Visual & Temporal Fidelity

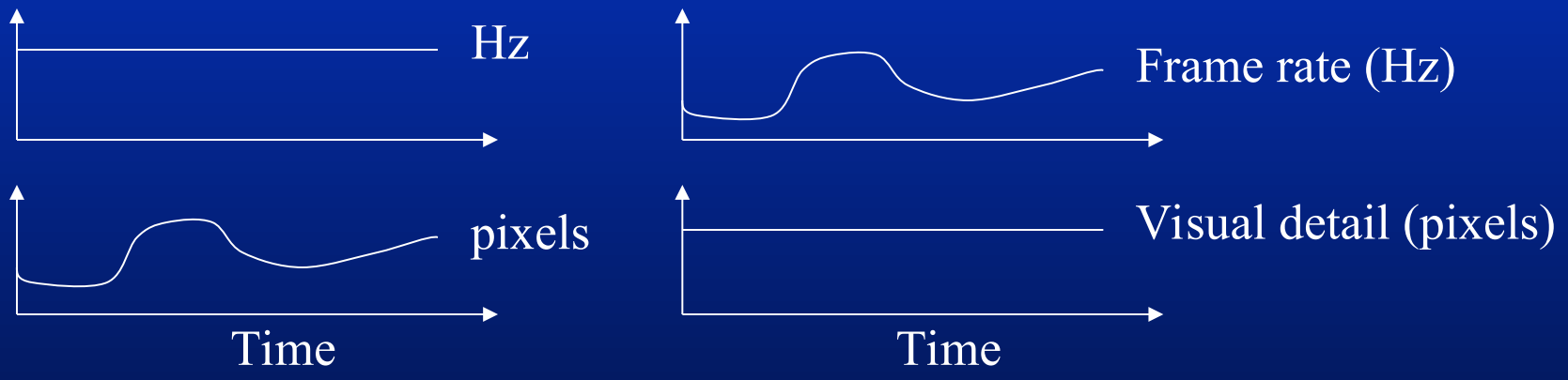
Benjamin Watson

Dept. Computer Science
Northwestern University

watson@cs.northwestern.edu

SAN ANTONIO
SIGGRAPH
2002

Visual-temporal tradeoff



LOD trades off visual & temporal fidelity

We examine both in isolation

Briefly discuss them in unison

What's coming

- Visual fidelity
  - Experimental studies
  - Semiautomatic simplification
- Temporal Fidelity
  - Measurement & control
  - Importance to users
- Visual vs. Temporal Fidelity

Experiment: *motivation*

We use visual fidelity measures

- *During simplification*
- *During runtime LOD control*

So when measuring fidelity

- *How well are we doing?*
- *How might we do better?*

Experiment: *overview*

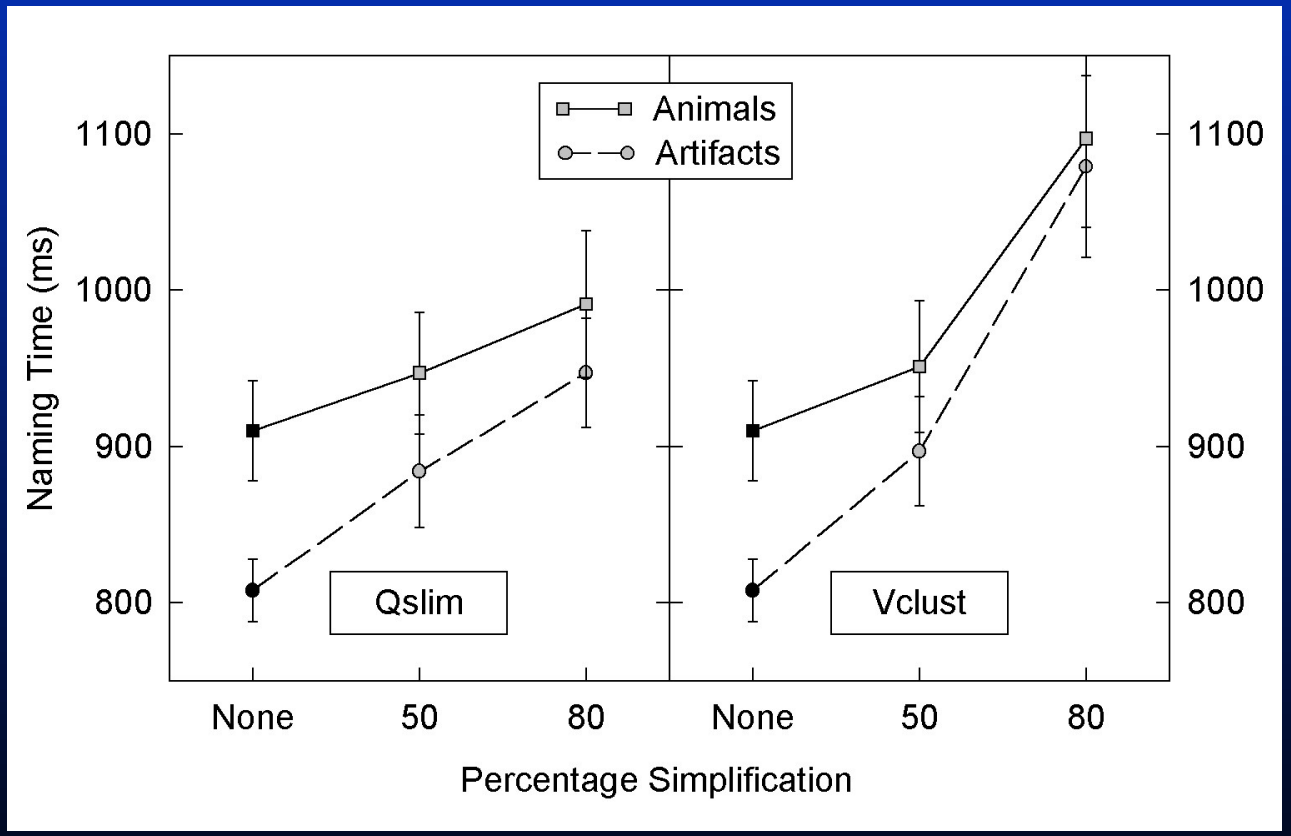
So we ran an experiment

- *36 people looked at 36 models*
- *Simp'd with 2 algorithms*: *QSlim, Cluster*
- *Responded with naming times, ratings*

