Algorithms and techniques for virtual camera control

Session 3: Interactive Camera Control

M. Christie, Univ. Rennes 1
C. Lino, Univ. Rennes 1
R. Ranon, Univ. Udine

When a camera becomes interactive…

…we need to understand:

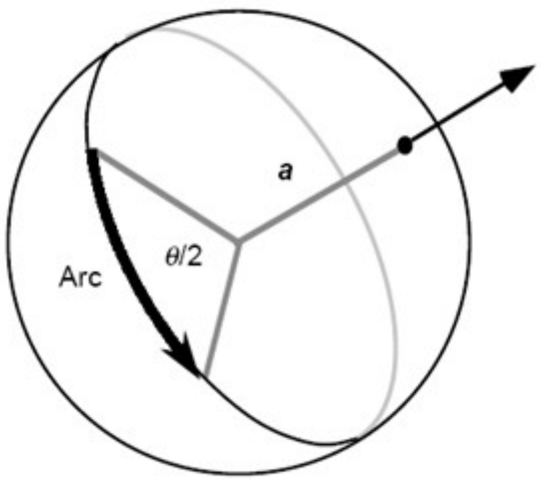
- the nature of the mapping between the user inputs and the camera parameters (internal constraints)

- the effect of other constraints on the camera parameters (i.e. external constraints such as visibility or surface of objects)

Interactive camera control

4 properties broadly characterize the space of interactive camera control techniques:

- degrees of freedom of the input device
  - low degree of freedom input devices (e.g. virtual arcball [Sho92], [CMS88])
  - 6 degree of freedom input devices (direct metaphors)
- directness of the mappings
  - control camera parameters, velocity, acceleration,…
- nature of the constraint on motion:
  - physical metaphors
  - geometrical
  - task
- world space vs. screen space based control



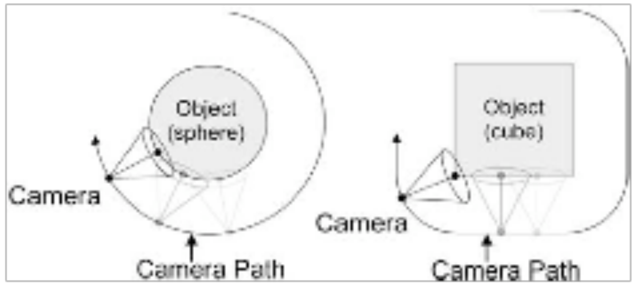
Enforcing usability

How? by reducing the dimensionality of the problem

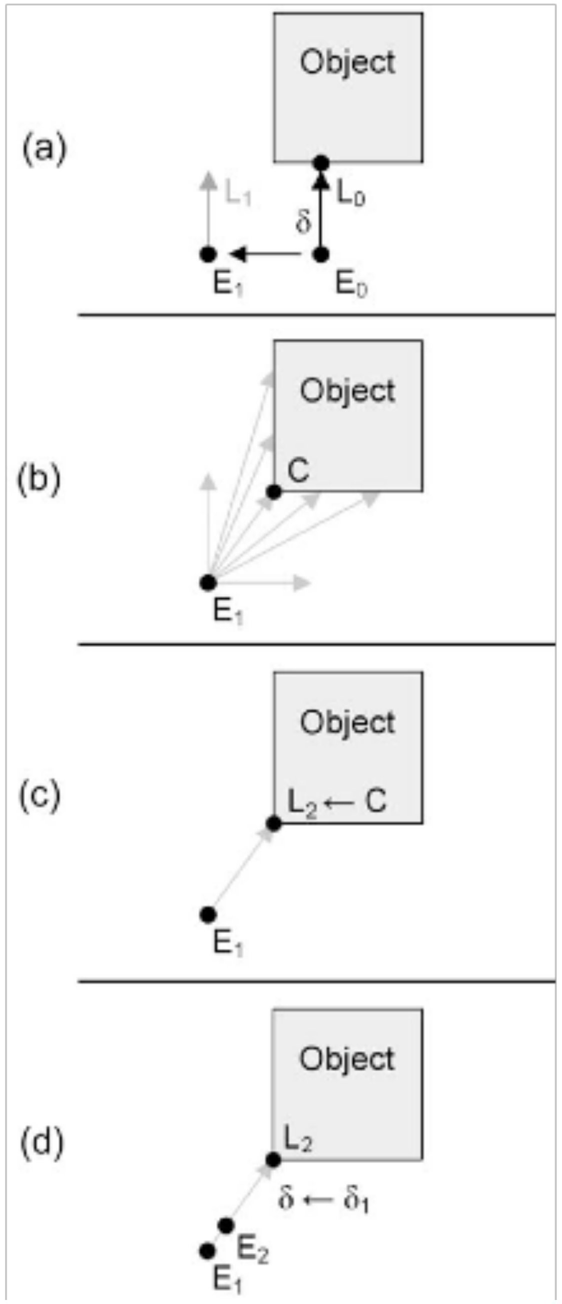
- Fixing camera parameters (e.g. roll parameter)
- Automatically computing camera parameters
  - Lookat of the camera fixed to a target
  - Adding physical constraints to the camera
- Constraining camera parameters to a sub-space of possible motions
- Exploiting alternative camera models

Constraints in proximal navigation

Khan et al [KKS*05] developed a "hovercam" metaphor for individual **object inspection**:

