Through the Lens of 25 Years: Through-the-Lens Camera Control *Revisited*

Michael Gleicher

Department of Computer Sciences

University of Wisconsin Madison

Caveat

I was asked to talk about old stuff

It seems arrogant to say something you did was really important or inspiring

I'll let you judge

It continues to inspire ***me***

I'll mention 3 unpublished papers

(2 to appear, 1 to be written)

Through-the-Lens Camera Control

Michael Gleicher and Andrew Witkin
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA
{gleicher|witkin}@cs.cmu.edu

Abstract

In this paper we introduce *through-the-lens camera control,* a body of techniques that permit a user to manipulate a virtual camera by controlling and constraining features in the image seen through its lens. Rather than solving for camera parameters directly, constrained optimization is used to compute their time derivatives based on desired changes in user-defined controls. This effectively permits new controls to be defined independent of the underlying parameterization. The controls can also serve as constraints, maintaining their values as others are changed. We describe the techniques in general and work through a detailed example of a specific camera model. Our implementation demonstrates a gallery of useful controls and constraints and provides some examples of how these may be used in composing images and animations.

Keywords: camera control, constrained optimization, interaction techniques

1 Introduction

Camera placement and control play an important role in image composition and computer animation. Consequently, considerable effort has been devoted to the development of computer graphics camera models. Most camera formulations are built on a common underlying model for perspective projection under which any 3-D view is fully specified by giving the center of projection, the view plane, and the clipping volume. Within this framework, camera models differ in the way the view specification is parameterized. Not all formulations are equivalent—some allow arbitrary viewing geometries, while others impose restrictions. Even so, alternative models can be viewed to a great degree as alternative slicings of the same projective pie.

How important is the choice of the camera model's parameterization? Very important, if the parameters are to serve directly as the controls for interaction and keyframe interpolation. For example, the popular LOOKAT-/LOOKFROM/VUP parameterization makes it easy to hold a world-space point centered in the image as the camera moves without tilting. To do the same by manually controlling generic translation/rotation parameters would be all but hopeless in practice, although possible in principle.

The difficulty with using camera parameters directly as controls is that no single parameterization can be expected to serve all needs. For example, sometimes it is more convenient to express camera orientation in terms of azimuth, elevation and tilt, or in terms of a direction vector. These particular alternatives are common enough to be standardly available, but others are not. A good example involves the problem, addressed by Jim Blinn[3] of portraying a spacecraft flying by a planet. Blinn derives several special-purpose transformations that allow the image-space positions of the spacecraft and planet to be specified and solves for the camera position. The need for this kind of specialized control arises frequently, but we would rather not face the prospect of deriving and coding specialized transformations each time they do.

In short, camera models are inflexible. To change the controls, one must either select a different pre-existing model or derive and implement a new one. If this inflexibility could be removed, the effort devoted to camera control could be reduced and the quality of the result enhanced.

In this paper, we present a body of techniques, which we call *through-the-lens camera control,* that offer a general solution to this problem. Instead of a fixed set, the user is given a palette of interactive image-space and world-space controls that can be applied "on the fly," in any combination. For example, the image-space position of an arbitrary world-space point can be controlled by interactive dragging, or pinned while other points are moved. Image-space distances, sizes, and directions can also be

Computer Graphics 26(2), July, 1992.
pages 331–340.
Proceedings SIGGRAPH '92.

Outline

Through-the-Lens Camera Control

History

Critique

Reflection

Stuff it inspired

Through-the-Lens Camera Control

Michael Gleicher and Andrew Witkin
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA
{gleicher|witkin}@cs.cmu.edu

Abstract

In this paper we introduce *through-the-lens camera control,* a body of techniques that permit a user to manipulate a virtual camera by controlling and constraining features in the image seen through its lens. Rather than solving for camera parameters directly, constrained optimization is used to compute their time derivatives based on desired changes in user-defined controls. This effectively permits new controls to be defined independent of the underlying parameterization. The controls can also serve as constraints, maintaining their values as others are changed. We describe the techniques in general and work through a detailed example of a specific camera model. Our implementation demonstrates a gallery of useful controls and constraints and provides some examples of how these may be used in composing images and animations.

Keywords: camera control, constrained optimization, interaction techniques

1 Introduction

Camera placement and control play an important role in image composition and computer animation. Consequently, considerable effort has been devoted to the development of computer graphics camera models. Most camera formulations are built on a common underlying model for perspective projection under which any 3-D view is fully specified by giving the center of projection, the view plane, and the clipping volume. Within this framework, camera models differ in the way the view specification is parameterized. Not all formulations are equivalent—some allow arbitrary viewing geometries, while others impose restrictions. Even so, alternative models can be viewed to a great degree as alternative slicings of the same projective pie.

How important is the choice of the camera model's parameterization? Very important, if the parameters are to serve directly as the controls for interaction and keyframe interpolation. For example, the popular LOOKAT-/LOOKFROM/VUP parameterization makes it easy to hold a world-space point centered in the image as the camera moves without tilting. To do the same by manually controlling generic translation/rotation parameters would be all but hopeless in practice, although possible in principle.

The difficulty with using camera parameters directly as controls is that no single parameterization can be expected to serve all needs. For example, sometimes it is more convenient to express camera orientation in terms of azimuth, elevation and tilt, or in terms of a direction vector. These particular alternatives are common enough to be standardly available, but others are not. A good example involves the problem, addressed by Jim Blinn[3] of portraying a spacecraft flying by a planet. Blinn derives several special-purpose transformations that allow the image-space positions of the spacecraft and planet to be specified and solves for the camera position. The need for this kind of specialized control arises frequently, but we would rather not face the prospect of deriving and coding specialized transformations each time they do.

In short, camera models are inflexible. To change the controls, one must either select a different pre-existing model or derive and implement a new one. If this inflexibility could be removed, the effort devoted to camera control could be reduced and the quality of the result enhanced.

In this paper, we present a body of techniques, which we call *through-the-lens camera control,* that offer a general solution to this problem. Instead of a fixed set, the user is given a palette of interactive image-space and world-space controls that can be applied "on the fly," in any combination. For example, the image-space position of an arbitrary world-space point can be controlled by interactive dragging, or pinned while other points are moved. Image-space distances, sizes, and directions can also be

Computer Graphics 26(2), July, 1992.
pages 331–340.
Proceedings SIGGRAPH '92.

Through-The-Lens Camera Control

Michael Gleicher
Andrew Witkin

School of Computer Science
Carnegie Mellon University

May, 1992

Some context...


August 1991, Yosemite
Driving back to Pittsburgh

A Conversation with my Advisor

The Monday after Thanksgiving, 1991

Mike: shouldn't the inertia tensor of a camera involve what it sees?

Andy: (math involving projections and integrals over images)

Mike: can't we just approximate that by summing over a few points?

Andy: yeah, but why would you want to?

Mike: I could use those image points as controls for the camera

(pause)

Andy: What are you doing for the next six weeks?

I had been building Bramble for a while.
A lot of infrastructure was in place.
TTL really was not a 6 week project

What was I trying to do?

And finally did by the end of my thesis in 1994.

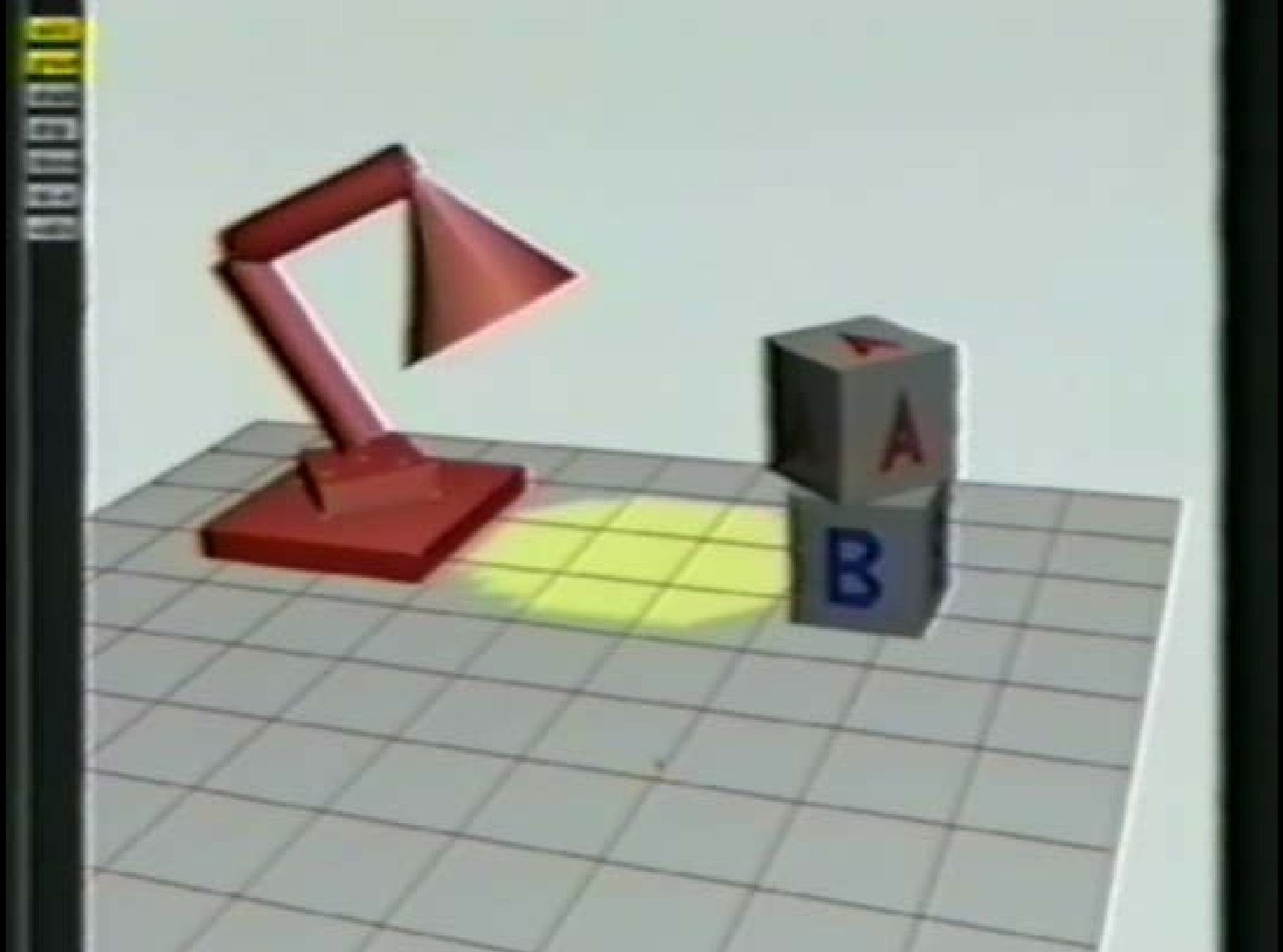
An Application:

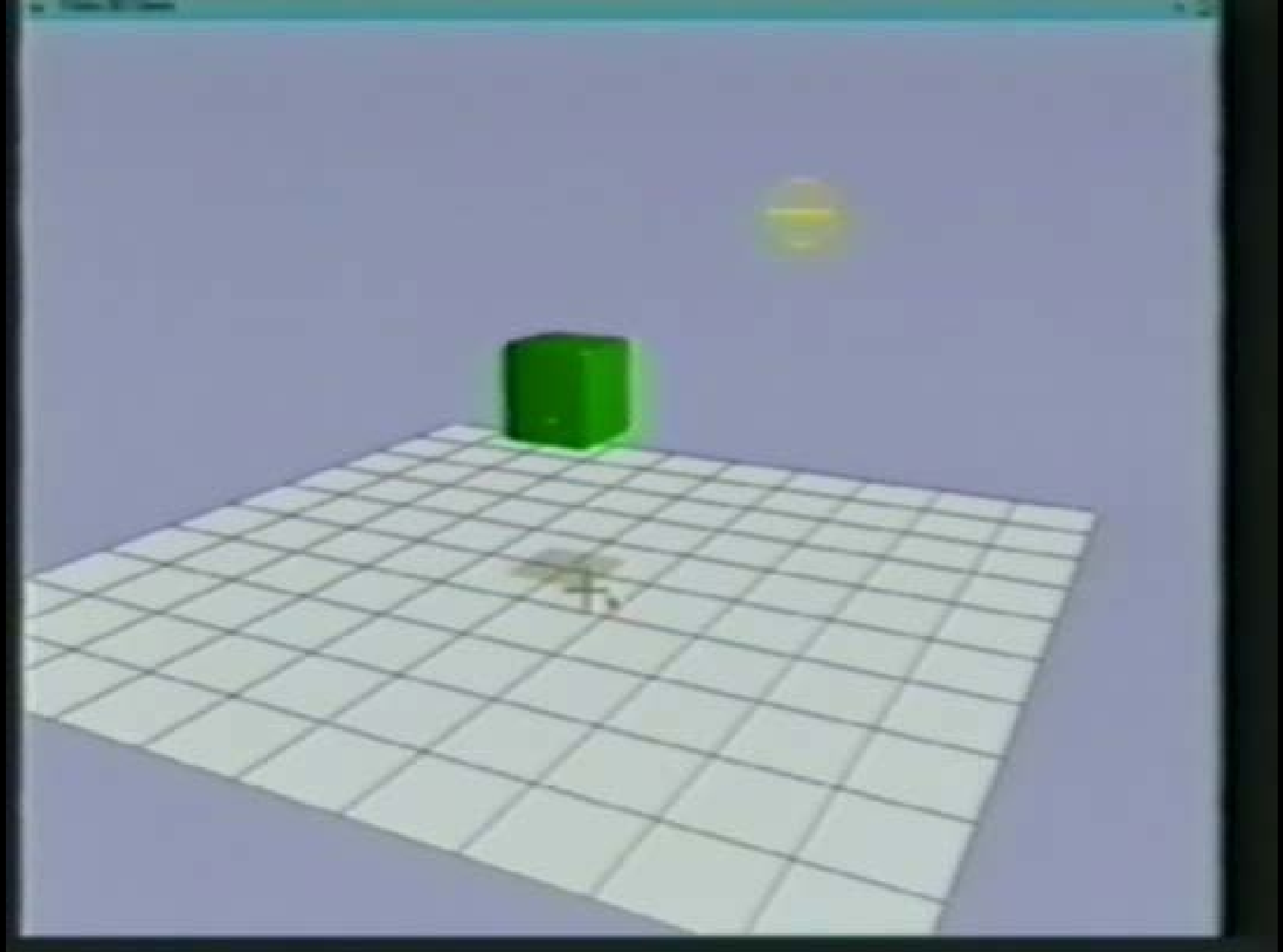
An example 3D application illustrating how the approach can be employed.

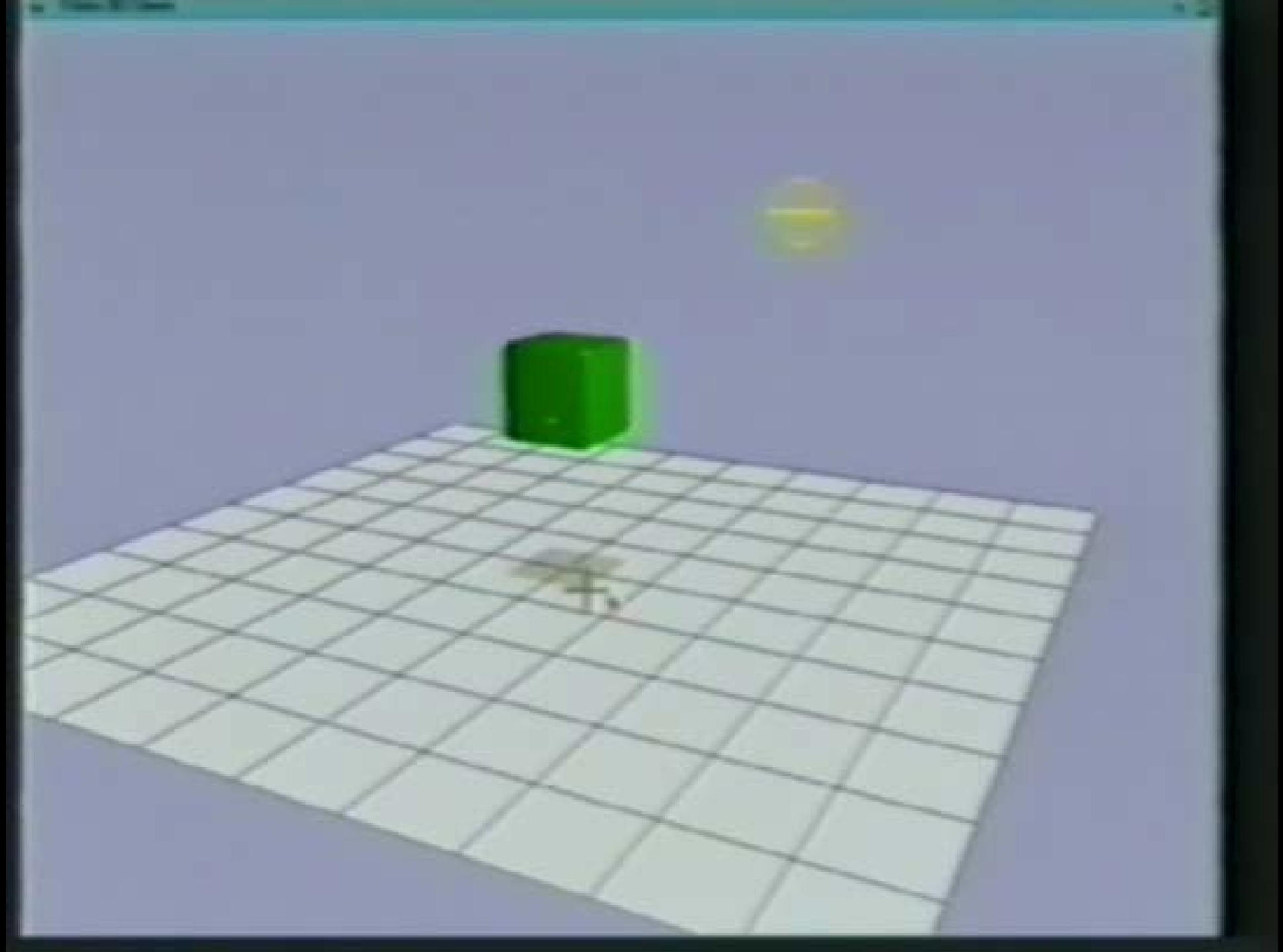
What I thought I was doing

My thesis: A Differential Approach to Graphical Interaction

Through-the-Lens Controls as a flexible building block


A
B





A critique

Looking back after 25 years… is it still any good?