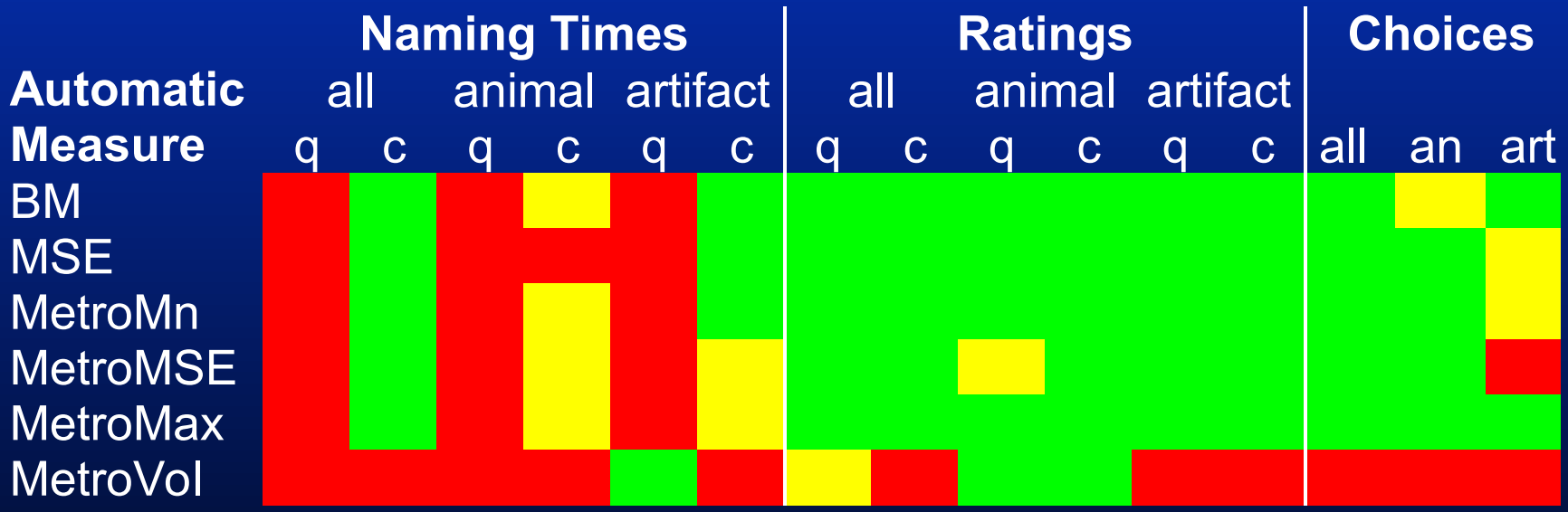
Compared to automatic measures

- *Metro, image MSE, Bolin & Meyer*

Experiment: *results*



Experiment: *results*



Stat Sig Correlation

Correlation < .2

Stat Insig Corr > .2

SAN ANTONIO SIGGRAPH 2002

Experiment: *implications*

For simplification:

- *The hard part is producing small models*
- *Quadric simplification is better - curvature?*
- *Effectiveness varies by model type - specialize?*
- *Some smaller models are **better** - distillation?*

For runtime LOD:

- *Simple image based measures work well*
- *Do projected distance measures work?*

Semisimp: *motivation*

Problems with automatic simplification

- *No sensitivity to semantics*
- *No knowledge of target application*
- *Very little user control*
- *Poor quality at small output sizes*

Solution: semiautomatic simplification

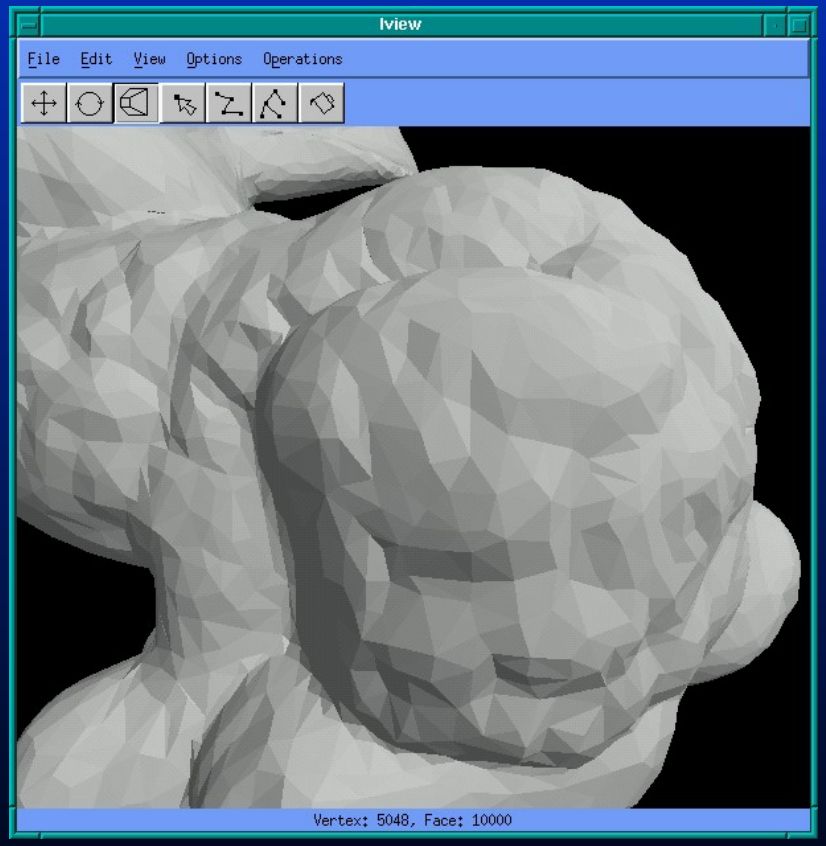
`Semisimp`: *overview*

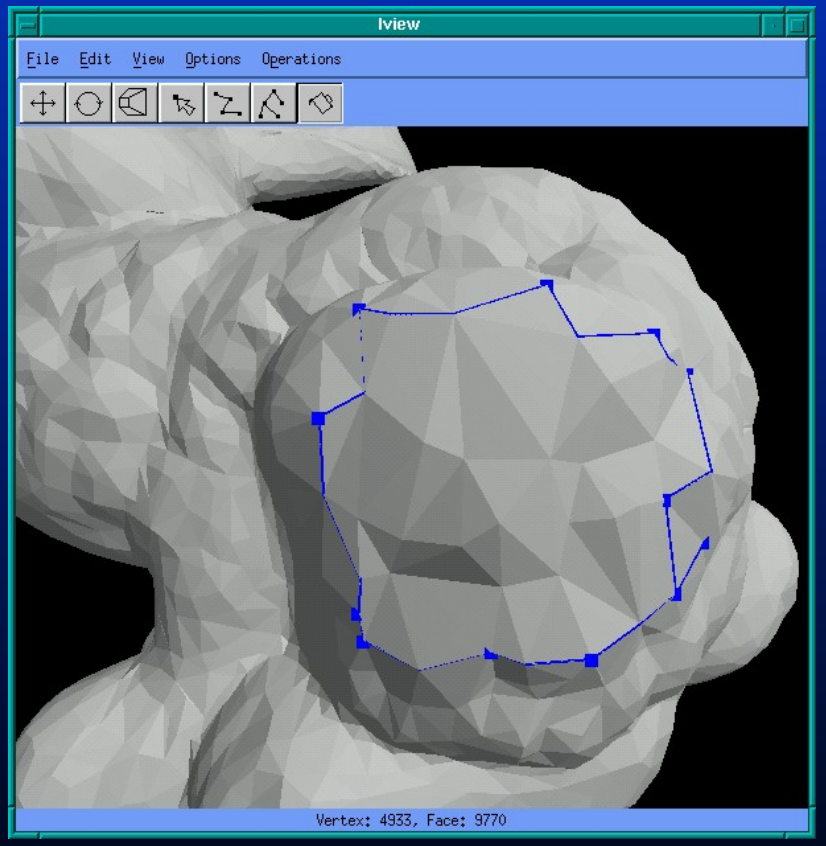
New tool `semisimp` allows user to

- *Reorder primitive simplifications*
- *Reshape simplified model patches*
- *Edit simplified model, propagate results*