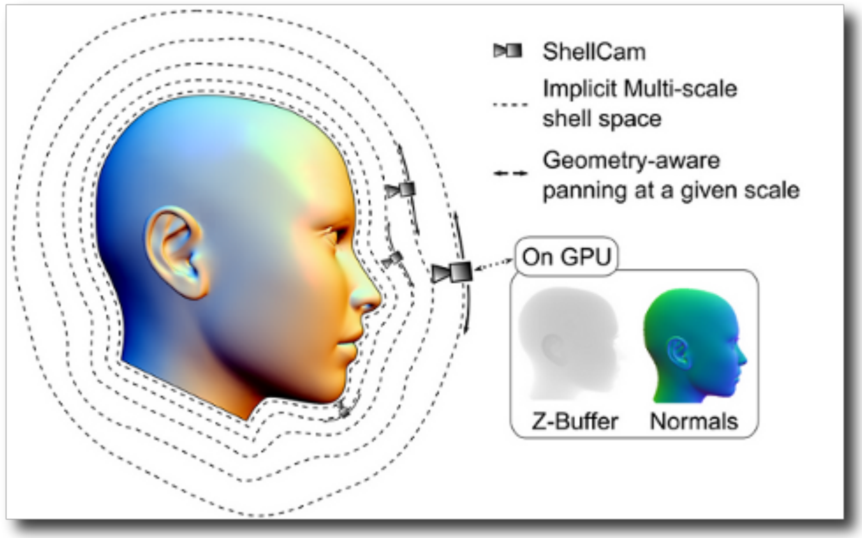


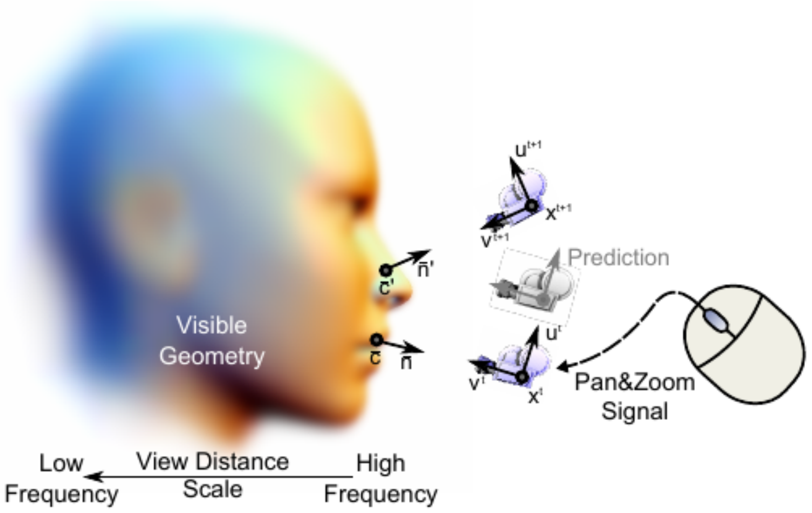
- apply user input to the eye point E0 (current camera position) and look-at point L0, to create E1 and L1;
- search for the closest point C on the object from the new eye position E1;
- turn the camera to look at C, and
- correct the distance δ1 to the object to match the original distance to the object δ to generate the final eye position E2
- clip the distance travelled



Constraints: Shellcam [Bbk14]

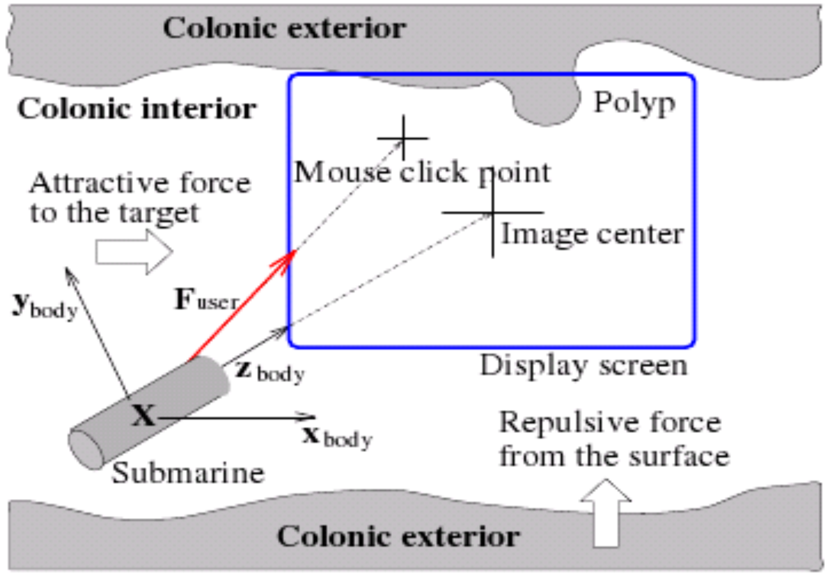
- Boubakeur extended the approach using a smooth motion subspace on arbitrary objects
- A scale-dependent offset shell is computed around the geometry
  - it provides tangent directions for pan/tilt camera motions
  - the zoom changes the offset shell
- The *shell* is a low frequency offset of the geometry