The Math…

Pose controls as non-linear constraints

Key good idea – and required!

Differential approach – steps towards the goal
Solve linear problem on each step
Hacky active set methods for inequalities

Computers are faster
Optimization is better
Just solve the NLP
Use a real QP/NLP solver

Deriviatives of normalized Quaternions

Not so bad – maybe use exponential maps instead

The math hasn’t changed much

I thought real non-linear solvers were

- too unreliable
- too slow
- too hard to implement

If I had to do it again today…

Fixed step sizes vs. solve and go to solution

- “Fixed sized steps” doesn’t give dynamics
- Probably better to solve and interpolate (or just track mouse)

Use a real QP solver per step, and probably SQP to get solution

Normalized Quaternions vs. Exponential maps

- Exponential maps are “right”
- Normalized Quaternions hacky – but integrate with numerics

Implementation would be completely different

- dense solves, simpler automatic differentiation, …

The Implementation

Automatic differentiation

Took a while to catch on, but now good Python libraries

Object-oriented math “blocks”

Lots of caching, deferred computation

Symbolic “compiler”

Computers have changed
This doesn’t make sense

Sparse data structures

Problems are too small –
modern architectures like dense matrices

Computers have changed

A lot

What was I computing on in 1991?

SGI Personal Iris

20mhz MIPS R3000

MIPS R3010 FPU

16MB RAM

16 MIPS, 1.6 MFLOPS

For that video?

SGI 4D/210GTX

Basically…

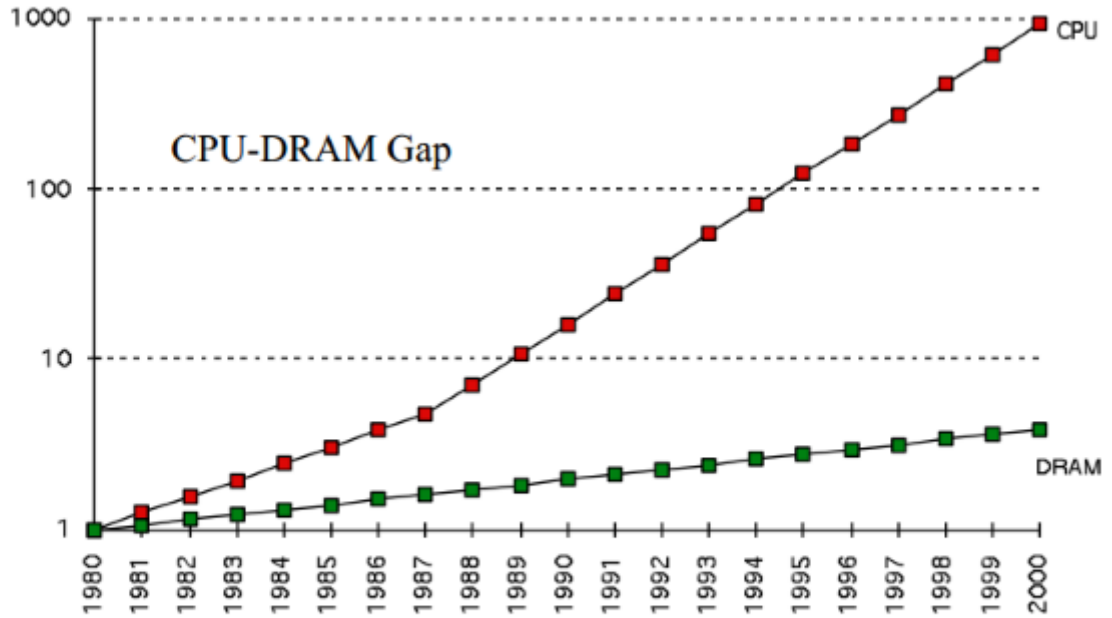
Same CPU/GPU

Better graphics card

Video I/O

Its more than just 1000x Faster

- Processor vs Memory Performance



1980: no cache in microprocessor;

1995 2-level cache

Its more than just 1000x Faster

1991 – MIPS R3000

20-25 MHZ

1 integer instruction per cycle

FP Multiply & Add – (muti-cycle)

Memory costs 2-3 cycles (hidden)

Cache miss about 10-20 cycles

2016 – Intel Skylake (per core)

3.2 – 4 GHZ (100x or more faster)

Multiple integer issue

Vector FP issue (32+ per cycle!)

Memory costs huge

Cache misses take forever

About that demo

The match-moving demo was important

Caught people's attention

Interactive tools inspired people

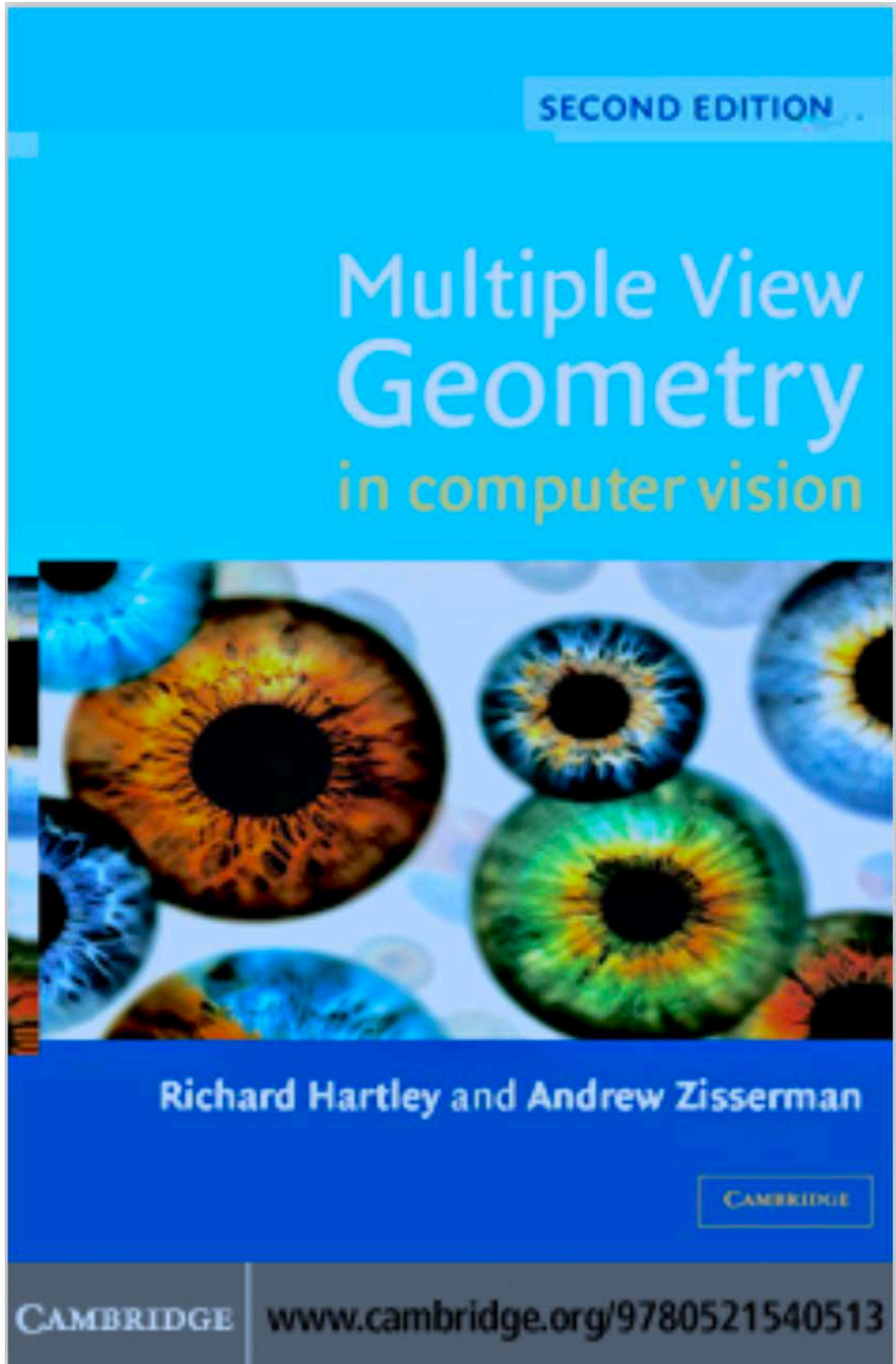
Match-moving is a big business!

How does it work today?

“Gold Standard Algorithm”

Use the “linear method” to get initial guess

Non-linear optimization to match control points



How off was I?

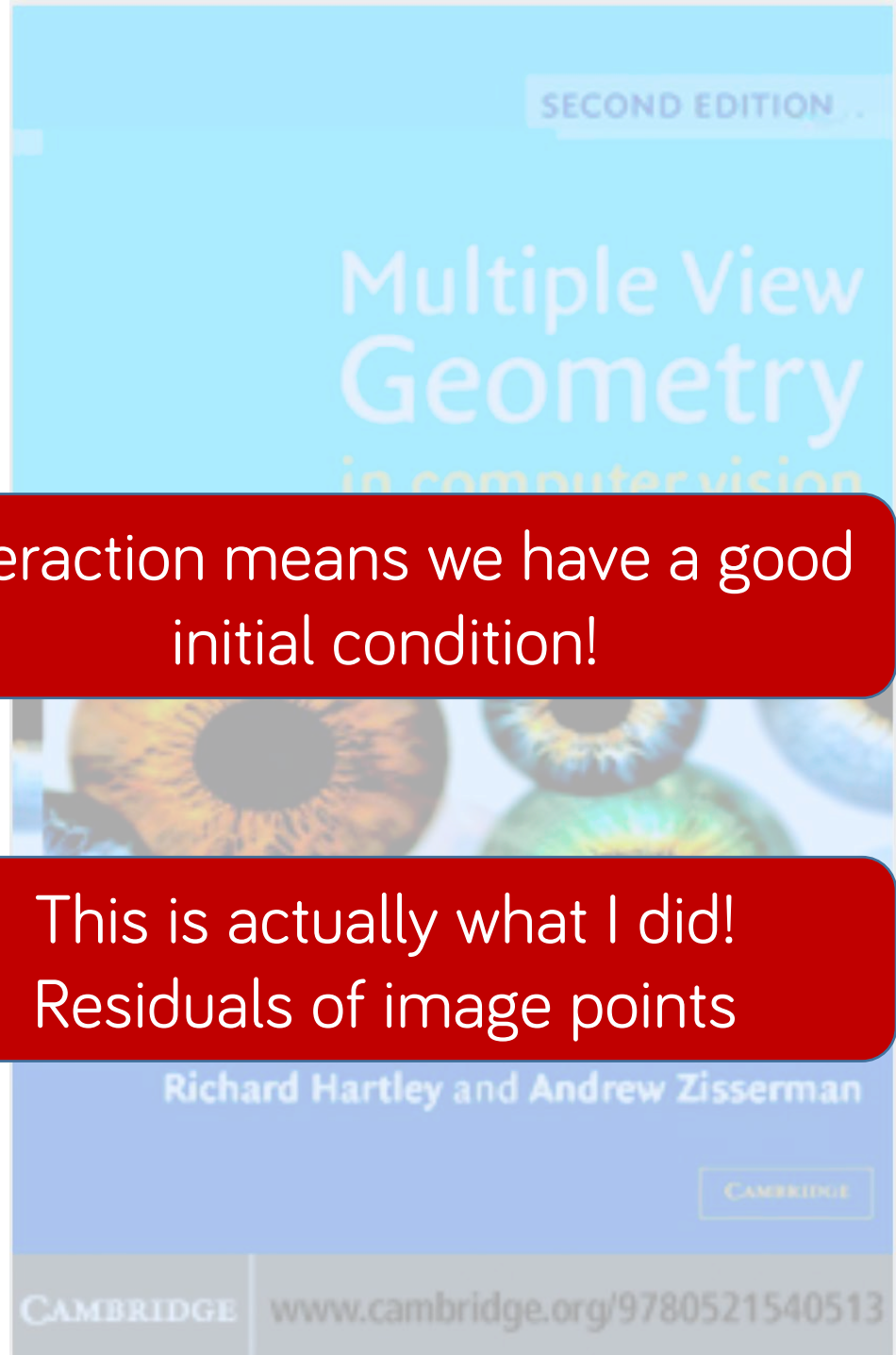
"Gold Standard Algorithm"

Use the "linear method" to get initial guess

Non-linear optimization to match control points

Interaction means we have a good initial condition!

This is actually what I did!
Residuals of image points



The linear method

For the 11 entries of the camera matrix

For more than 5 points (need enough to fully determine)

There is a linear solution to the problem!

(least squares for more than 5 ½ points)

But…

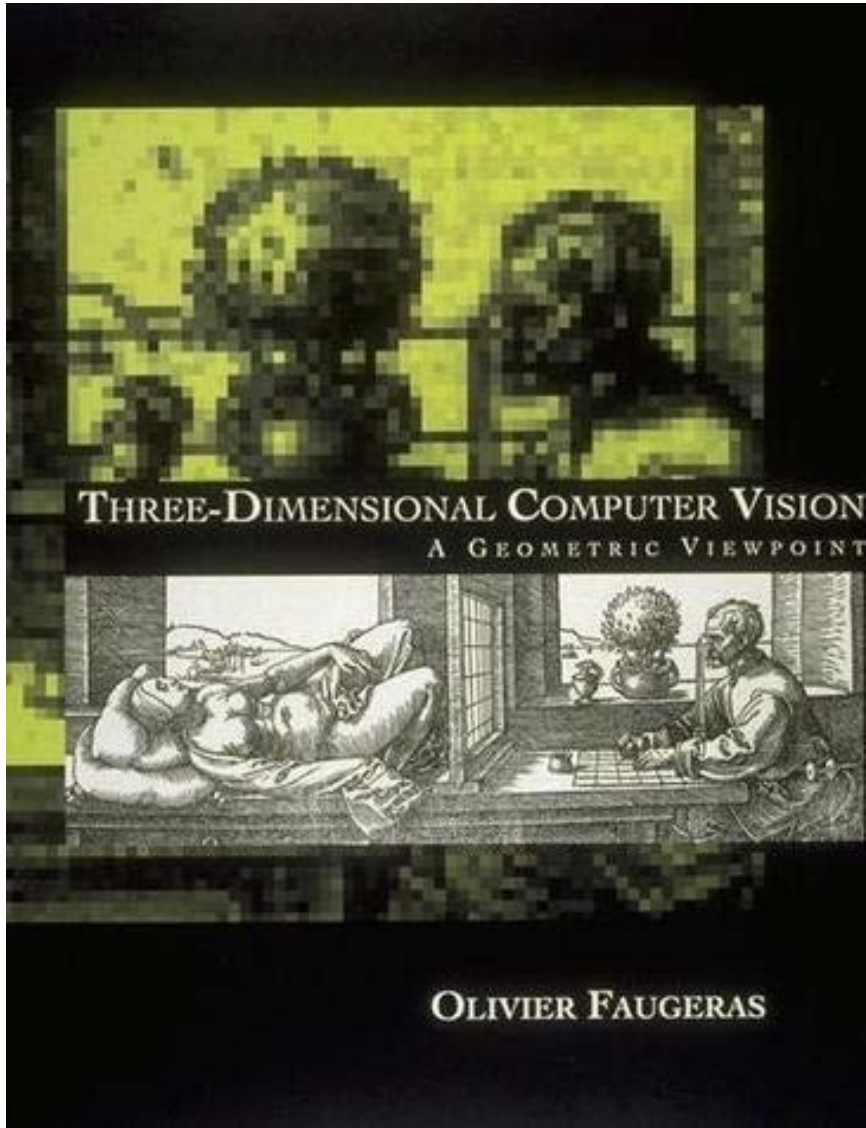
Can’t constrain to “real” camera (e.g. square pixels, simple projections)