All with an automatic assist

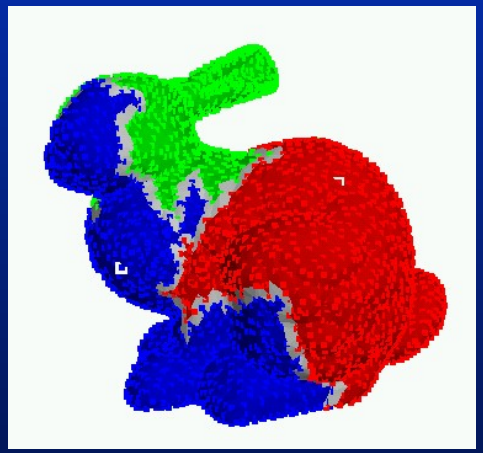
Semisimp: *reordering*



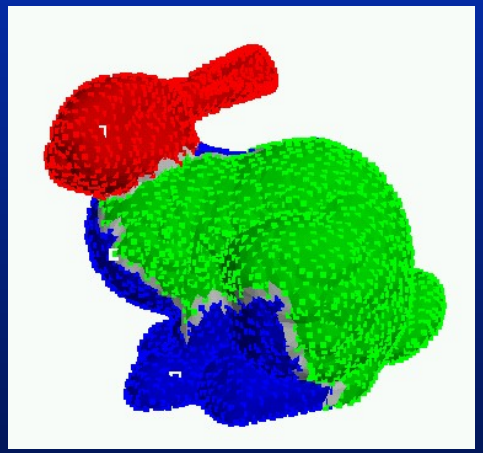


Semisimp: *patch reshaping*

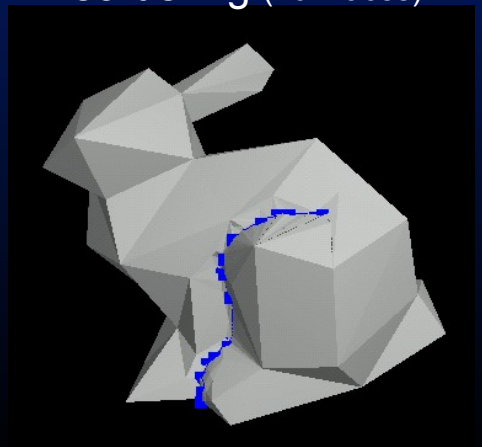
automatic patches



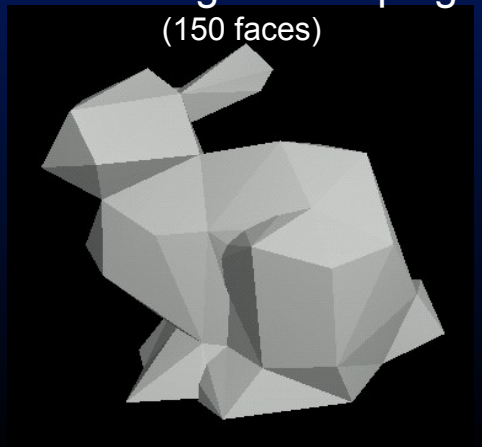
user's patches



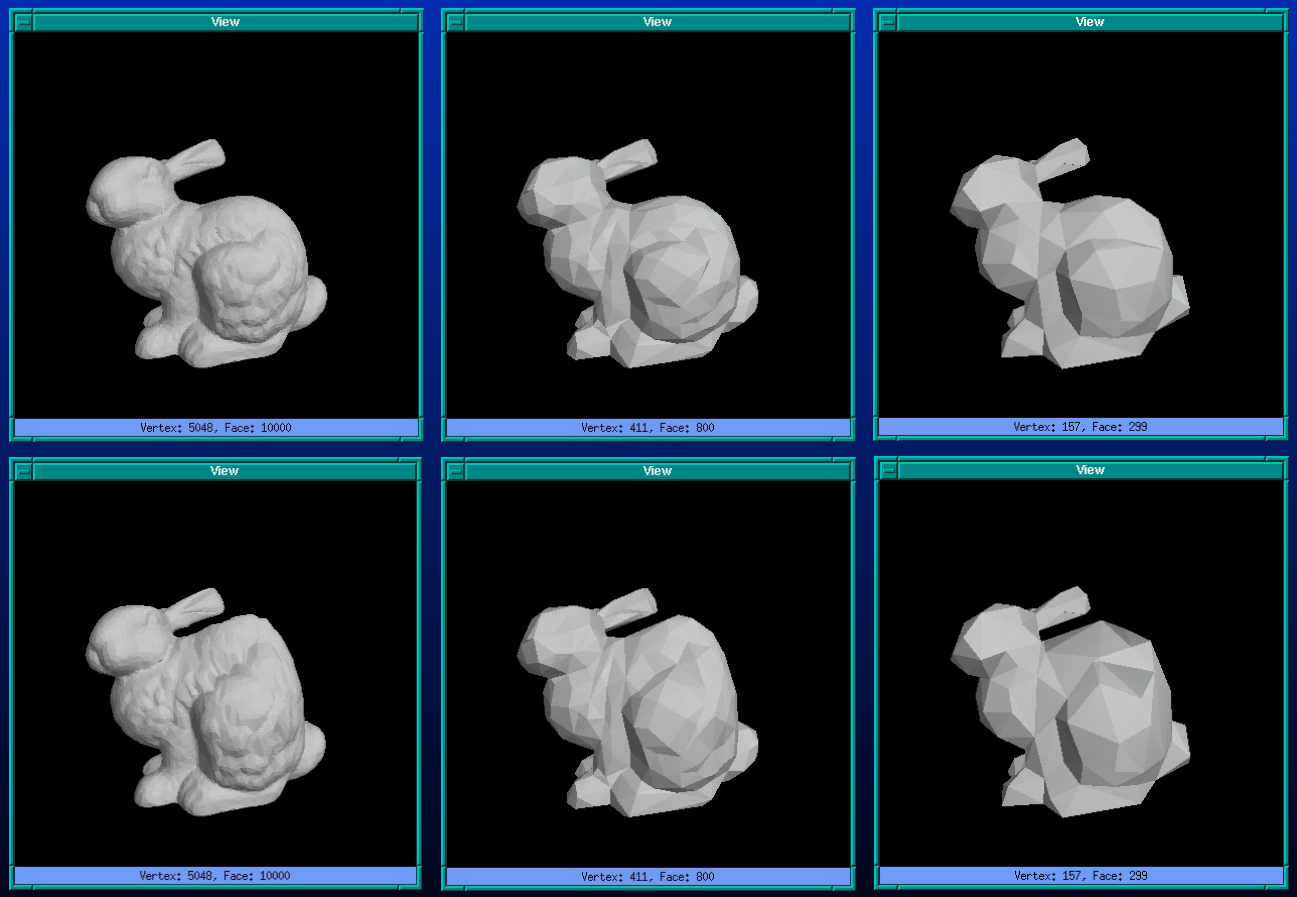
reordering (262 faces)



reordering & reshaping
(150 faces)



