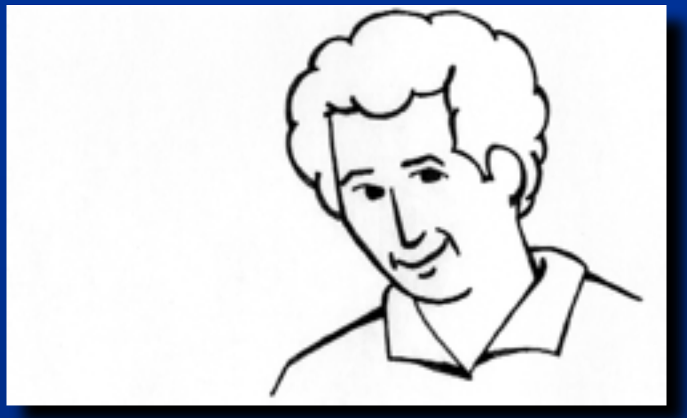


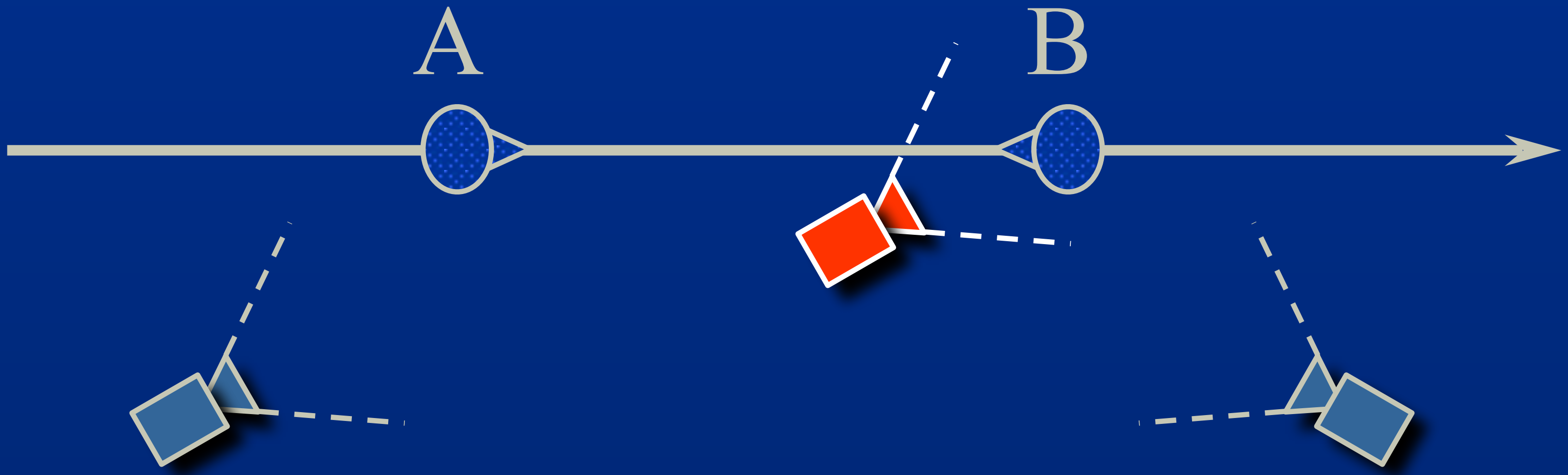


# External camera

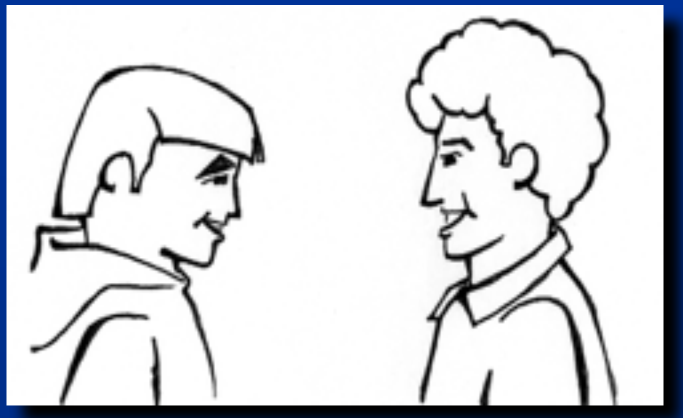


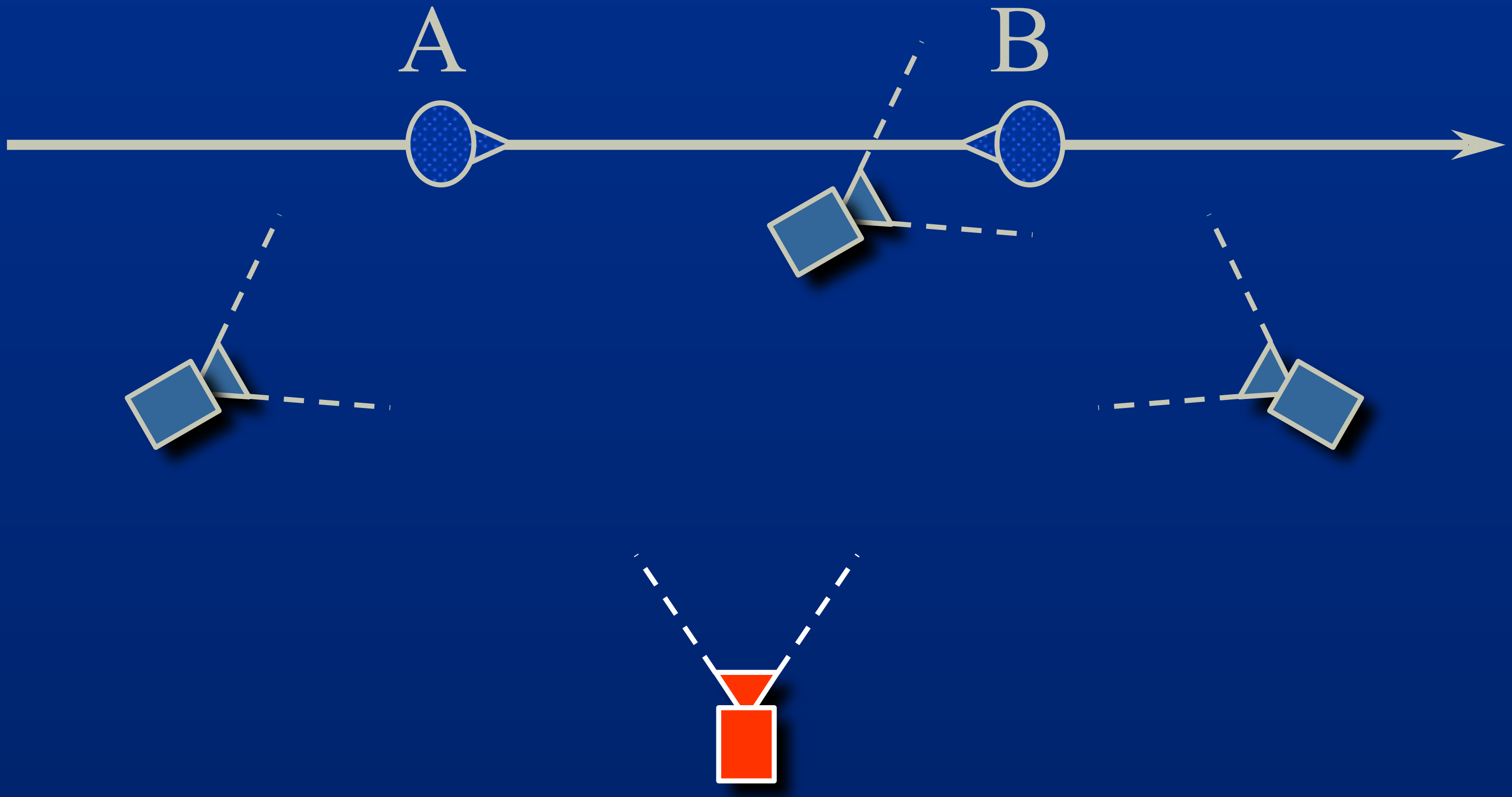
# Internal camera



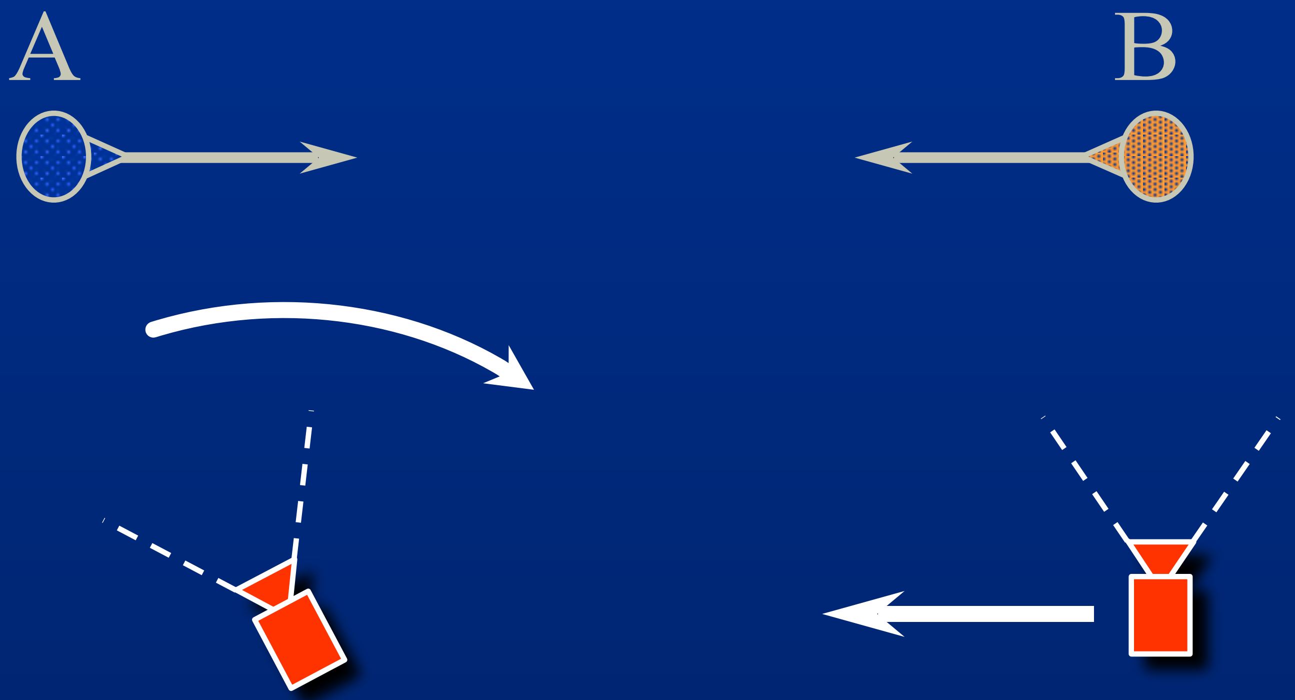


# Apex camera





# Moving cameras



A

B

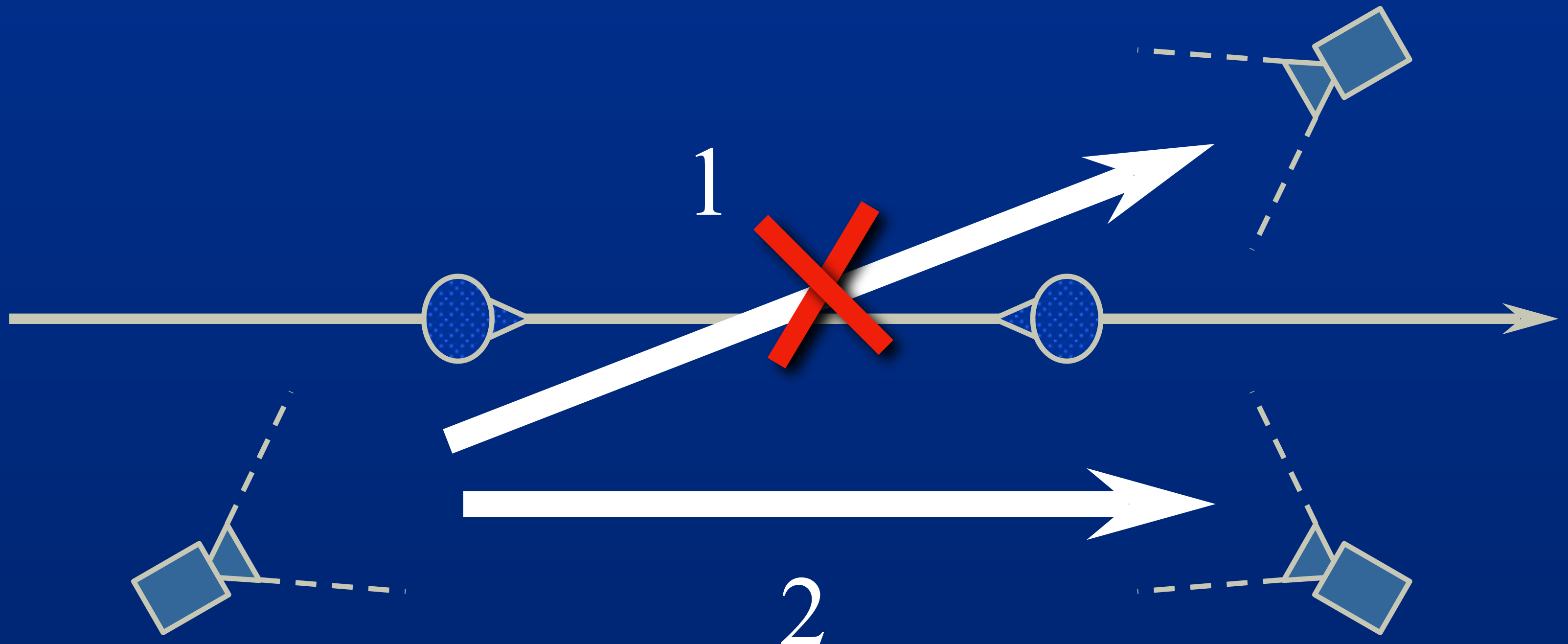
pan(A)

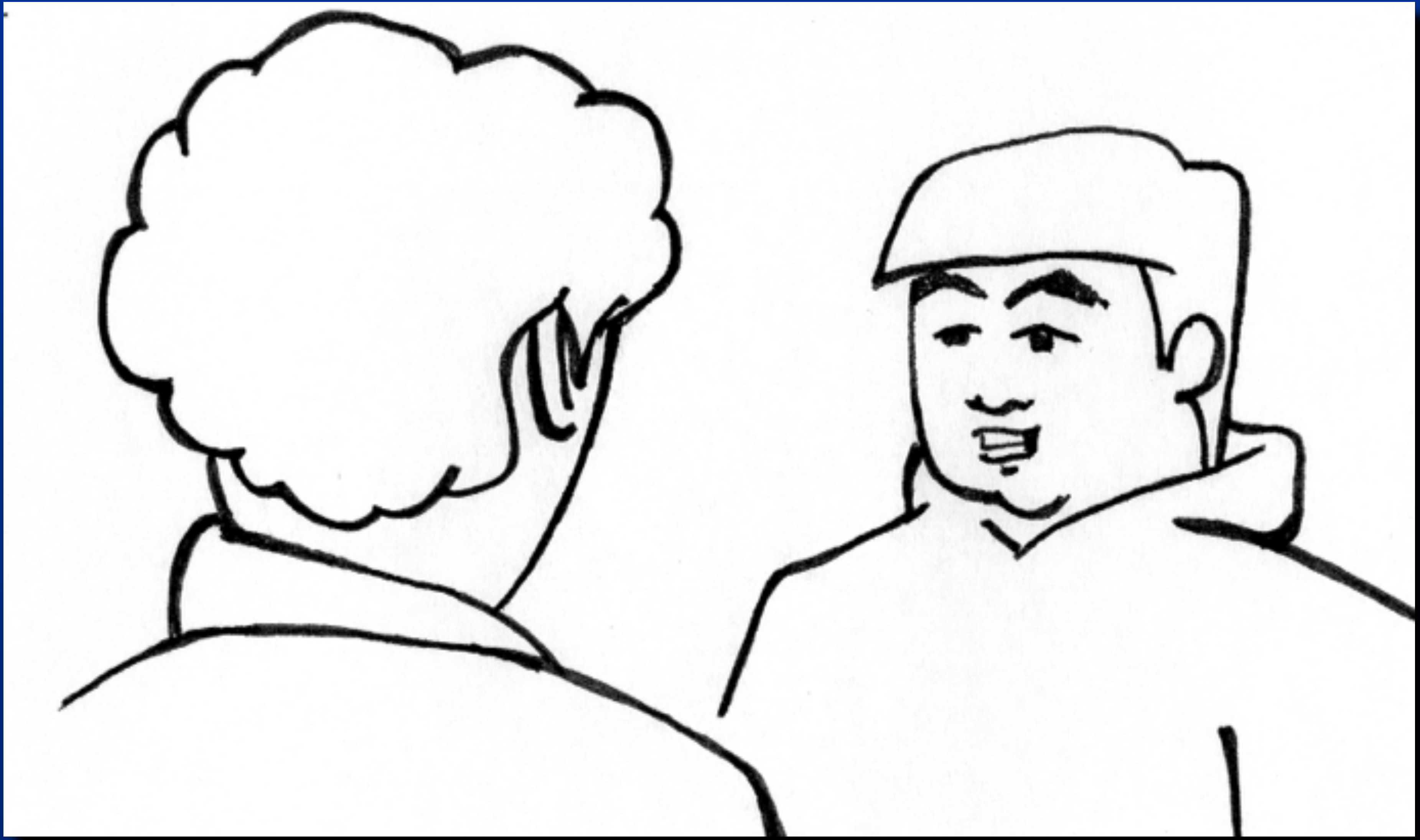
track(B)