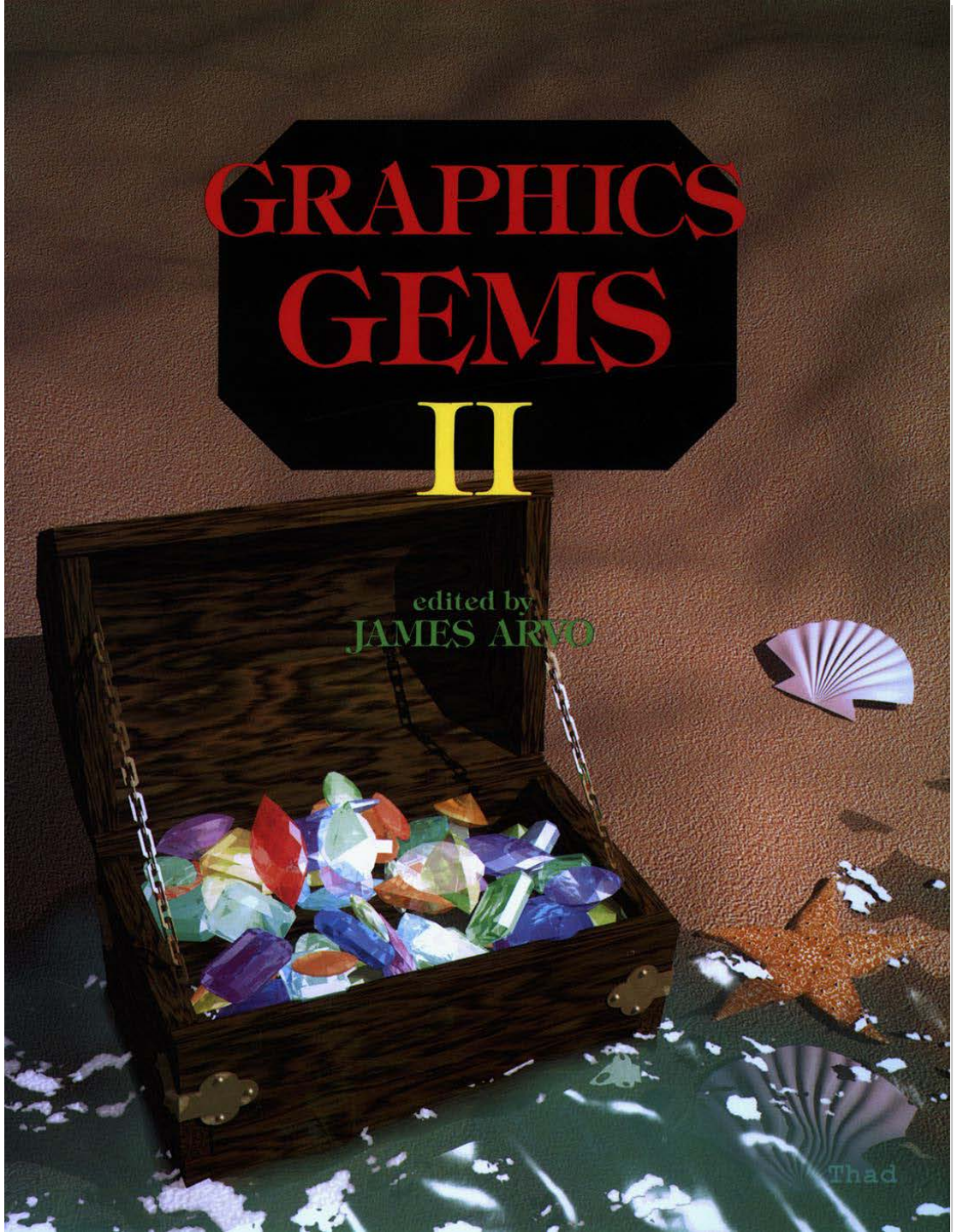
Measures error in wrong space (so small errors mean big problems)

But this was known in 1991 (just not by me!)

Faugeras' paper was 1986

Faugeras' book was 1993





IV.5

VIEW CORRELATION

Rod G. Bogart
University of Michigan
Ann Arbor, Michigan

To combine computer-generated objects into a photographic scene, it is necessary to render the objects from the same point of view as was used to make the photo. This gem describes an iterative technique for correlating view parameters to a photograph image. The method is implemented in C (Appendix 2) as a user function with a simple driver program for testing. The following sections describe the math behind the iterative technique, some specifies about the given implementation, and an example.

This method requires that at least five points are visible in the photo image, and that the 3D coordinates of those points are known. Later, when a computer-generated object is modeled, it must be in the same 3D space as the photo objects, and must use the same units. It is not necessary for the origin to be visible in the photo image, nor does it assume a particular *up* direction.

For each of the five (or more) data points, the 2D screen point must be found. This can be done simply by examining the photo image, or by employing more advanced image processing techniques. Because this is an iterative technique, the 2D point need not be accurate to sub-pixel detail. The final set of view parameters will project the given 3D points to 2D locations that have the minimum error from the given 2D screen points.

In addition to the data points, an iterative process needs a starting value. A set of view parameters must be given that approximate the correct answer. The method is extremely forgiving; however, it does help if the starting eye position is at least in the correct octant of 3-space, and the direction of view looks towards the center of the 3D data.

The basics of Match-Moving in 1991

About the same time as my paper

Has the same basic ideas (uses Newton iteration, works out derivatives)

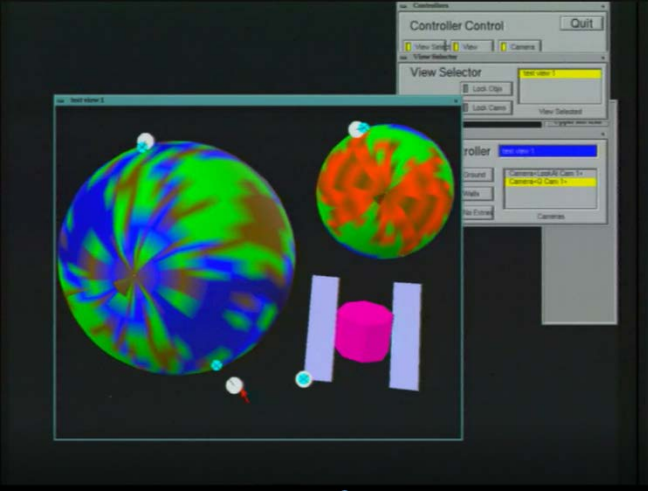
Also works out the derivatives

Thanks to:

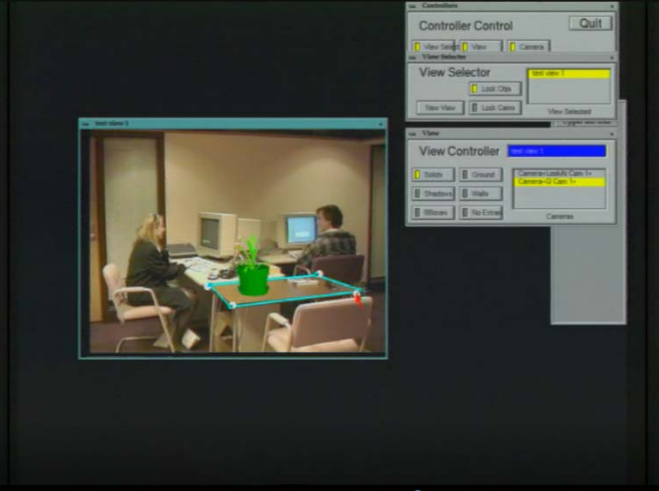
Doug Roble (Academy Award in 1998 for Match Moving)
Mohit Gupta (who teaches our Vision class, and pointed me at the HZ book)

But Match Moving wasn’t the point...

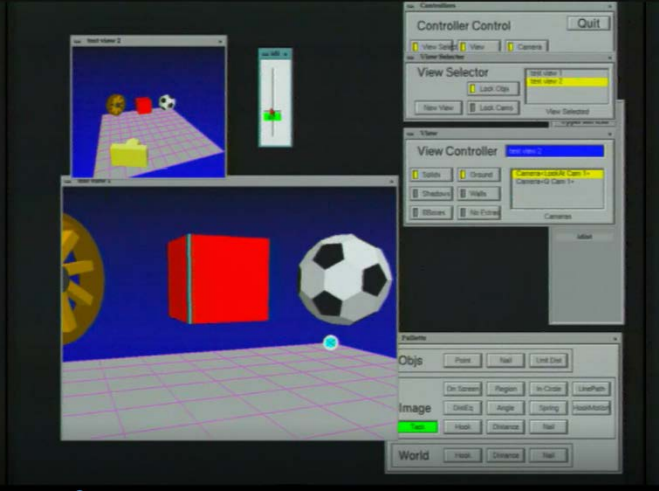
Maybe it should have been?



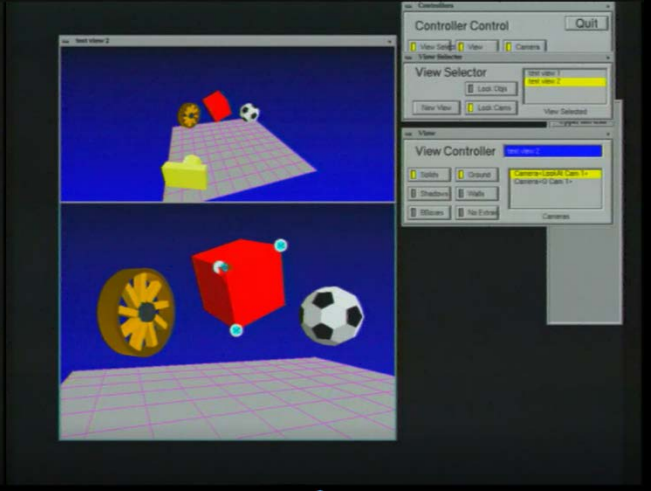
Specify by TTL



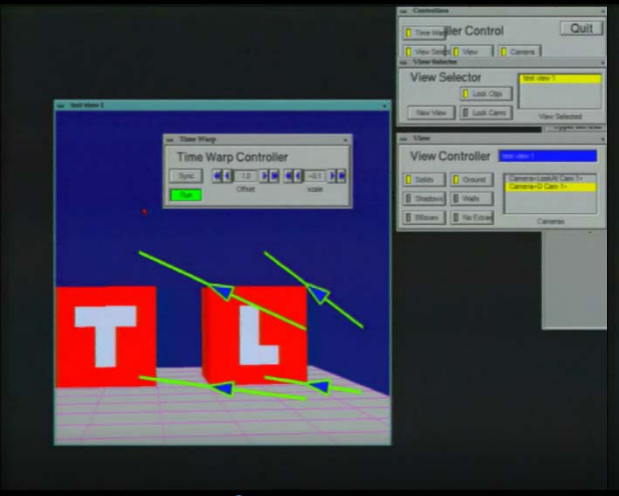
Specify (match-real image)



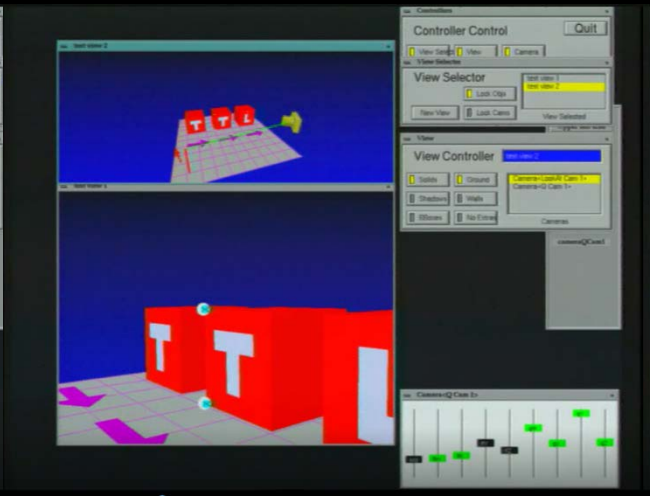
Specify (crazy controls)



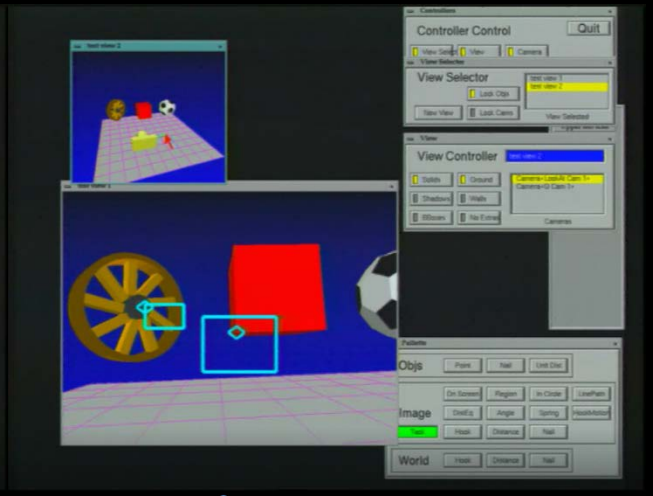
Specify (control objects)



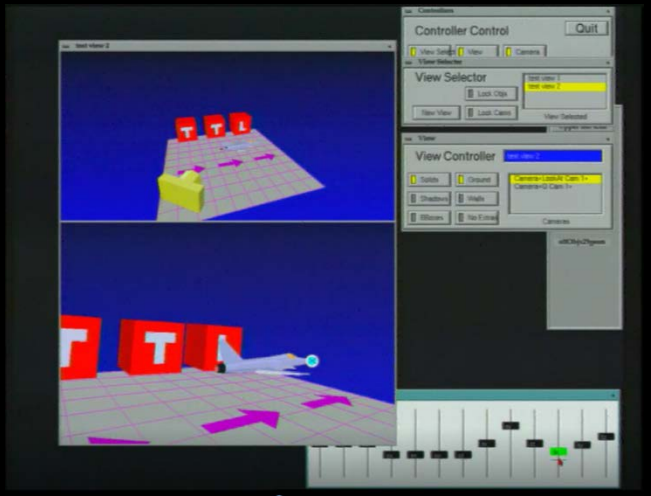
Specify by Motions



Constrain Motions



Constrain Motions



Tracking

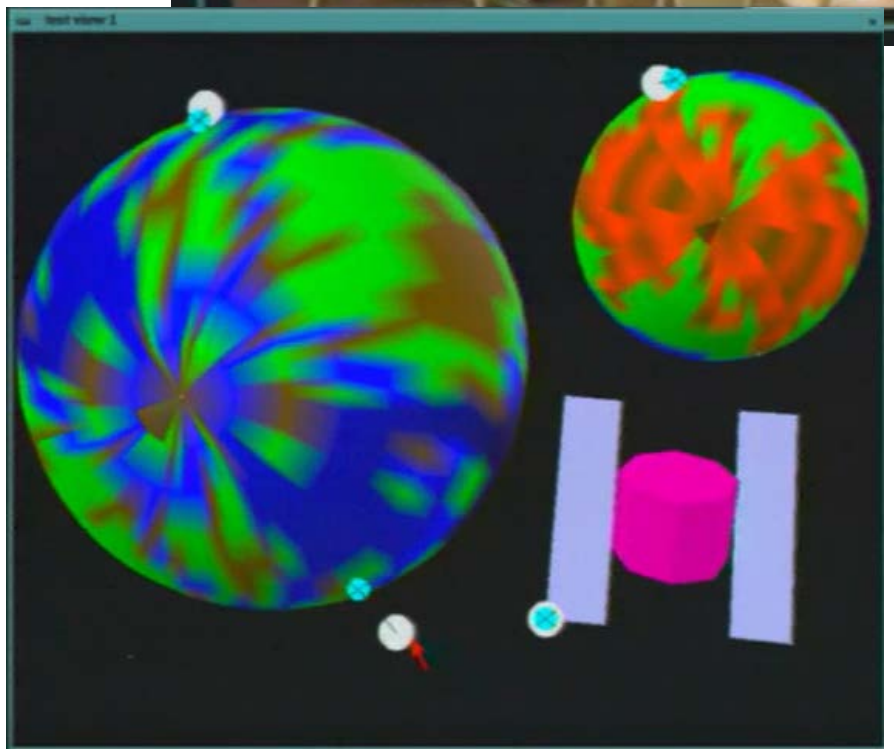
TTL Points as Interface

Is this a good interface?

Other than match-moving…
Do real people have problems this hard?

Not clear if anyone could use this…

(algorithms can!)



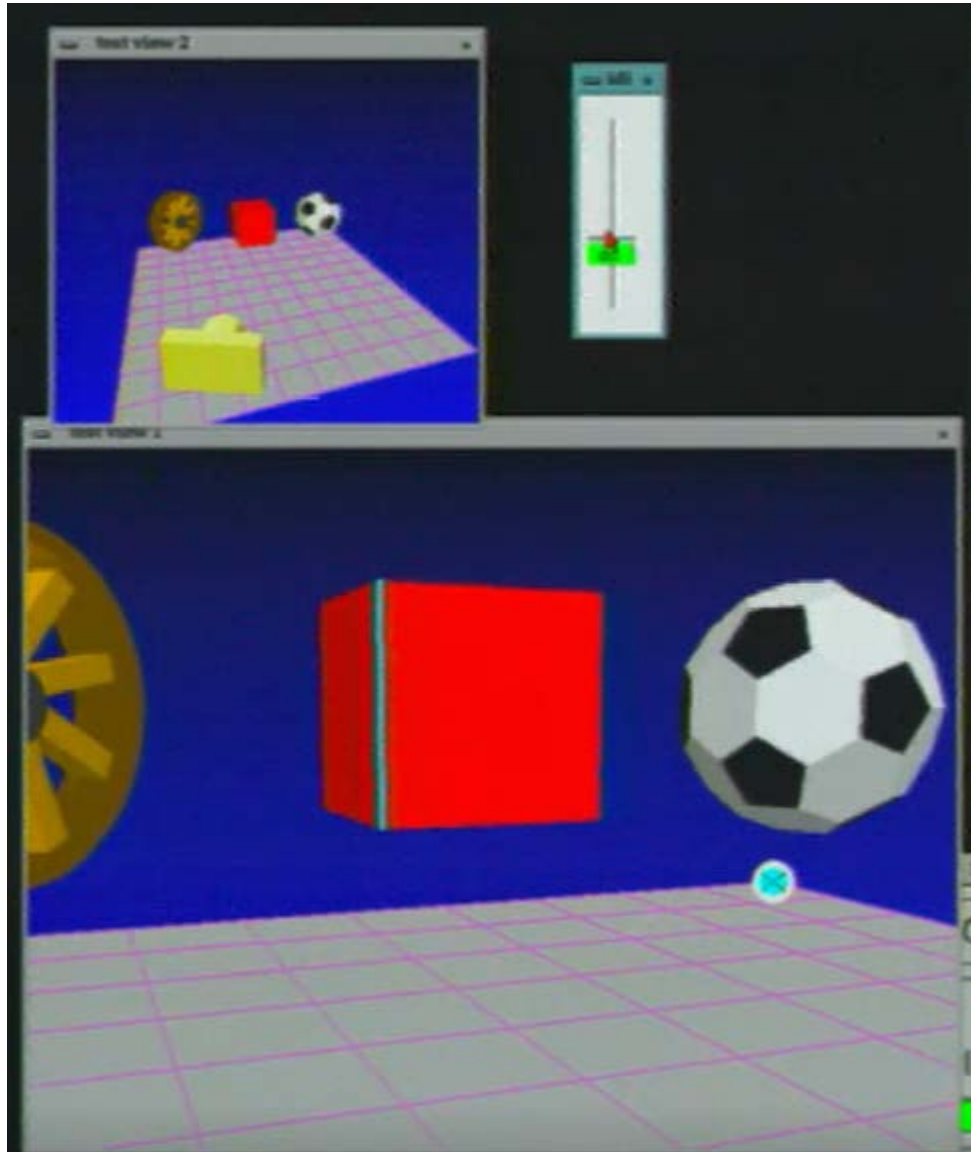
TTL Controls Beyond Points...