Semisimp: *editing*



before editing

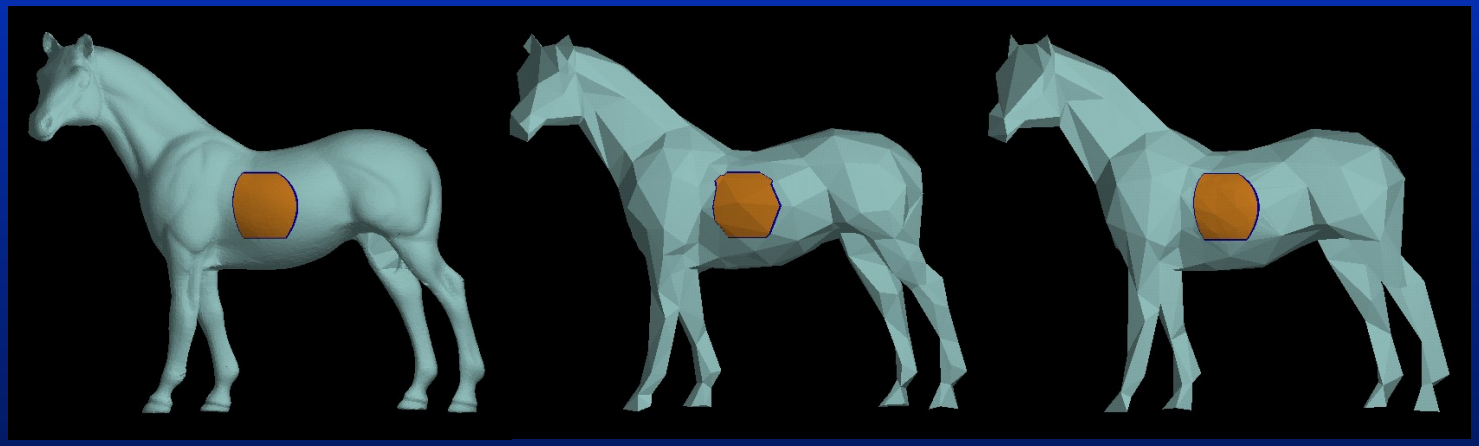
after editing

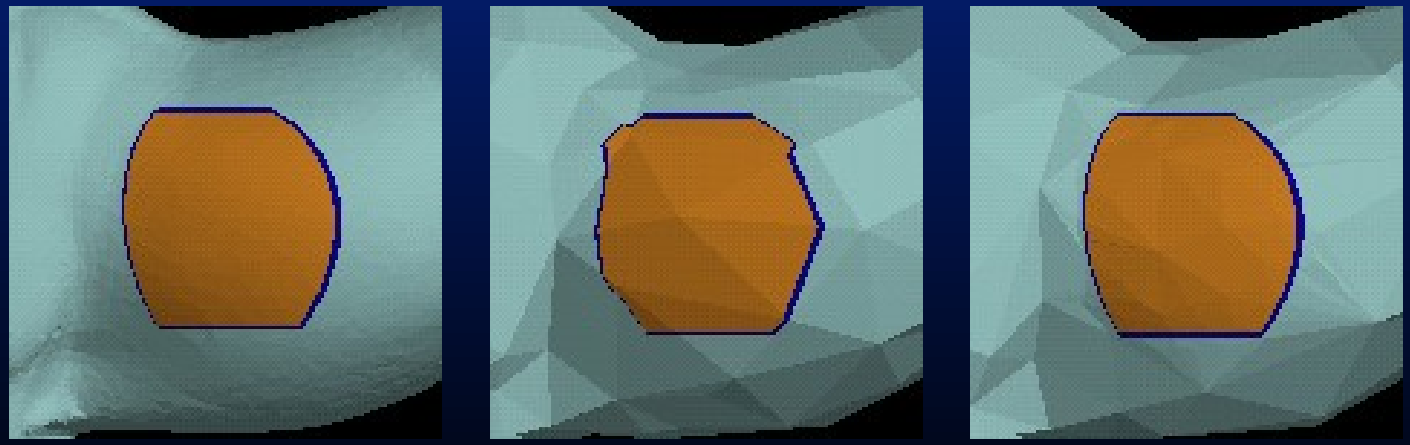
fine propagation
(10000 faces)

edited LOD
(800 faces)

coarse propagation
(300 faces)

Semisimp: *example use*





original
(30000 faces)

automatic
(588 faces)

semiautomatic
(588 faces)

Progress

Visual fidelity

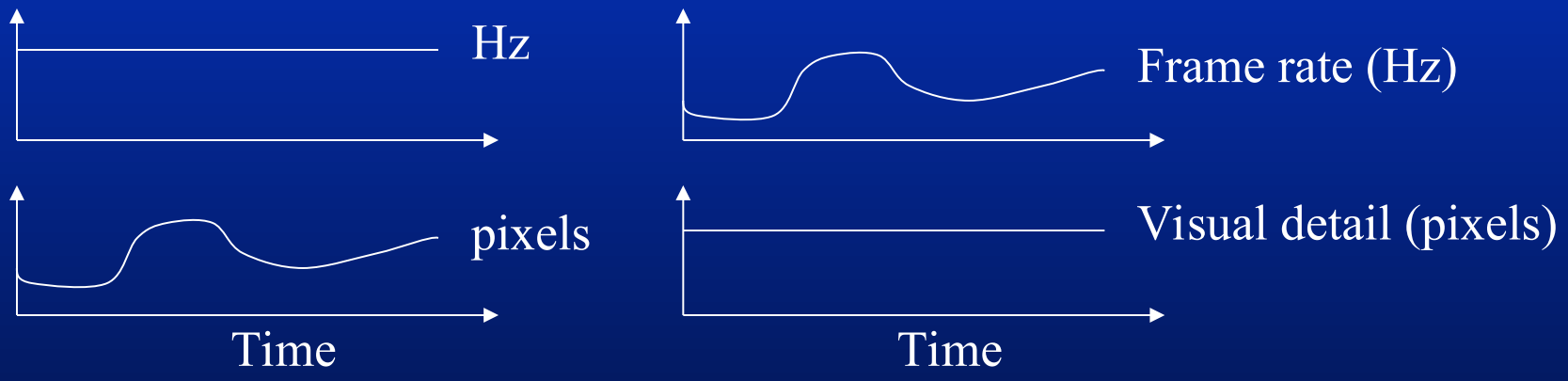
- *Experimental studies*
- *Semiautomatic simplification*