# 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

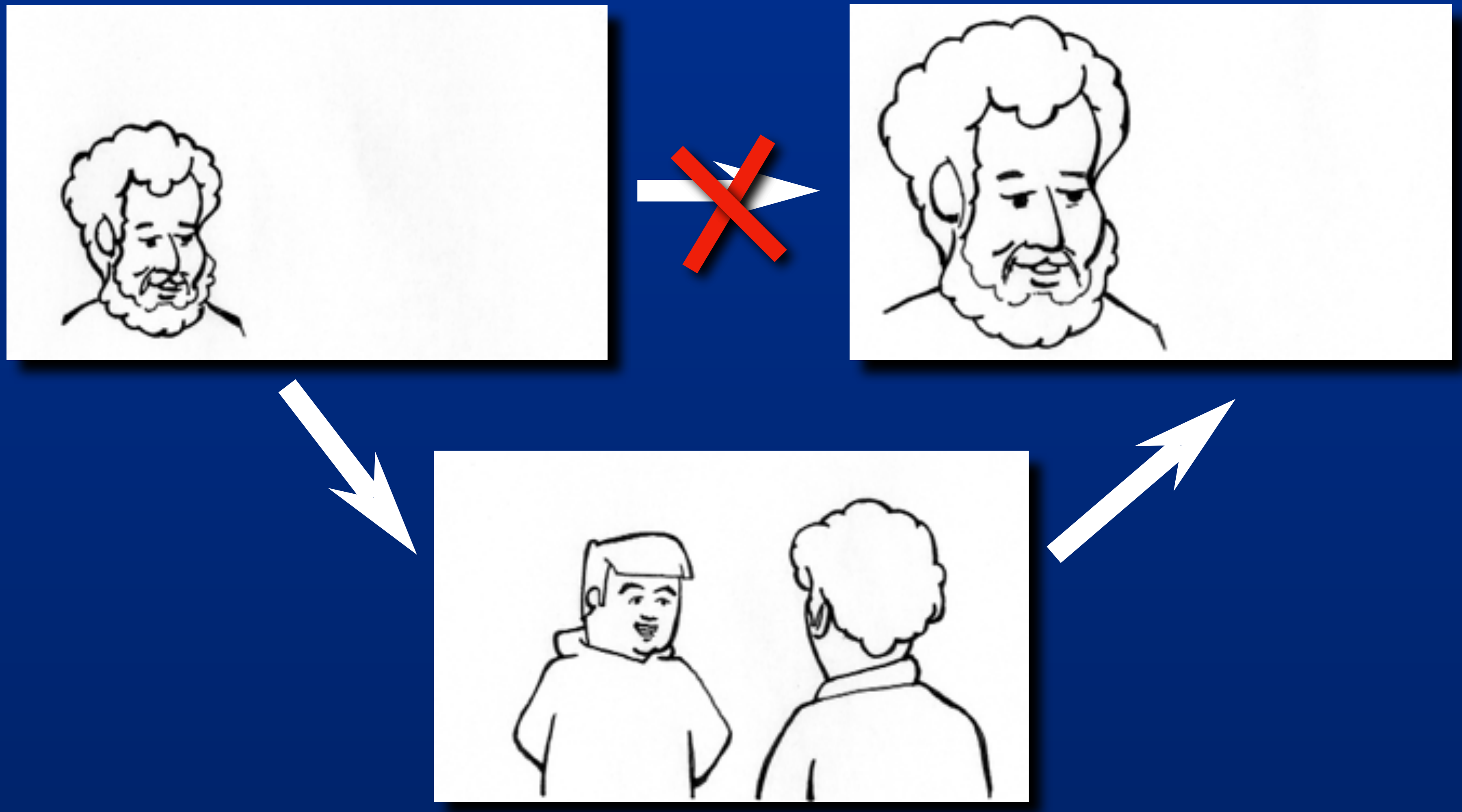




# Some rules in film editing

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

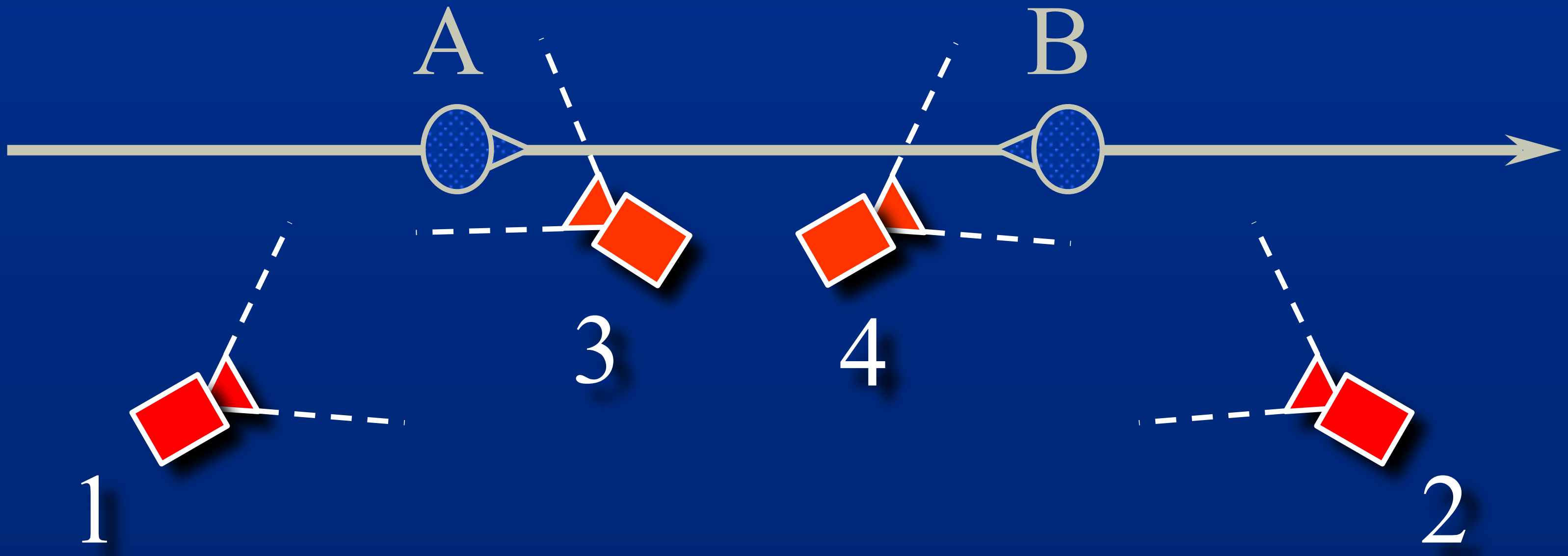
# Avoid jump cut

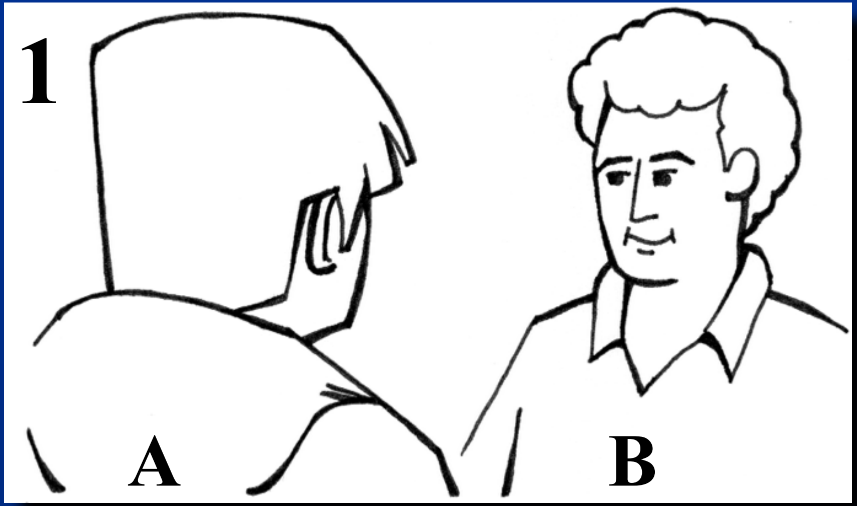


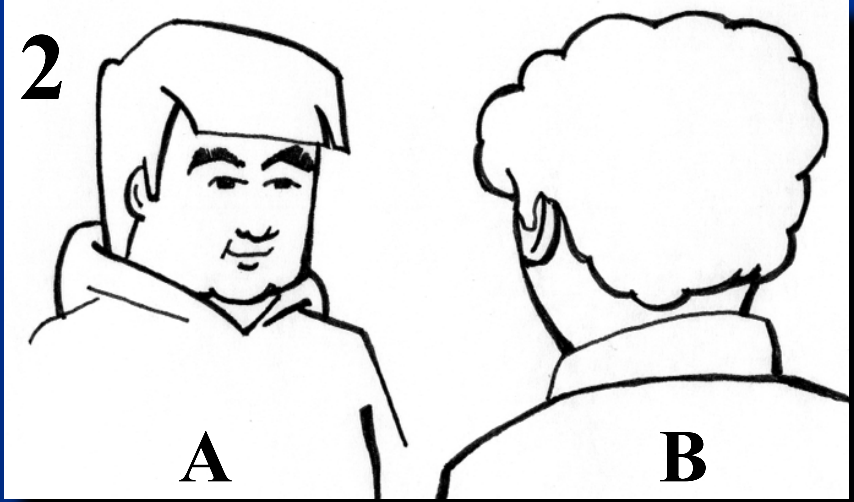
# Some rules in film editing

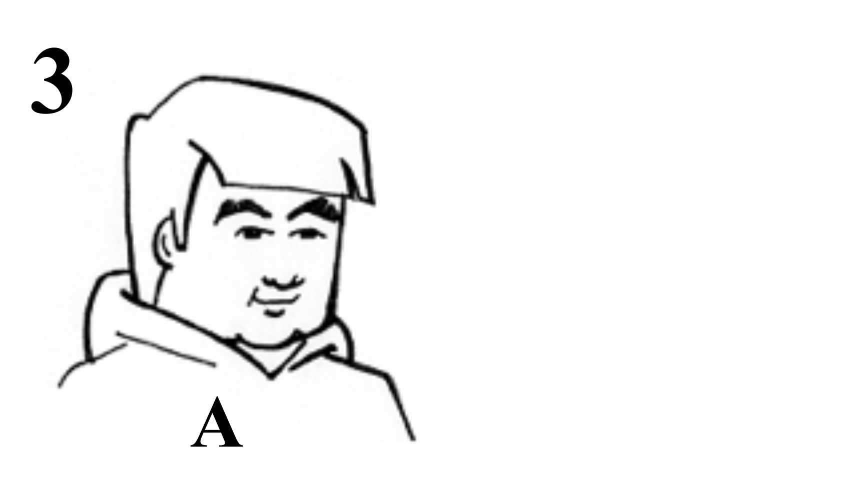
- Don't cross the line of interest
- Avoid jump cuts
- Let the actor lead
- Break movement