Mix and Match!

Who thinks that way?

Can people figure out what to use?

Maybe for declarative specification

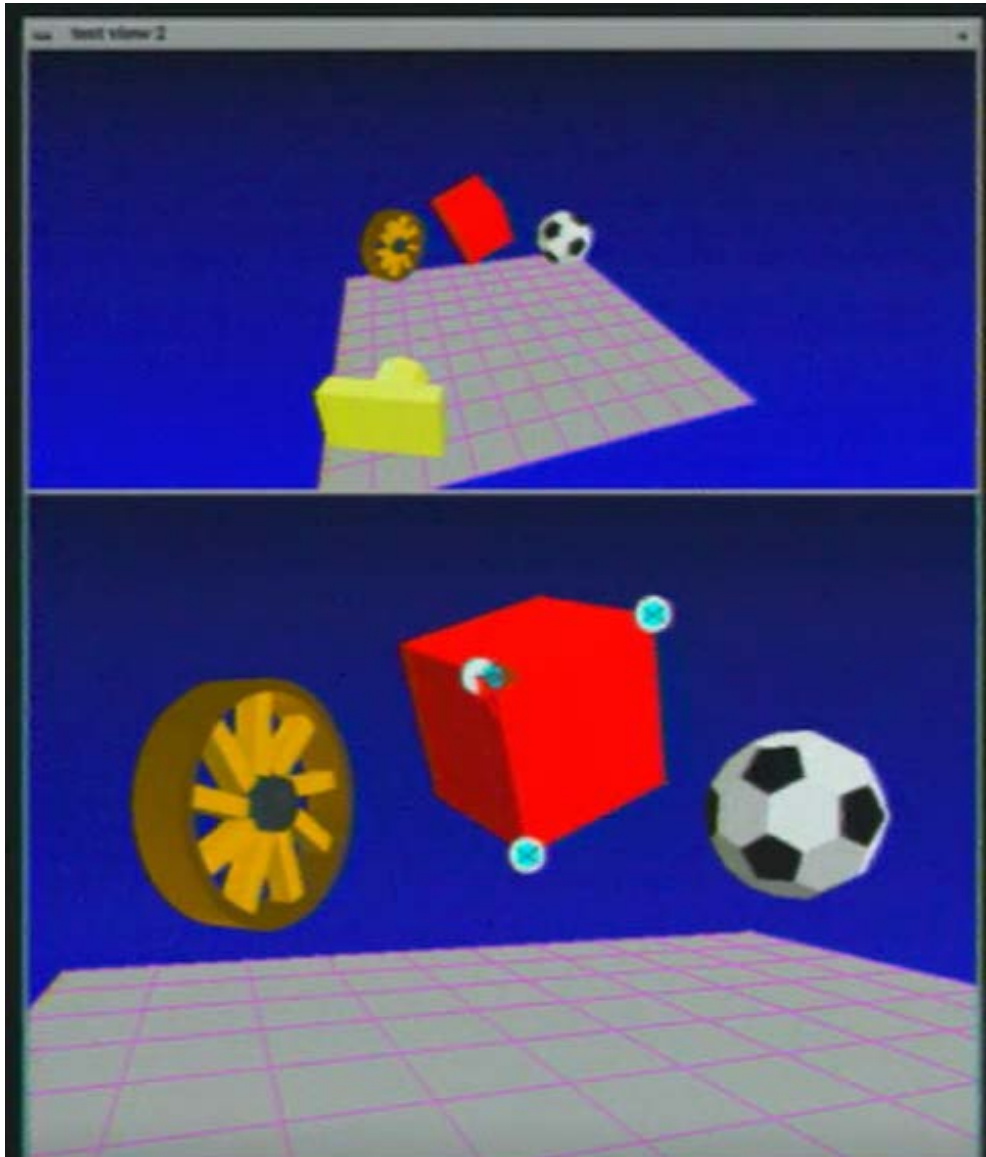


TTL Controls to Manipulate Objects

Even I couldn’t control this well

More for interaction prototyping

Sketch-based controls get this right



Avoids 3D inferences when sketching

Or, if you do, be very careful

gestural, stylized, …

Space-time sketching of character animation

Martin Guay*
Université de Grenoble
LJK, INRIA

Rémi Ronfard
Université de Grenoble
LJK, INRIA

Michael Gleicher
University of Wisconsin
Madison

Marie-Paule Cani
Université de Grenoble
LJK, INRIA

Abstract

We present a space-time abstraction for the sketch-based design of character animation. It allows animators to draft a full coordinated motion using a single stroke called the *space-time curve* (STC). From the STC we compute a dynamic line of action (DLOA) that drives the motion of a 3D character through projective constraints. Our dynamic models for the line's motion are entirely geometric, require no pre-existing data, and allow full artistic control. The resulting DLOA can be refined by over-sketching strokes along the space-time curve, or by composing another DLOA on top leading to control over complex motions with few strokes. Additionally, the resulting dynamic line of action can be applied to arbitrary body parts or characters. To match a 3D character to the 2D line over time, we introduce a robust matching algorithm based on closed-form solutions, yielding a tight match while allowing squash and stretch of the character's skeleton. Our experiments show that space-time sketching has the potential of bringing animation design within the reach of beginners while saving time for skilled artists.

CR Categories: I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

Keywords: Sketch-based animation, space-time, stylized animation, squash-and-stretch.

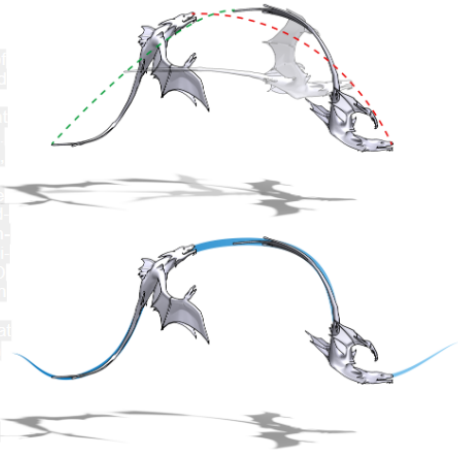


Figure 1: *Current shape interpolation techniques assume point-to-point blending (first row, result shown in grey), making it hard to create path-following motions. In contrast, our space-time sketching abstraction enables animators to sketch shapes and paths with a single stroke (second row).*

1 Introduction

Creating artistic and exaggerated styles of character animation requires flexible tools that allow expressive devices such as squash and stretch, as well as animating imaginary creatures such as dragons—often precluding the use of motion capture or database look-up. Yet, creating quality movements with current free-form animation technologies is a challenge.

The main approach to free-form motion design is keyframing: character poses at specific times are interpolated to produce motion. Over the years, significant advances have been made to more naturally specify key-poses. For example, by sketching skeletons or lines of action, or by handling deformations. However, the standard keyframing approach divides spatial and temporal control, making coordination of shape over time difficult. Hence, achieving quality results with the standard approach remains beyond the ability of unskilled artists and time consuming for skilled ones.

In this work, we introduce a novel space-time sketching concept enabling an animator to draft a full coordinated movement—that includes shape deformation over time—by sketching a single stroke. Further strokes can be used to progressively refine the animation. While strokes have been used in the past to specify both temporal and spatial iso-values of motion—with static lines of action (LOA) serving as shape abstraction at a given time as well as trajectories describing the successive positions over time of a single point—*space-time sketching* was never used to define animations. In our approach, we allow the user to control both the shape and trajectory of a character by sketching a single *space-time curve* (STC).

To illustrate our approach, consider the simple example of animating a flying dragon (Fig. 1). Animating the dragon requires the coordination of its shape over time as to follow the path's shape. With our approach, the basic animation can be created with a single sketched stroke. The stroke is used not only to provide the path of travel, but also to define how an abstraction of the character's shape (its line of action) changes over time. Additional strokes can be used to refine the movement, or add details such as the flapping of the wings. Creating such motion with existing approaches would require coordinating a large number of keyframes that specify deformations and positions along the path, or a method for puppeteering the degrees of freedom of the dragon.

The key to our approach is an interpretation of the space-time curve to define a 2D dynamic skeletal line abstraction, or *dynamic line of action* (DLOA). From the STC, we extract the DLOA's

*lemailamartin@gmail.com

ACM Reference Format
Guay, M., Ronfard, R., Gleicher, M., Cani, M. 2015. Space-time Sketching of Character Animation. ACM Trans. Graph. 34, 4, Article 118 (August 2015), 10 pages. DOI = 10.1145/2766893 http://doi.acm.org/10.1145/2766893.

ACM 978-1-4503-3331-3/15/08 ... $15.00.
DOI: http://dx.doi.org/10.1145/2766893

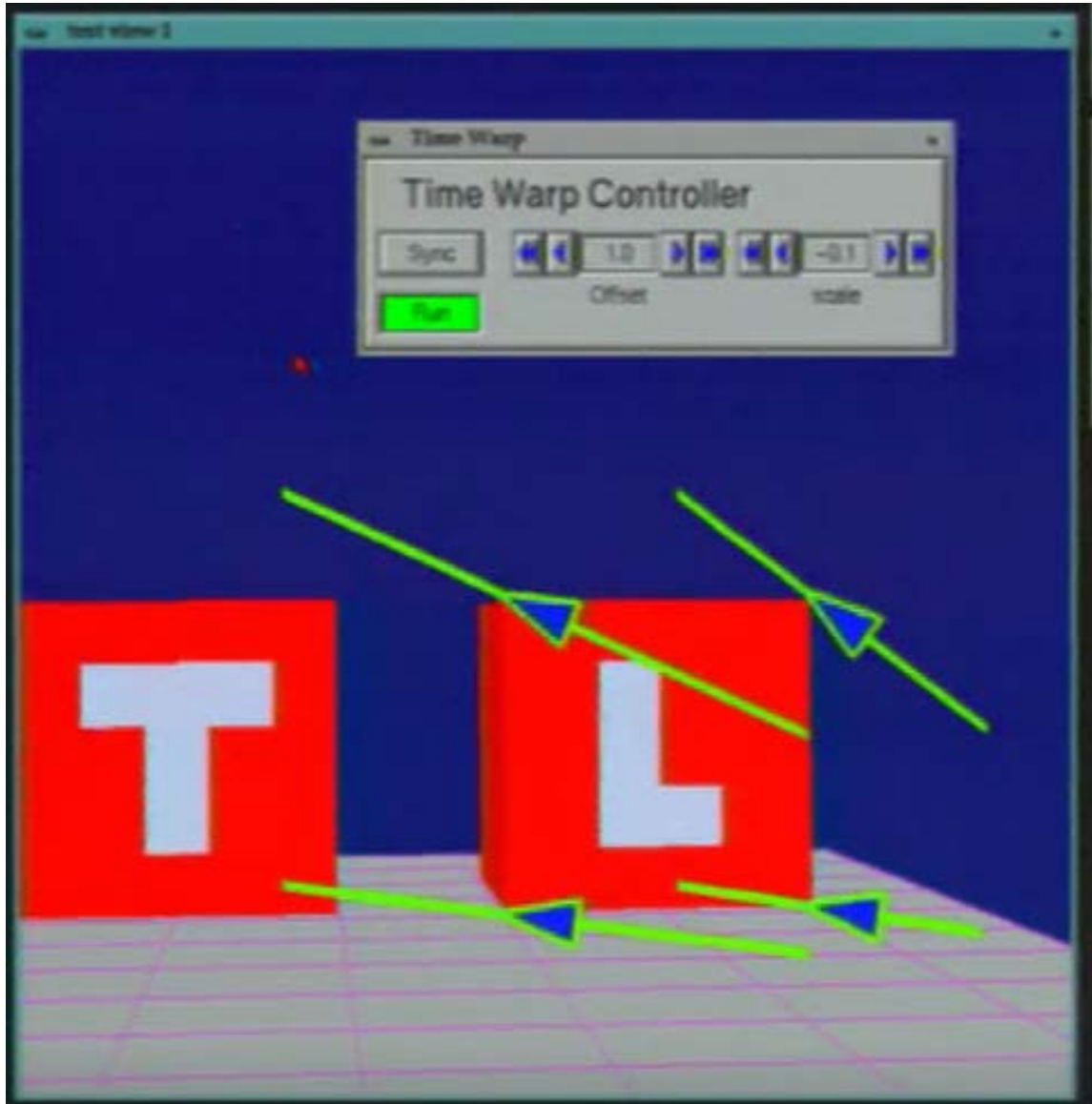
TTL specification of Motion

Hard to do for interesting motions

Very non-intuitive

Probably OK in special cases

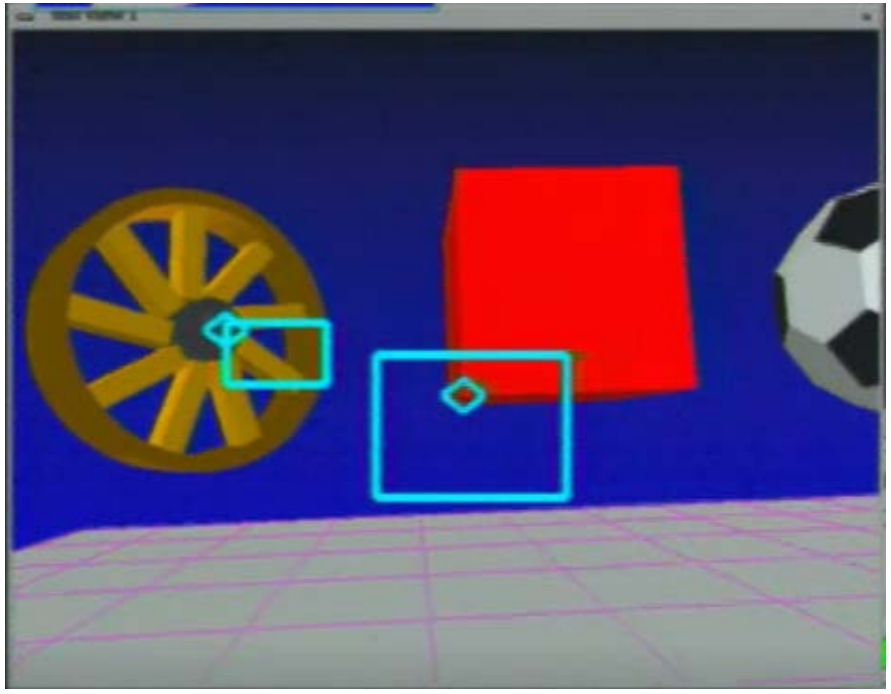
Hard to get **dynamics** right



TTL controls during motion...

Seems useful

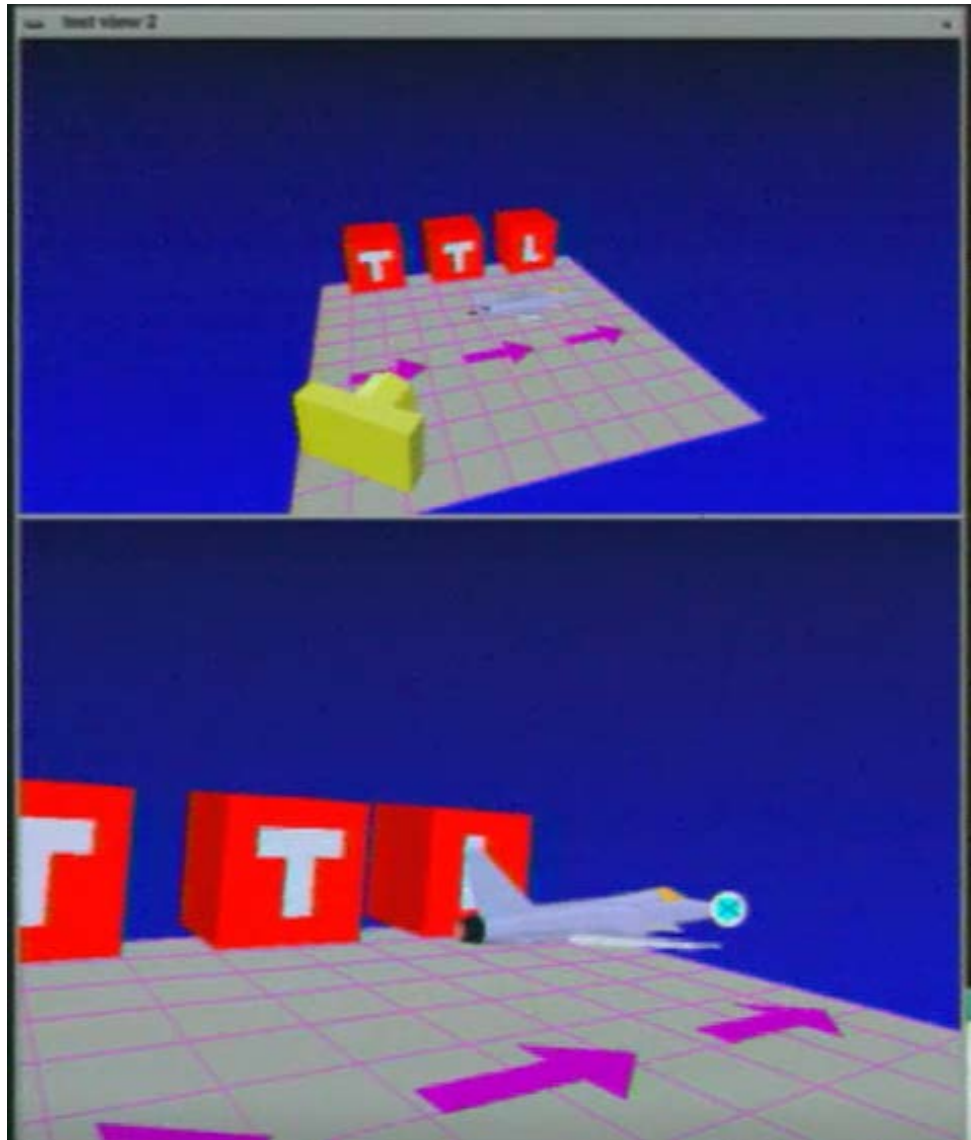
Might be overkill



TTL Constraints for moving objects

This seems more useful…

Not clear how generality is needed



Constrained Optimization is a great hammer

Good parameters != Good controls (so decouple!)