Environment-based control

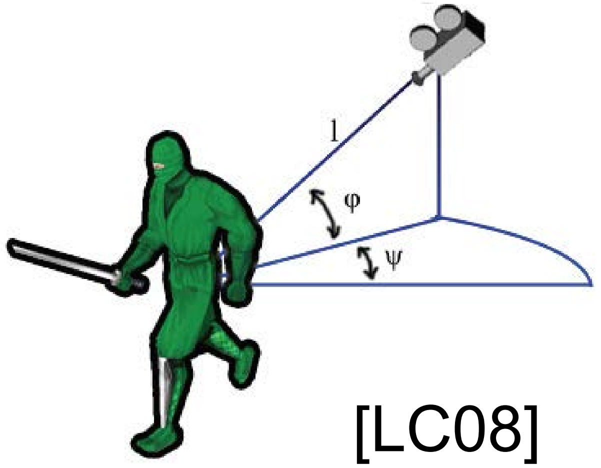
- methods to assist navigation/exploration are mostly based on motion planning techniques from the field of robotics:
  - e.g., potential fields and vector fields
- methods require significant pre-computations



example: application to virtual colonoscopy [HMK97]

Towards indirect interaction

- multiple approaches implement more elaborate interactions with the camera (*i.e.* from parameters manipulation to properties manipulation)
  - **through-the-lens techniques:**
    interaction is performed on the content of the screen
    (for specifying camera motions, or screen composition)
  - **reactive techniques:**
    control is operated over targets which indirectly control the camera motions
    (typically following avatars [LC08,HHS01])

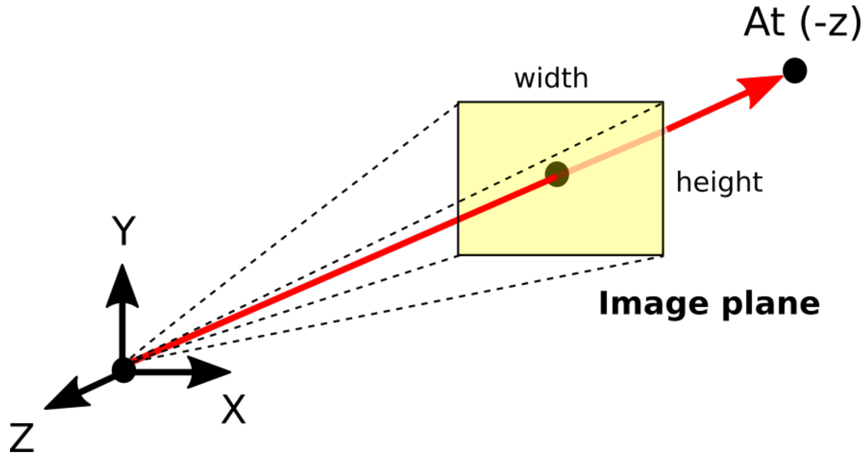


[LC08]

"Through the lens" control

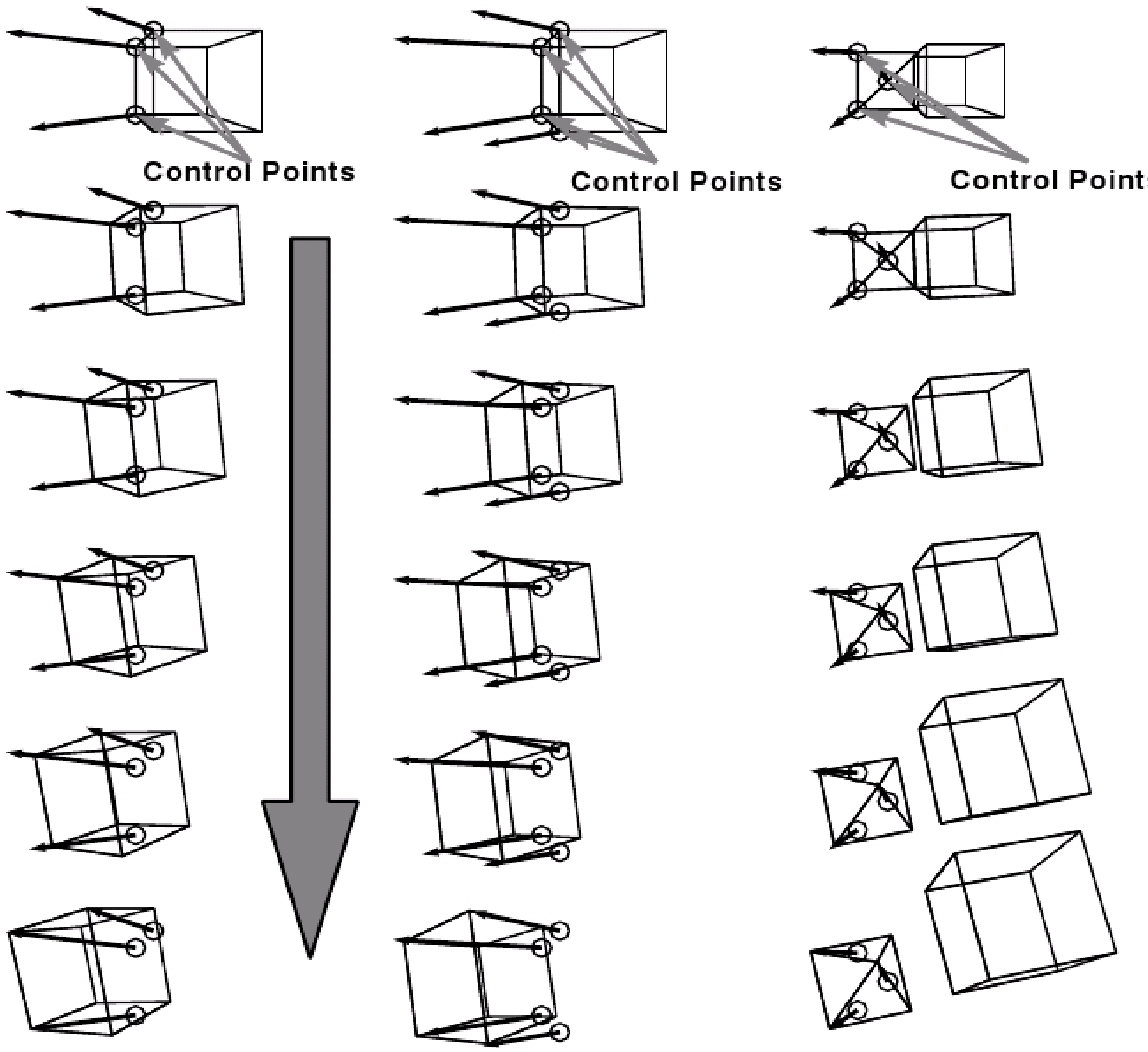
- indicate desired positions of objects on the screen: **Through-the-lens camera control** (Gleicher & Witkin [GW92])
- difference between the actual screen locations and the desired locations indicated by the user is treated as a velocity
- **relationship** between
  - the **velocity** $(\dot{\mathbf{h}})$ **of m displaced points** on the screen
  - and the **velocity** $(\dot{\mathbf{q}})$ **of camera parameters**
- **expressed with the Jacobian** J that represents the perspective transformation:

$$\dot{\mathbf{h}} = J\dot{\mathbf{q}}$$



In the camera space (perspective projection)

"Through the lens" control



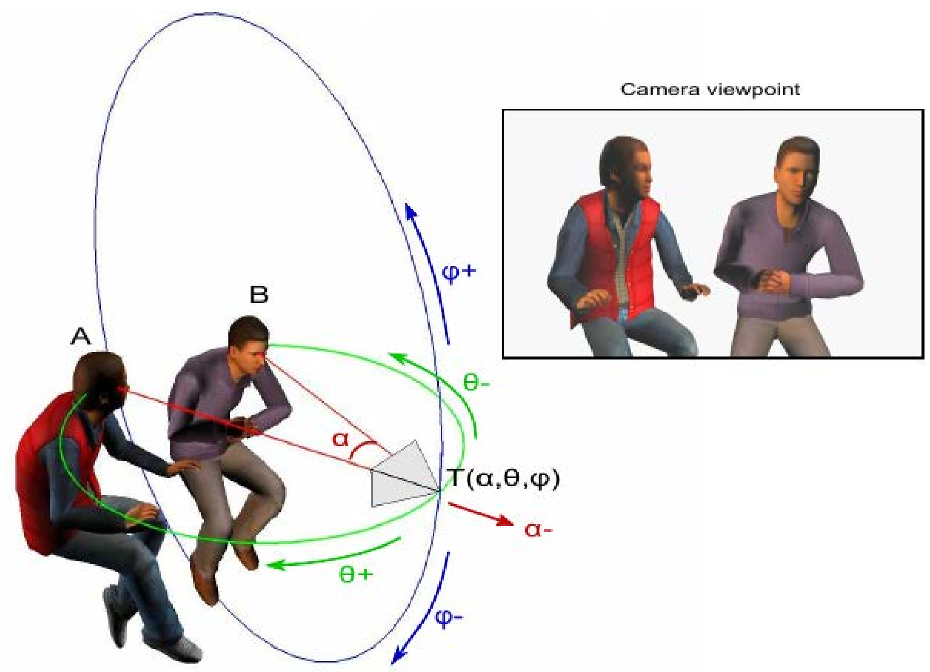
“Through the lens” control

- The Jacobian is generally non-square (m x n)
  - m: dofs of the camera
  - n: parameters of the visual features in 2D
- Invert of the Jacobian?
  - compute its pseudo inverse with a Singular Value Decomposition (SVD)
    - $O(mn^2)$ complexity
  - or use some optimization process (e.g. [GW92])
- Visibility needs to be handled separately…
  - by excluding some areas from the camera *dofs*

Though the lens control with The Toric Space

- Introducing a novel 3DOF representation of a camera [LC15]
  - dedicated to viewpoint manipulation of two targets
- Three parameters to control the position:
  - α : angle between targets A and B
  - θ : horizontal angle
  - φ : vertical angle
  - the framing of the two targets is implicitly defined in the model

(Unity and C++ code available: ToricCam)