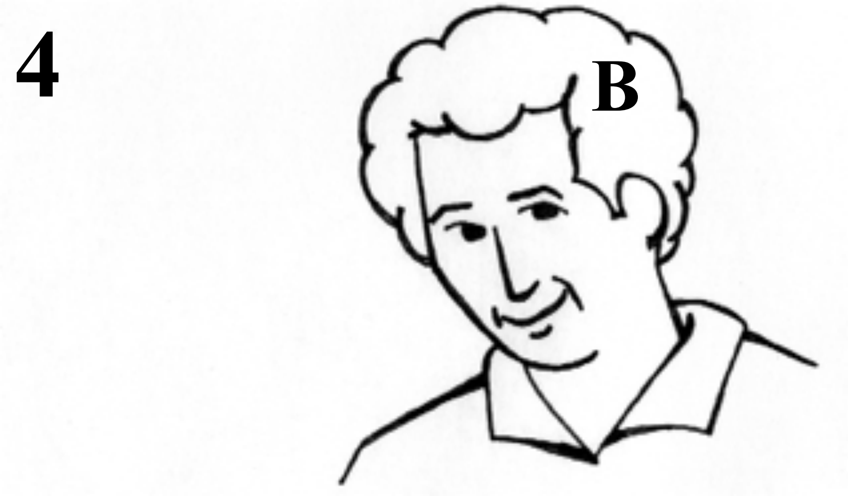
# Two-person conversation



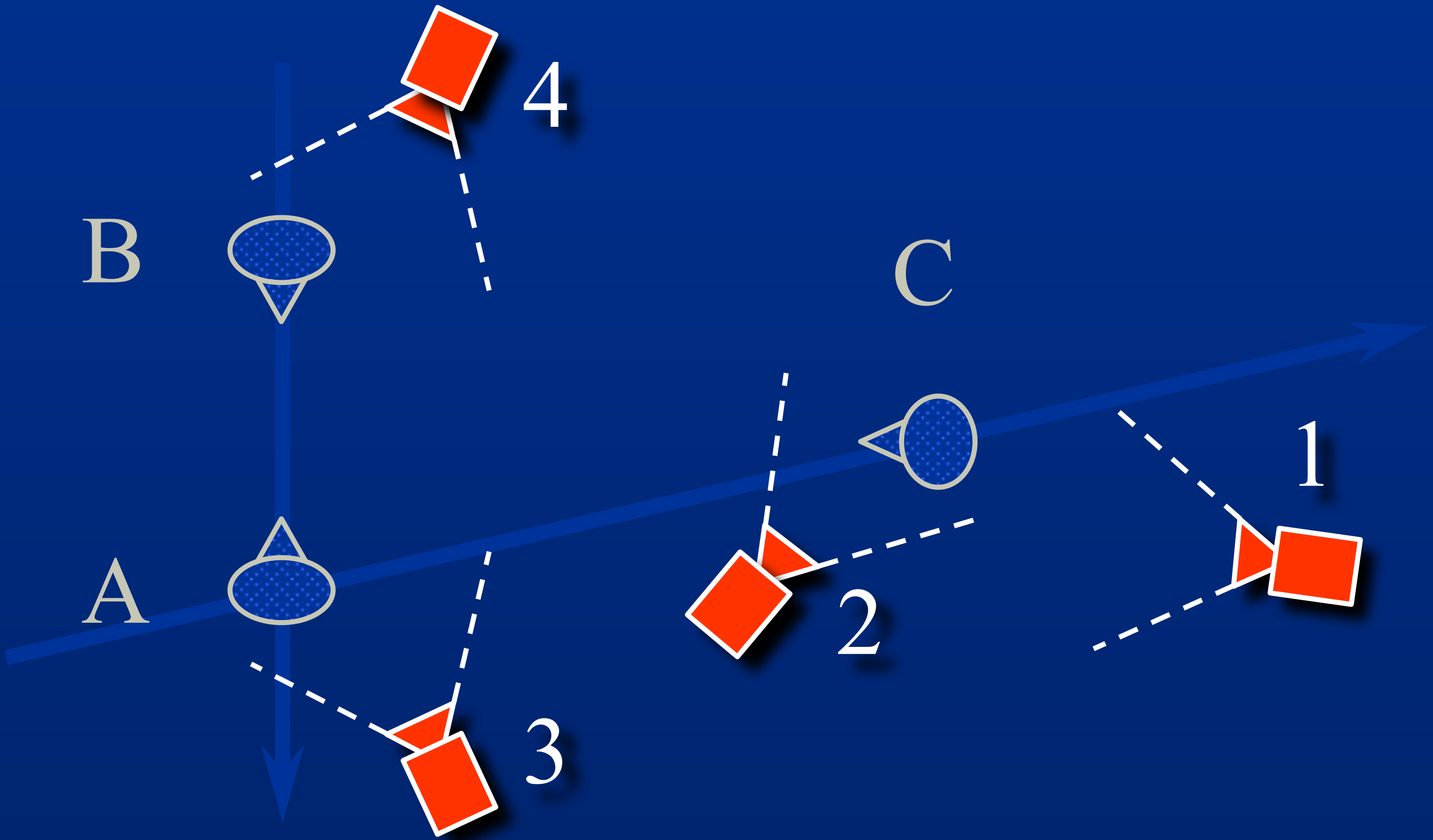

1
A
B

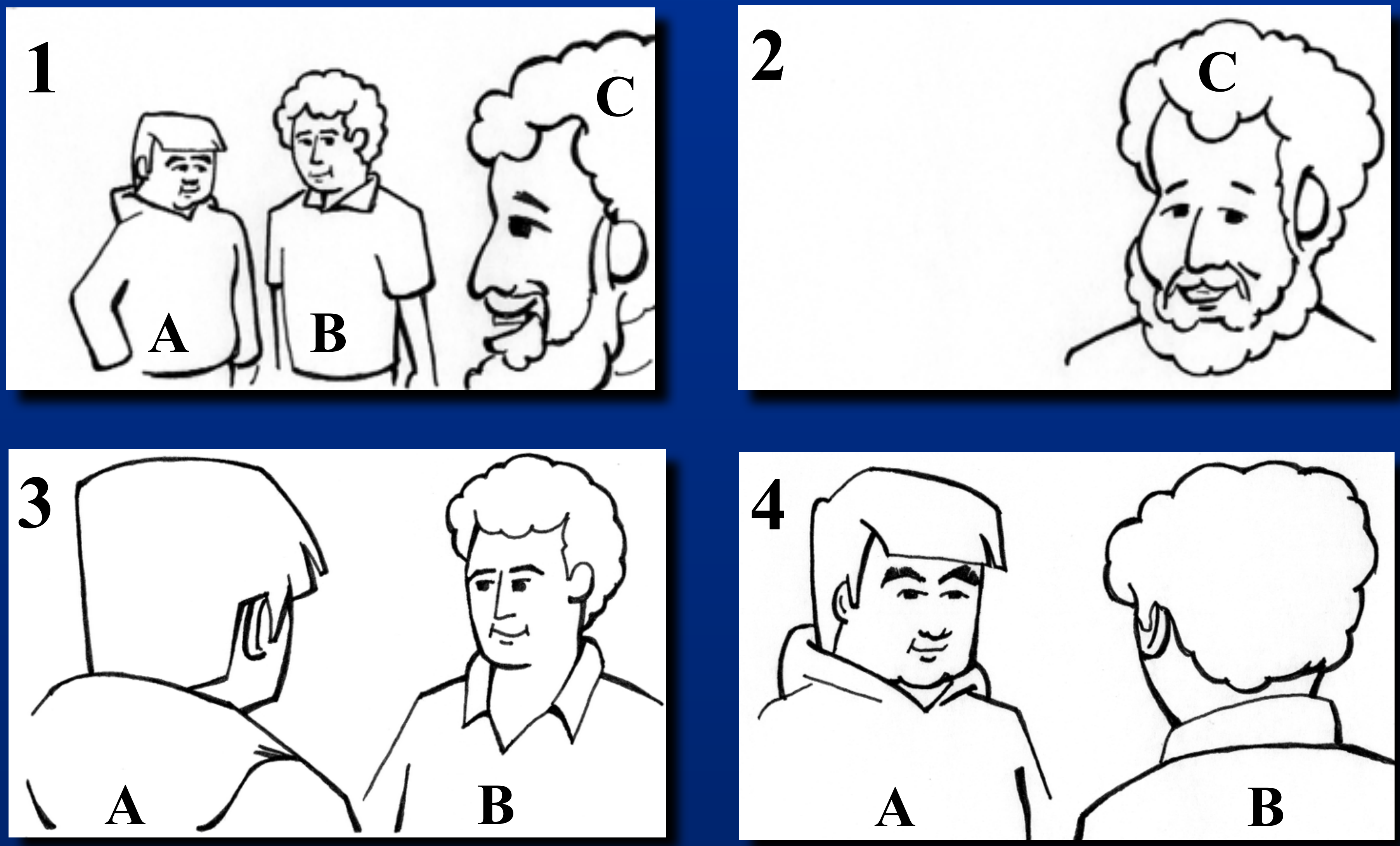

2
A
B

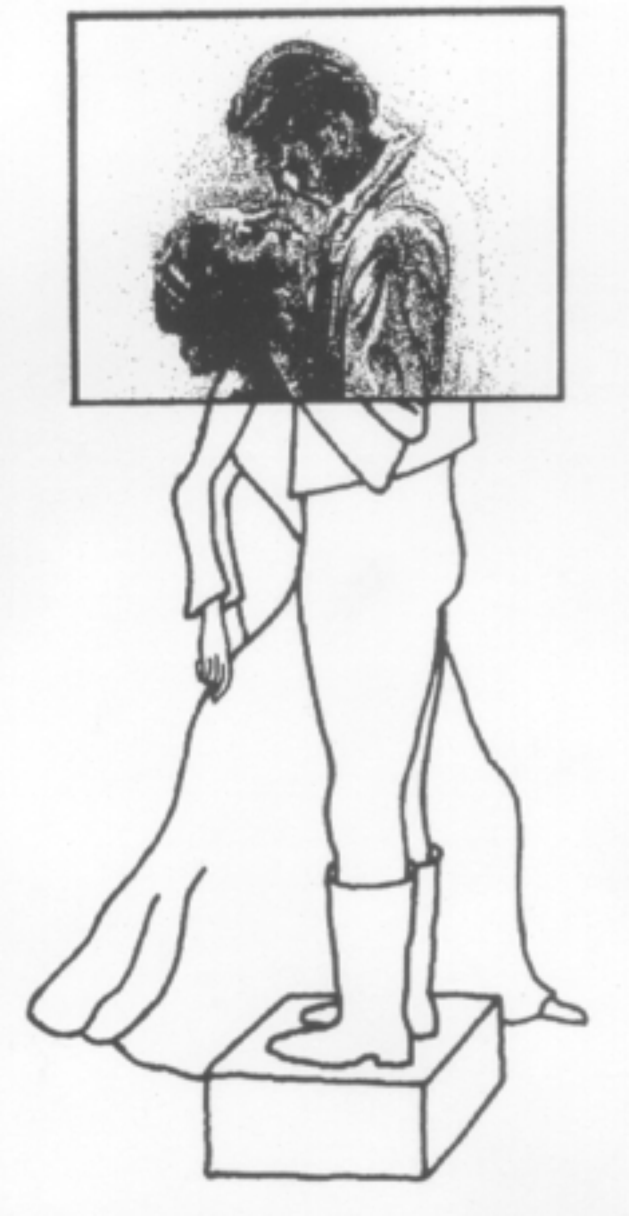

3
A


4
B

# Three-person conversation




1
A
B
C
2
C
3
A
B
4
A
B

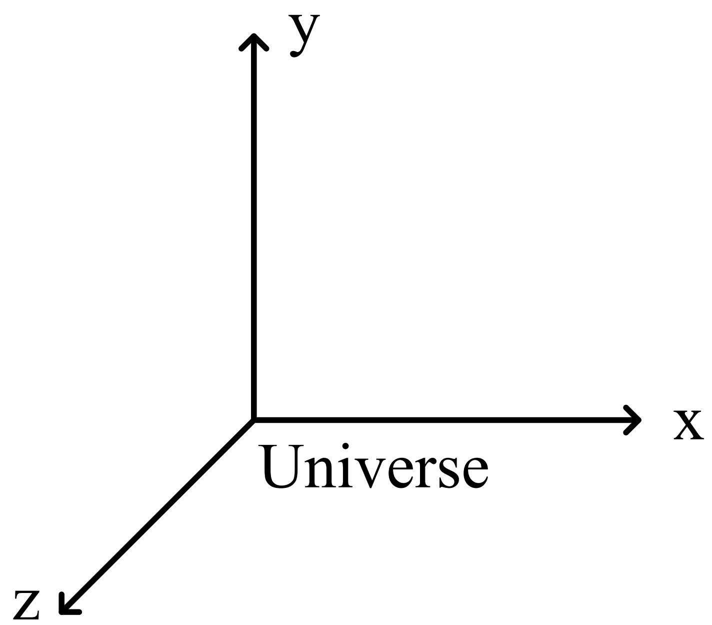


# Bad

Good

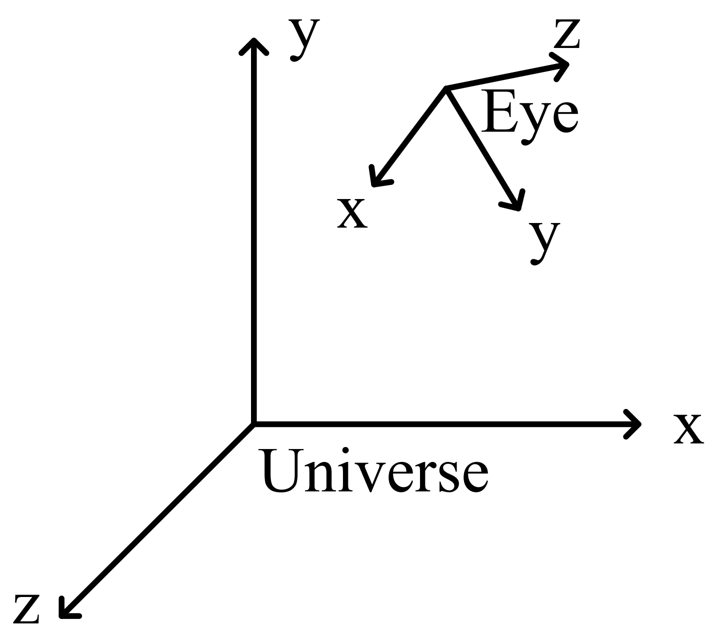
# Three spaces

The universe space



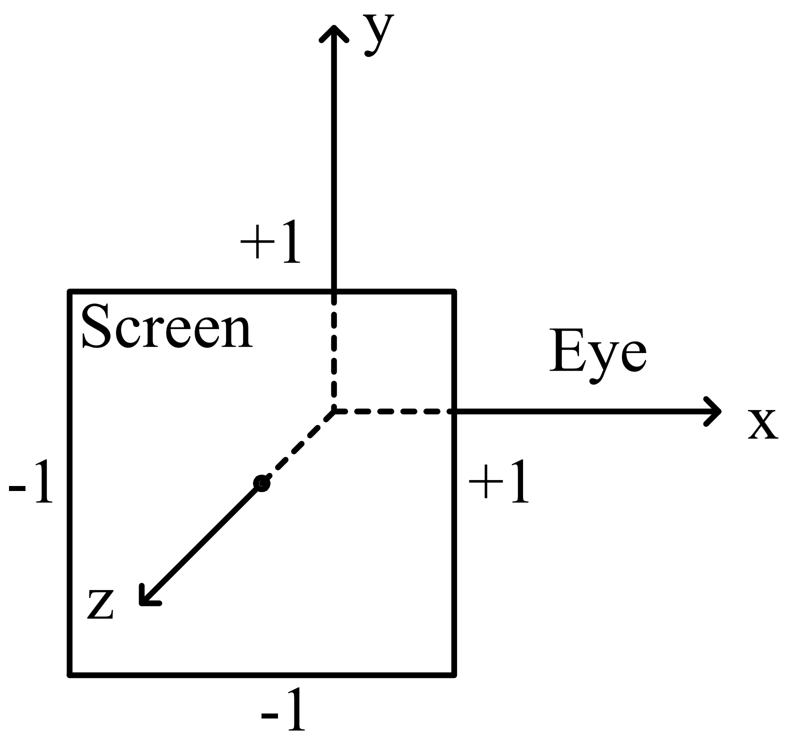
# Three spaces

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



# Three spaces

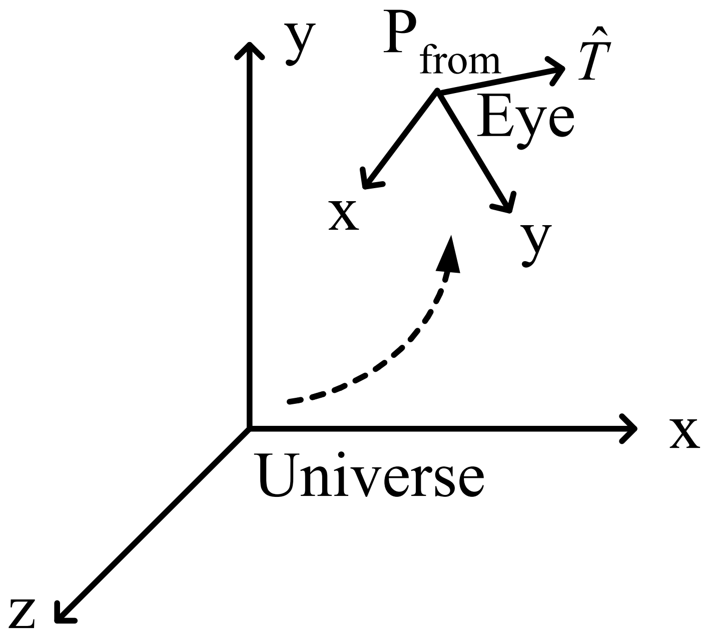
The screen space (screen is ctan(fov/2) away from eye point)



# Standard look-at transformation

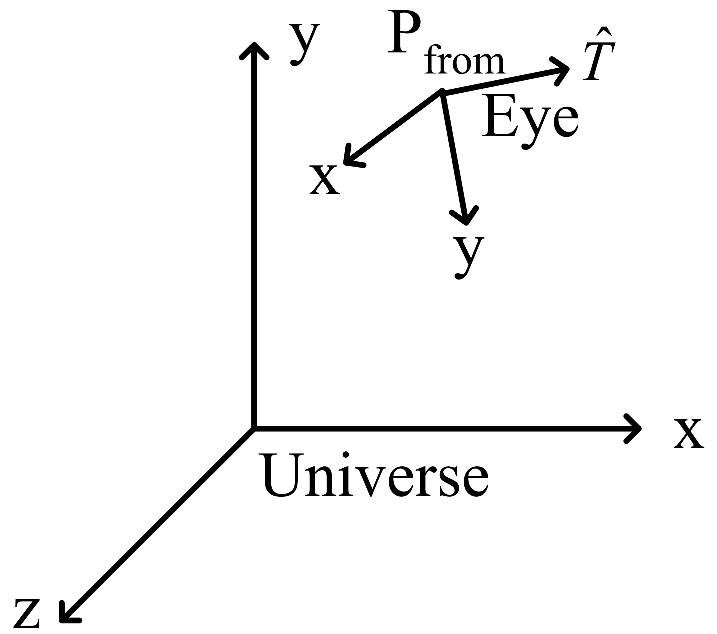
Given eye position $P_{\text{from}}$, a look at direction $\hat{T}$

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



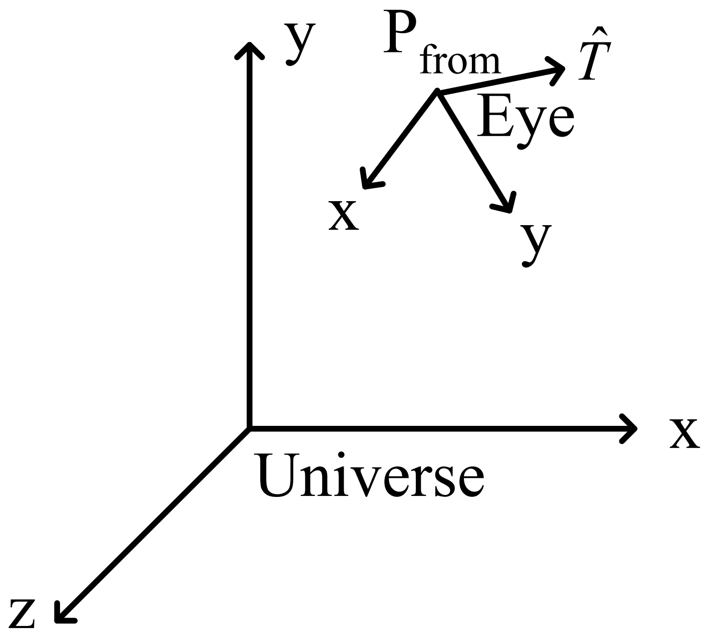
# Standard look-at transformation

But this is not unique



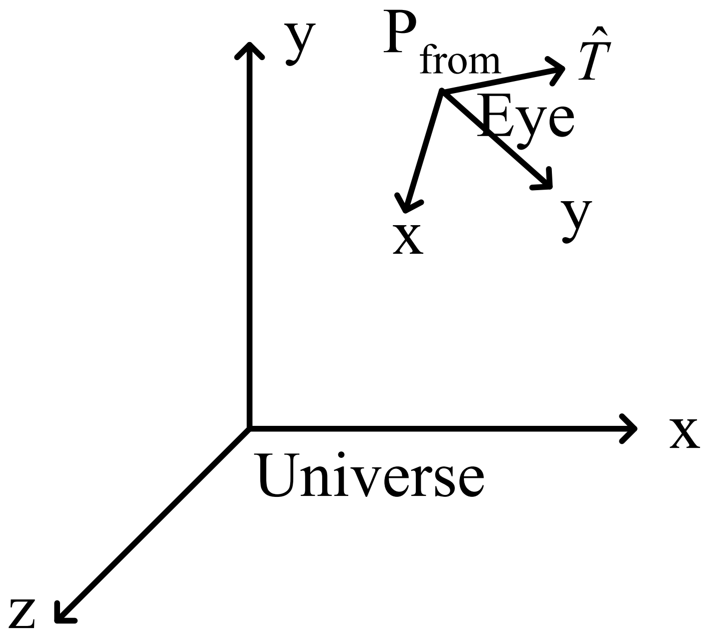
# Standard look-at transformation

But this is not unique



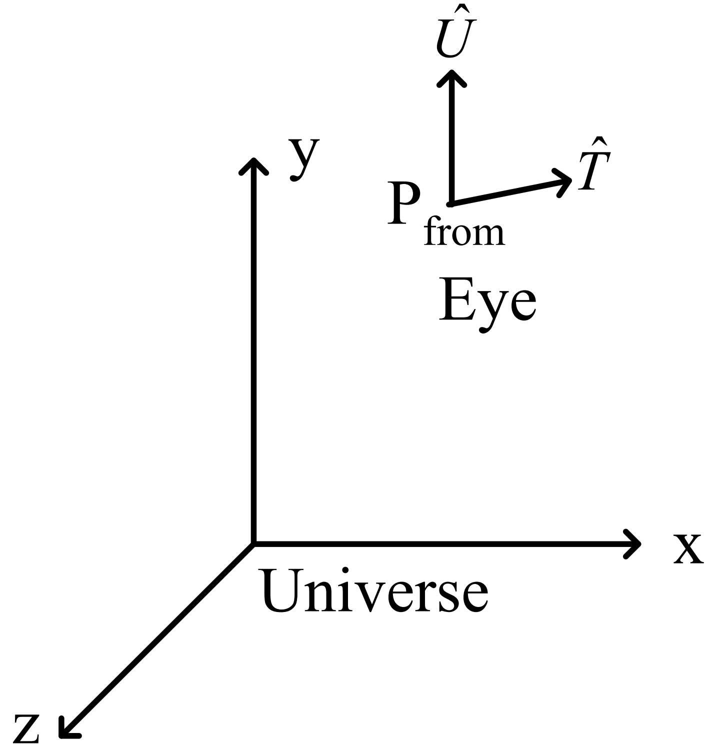
# Standard look-at transformation

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

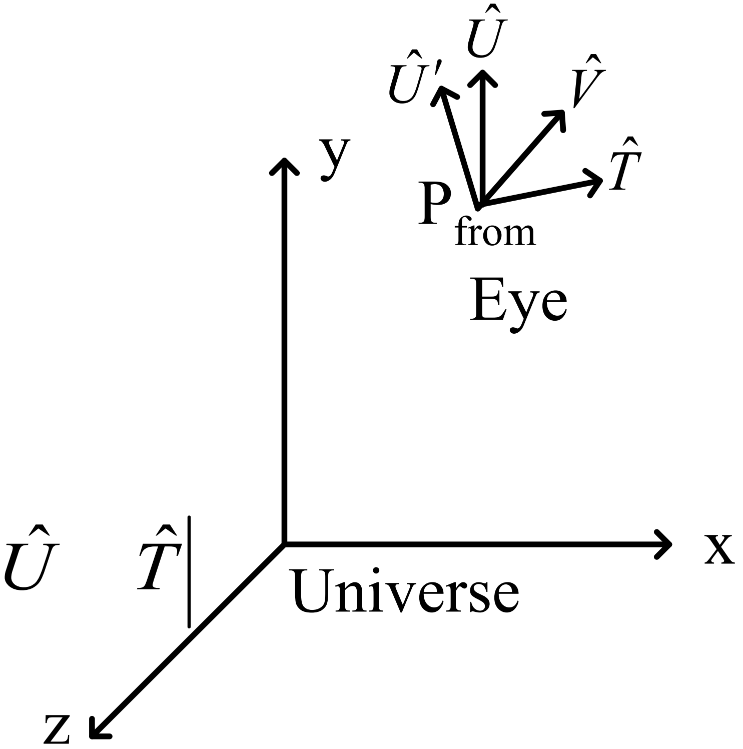
Z-axis is $\hat{T}$

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

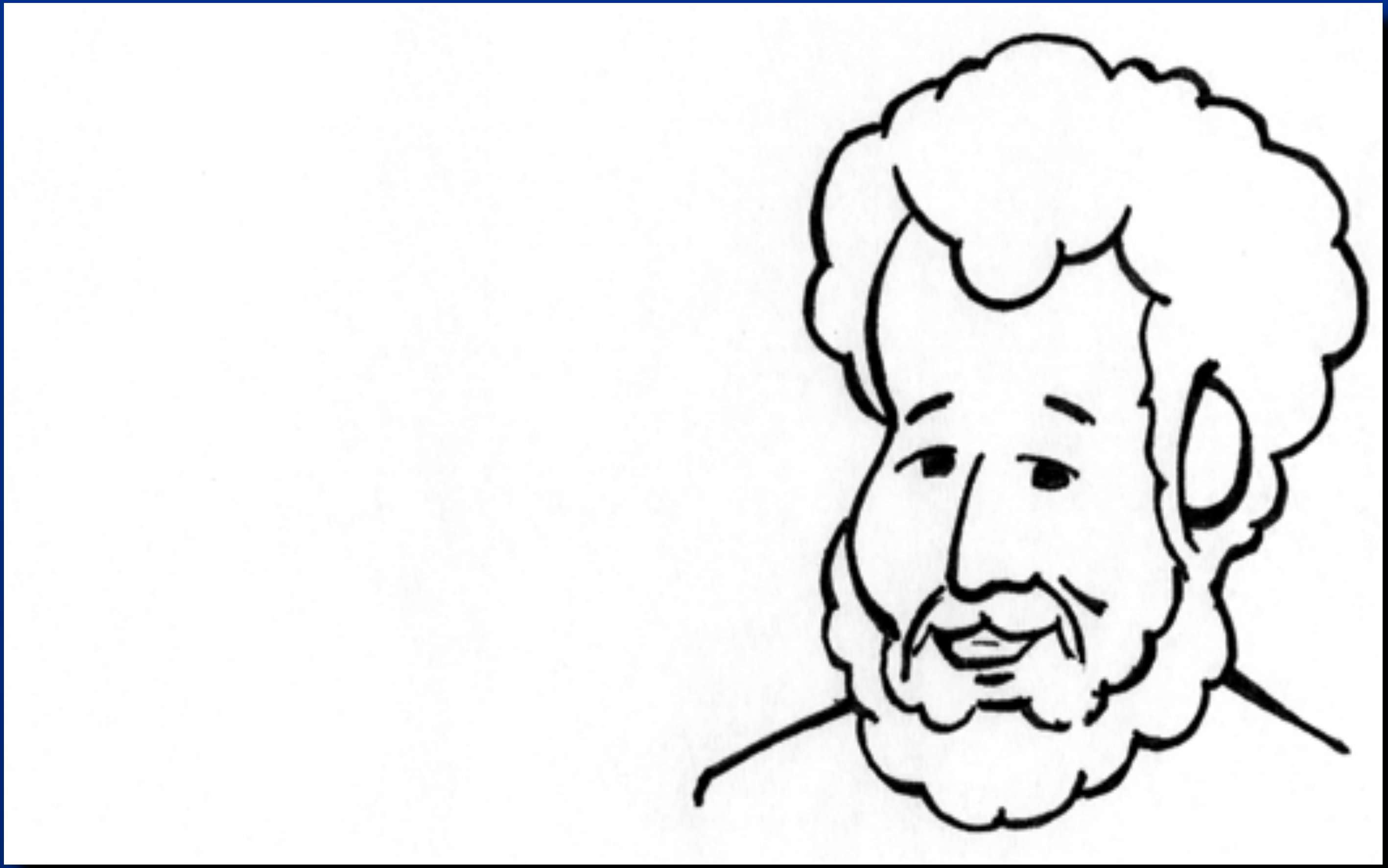
Y-axis is $\hat{U}' = \hat{T} \times \hat{V}$

Rotation matrix $R_{3\times3} = \left| \hat{V} \quad \hat{U} \quad \hat{T} \right|$

Translation is $-P_{\text{from}}$



# Internal close-up of actor A



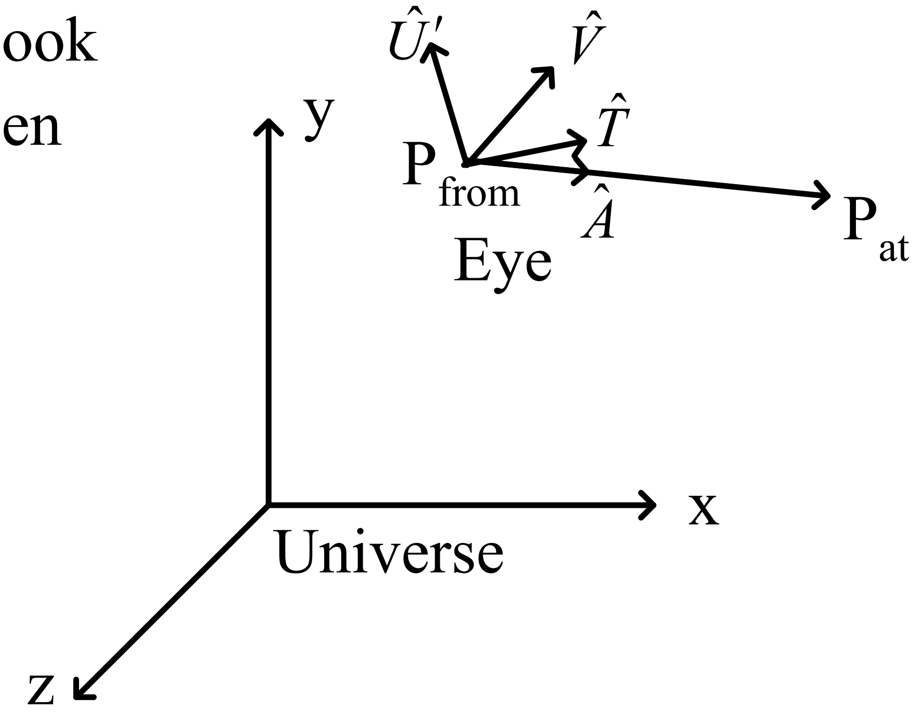
# Look at $[x_{at}, y_{at}]$ transformation

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

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



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

In screen space:

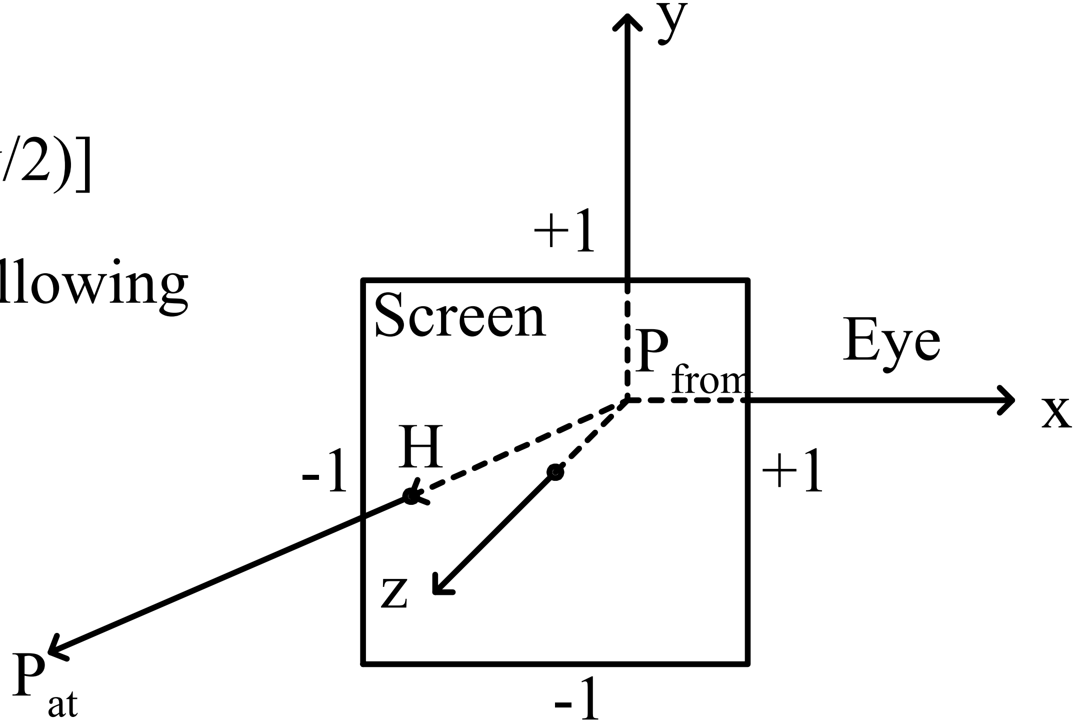
$H = [x_{at}, y_{at}, \mathrm{ctan}(fov/2)]$