A single control is OK

Mixing-and-Matching controls, less OK (as a UI)

- Understand what a control does?
- Understand what combinations will do?
- Knowing how to choose from a broad palette?

Useful to talk in terms of what you see (controls) not what causes it

A little reflection… What can I learn?

1. Have a Killer Demo!
2. Solve a problem people have
3. Provide a solution that people can use
4. Provide a vision beyond 1,2,3

The cool problem you want to solve

may not be

The real problem people want a solution to

Two kinds of papers (there are others)

Have a novel (and good) problem – and a “good enough” solution

Have an existing problem – and have a “better” solution

Evaluation: how to convince people:

1. Problem is good
2. Solution is good enough

or

1. Solution is better

Jarke van Wijk – VIS Capstone, 2013

Lesson 2 on Evaluation

- Avoid evaluation. It is difficult, takes lots of effort, and is not always interesting.
- Take a short-cut.

Develop new methods that are so awesome, cool, impressive, compelling, fascinating, and exciting that reviewers, colleagues, users are totally convinced just by looking at your work and some examples.

Recipe for rejection?

What reviewers write:

The paper is not acceptable because a controlled user-study is missing.

What reviewers mean:

I don't believe this work makes any sense at all, but be my guest to convince me otherwise.

Good problems lead to better solutions

Motion Retargeting -> Lee&Shin ’99 Hierarchical Approach

Motion Graphs -> Reinforcement Learning Approaches

Parametric Motion Graphs -> Motion Fields

Image Retargeting -> Seam Carving, Image Deformation Approaches, …

Video Retargeting -> [whole literature]

Re-Cinematography (camera dynamics) -> L1 Minimization

Re-Cinematography (large deformations) -> Deformation-based stabilization

Critique of 2017...

CHI 2017

Honorable Mention Award

Yes, TTL inspired...

Measurable progress towards goal
(fluent conversational characters)

or

Totally new thing (bi-directional gaze)
(that others may do better in the future)

Looking Coordinated: Bidirectional Gaze Mechanisms for Collaborative Interaction with Virtual Characters

Sean Andrist,[1] Michael Gleicher,[2] Bilge Mutlu[2]
(1) Microsoft Research, Redmond, WA, USA
(2) Department of Computer Sciences, University of Wisconsin–Madison, Madison, WI, USA
sandrist@microsoft.com; gleicher@cs.wisc.edu; bilge@cs.wisc.edu

Figure 1. Left: A user wears eye-tracking glasses to collaboratively assemble a sandwich with a virtual character. Middle: The virtual character produces gaze cues to relevant task objects. Right: A user interacting with the virtual character in head-mounted virtual reality.

ABSTRACT
Successful collaboration relies on the coordination and alignment of communicative cues. In this paper, we present mechanisms of *bidirectional gaze*—the coordinated production and detection of gaze cues—by which a virtual character can coordinate its gaze cues with those of its human user. We implement these mechanisms in a hybrid stochastic/heuristic model synthesized from data collected in human-human interactions. In three lab studies wherein a virtual character instructs participants in a sandwich-making task, we demonstrate how bidirectional gaze can lead to positive outcomes in error rate, completion time, and the agent's ability to produce quick, effective nonverbal references. The first study involved an on-screen agent and the participant wearing eye-tracking glasses. The second study demonstrates that these positive outcomes can be achieved using head-pose estimation in place of full eye tracking. The third study demonstrates that these effects also transfer into virtual-reality interactions.

ACM Classification Keywords
H.5.2 **Information Interfaces and Presentation**: User Interfaces—*evaluation/methodology, user-centered design*

Author Keywords
Bidirectional gaze; gaze coordination; interactive gaze; dyadic gaze; joint attention; embodied agents; verbal referencing

CHI 2017, May 06 - 11, 2017, Denver, CO, USA.

ACM 978-1-4503-4655-9/17/05$15.00
DOI: http://dx.doi.org/10.1145/3025453.3026033

INTRODUCTION
When people interact they use a number of verbal and nonverbal communication mechanisms to coordinate. Gaze is a particularly important cue in both directions; people use it to indicate their attention as well as sense the attention of others. For example, an instructor might observe the gaze of their student to see that they are looking in the wrong place or are seeking help and then use their own gaze to capture the student's attention in order to guide it to the correct place. Such bidirectional gaze mechanisms can improve coordination in interaction by correcting potential failures before they occur in a subtle way, avoiding interruptions in the flow of activity.

Interfaces utilizing virtual embodied agents hold great promise for situated interaction in domains such as work training, occupational therapy, rehabilitation, counseling, retail, education, entertainment, and more. To build rich, immersive, and fluent interactive experiences with agents in these settings, we must build models that they can use to coordinate their actions and behaviors with their users and shared objects in the task environment. In this paper, we present techniques that improve the quality of human-agent interactive experiences through the use of *bidirectional gaze*—the coordinated production and responsiveness to social gaze cues—and demonstrate that these techniques indeed achieve positive interaction outcomes.

Bidirectional gaze is particularly important when people collaborate over a shared visual space [11, 13, 15, 45]. Coordinated gazing allows conversational participants to monitor their interlocutor for understanding, regulate the amount of mutual gaze and averted gaze, quickly pass and receive the conversational floor, disambiguate verbal references early in their production, and so on. Virtual agents currently lack sophisticated models that would allow them to engage in similar

Has TTL done anything for you since then?

What has happened since then?

How Through-the-Lens continue to inspire me

What Mike Does...

Human Data Interaction (visualization)

Human Graphics Interaction (media authoring)

Human Robot Interaction (robots!)

Video Stabilization

Moving a camera requires getting the dynamics right

Gleicher and Liu. Re-Cinematography: Improving the Camerawork of Casual Video. ACM TOMMCAP 2008.
(extended from ACM Multi-Media 2007 best paper)
Liu, Gleicher, Jin, and Agarwala. Content Preserving Warps for 3D Stabilization. SIGGRAPH 2009.
Liu, Gleicher, Wang, Jin, and Agarwala. Subspace Video Stabilization. ACM ToG 2011.

Source motion

Re-Cinematography

Video Stabilization

Problem:

Shaky video in, less shaky (good?) video out

Art (and Perception)

What is good camera movement?

Perception (and Art)

How can we avoid impossible computer vision?

Three Projects

Re-Cinematography

What can you do beyond removing jitter?

Stabilization by 3D Warping

How can you make bigger changes?

Stabilization by Subspace Constraints

How do you make it practical?

Work with Feng Liu and Adobe

Lessons from Filmmaking 101

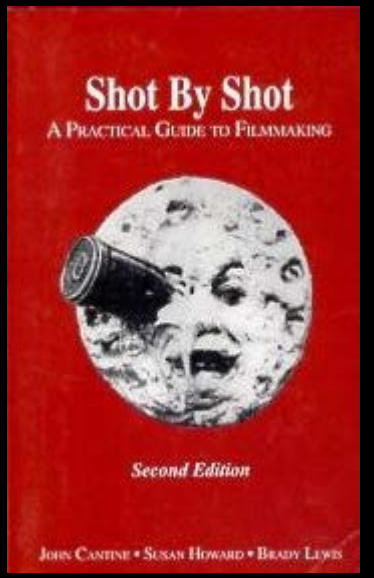
Direct the viewer's attention

Use a tripod! (a damped one)

Avoid distracting the viewer (unless…)

Camera movements should be motivated

Smooth movements to make connections



Pittsburgh Filmakers
circa 1991

The key insight:
Translate cinematography to implementation

Motion should be intentional

- Static shots should be static
- Moving shots are goal directed
  - Constant velocity with ease in/out

What the art of cinematography tells us about camera motion

Camera motions should be intentional

- Avoid movement if not necessary
- Move in directed ways

Re-Cinematography:

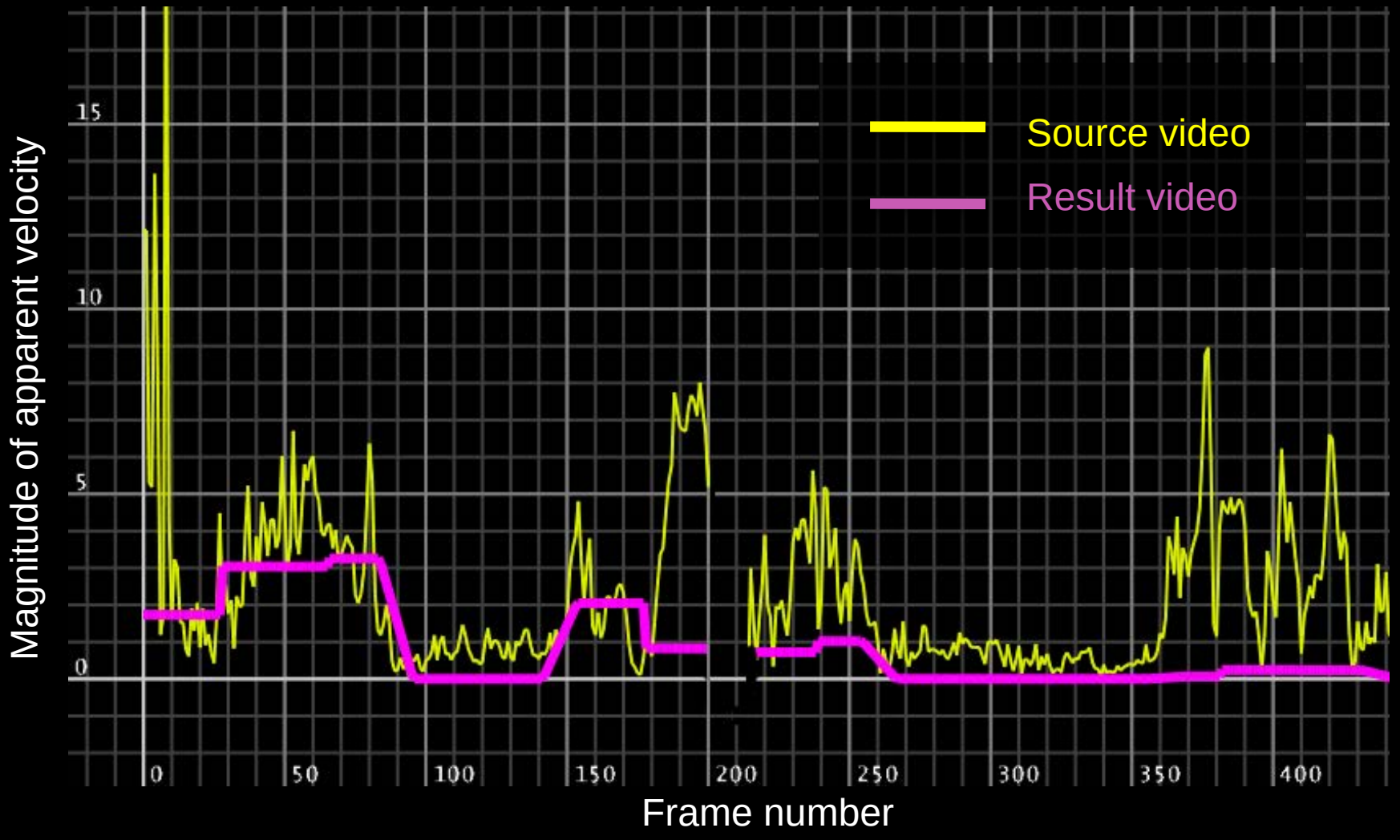
Post-process video clips so that the camera motions appear to better follow the rules.

What paths do we want?

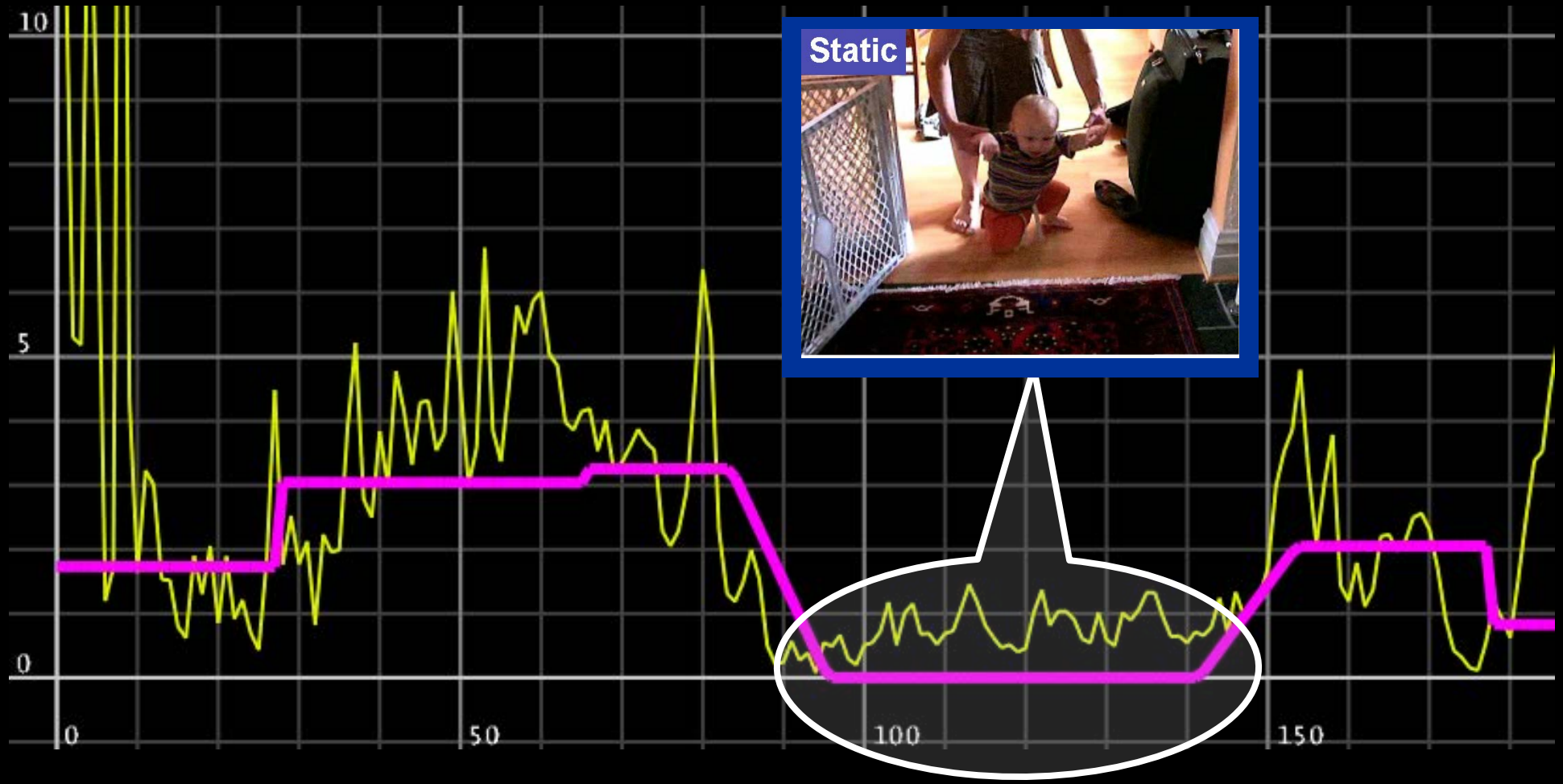
1. Preserve the intent of the source
2. Obey **the** rule of cinematography:

   Camera motion should be intentional

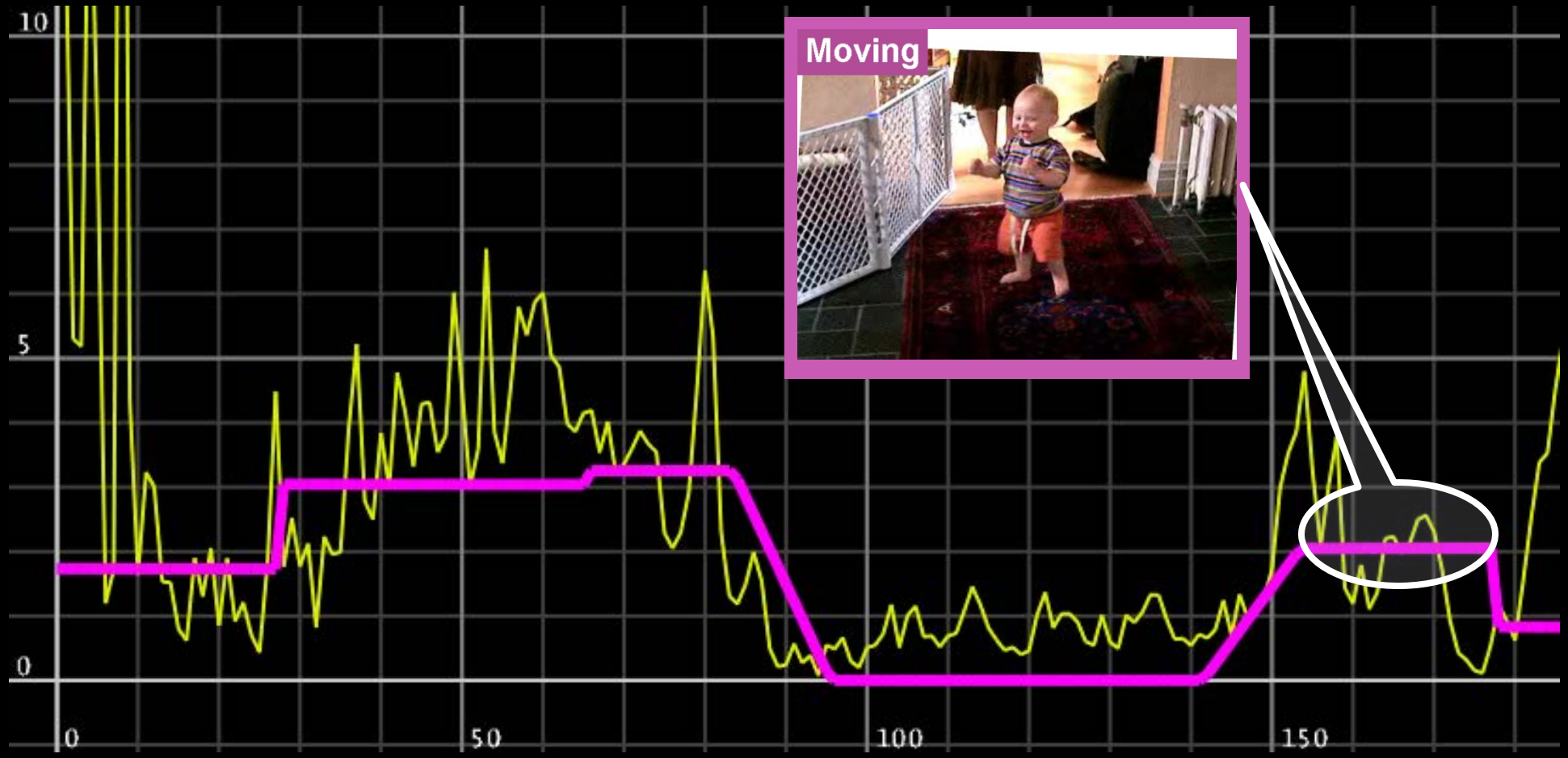
Re-Cinematography “Works”
Velocity profiles meet goals



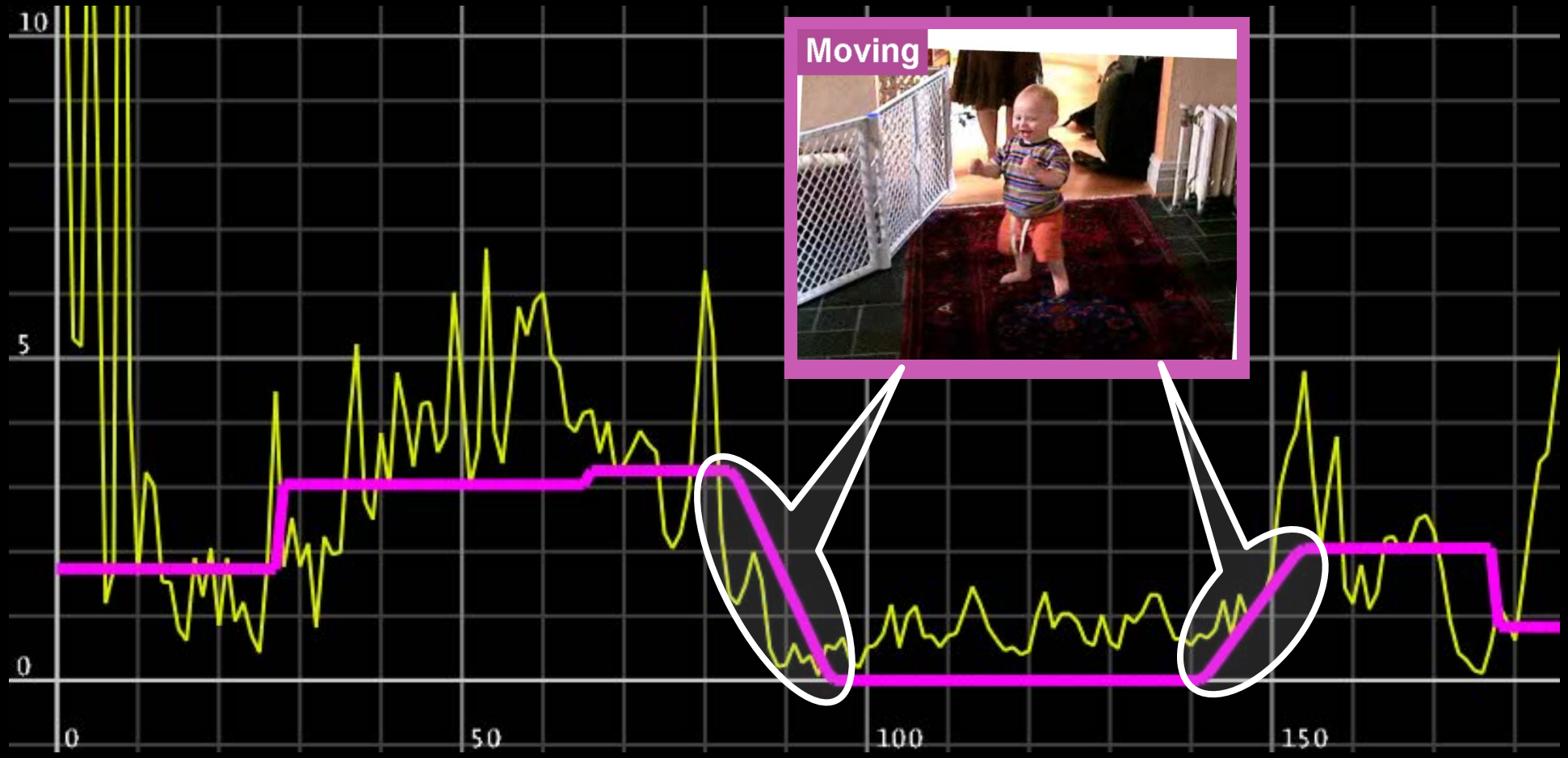
Static segments are static



Moving segments have piecewise constant velocity



Ease in and out



(at least)
Two Big Problems...

The motion “planning” is quite hacky – and can’t trade off quality

Answer: L1 optimal paths (see also our paper)

Grundmann, Kwatra, Essa. Auto-Directed Video Stabilization with Robust L1 Optimal Camera Paths, CVPR 2011.

The large deformations distort horribly

Answer: Liu et. al 2009, 2011

Content-Preserving Warps (2009)

Subspace Methods (2011)

Input

Our
Output

More problems...

To swing or not to swing...

Stylistic choices?

Build smart systems

no “right answer”

depends on…

taste, judgment, style, …

intended message

Avoid making decisions

EUROGRAPHICS 2017 / L. Barthe and B. Benes
(Guest Editors)

Volume 36 (2017), Number 2

Zooming On All Actors: Automatic Focus+Context Split Screen Video Generation

Moneish Kumar[1], Vineet Gandhi[1], Remi Ronfard[2] and Michael Gleicher[3]

[1]IIIT Hyderabad, [2]Univ. Grenoble Alpes/INRIA/LJK, [3]University of Wisconsin-Madison.

Abstract

Recordings of stage performances are easy to capture with a high-resolution camera, but are difficult to watch because the actors' faces are too small. We present an approach to automatically create a split screen video that transforms these recordings to show both the context of the scene as well as close-up details of the actors. Given a static recording of a stage performance and tracking information about the actors positions, our system generates videos showing a focus+context view based on computed close-up camera motions using crop-and zoom. The key to our approach is to compute these camera motions such that they are cinematically valid close-ups and to ensure that the set of views of the different actors are properly coordinated and presented. We pose the computation of camera motions as convex optimization that creates detailed views and smooth movements, subject to cinematic constraints such as not cutting faces with the edge of the frame. Additional constraints link the close up views of each actor, causing them to merge seamlessly when actors are close. Generated views are placed in a resulting layout that preserves the spatial relationships between actors. We demonstrate our results on a variety of staged theater and dance performances.

Categories and Subject Descriptors (according to ACM CCS): I.2.10 [Computing Methodologies]: Vision and Scene Understanding—Video Analysis I.3.3 [Computing Methodologies]: Picture/Image Generation—Viewing Algorithms

1. Introduction

A video presenting a staged event, such as theatre or dance, must choose between providing a wide field of view of the whole scene or close-up views that show details. Recordings of staged performances typically use multiple cameras, with multiple camera operators, to capture multiple views which are edited together to make a single video. Alternatively, these views may be composited together to create a *split-screen composition* (SSC). Split-screen compositions give the user the decision of what to attend to, reducing the need for editorial decisions that can be difficult to automate. Creating good split-screen compositions requires creating a set of views that are good individually and can be used together, as well as creating layouts that correctly convey the scene and its details. In this paper, we present an approach for creating split-screen compositions of staged performances. We record a high-resolution, but wide field-of-view, video of the event with a static (unattended) camera. While this easy to create recording may capture detail, it does not necessarily provide a convenient way for a viewer to see both the whole scene and important details (like actors' facial expressions or gestures). Therefore, we provide an automatic system that transforms the video into a split-screen composition of both the wide view of the scene as well as detailed views of the actors' faces (Figure 1). The close-up views are created as virtual camera movements by applying panning, cropping and zooming to the source video. The key challenges are: (a) to compute appropriate virtual camera movements in a way that creates good close-ups which work together; and (b) to create proper layouts of these views that preserve the spatial relationships between actors.

The core of our approach is a novel method to determine the camera movements for close-up views of each actor given their position on stage (tracking information). Our method takes tracking information as input, and, therefore, is independent of the tracking algorithm. For example, our prototype implementation uses an interactive offline approach that provides acceptable actor position estimates with some manual annotations. Given this tracking information, our approach poses camera trajectory computations as a convex optimization, which aims at showing an actor as large and centred in the close-up as possible, given the constraints that: (a) the entire face must be visible; and (b) the frame should avoid cutting other actors' faces and torso. An $L(1)$ regularization term creates movements that are as smooth as possible but also follow the kinds of acceleration patterns preferred by cinematographers. When multiple actors come close together, the system combines them into a single close-up view to avoid cutting their faces (or torso) with the frame. By adding additional constraints between the camera movements, the different camera paths merge seamlessly as

Another Application...

How do we use Visual Simulation as a Behavioral Research Tool?

How to review Virtual Environment Experiences?

Ponto, Kohlmann, and Gleicher. *Effective Replays and Summarization of Virtual Experiences*. IEEE TVCG (VR 2012).

Head Tracked Virtual Experiences

What did they see?

Where were they “looking”?

A (3D) Video Stabilization Problem?

Easier than 3D Stabilization:

Have camera path

Have world geometry (to render novel views)

Higher Expectations!

More power for analysis

More flexibility for re-synthesis

Key Ideas

Cinematography Model

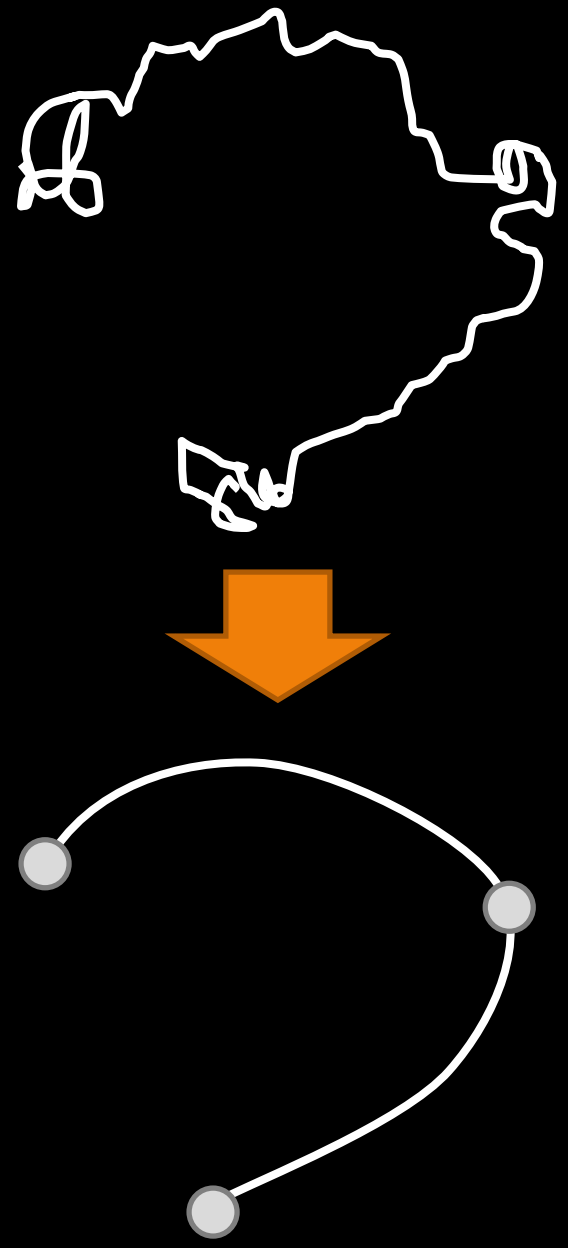
- from Re-Cinematography
- stable points + interpolations