Temporal Fidelity

- Measurement & control
- Importance to users

Visual vs. Temporal Fidelity

Temporal fidelity



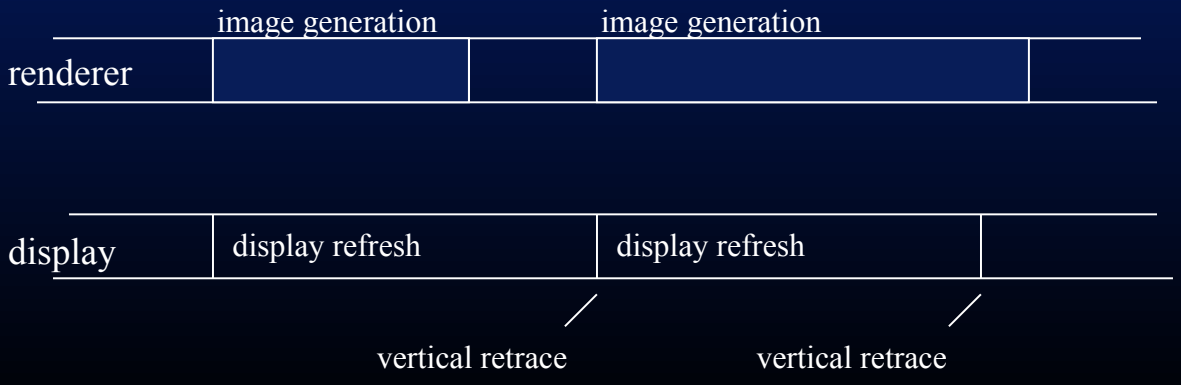
What about the temporal side of tradeoff?

Measures: *frame rate*

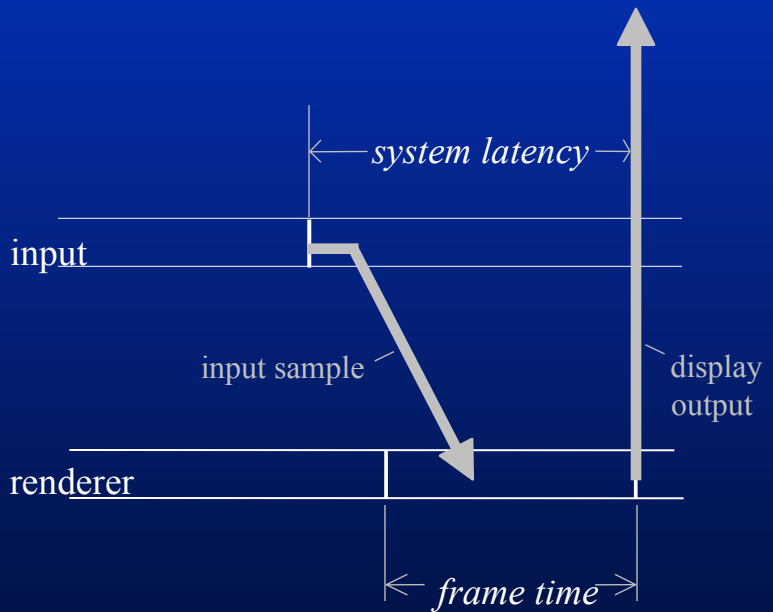
Frame time: inverse of frame rate

Refresh time: time for display refresh

- *Frame time a multiple of refresh time*
- *Mean frame time may not be*



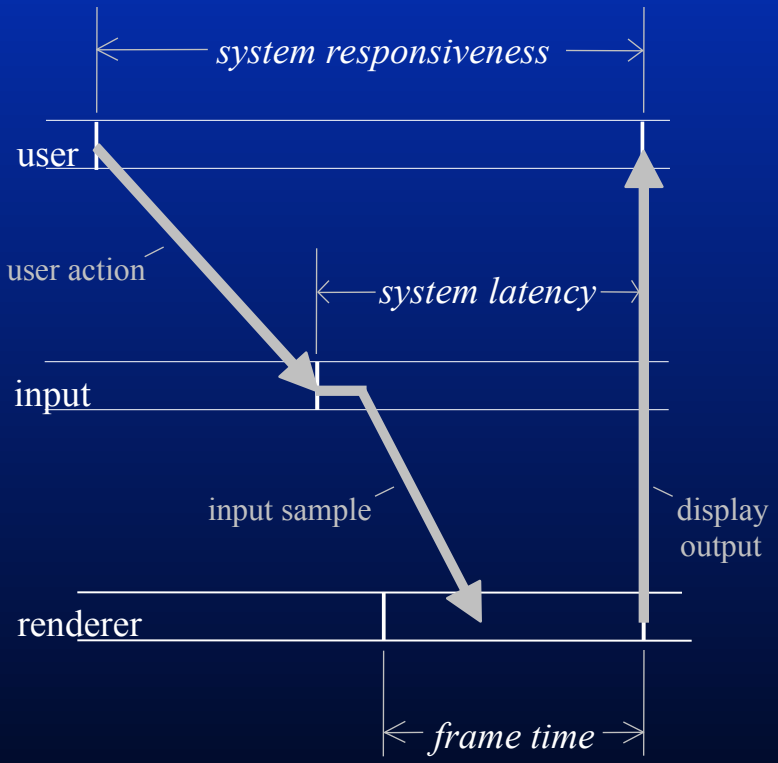
Measures: *latency*



System latency:

- *age of displayed sample*
- *some frame time + input gather time*

Measures: *responsiveness*



System responsiveness (SR):