Solve $\hat{T}$ from the following equations:

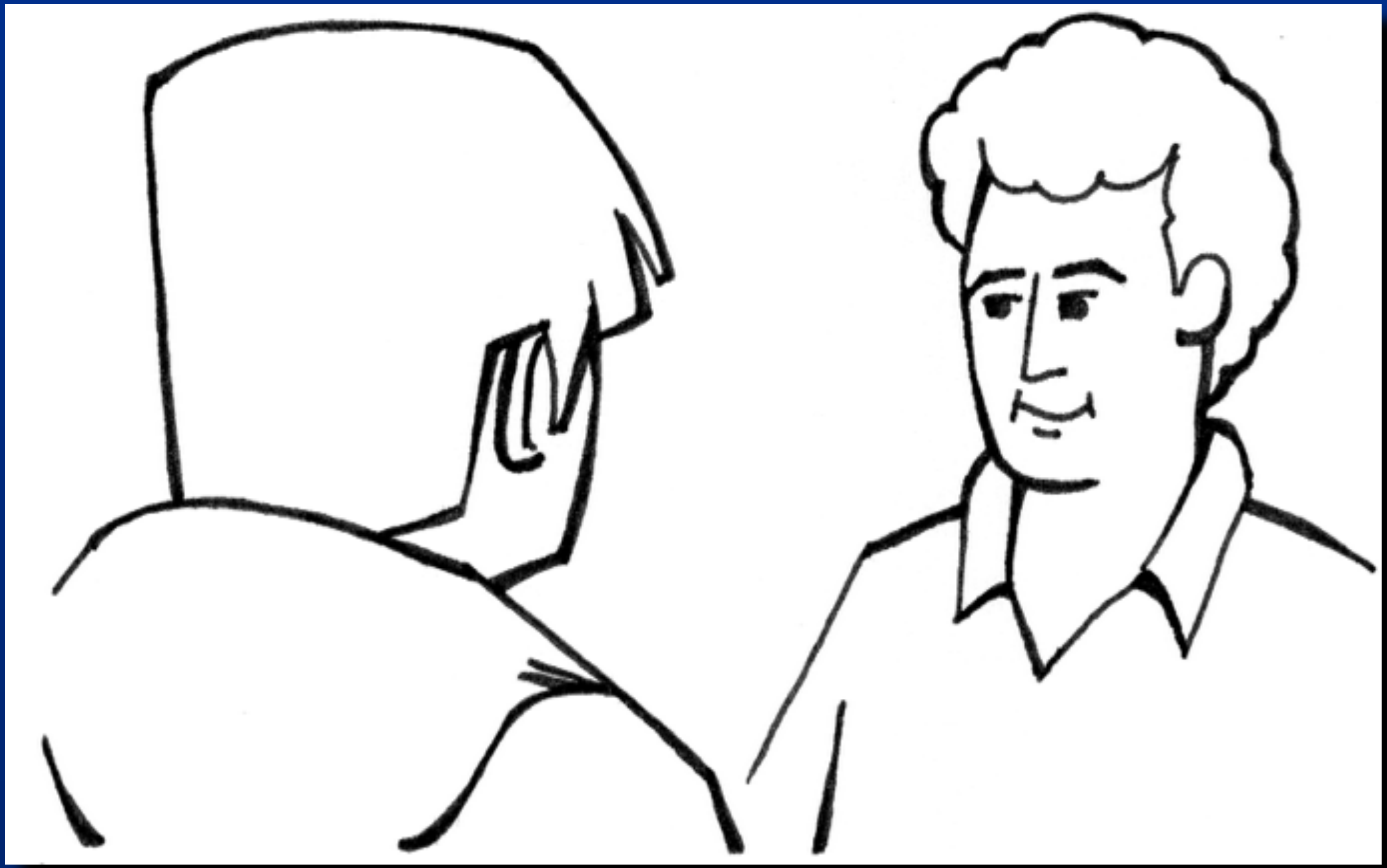
$$\hat{H}\cdot[0,0,1] = \hat{A}\cdot\hat{T}$$

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

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



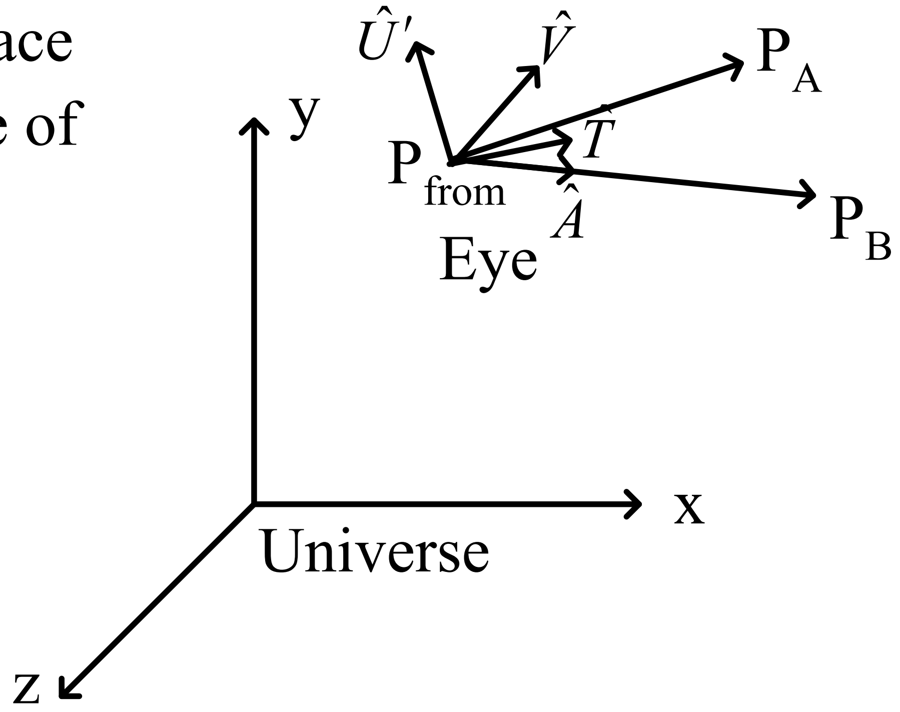
# External of actor A and actor B



# 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 $P_a$.

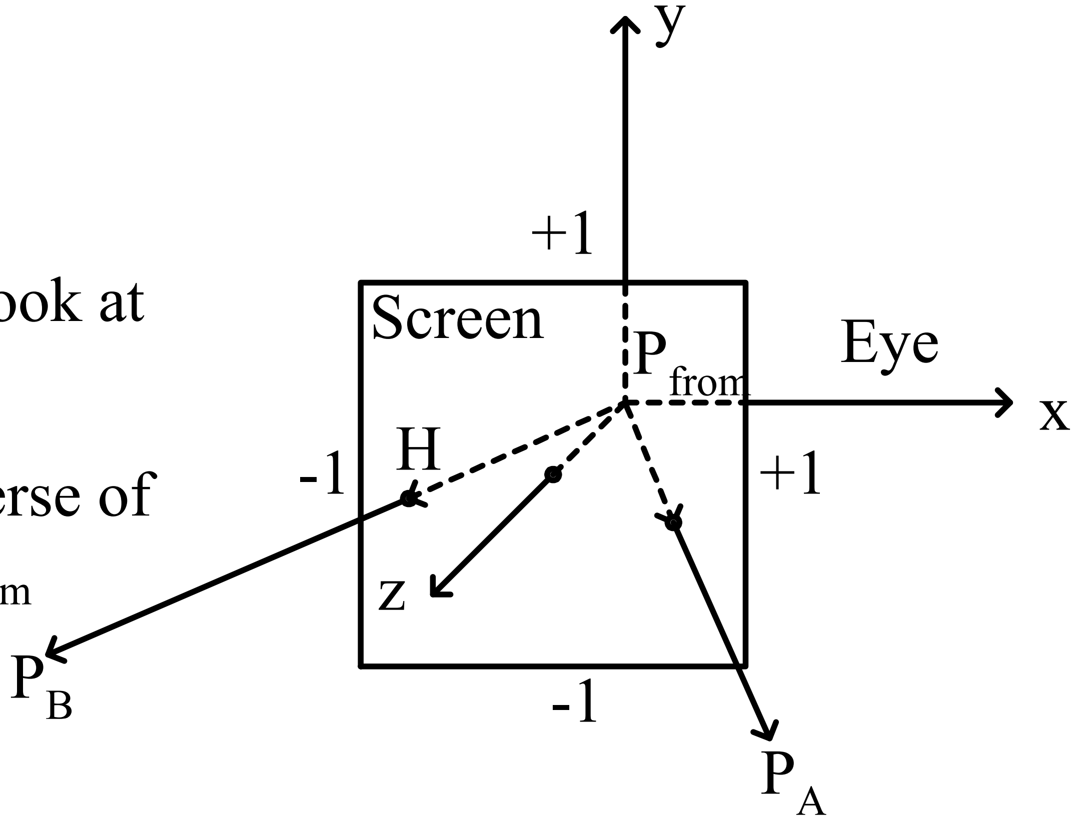
Want $P_{from}$ and $\hat{T}$



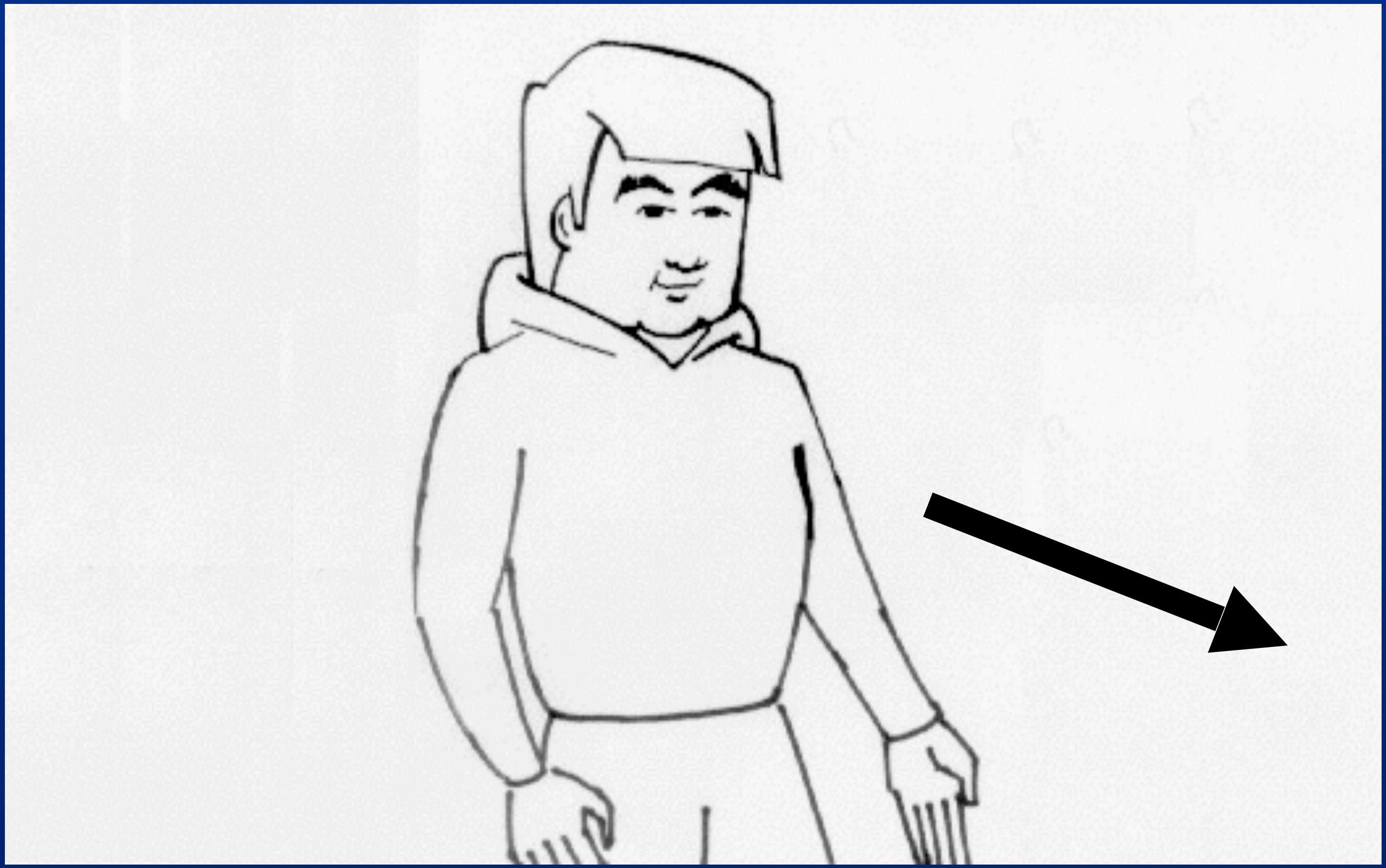
# Look at $[x_A,y_A]$ and $[x_B,y_B]$

Use numerical method:

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



# Medium panning shot of actor A

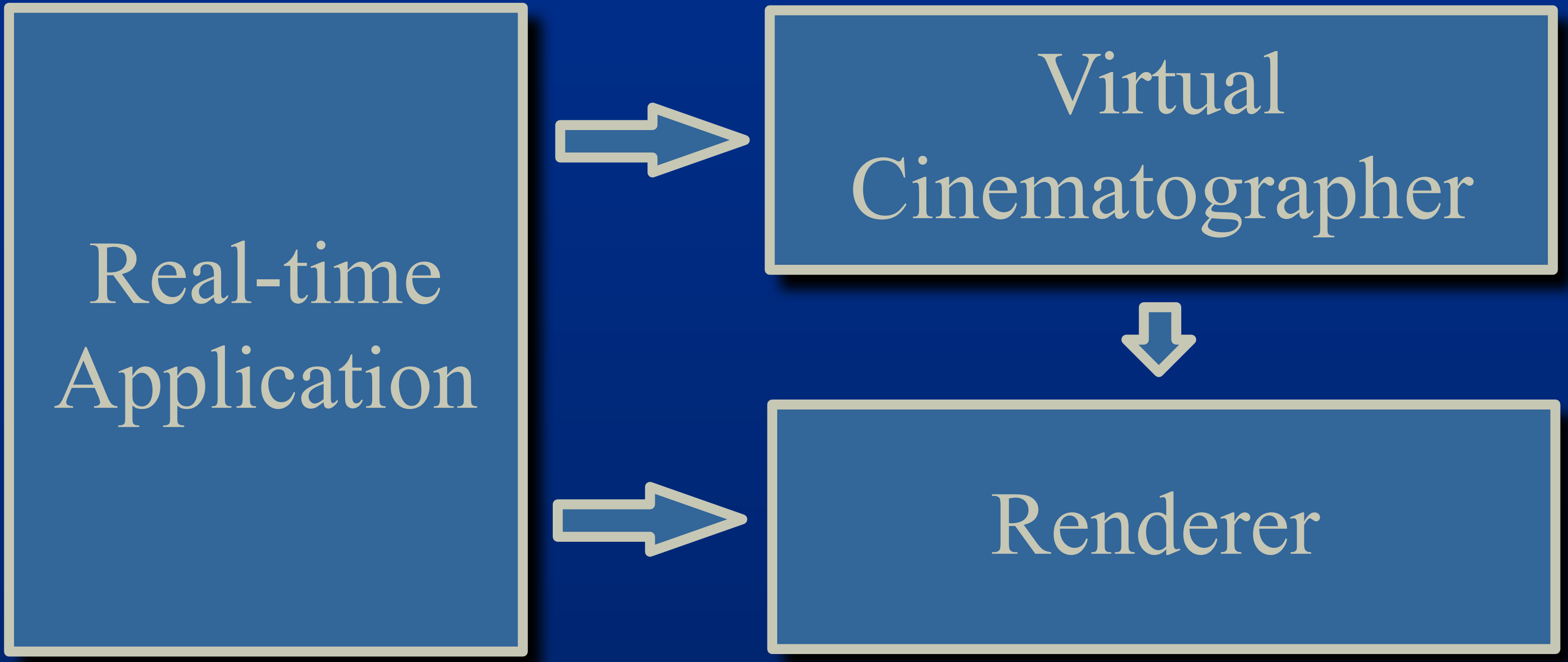


# References

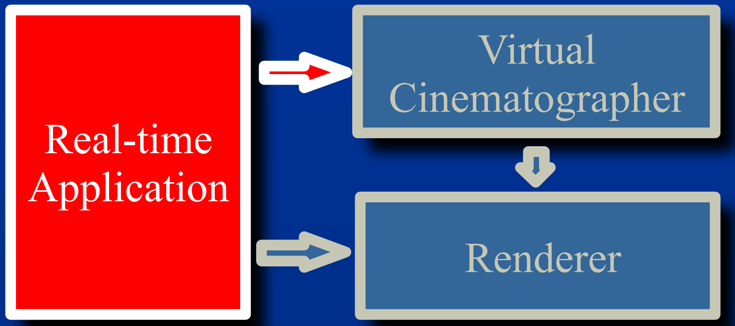
- Jim Blinn’s CG&A ’88 article
  - Where am I? What am I looking at?
- Steve Drucker’s Interactive 3D ’92, ’94, ‘95 papers
  - Cast camera positioning as an optimization problem

# System Design

# Overall system diagram

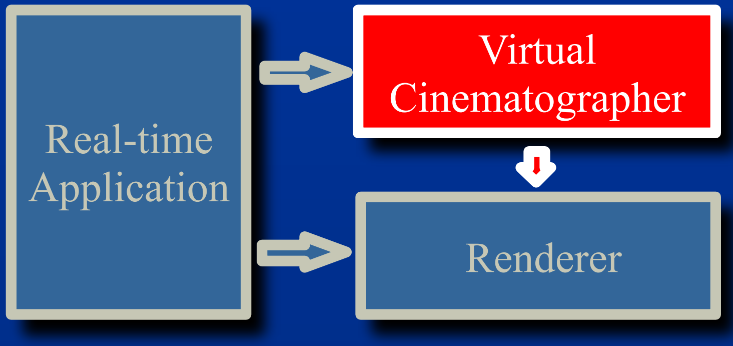


# At each time step



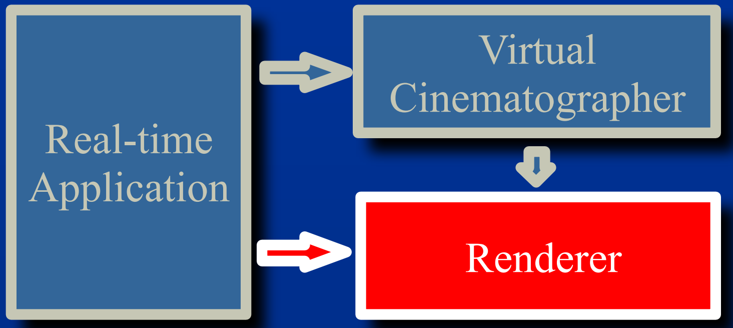
1. Application generates events to VC
   – ( subject, verb, object )
2. VC determines
   – camera specifications
   – acting hints
3. Renderer outputs the image

# At each time step



1. Application generates events to VC
   – ( subject, verb, object )
2. VC determines
   – camera specifications
   – acting hints
3. Renderer outputs the image