Content-Dependent Metric

Segmentation

Path Generation

Reproject Original Viewpoint



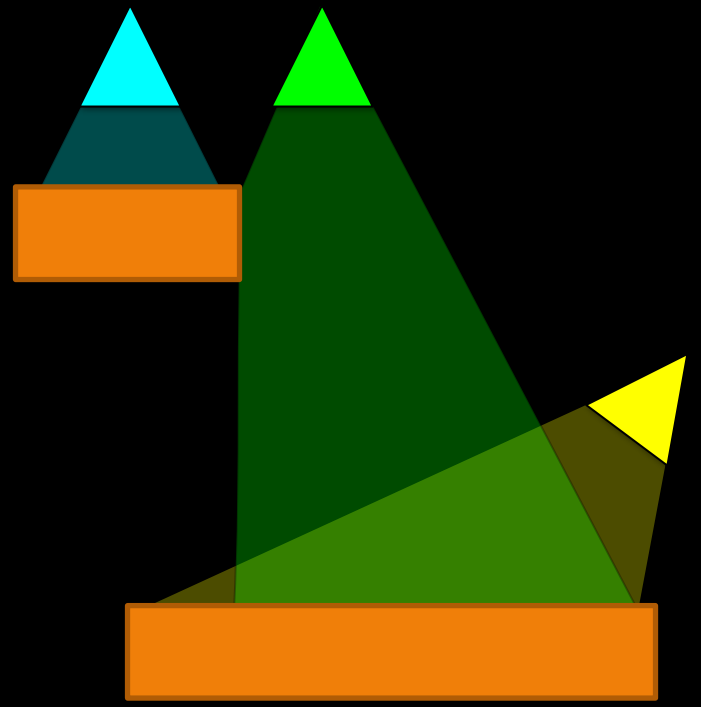
Key Ideas

Cinematography Model

Content-Dependent Metric

- cameras see the same things?
- efficient GPU implementation

Segmentation



Path Generation

Reproject Original Viewpoint

Key Ideas

Cinematography Model

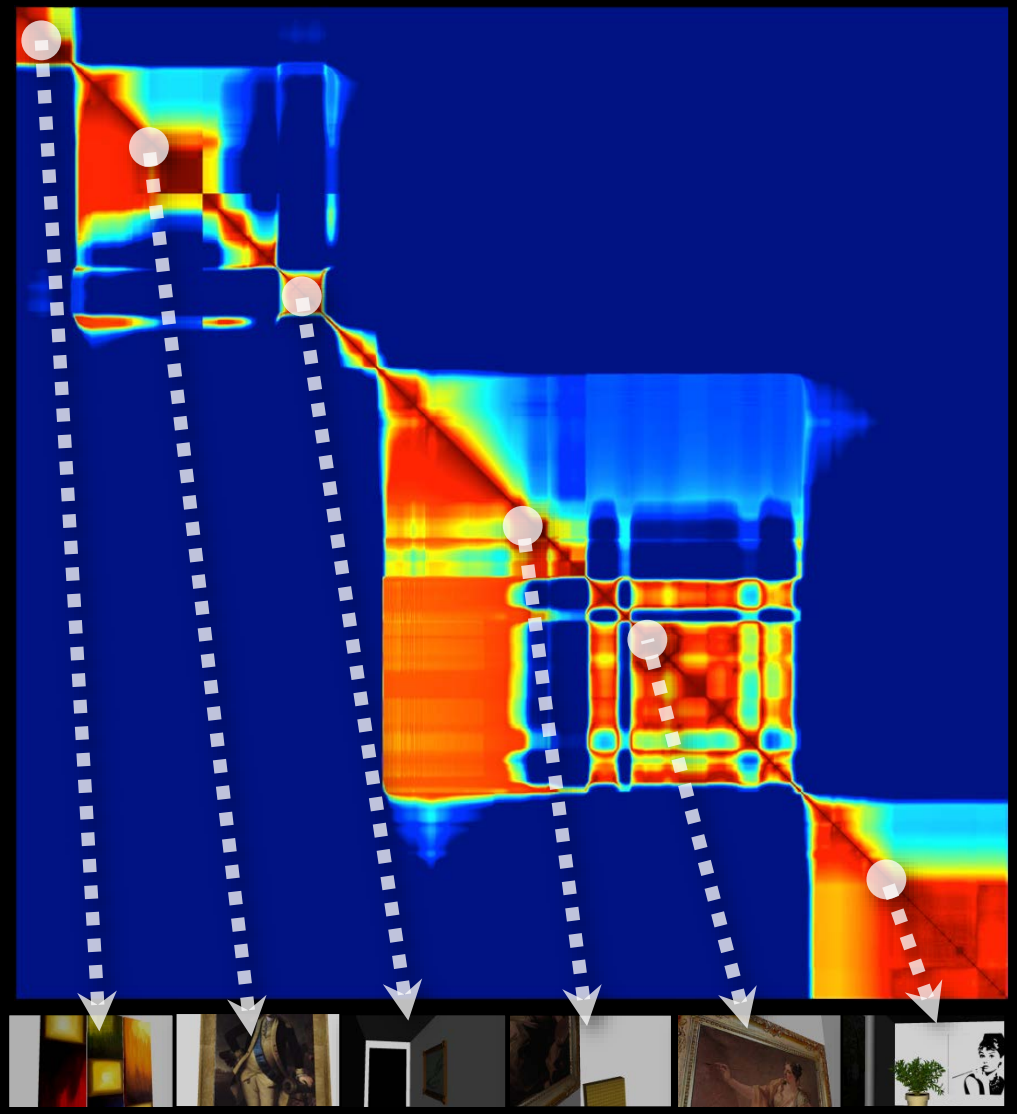
Content-Dependent Metric

Segmentation

- find key “fixations”
- serves as compact summary

Path Generation

Reproject Original Viewpoint



Key Ideas

Cinematography Model

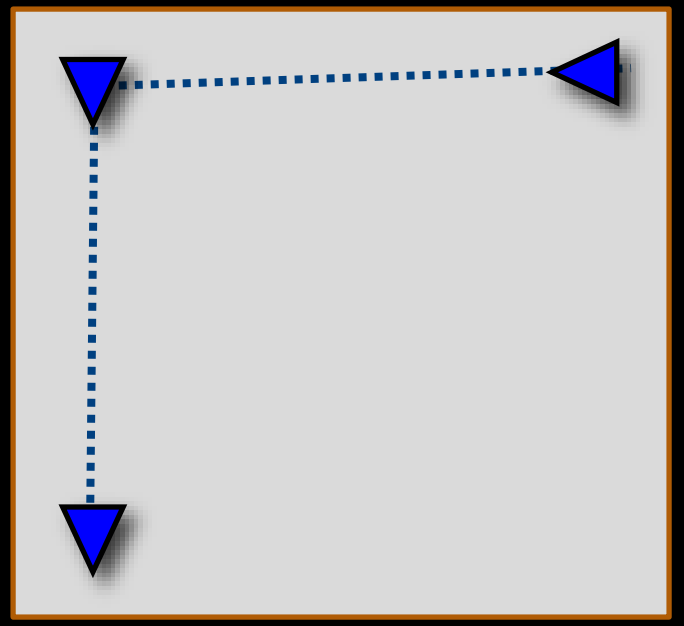
Content-Dependent Metric

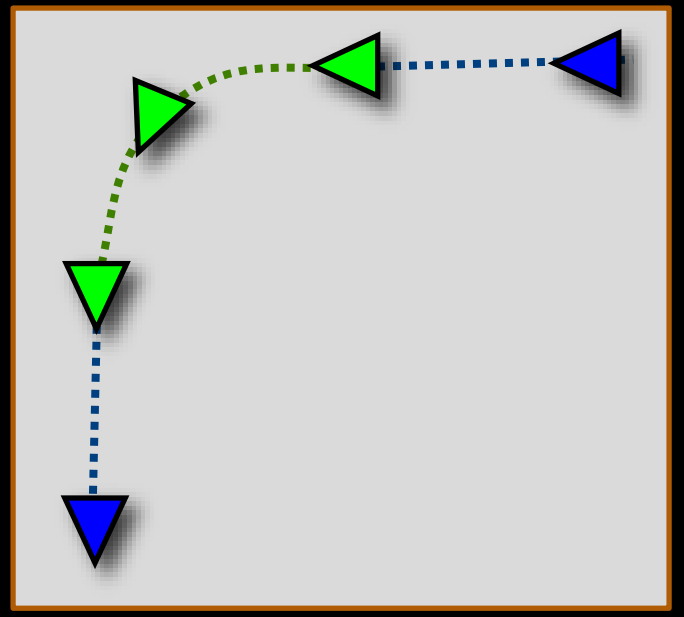
Segmentation

Path Generation

- cinematic smooth arcs
- exp. Coord interpolation

Reproject Original Viewport





Key Ideas

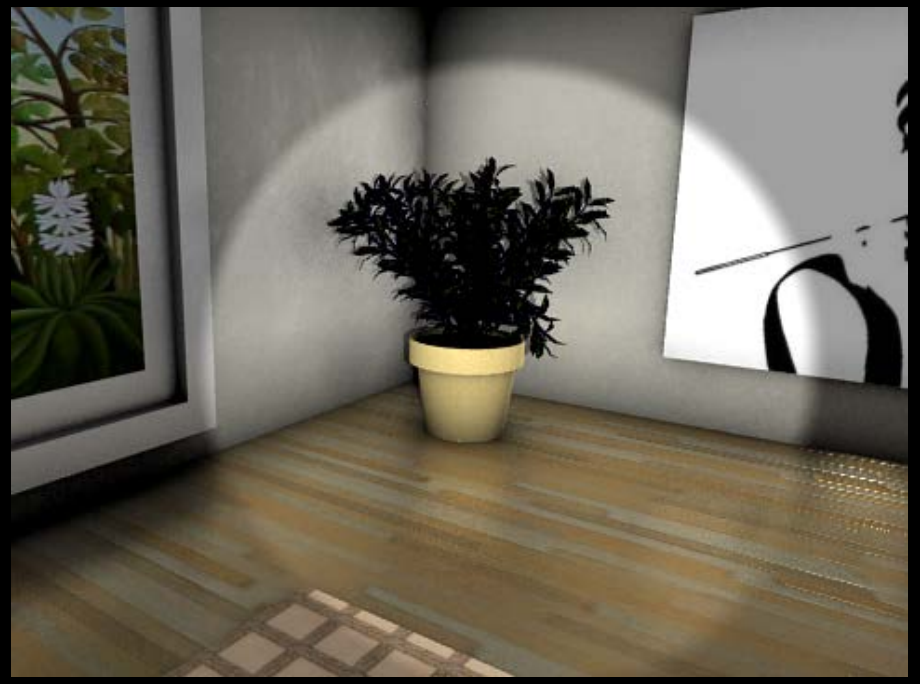
Cinematography Model

Content-Dependent Metric

Segmentation

Path Generation

Reproject Original Viewport



Original

Our Method

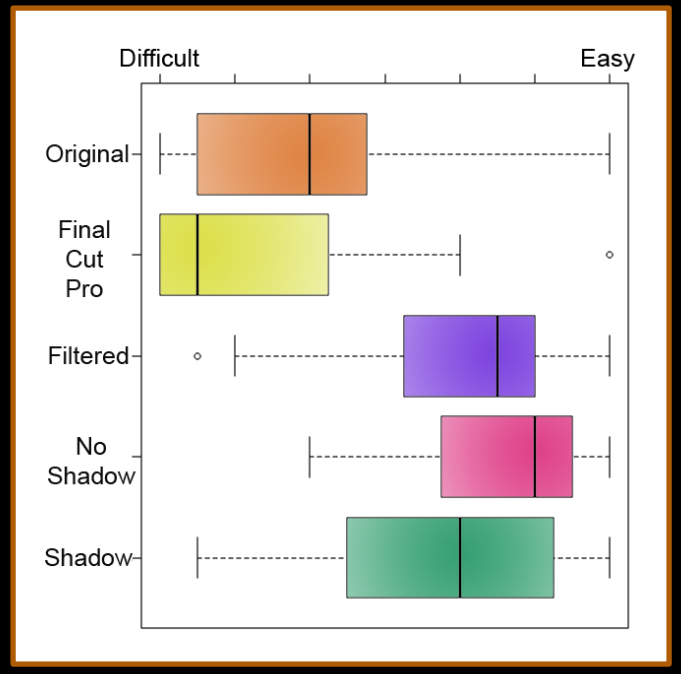
Evaluation

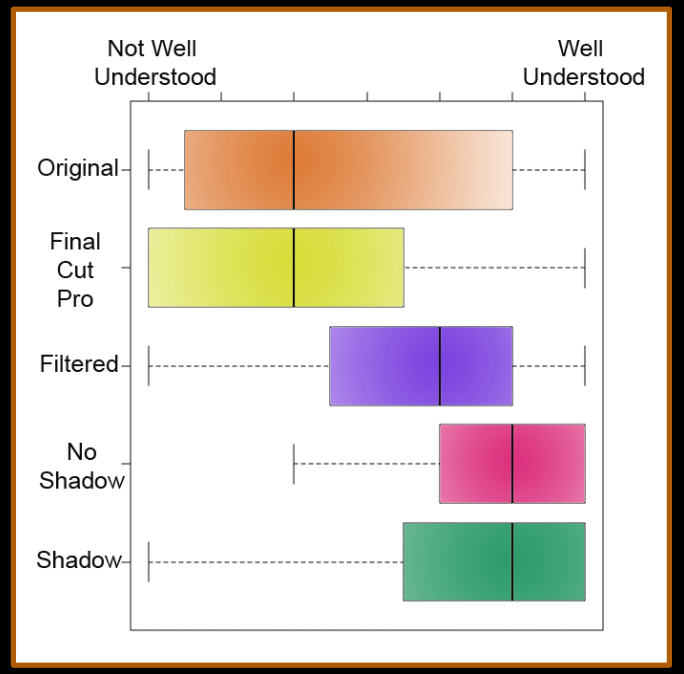
Three Initial Studies

Objective: VR Participant Viewed Objects

Subjective: Viewer Prefers

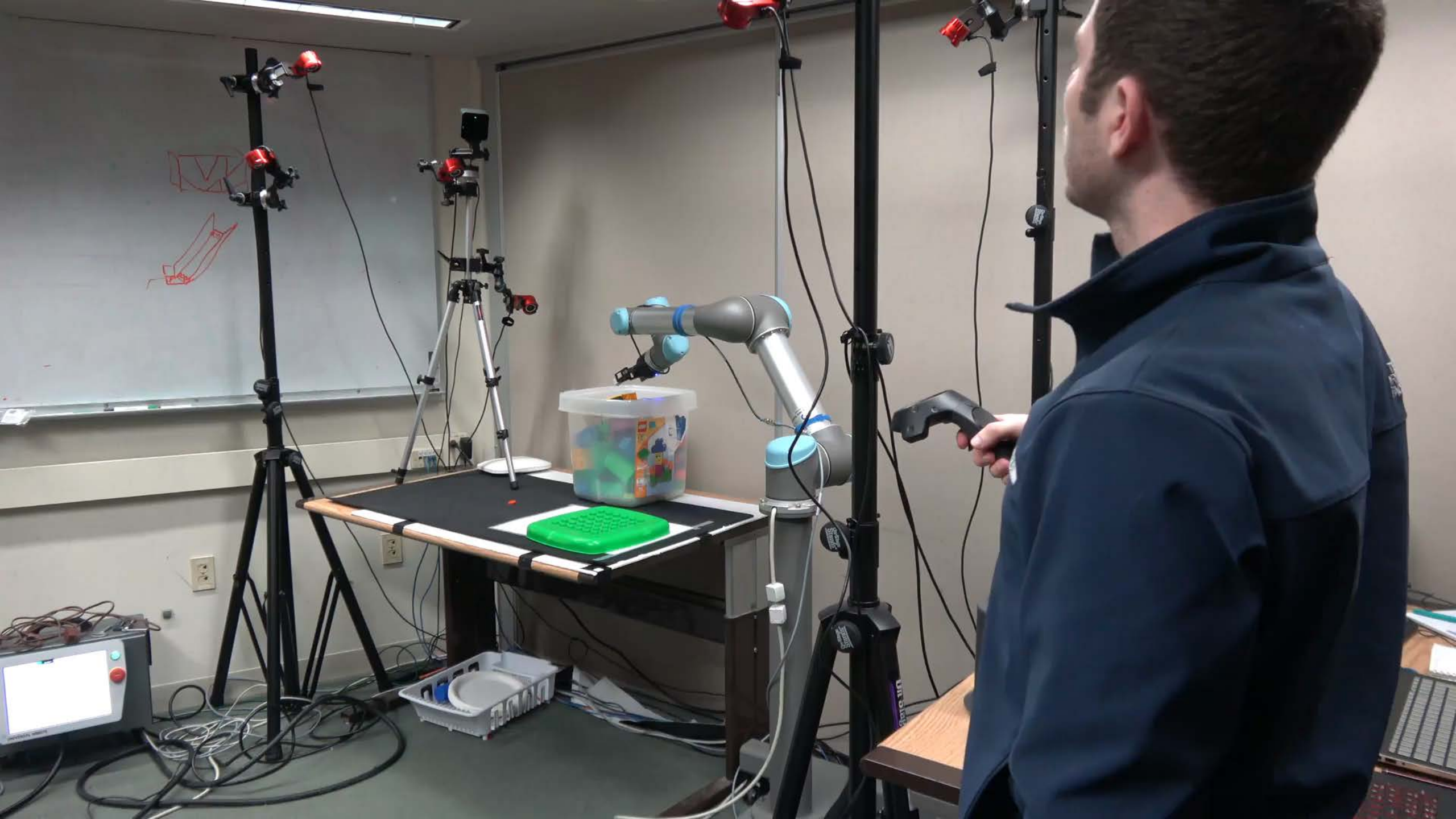
Objective: Viewer Comprehends





A Robotics Application

Warning this is really new – unpublished, unfinished, not yet worked out, …

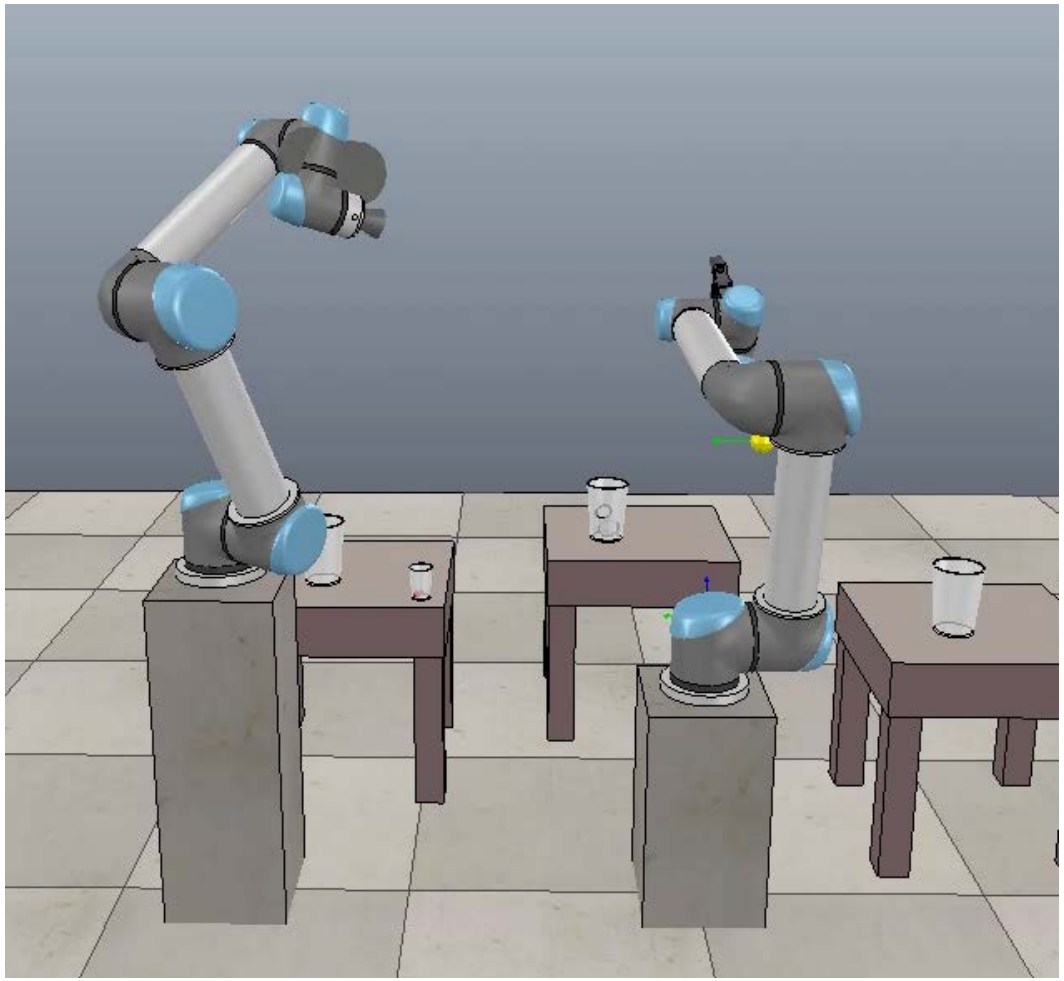


The Problem. . .