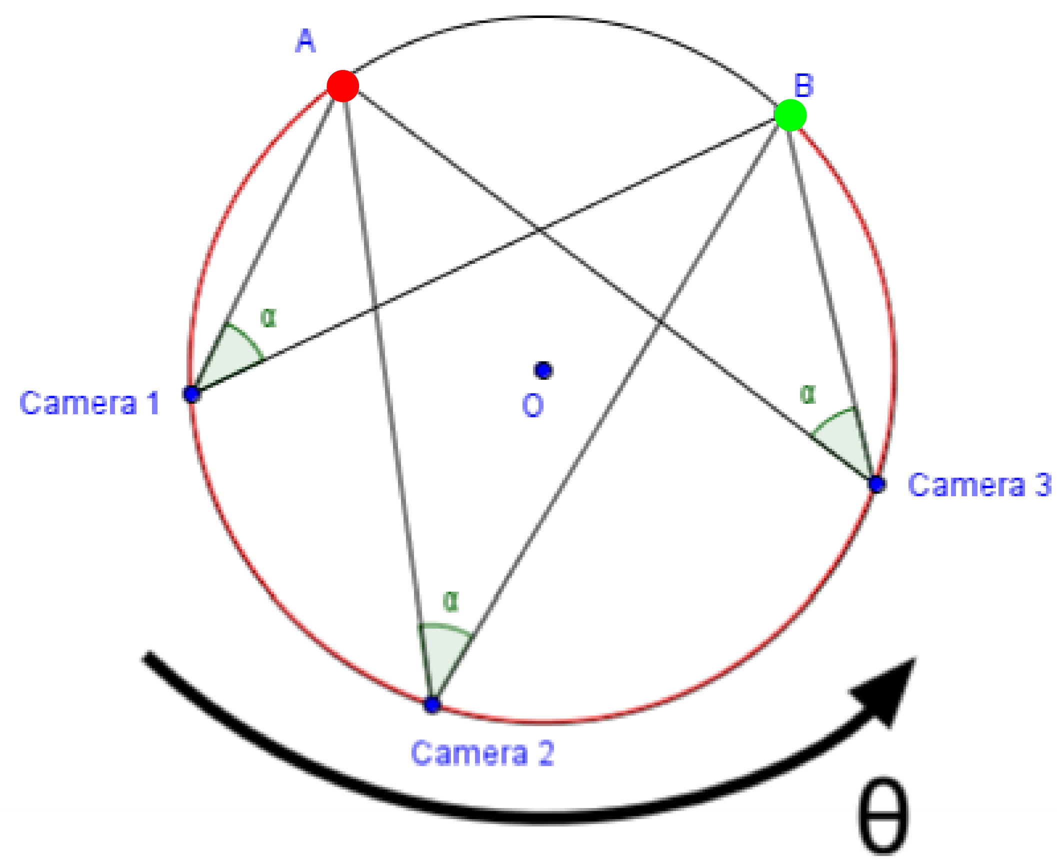


Composition : intuition (2D environment)

Desired on-screen
Composition
(1D)

Camera: C
$\alpha = (CB, CA)$

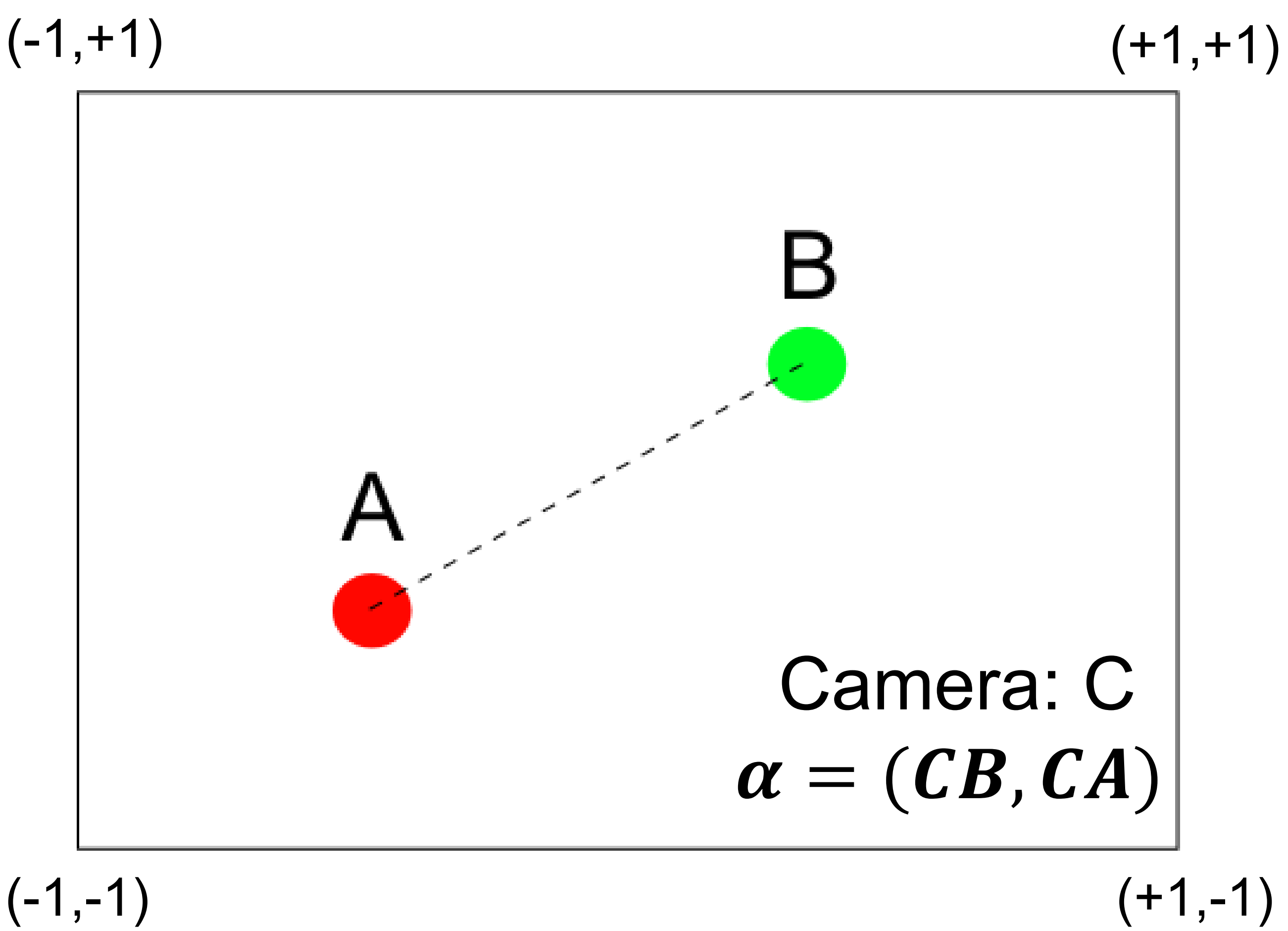
Solution = 1-parametric
manifold (θ)