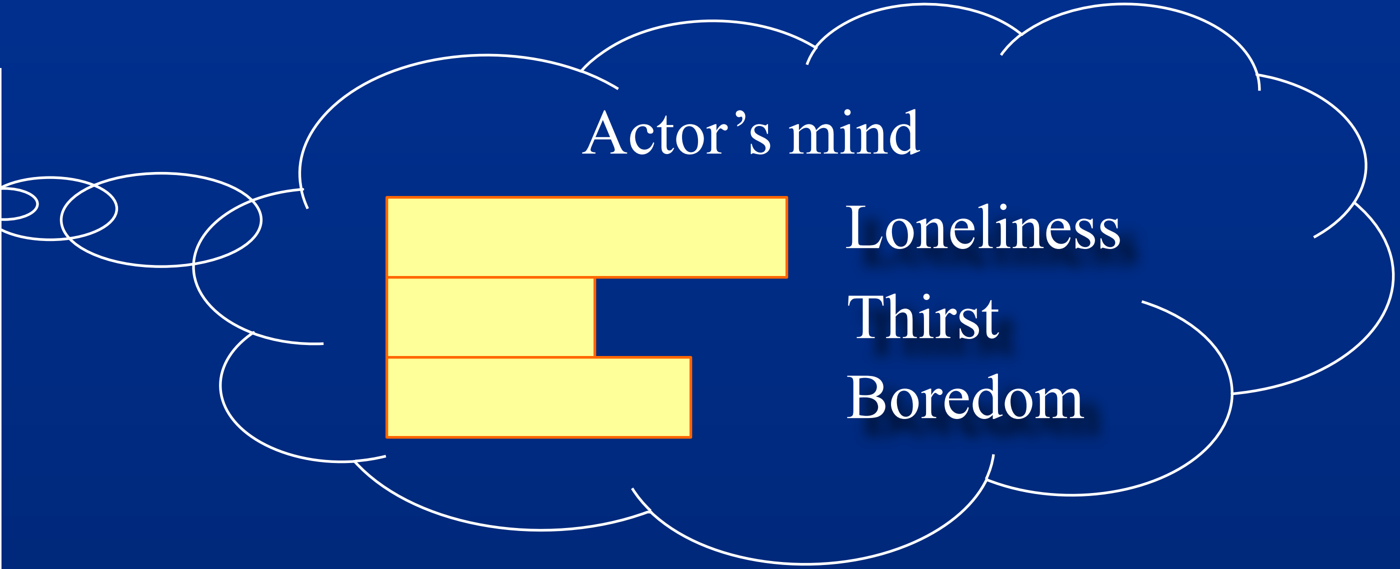
# At each time step



1. Application generates events to VC
   - ( subject, verb, object )
2. VC determines
   - camera specifications
   - acting hints
3. Renderer outputs the image

# A networked virtual party game

# Actors are simulated



Actor’s mind

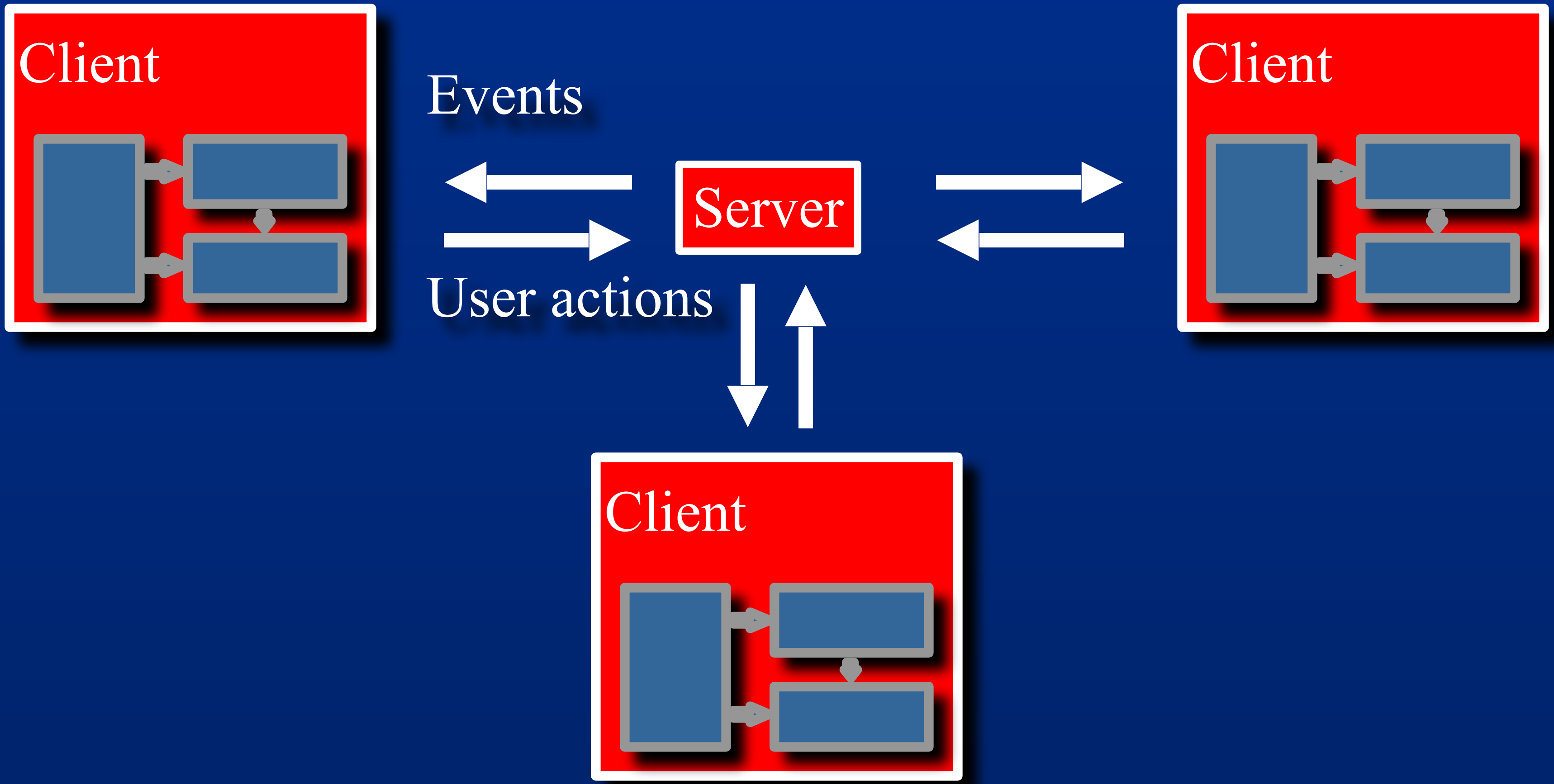
Loneliness

Thirst

Boredom

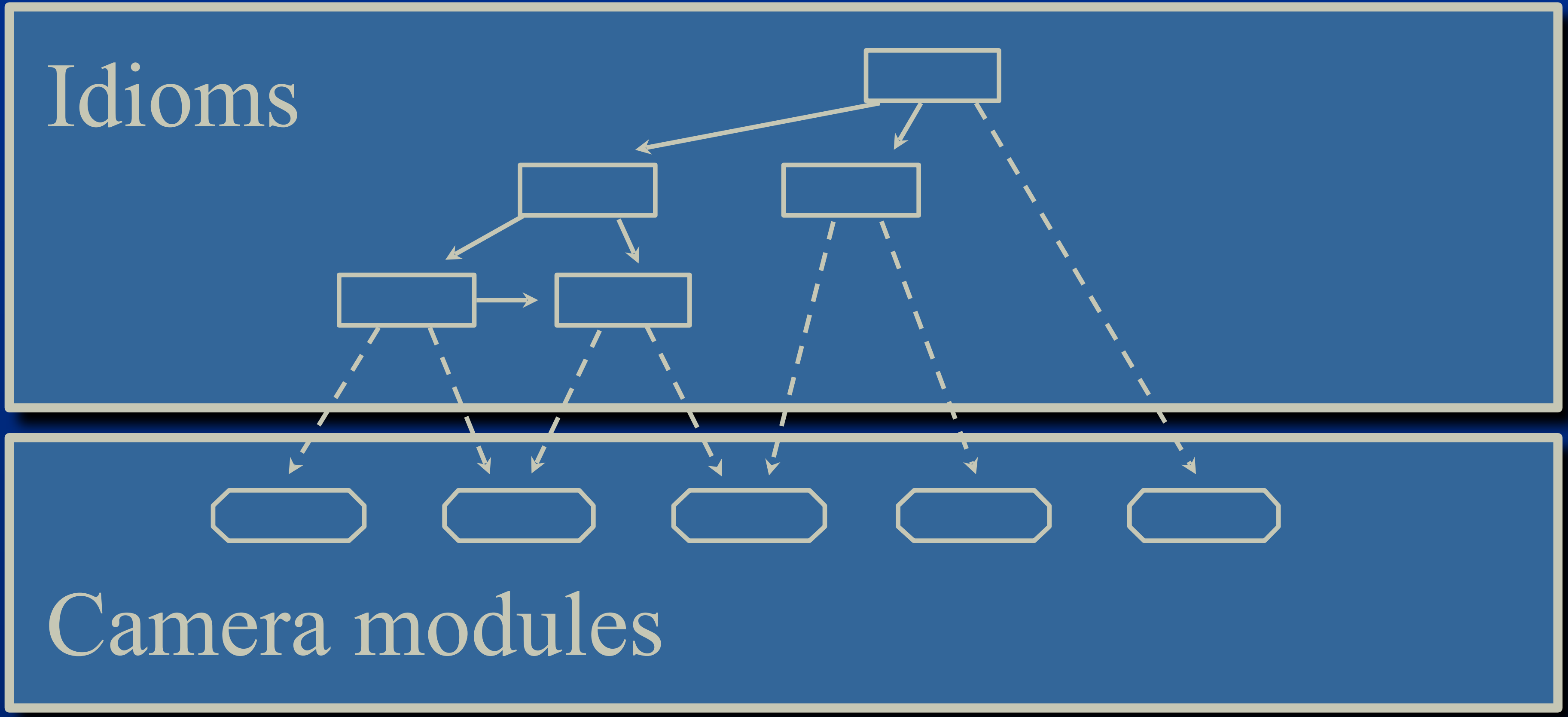
- Walk, converse, look around, drink, etc.
- Can be controlled by the users

# Networked virtual party game

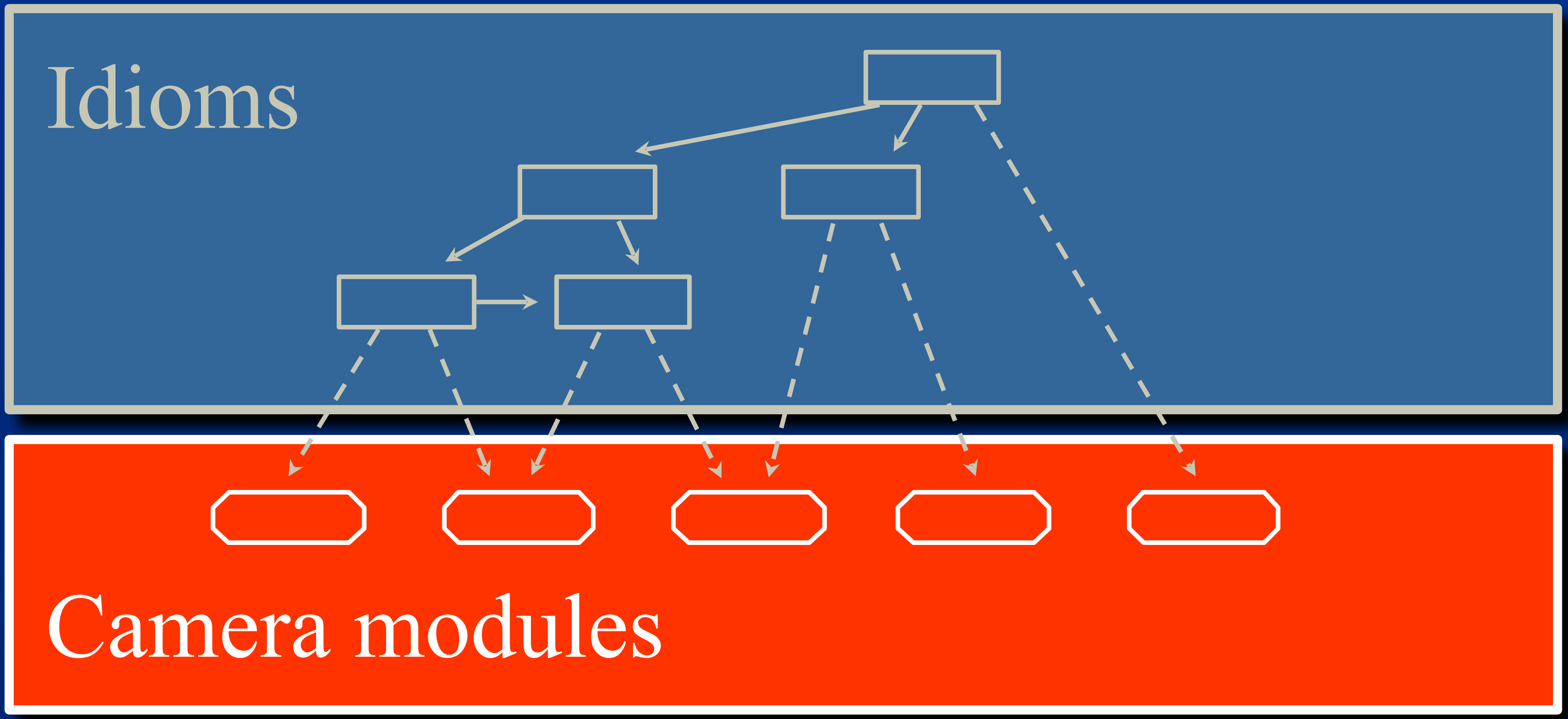


# Inside the Virtual Cinematographer

# VC architecture



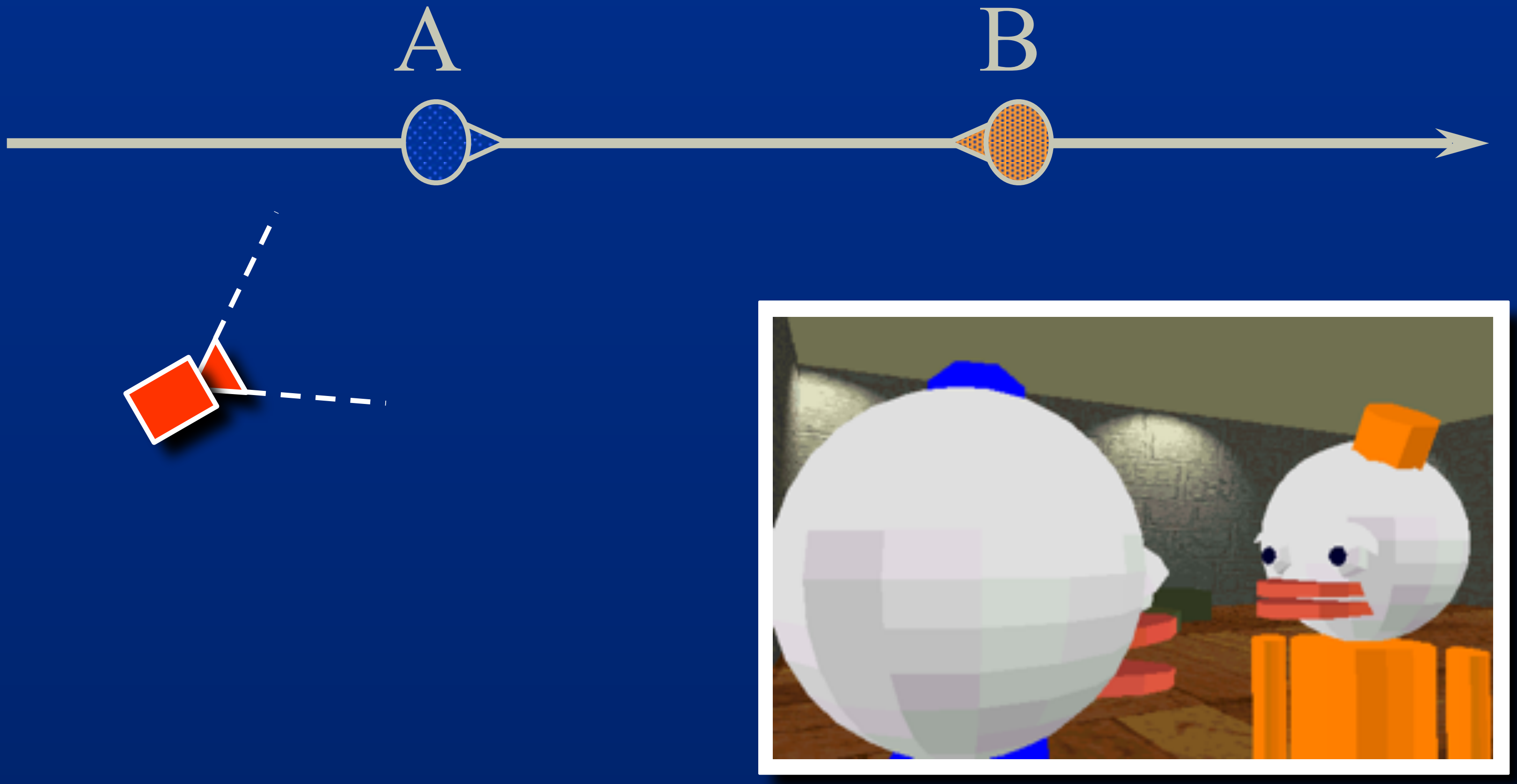
# VC architecture



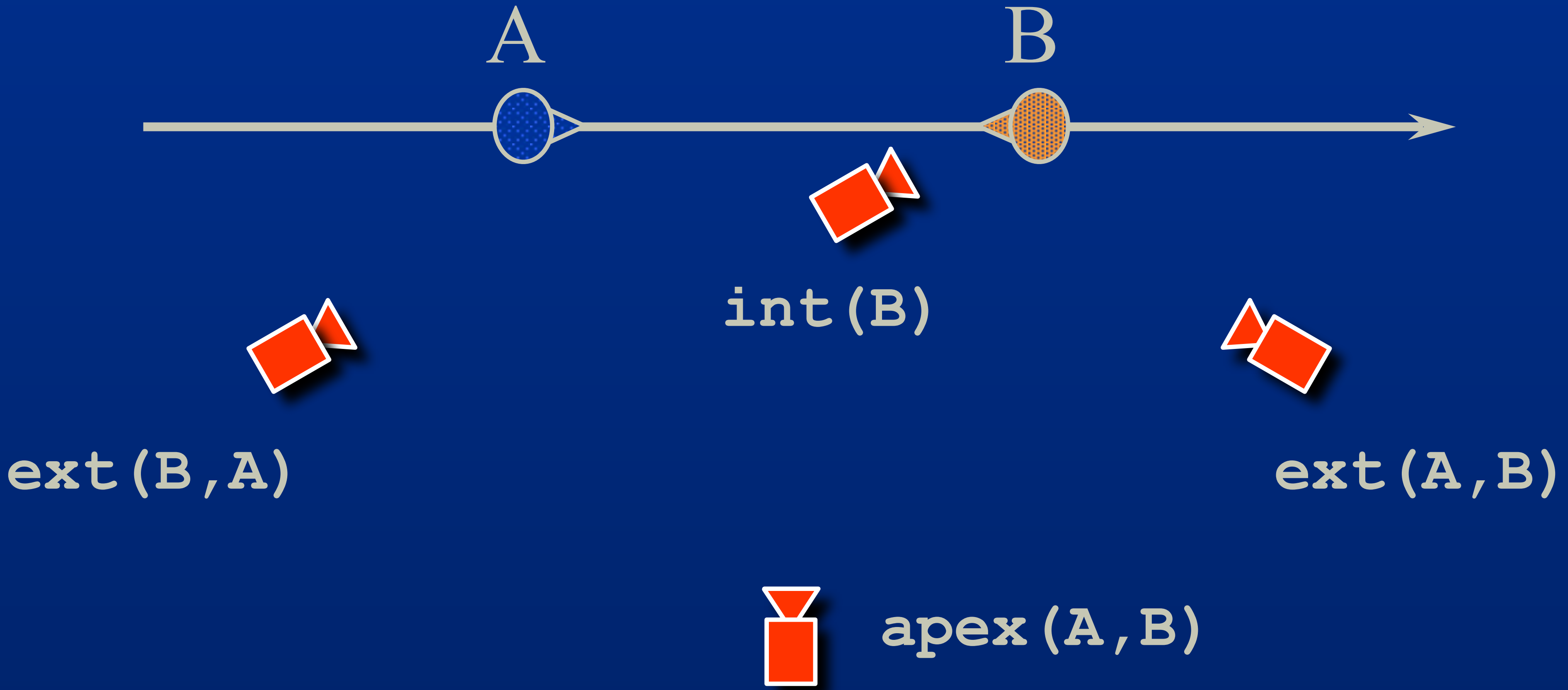
# Camera modules

- Geometric placement of specific cameras for each shot
- Choose the side of the line of interest
- Influence acting