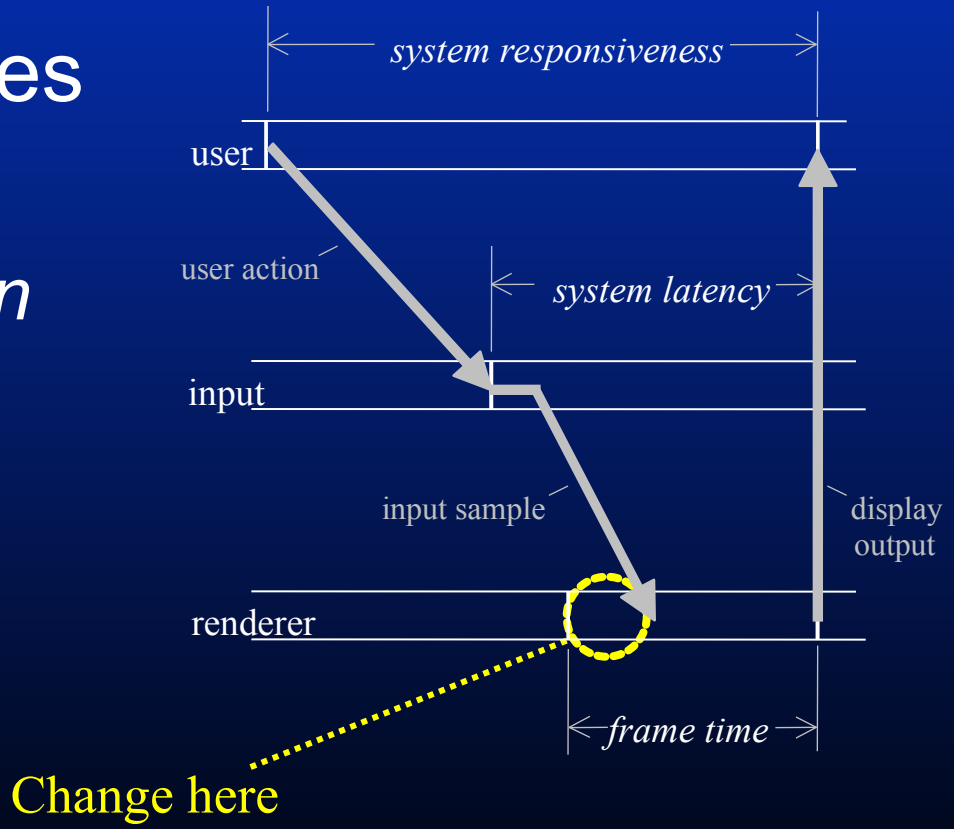
delay from input to display

latency + delay from action to sample

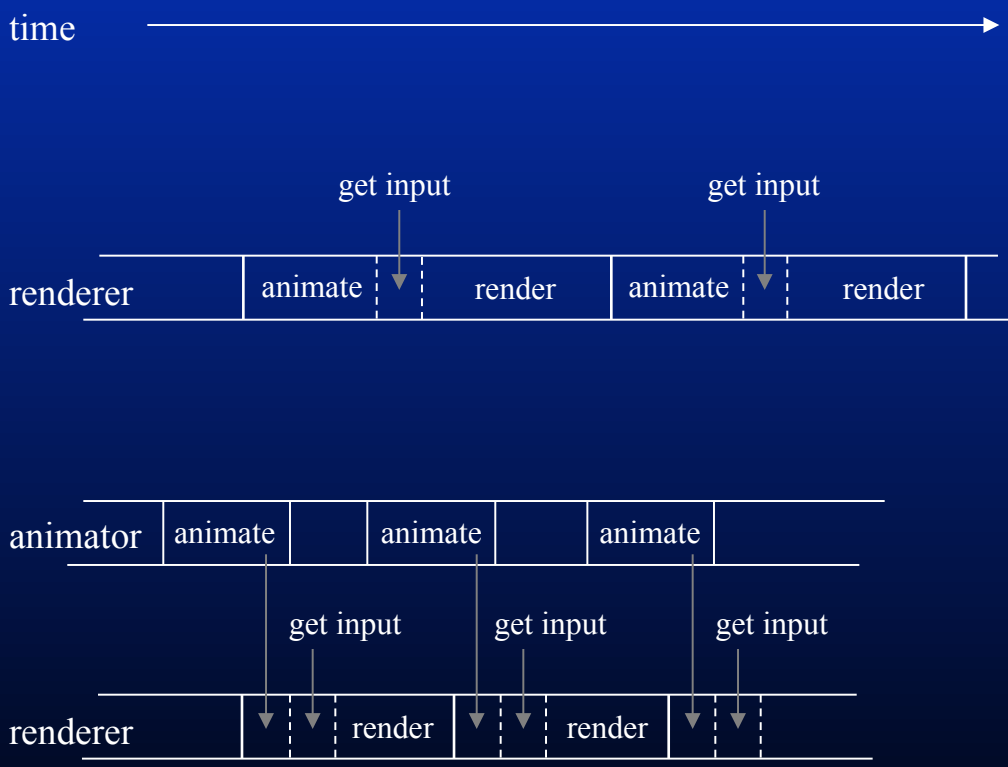
Control: *frame-only (FO)*

Only frame time changes

- *SR += 1/2 FO change*
- *e.g. parallelize animation*
- *e.g. parallelize collison*
- *e.g. motion LOD*