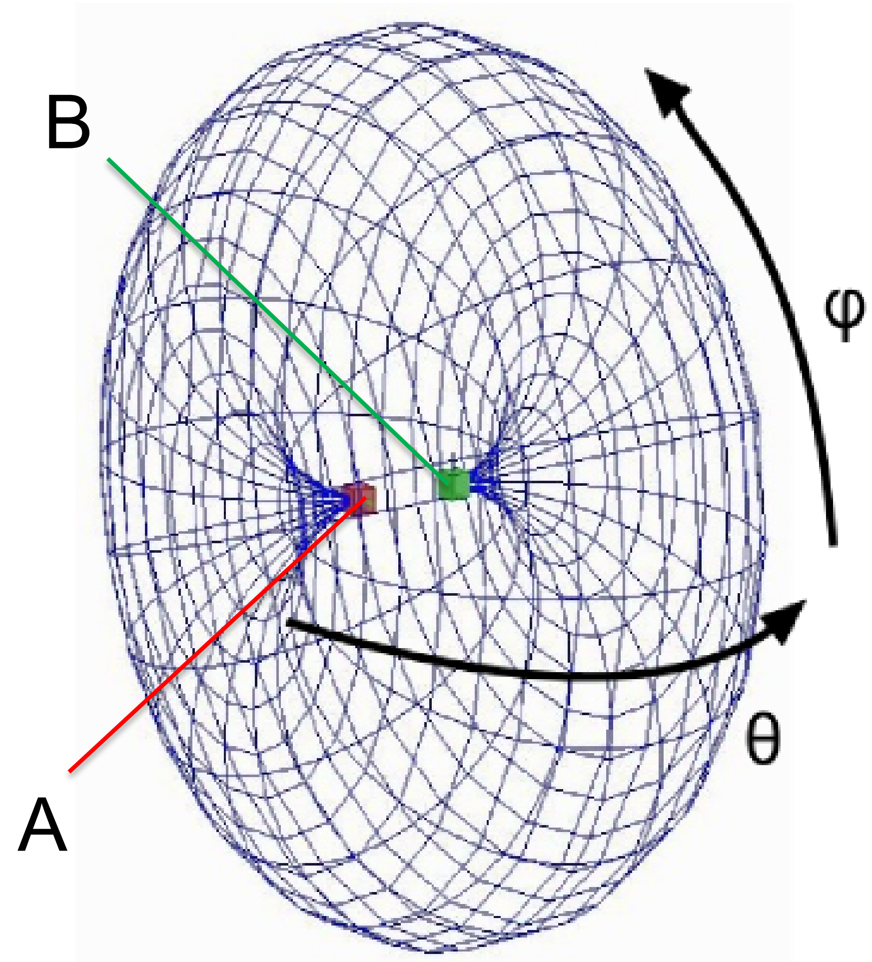
Any configuration c(θ) satisfies the 1D composition

Composition: 3D environment

Desired on-screen
Composition
(2D)



Solution = 2D manifold
surface (θ,φ)
(subset of a spindle torus)