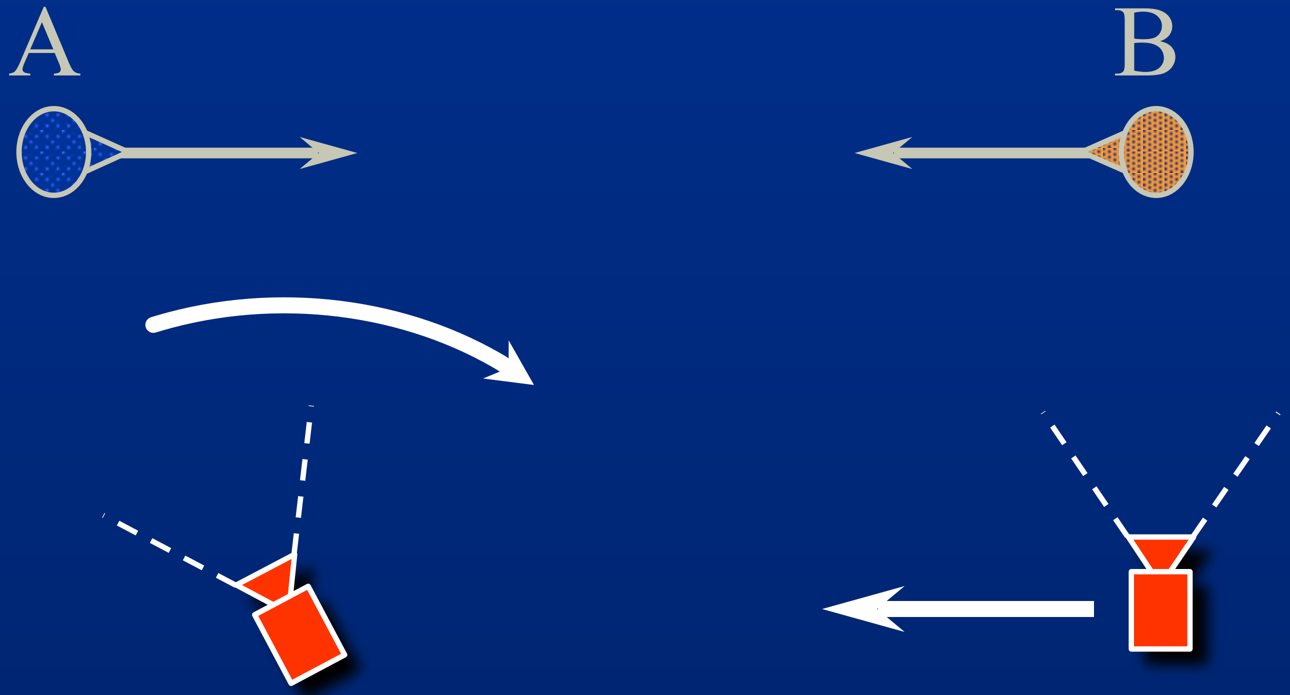
# Camera module: `ext(B,A)`



# Static camera modules



# Moving camera modules

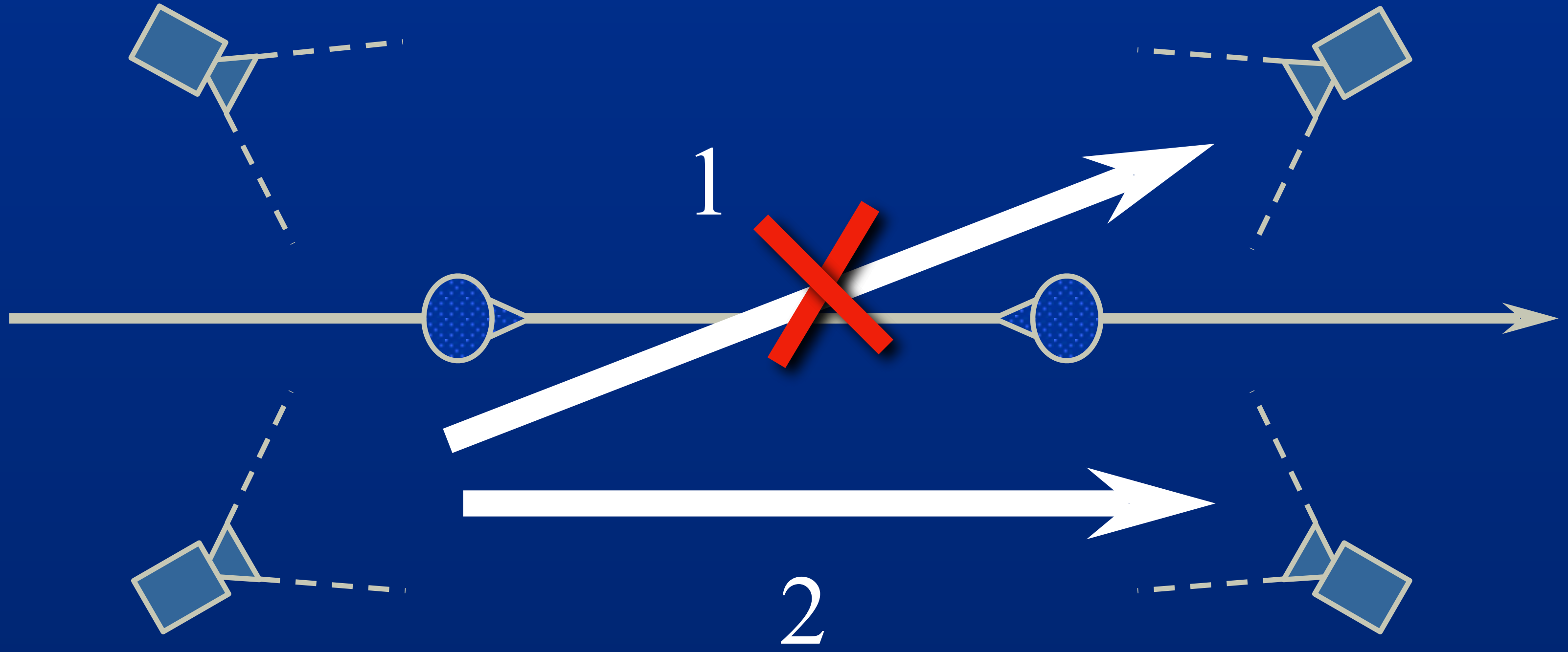


`pan(A)` `track(B)`

# Camera modules

- Geometric placement of specific cameras for each shot
- Choose the side of the line of interest
- Influence acting

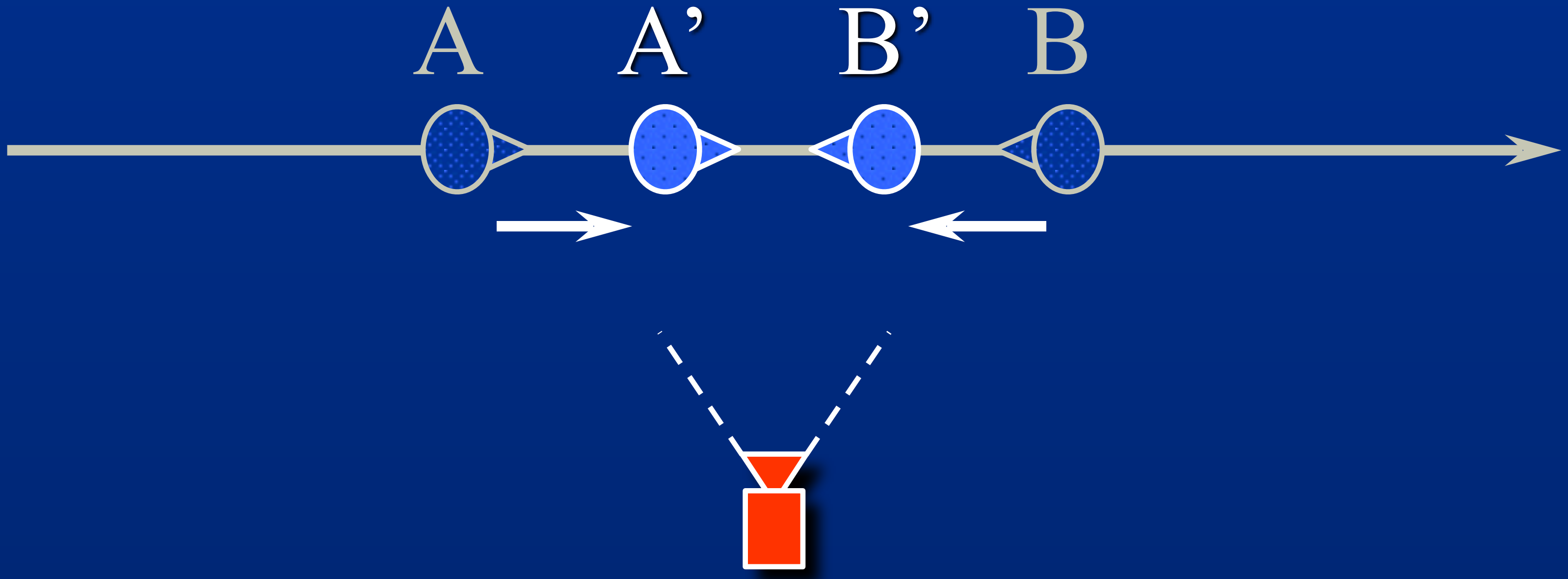
# Don’t cross the line of interest



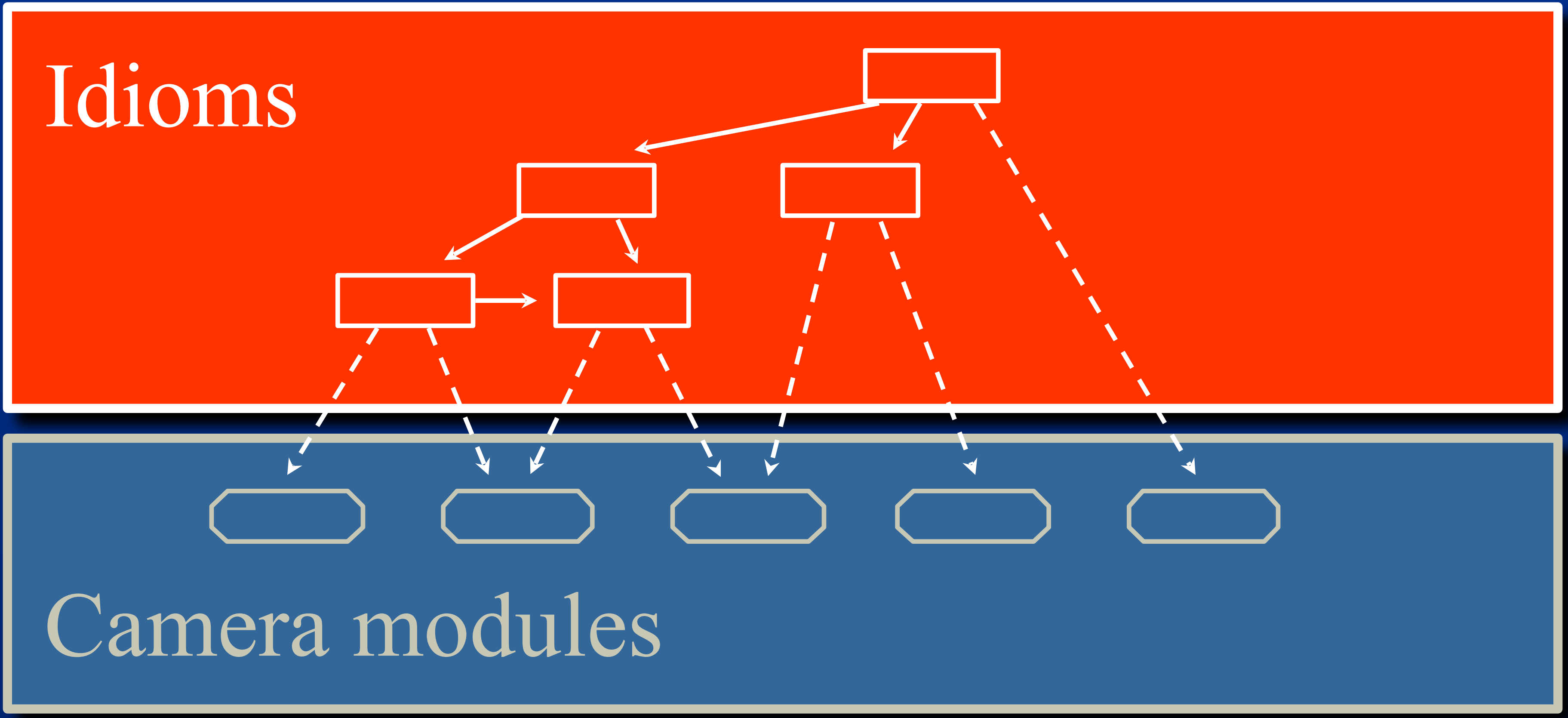
# Camera modules

- Geometric placement of specific cameras for each shot
- Choose the side of the line of interest
- Influence acting

# Acting hints



# VC architecture



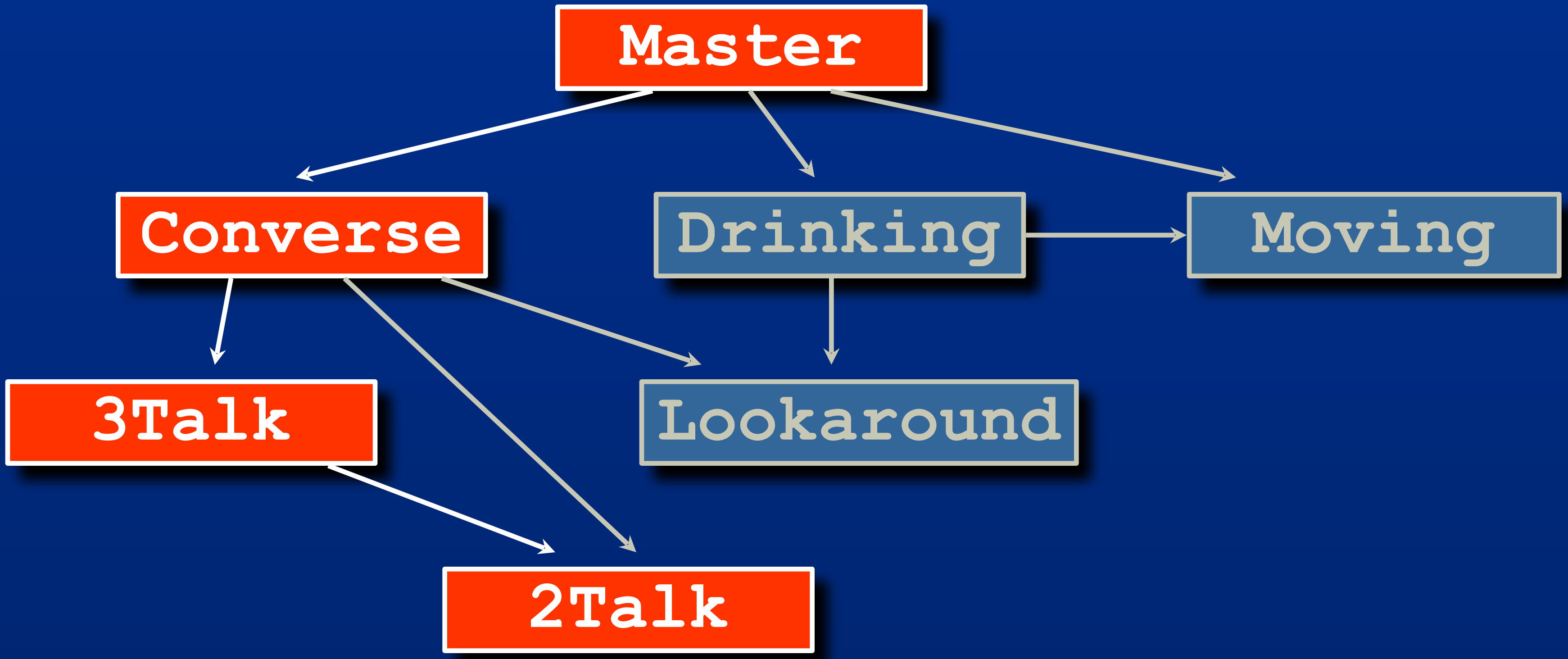
# Previous works

- Dave Christianson’s AAAI ’96 paper
  - Declarative camera control for automatic cinematography
  - An off-line algorithm
  - Need to know all events ahead
  - Use plan algorithms in AI

# Film idioms

- Capture a particular type of scene
- Register relevant events
- Select shot types
- Determine the transitions between shots

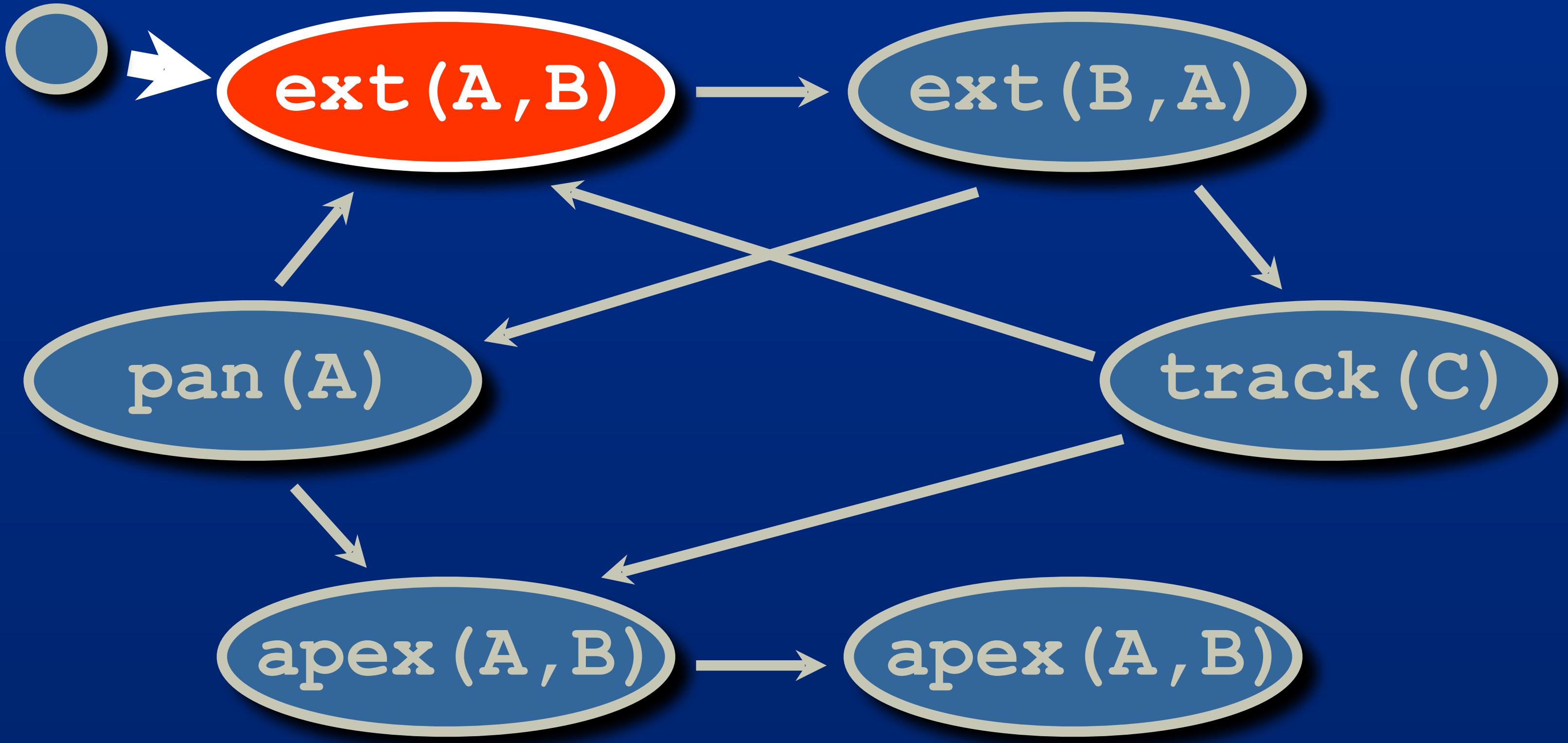
# Hierarchical idiom structure



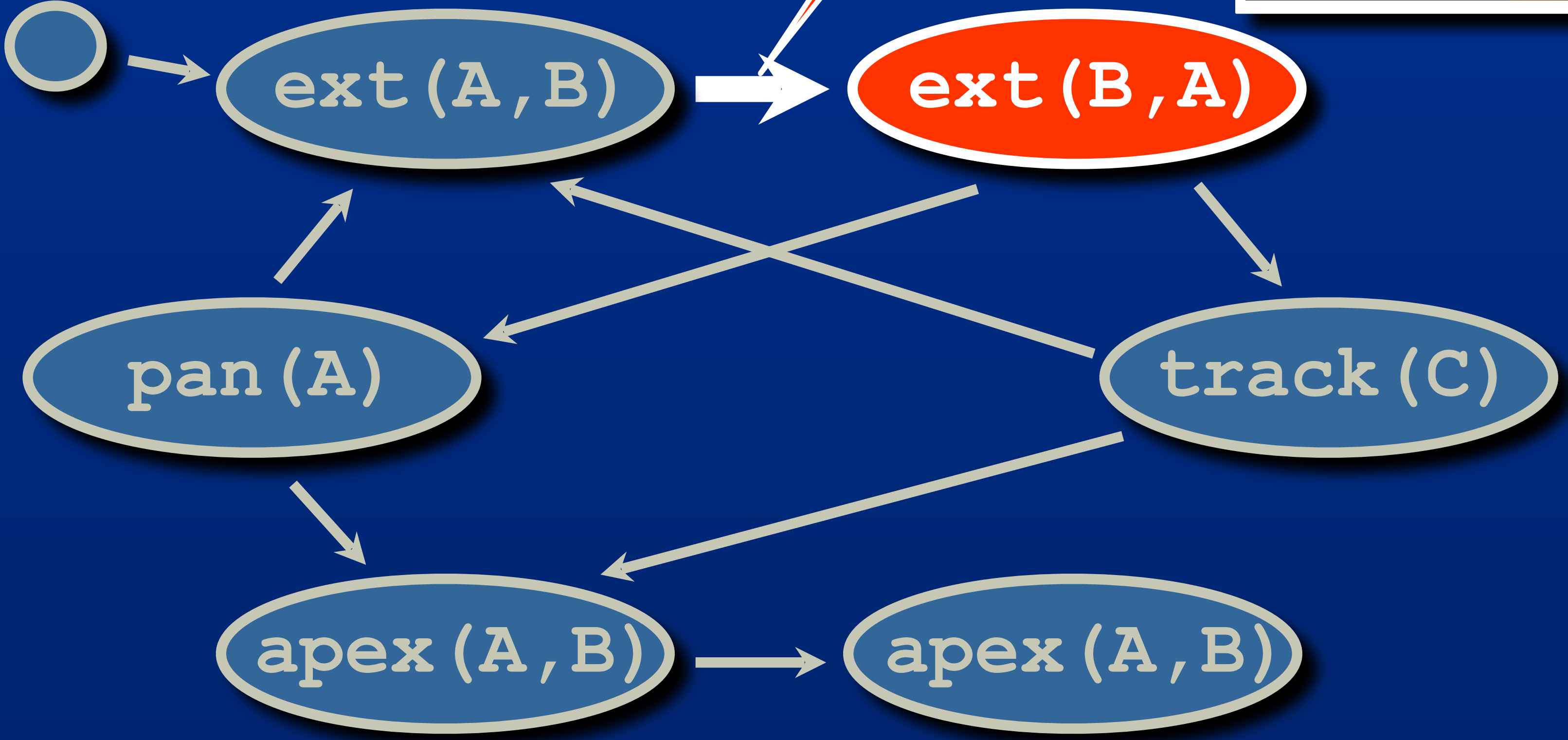
# Hierarchical idiom structure

- Avoids exponential growth of states
- Allows expertise to be reused
- Provides robustness