Control: *frame-only (FO)*



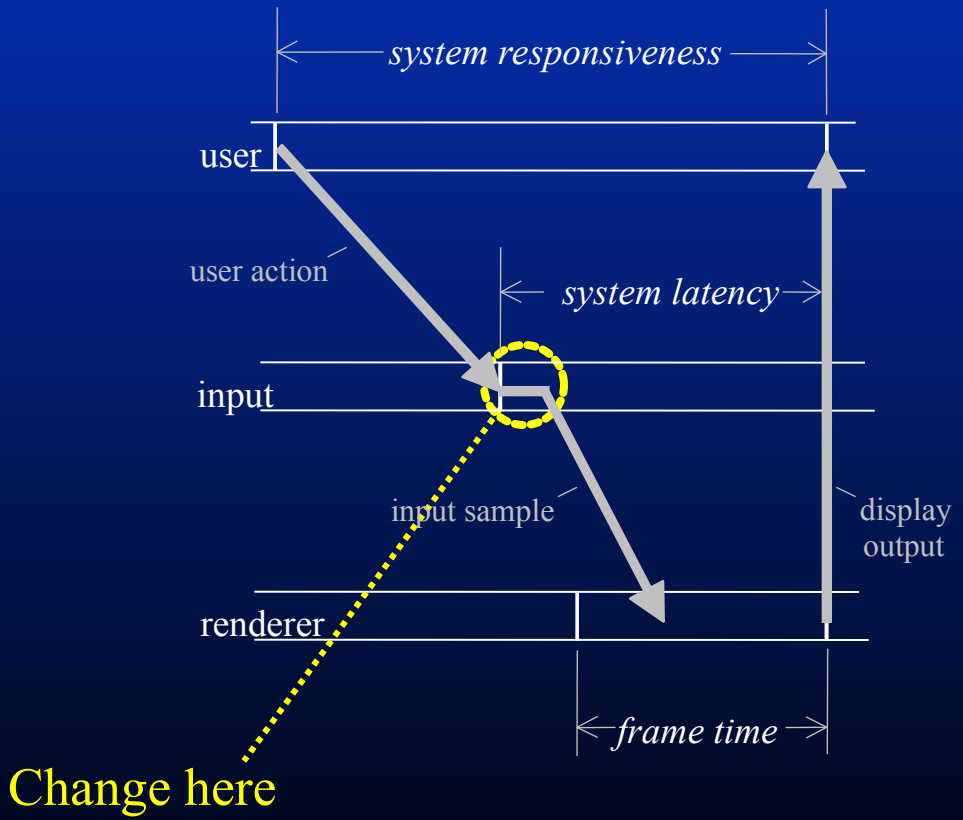
sequential animation

parallel animation

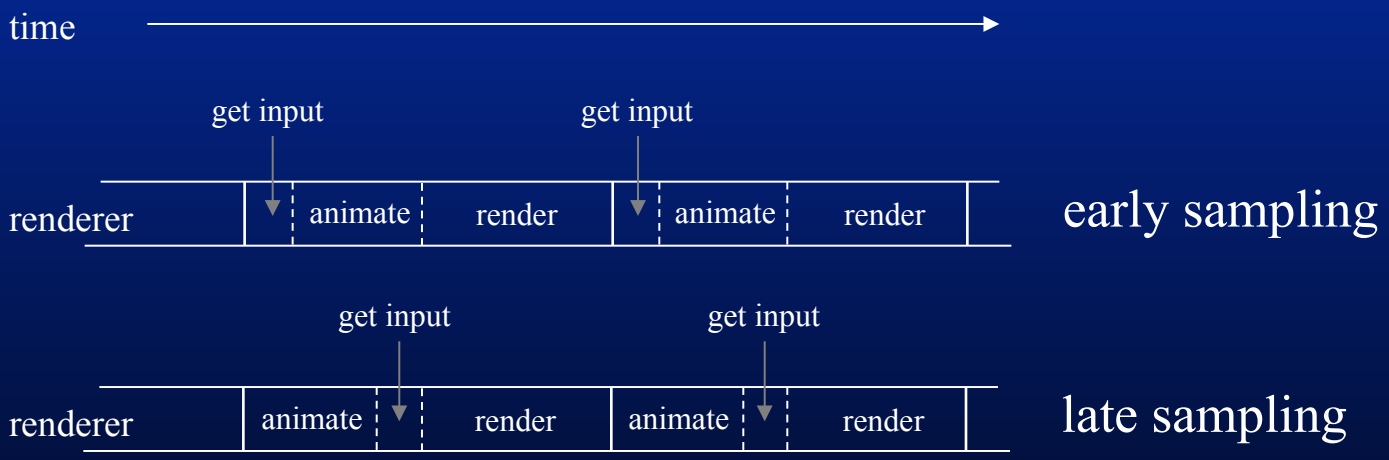
Control: *latency-only (LO)*

Only latency changes

- *SR += LO change*
- *e.g. prediction*
- *e.g. late sampling*



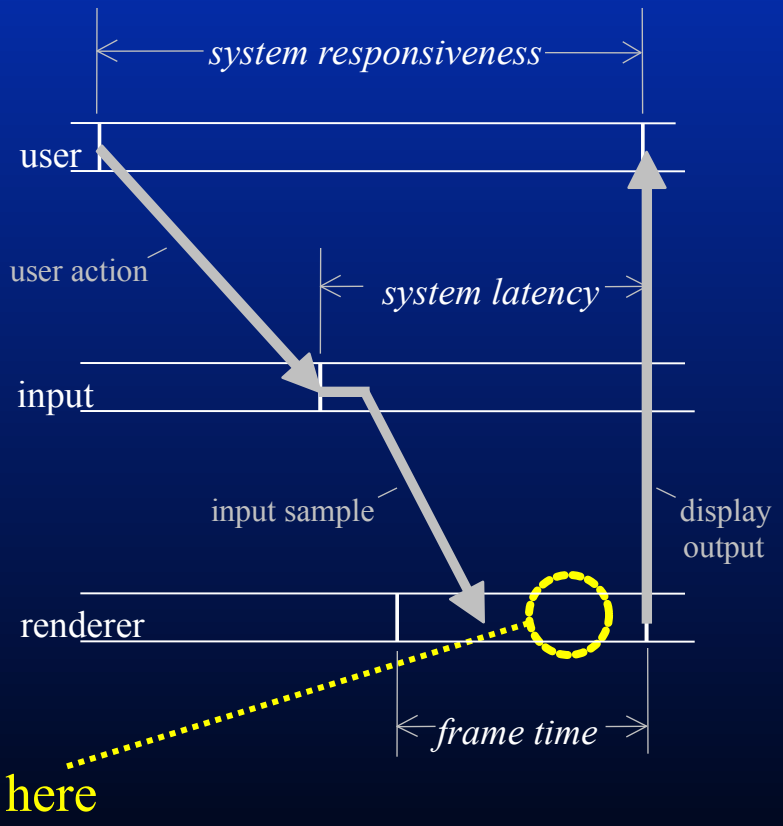
Control: *latency-only (LO)*



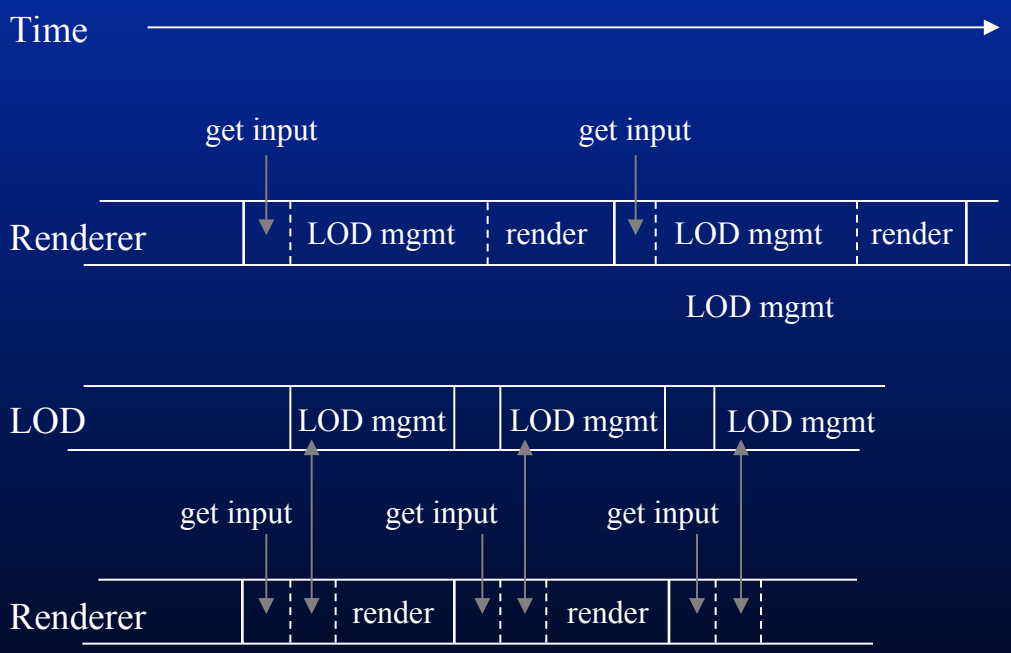
Control: *frame-latency (FL)*

Frame time & latency chg

- *SR += 3/2 FL change*
- *e.g. LOD*
- *e.g. parallelize LOD mgmt*
- *e.g. vertex caching*
- *e.g. other rendering optims*