Any configuration c(θ,φ) satisfies the 2D composition

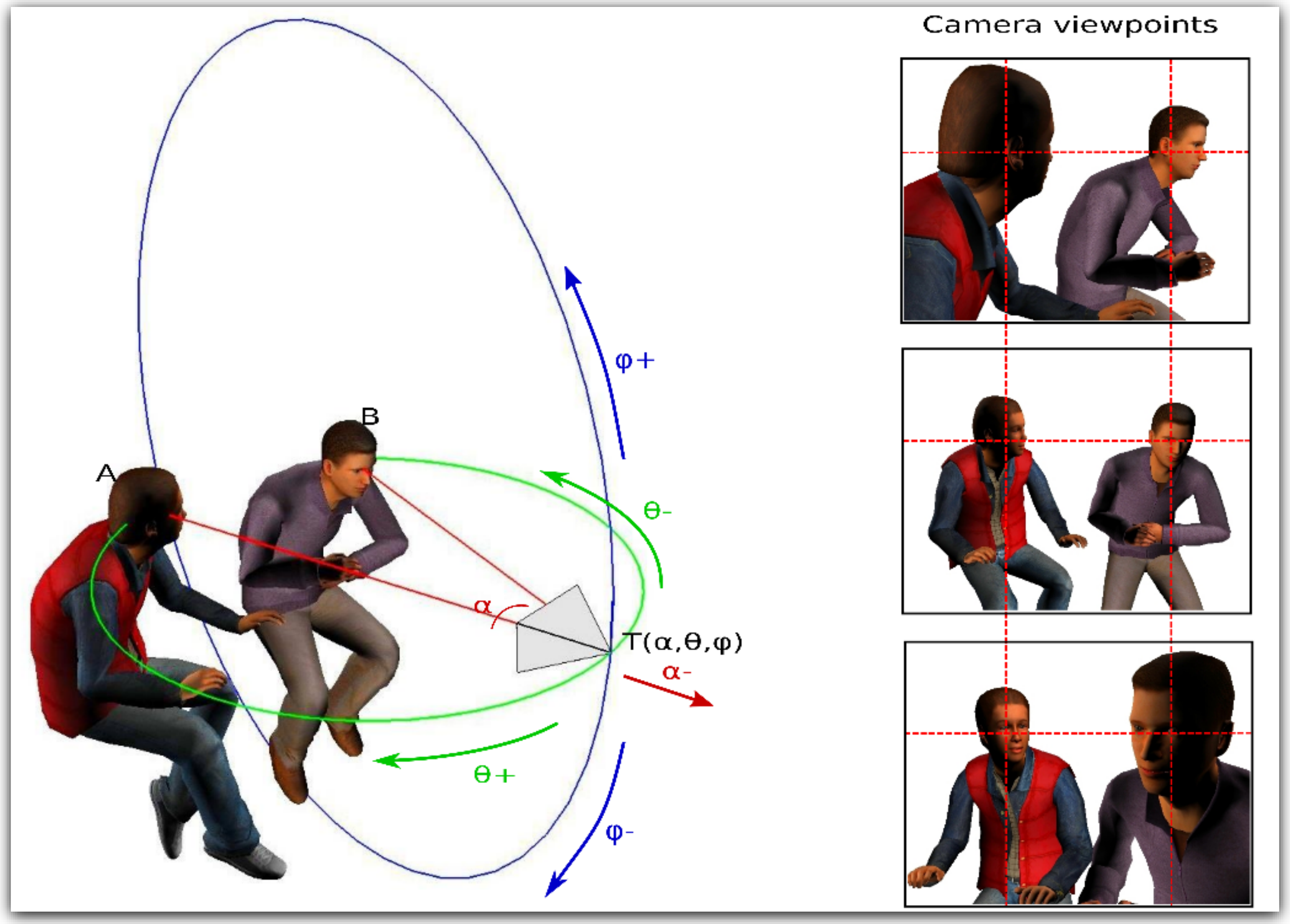
Extension : 3D Toric space

- More evolved problems:

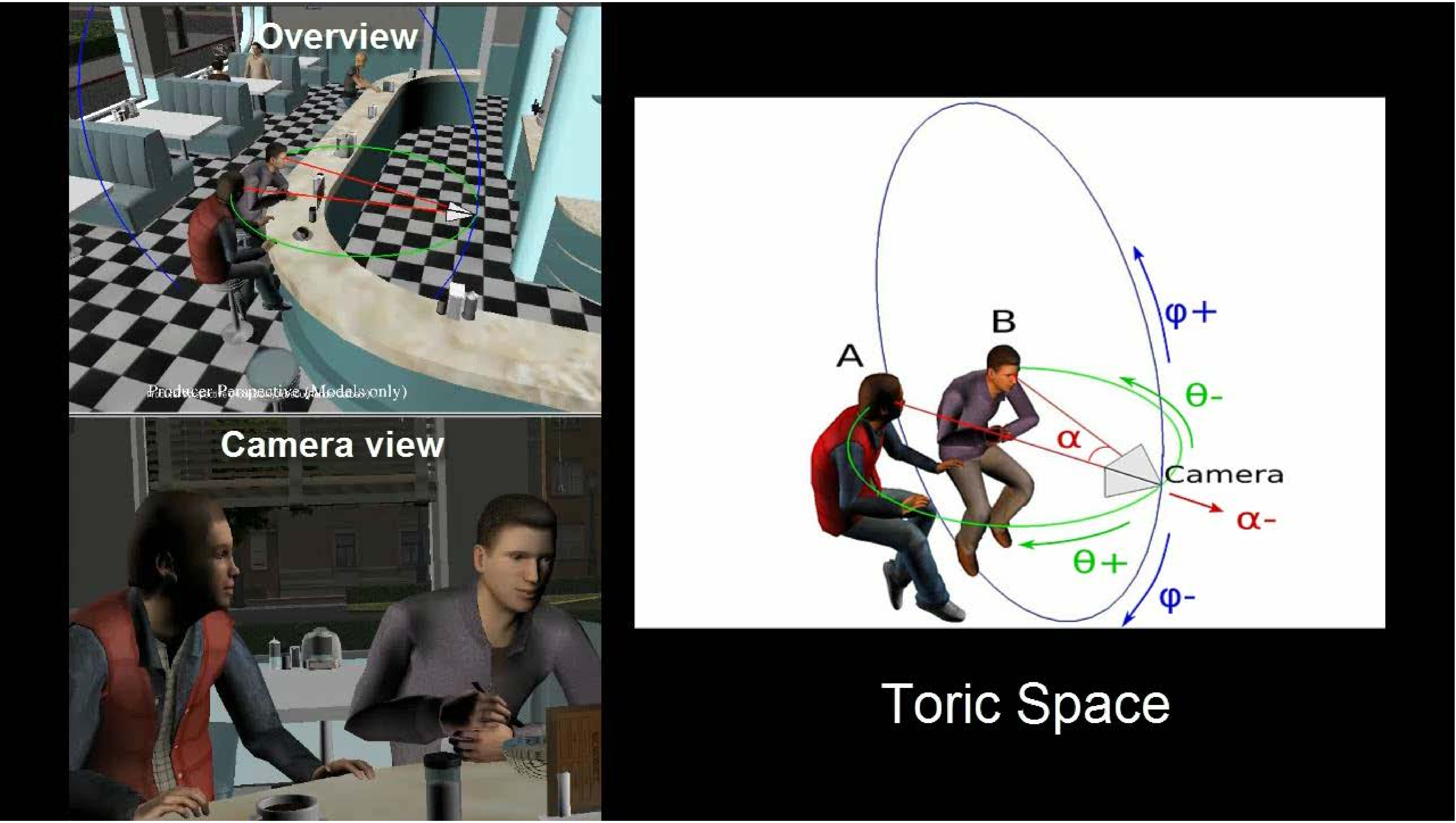
⇒ **relax the positioning constraint**

- Generalized model of camera:
  - **3-parametric space** (α, θ, φ)