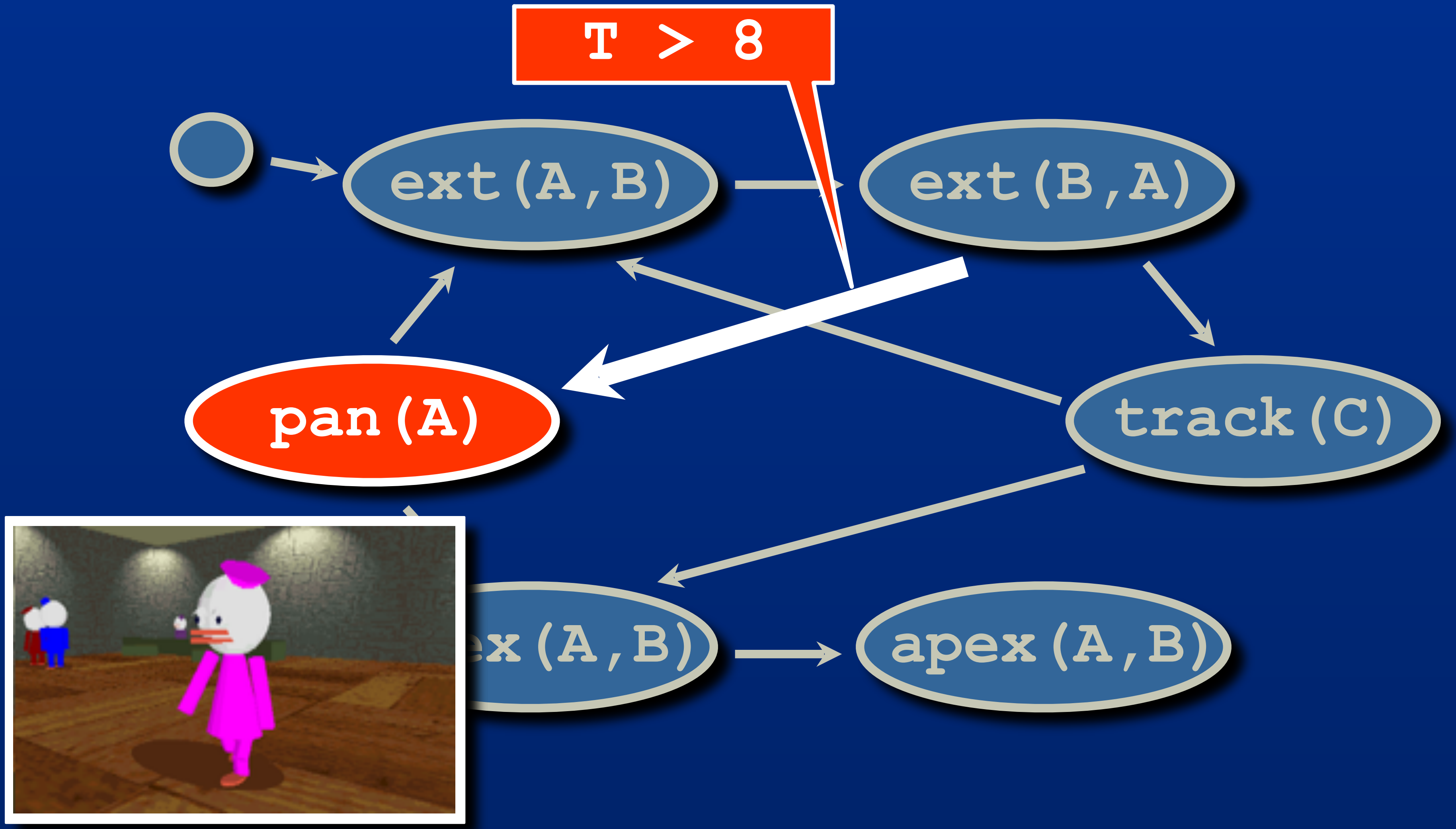
# Moving idiom



# Moving idiom



# Moving idiom



T > 8

ext(A,B)

ext(B,A)

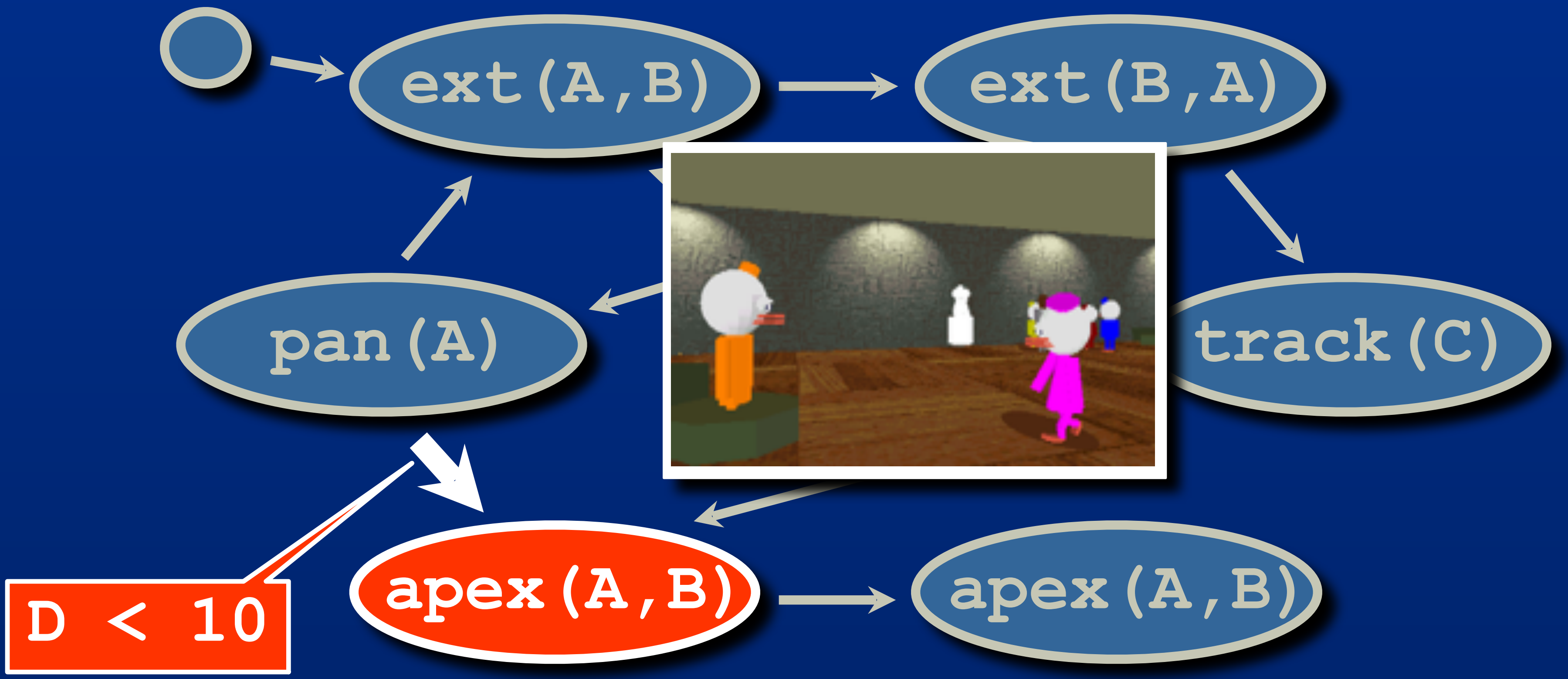
pan(A)

track(C)

ex(A,B)

apex(A,B)

# Moving idiom



ext(A,B)

ext(B,A)

pan(A)

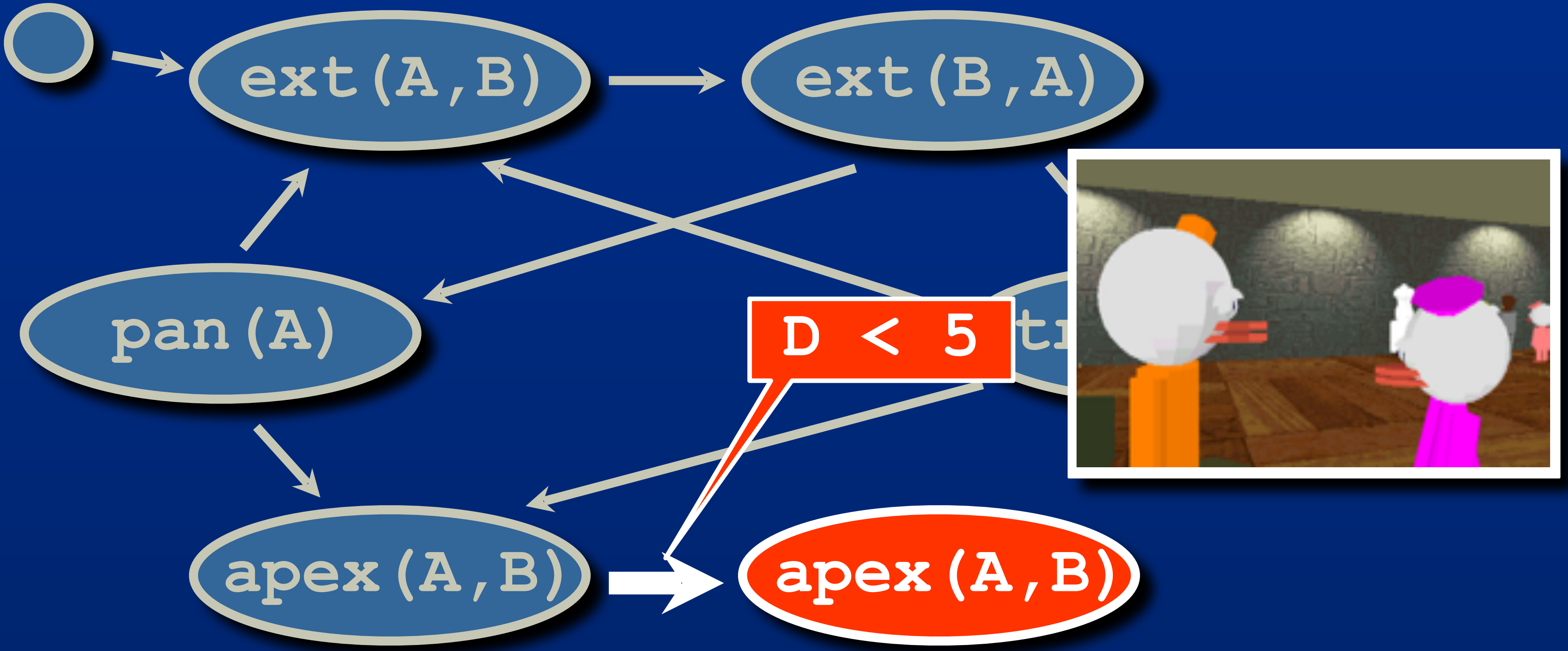
track(C)

D < 10

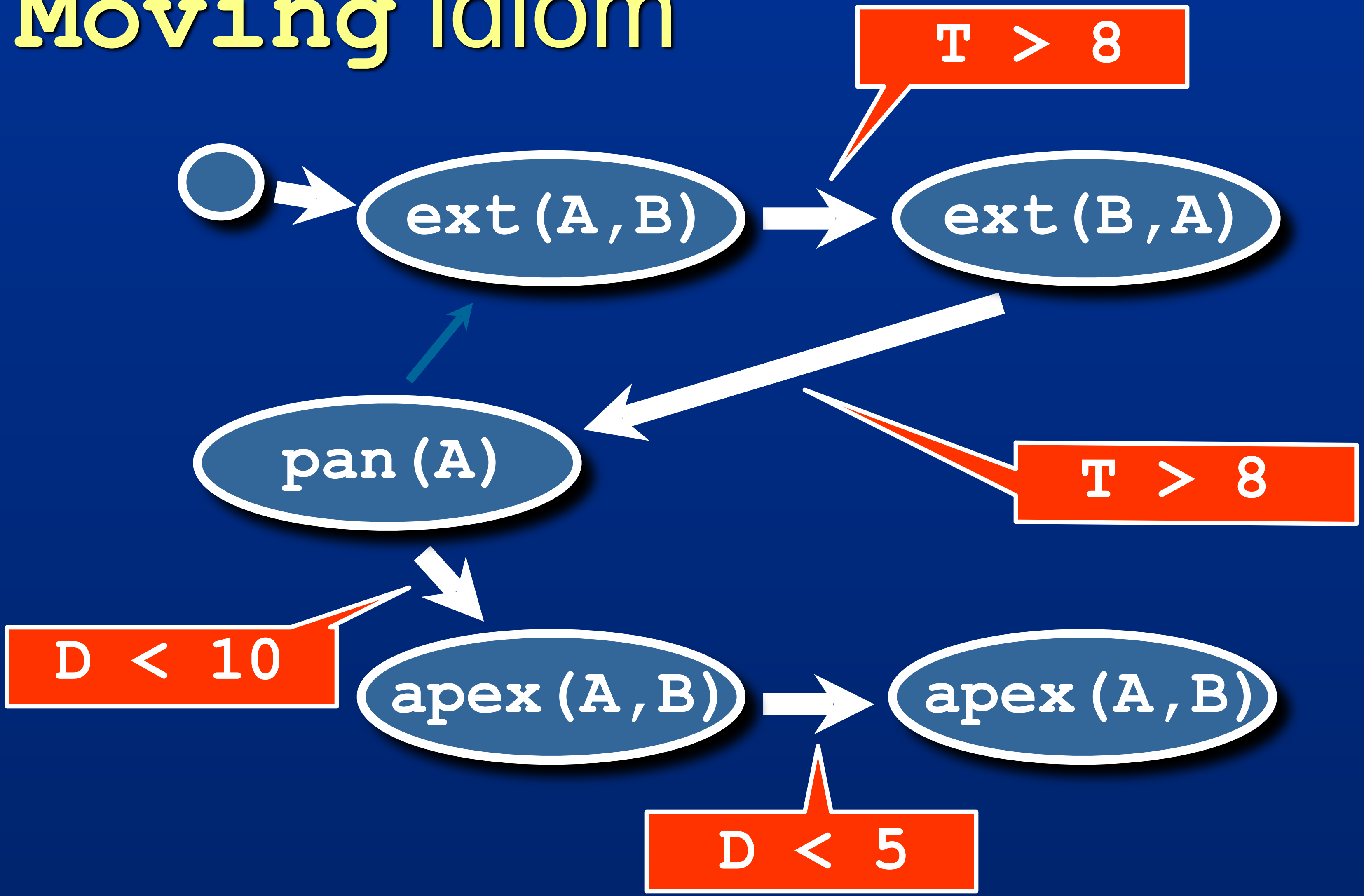
apex(A,B)

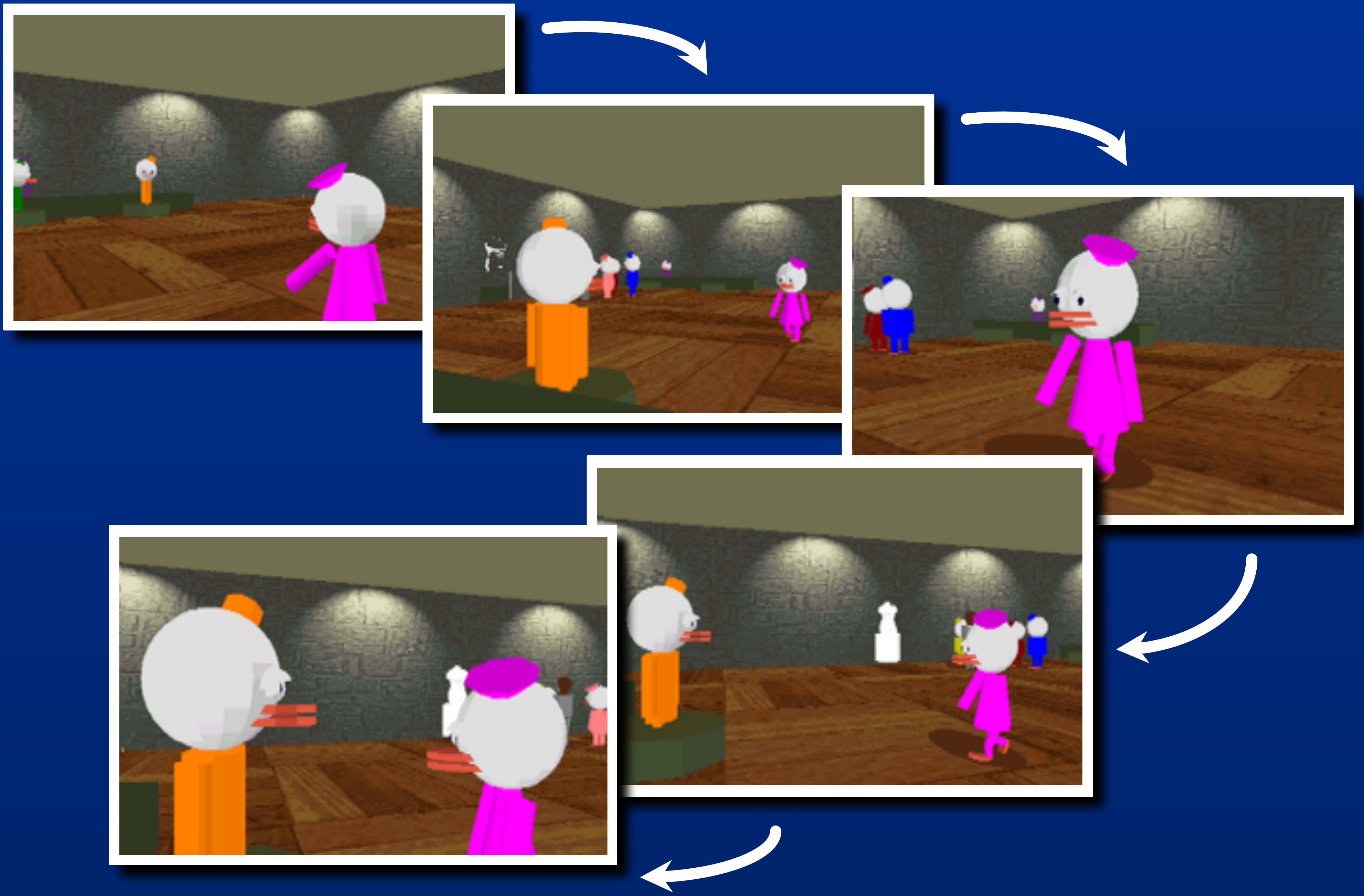
apex(A,B)