What to view to show when doing tele-operation

In interesting applications the view is remote

One robot watches the other...



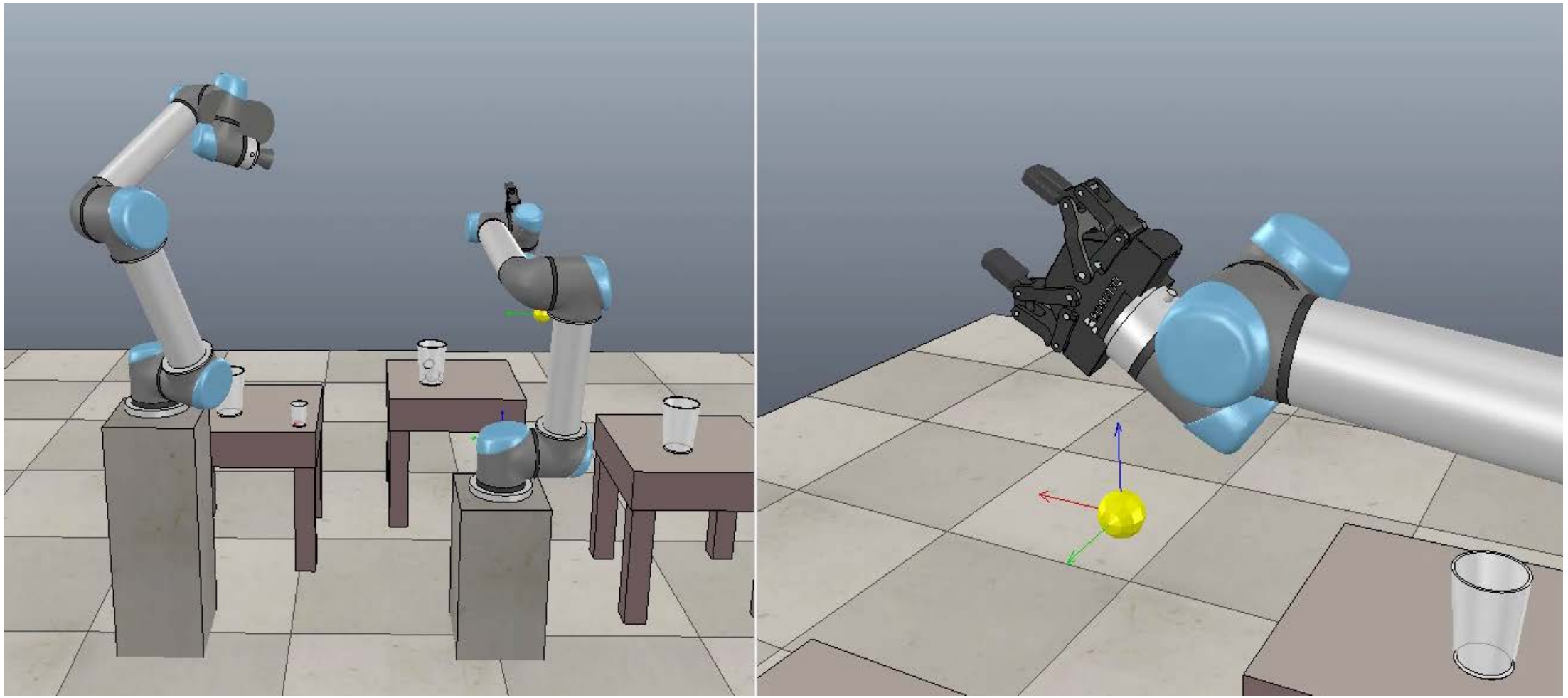
Get a second robot

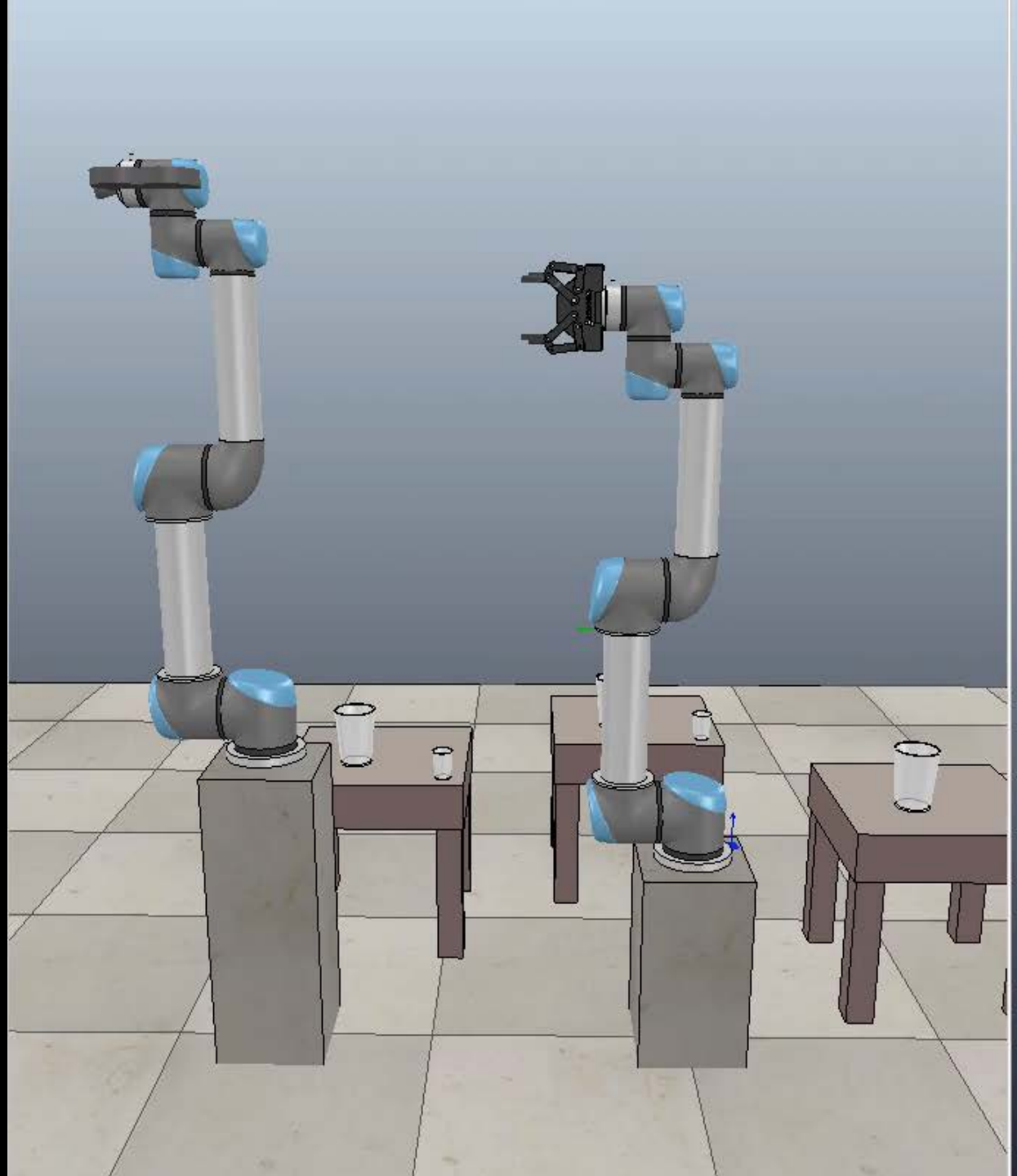
Put a camera in its hand

No need to pick “best view”

Need Through-the-Lens!

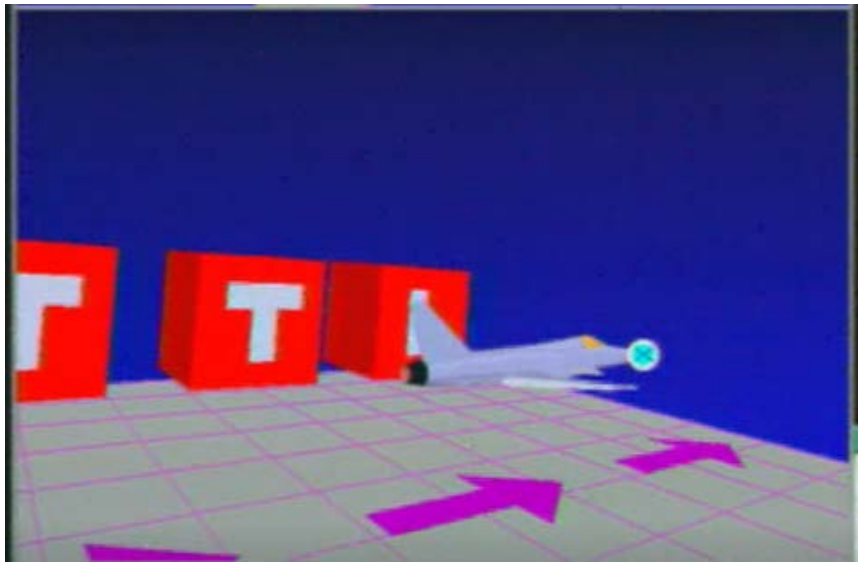
One robot watches the other...





What do you need to do this?

Through the lens controls! (keep your eye on the hand)



Good camera dynamics (don't disorient the viewer)

Prediction (where is the user going)

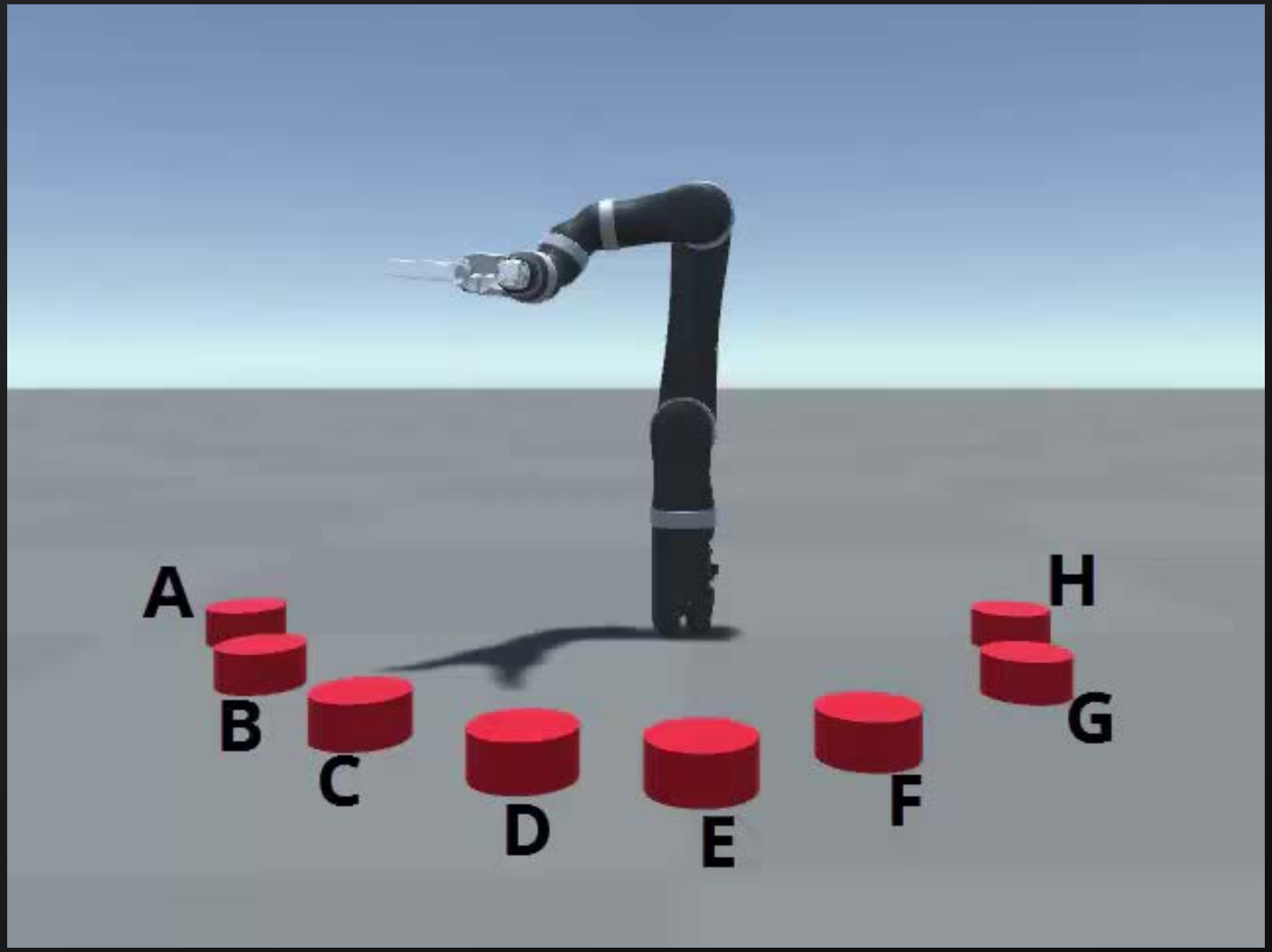
Occlusion avoidance (a big problem)

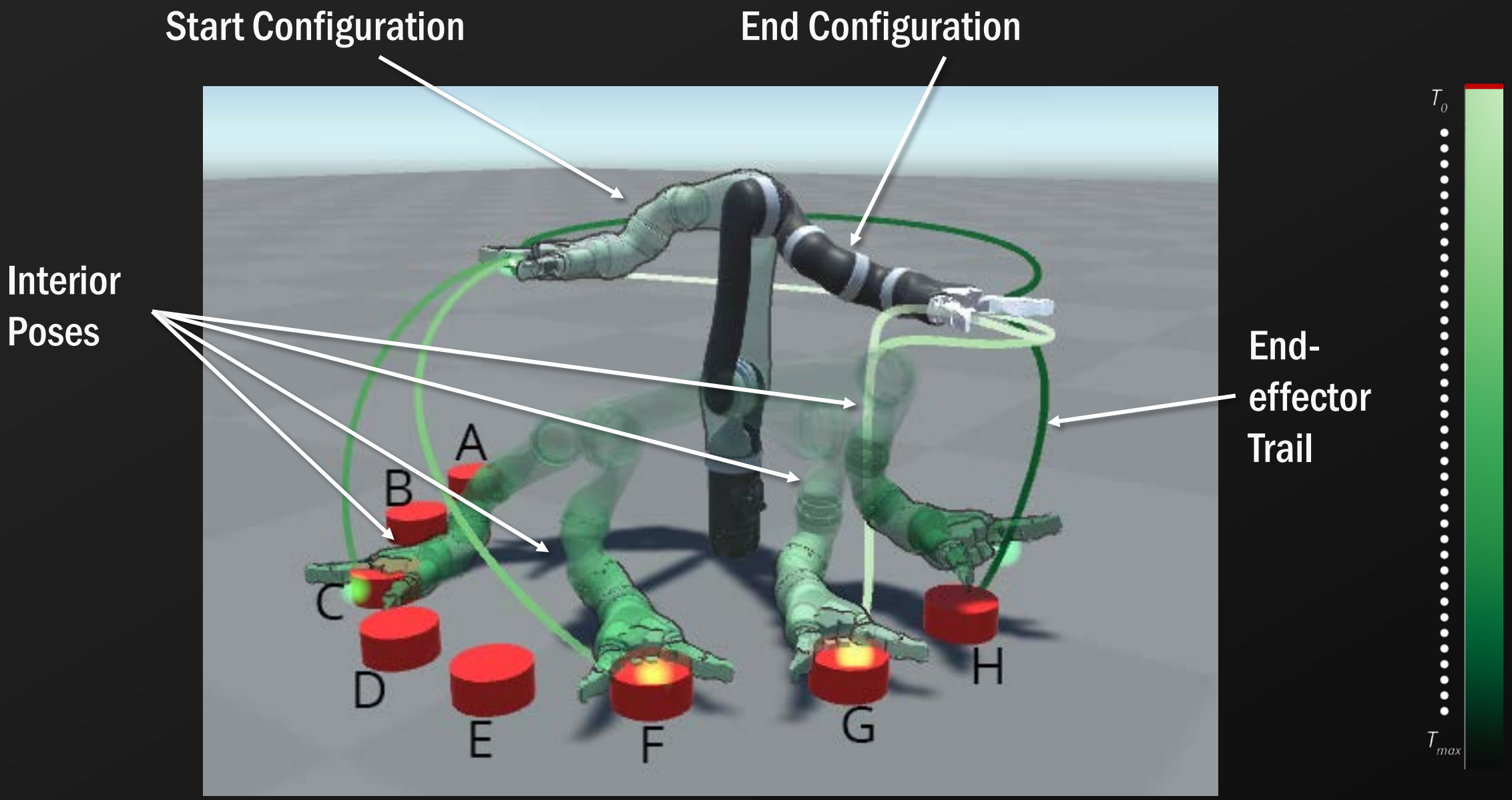
Low latency control

Payoff: useful tele-operation in complicated environments

Another robotics problem (that roboticists seem to ignore)

Rakita, Mutlu, Gleicher @ RO-MAN 2017


A
B
C
D
E
F
H
G


Start Configuration
End Configuration
Interior Poses
End-effector Trail
A
B
C
D
E
F
G
H
T_0
T_{max}

Summary?

You can learn some stuff from old papers

be careful, since computers change!

Old ideas inspire new ones

Thinking about how to look at things is useful

Thanks!

To you for listening.
To the organizers for inviting me.
To my mentors, students and collaborators.
To the funding sources over the years (NSF, UW, …).

Through the lens of 25 years… Through-the-Lens Camera Control Revisited

Michael Gleicher

University of Wisconsin Madison
gleicher@cs.wisc.edu