- Defines the **range of all possible manifolds** around two targets



(Algebraically) casts 7D camera problems to 3D

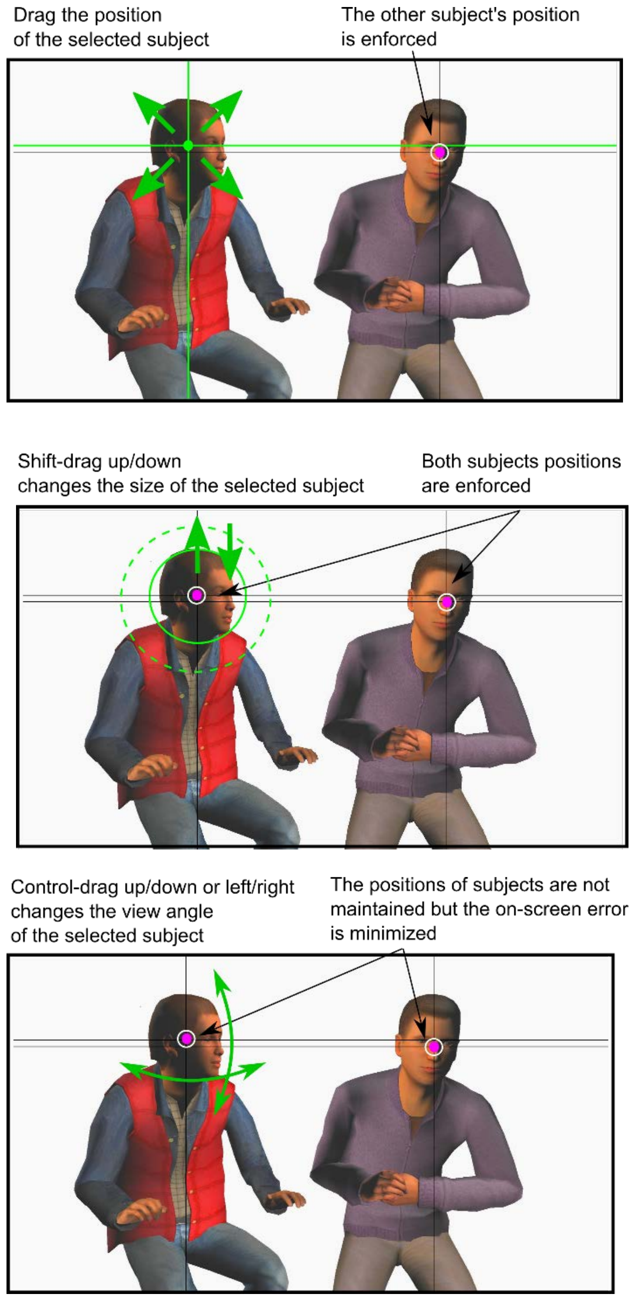


Video : https://www.youtube.com/watch?v=N-hEPkvGSf4

Manipulations in the Toric Space

Principle:

- *Manipulation of one target:*
  - *while the other is constrained in the screen-space*
  - *and roll is constrained to 0 (or a fixed value)*
- *Interactions:*
  - *change on-screen positions, distances, and vantage angles*
  - *example for on-screen positions:*
    - *we search for a position on the manifold surface where roll is null and minimizes the change in on-screen position*



$$\min_{(\theta,\varphi)} \left(p_A - p'_A\right)^2 + \left(p_B - p'_B\right)^2$$

Demonstration

Video : https://www.youtube.com/watch?v=3kFAIaihlX8

Bibliography

- [XH98] D. Xiao and R. Hubbold, *Navigation Guided by Artificial Force Fields*, Proceedings of ACM CHI 98, 1998
- [YCL08] Y. Tao, M. Christie, and X. Li, Through-the-lens Scene Design, Proc. of 8th International Conference on Smartgraphics, 2008
- [MC02] E. Marchand and N. Courty, *Controlling a camera in a virtual environment*. The Visual Computer Journal, 18(1):1-19, February 2002
- [Sho92] Shoemake K.: *Arcball: a user interface for specifying three-dimensional orientation using a mouse.* In Proc. of Graphics Interface, 1992.
- [KKS 05] Khan A., Komalo B., Stam J., Fitzmaurice G., Kurtenback G.: *Hovercam: interactive 3d navigation for proximal object inspection.* In Proc. of the 2005 symposium on Interactive 3D graphics, 2005.
- [BKF∗ 02] Burtnyk N., Khan A., Fitzmaurice G., Balakrishnan R., Kurtenbach G.: *Stylecam: interactive stylized 3d navigation using integrated spatial & temporal controls.* In Proc. of the ACM symposium on User interface software and technology, 2002.
- [HW97] Hanson A., Wernert E.: *Constrained 3d navigation with 2d controllers*. In Proc. of the IEEE Visualization Conference, 1997.
- [HMK∗ 97] Hong L., Muraki S., Kaufman A., Bartz D., He T.: *Virtual voyage: interactive navigation in the human colon*. In Proceedings of the Siggraph Conference, 1997.
- [Bbk15] Boubekeur, T. ShellCam: Interactive Geometry-Aware Virtual Camera Control IEEE International Conference on Image Processing 2014
- [LC15] Lino C. and Christie, M. "*Intuitive and Efficient Camera Control with the Toric Space*" Proceedings of SIGGRAPH 2015.