# Moving idiom

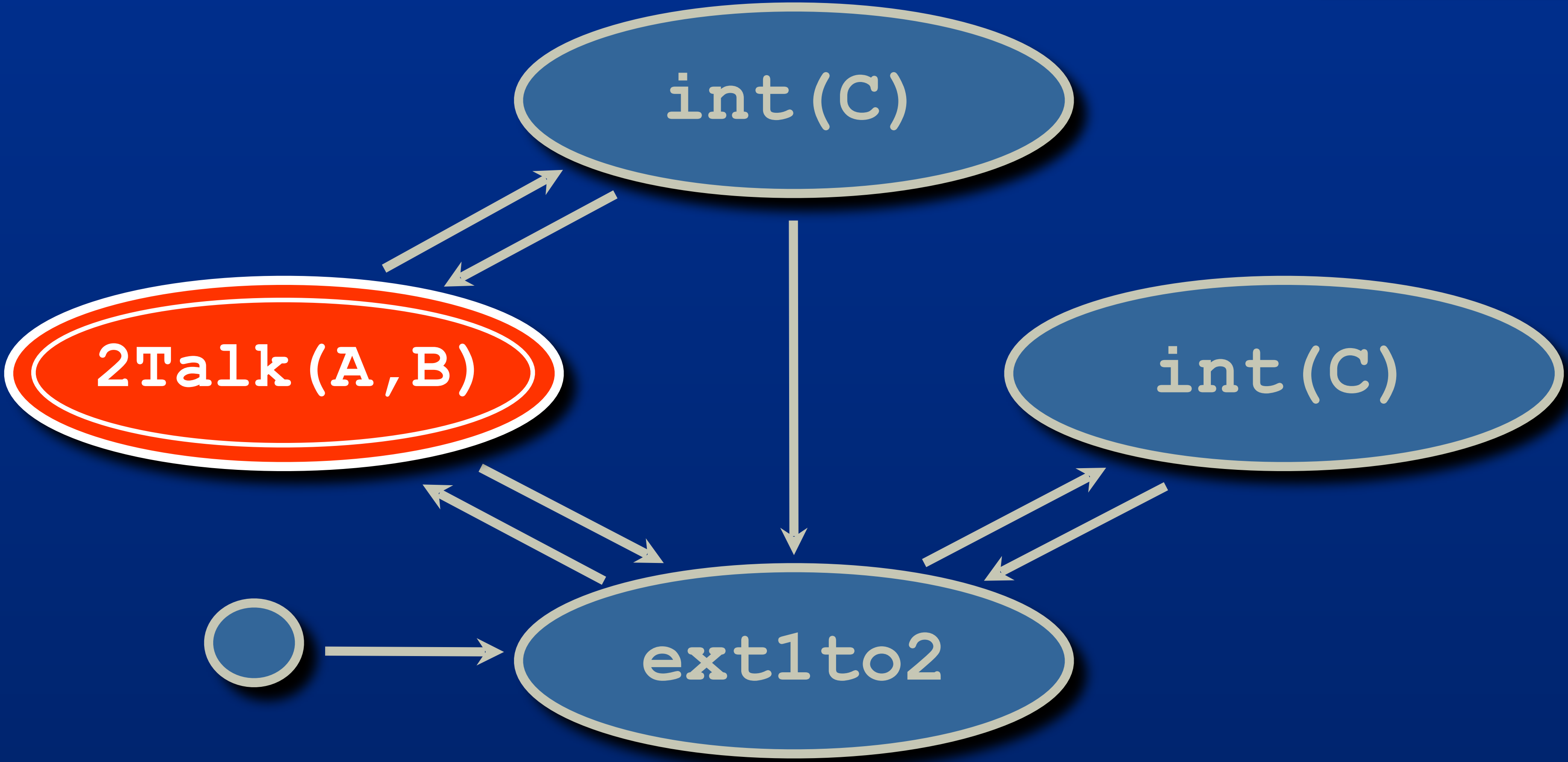


# Moving idiom

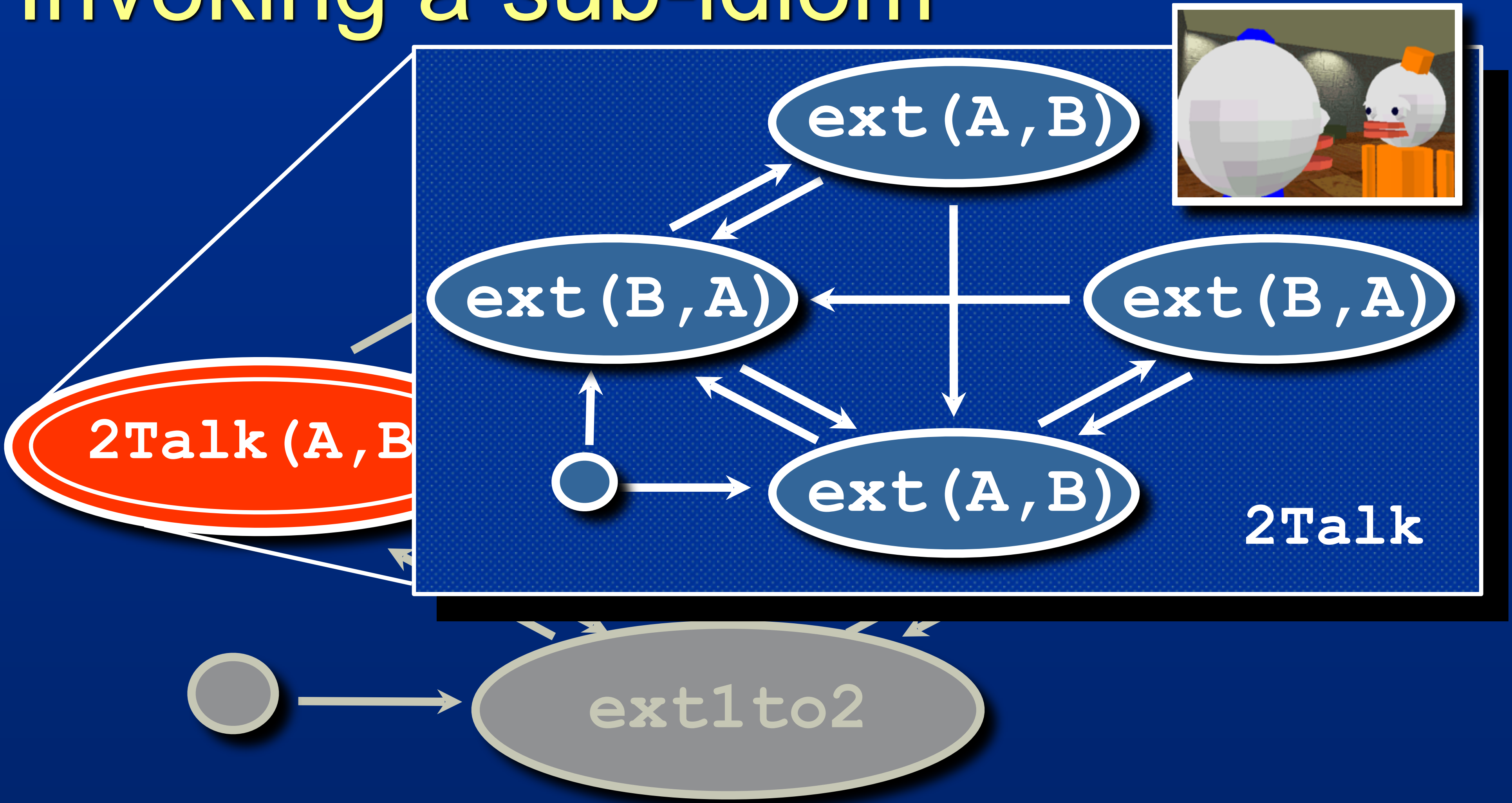




# 3Talk idiom



# Invoking a sub-idiom

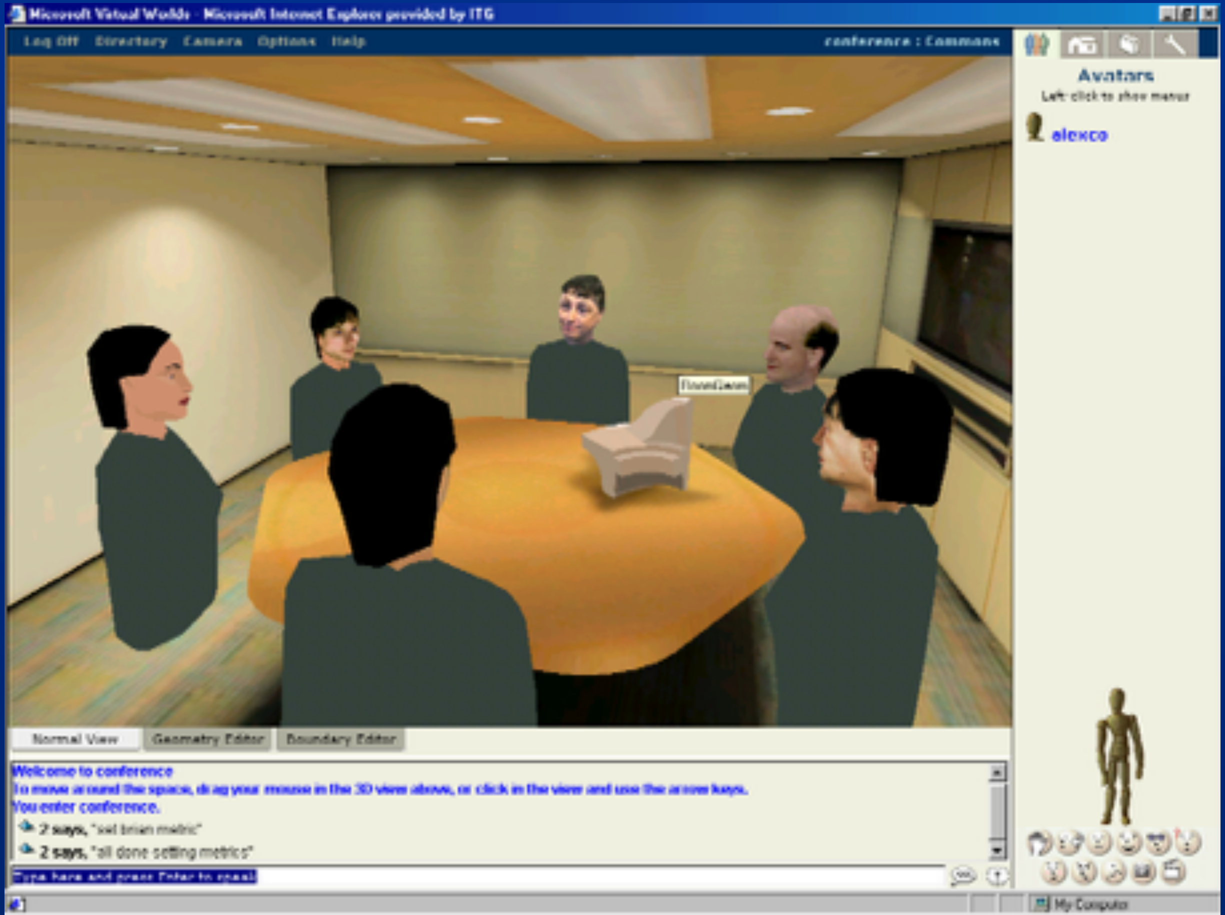


# Contributions

- Real-time camera control
  - lightweight
  - automatic camera placement
  - automatic shot transitions
- Do a reasonable job
  - not to replace human in computer animation

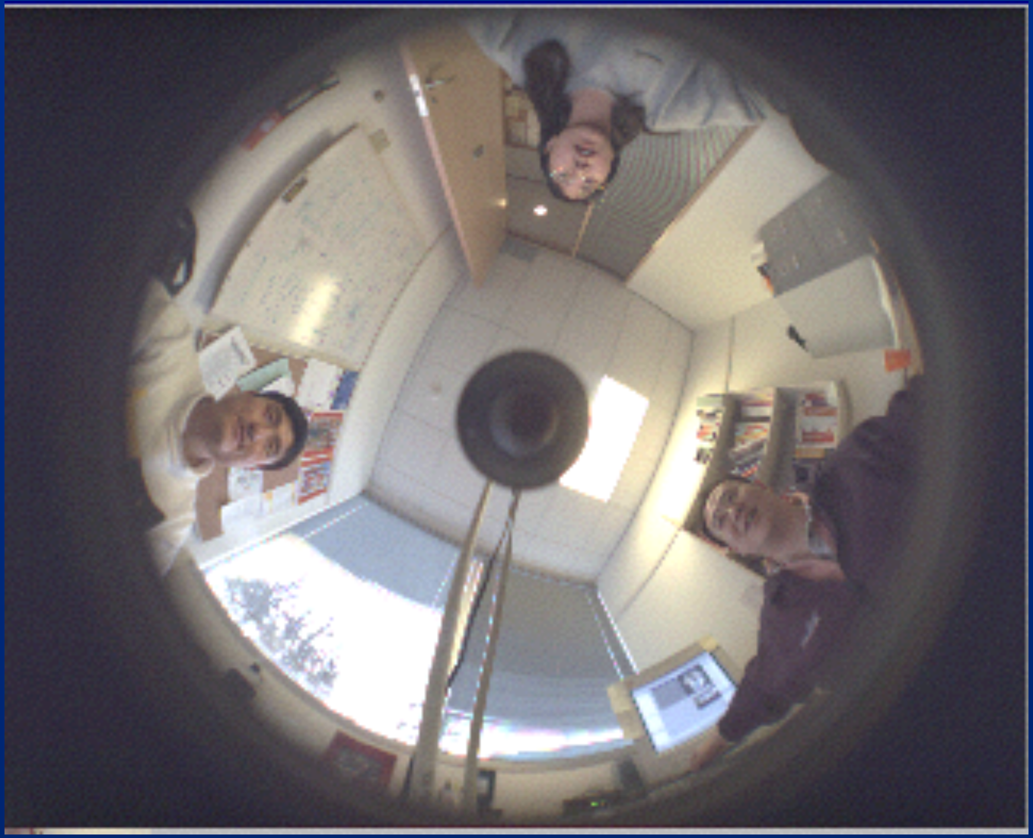
# Work in Progress

# Camera control for chat

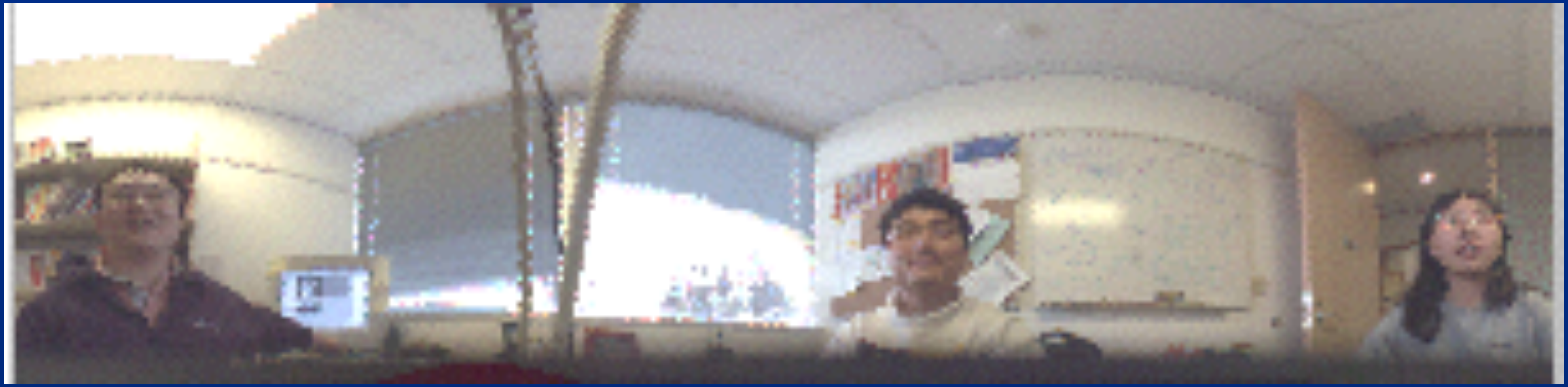


Welcome to conference
To move around the space, drag your mouse in the 3D view above, or click in the view and use the arrow keys.
You enter conference.
2 says, "set brian metric"
2 says, "all done setting metrics"

# Camera control for live meeting



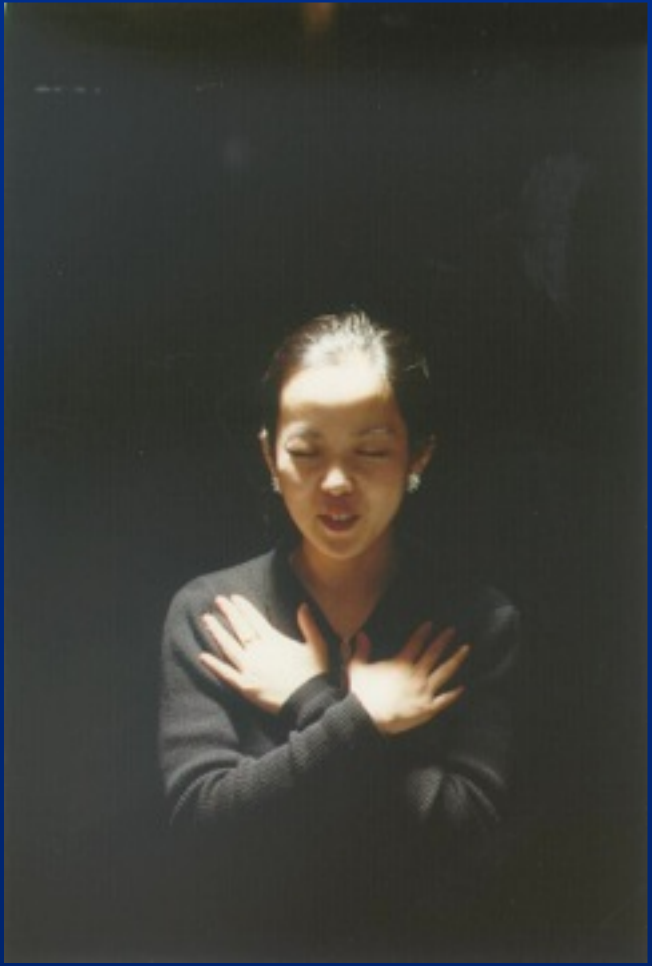
# Image after warping

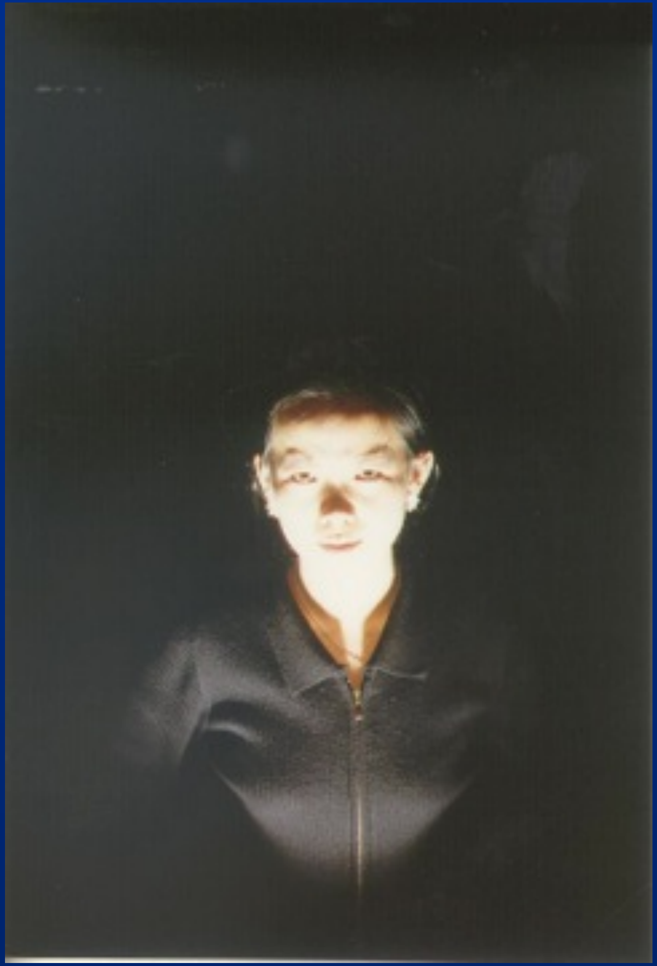


# Camera control for classroom

- Multiple cameras per classroom
  - Lecturer cameras
  - Audience camera
- Use audio and vision techniques to
  - Track the lecturer
  - Activate the audience camera

# Dramatic lighting





# Virtual Cinematography

## Theory and Practice for Automatic Real-Time Camera Control and Directing

Liwei He

Microsoft Research

http://research.microsoft.com/users/lhe