Control: *frame-latency (FL)*



sequential
LOD mgmt

parallel
LOD mgmt

SAN ANTONIO
SIGGRAPH
2002

Control: *summary*

A one ms temporal manipulation brings:

Manipulation	Abbr	Ftm	Ltcy	SR
Frame only	FO	1 ms	0 ms	0.5 ms
Latency only	LO	0 ms	1 ms	1.0 ms
Frame latency	FL	1 ms	1 ms	1.5 ms

Speedups: FL most effective (esp. if no tradeoff)

Overhead: FO least harmful

Progress

Visual fidelity

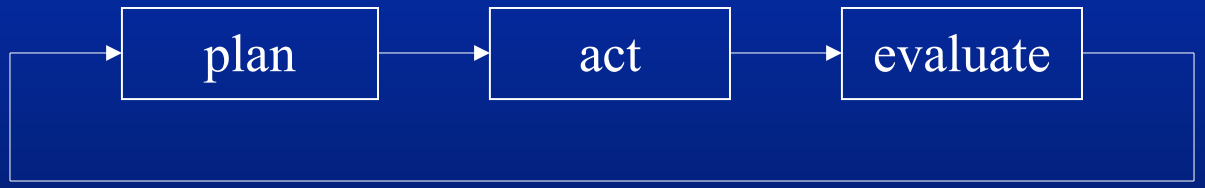
- *Experimental studies*
- *Semiautomatic simplification*

Temporal Fidelity

- *Measurement & control*
- Importance to users

Visual vs. Temporal Fidelity

Usability: *closed loop*



Closed loop tasks require regular feedback

- *e.g. real world: driving, watering with a hose*
- *e.g. computer: drawing a line, navigating in 3D*

Particularly sensitive to temporal detail

Usability: *open loop*

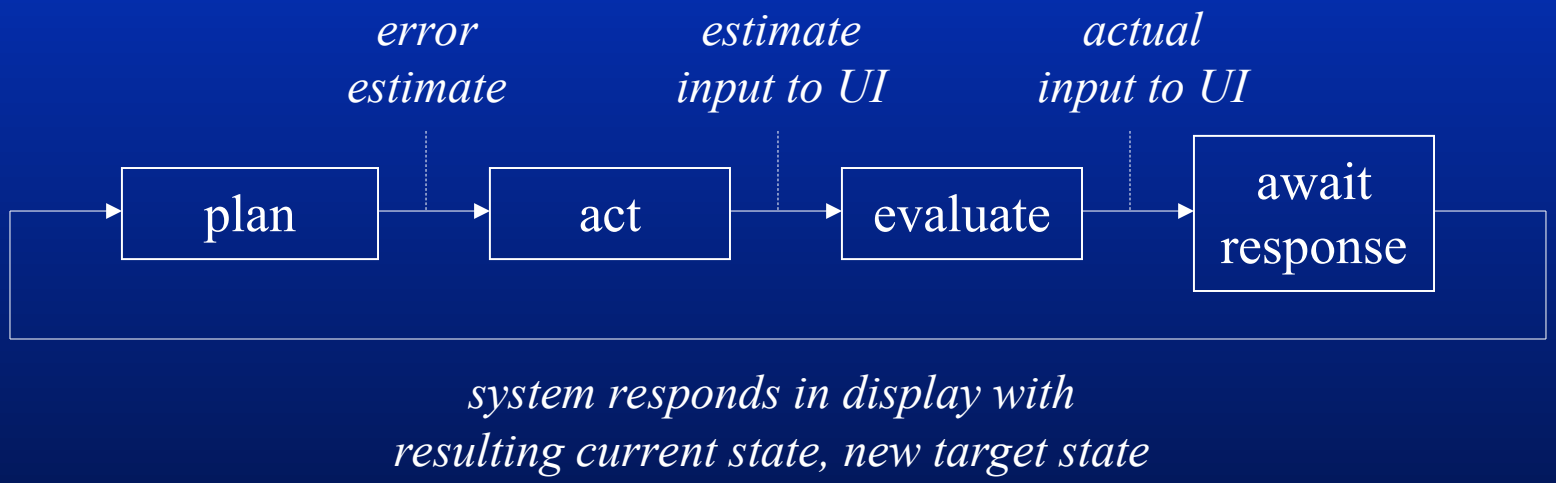
Open loop tasks require little/no feedback

- *e.g. real world: assembly line, pitching*
- *e.g. computer: typing, clicking known button*

Limited sensitivity to temporal detail

In fact, open-closed is a continuum

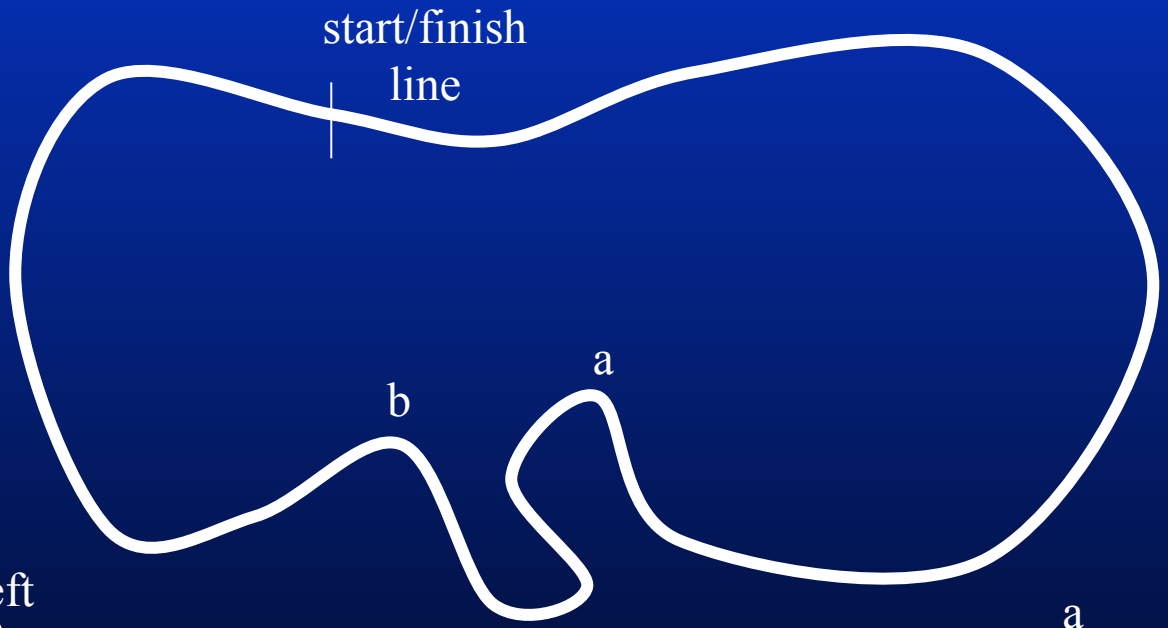
Usability: *dynamic control*



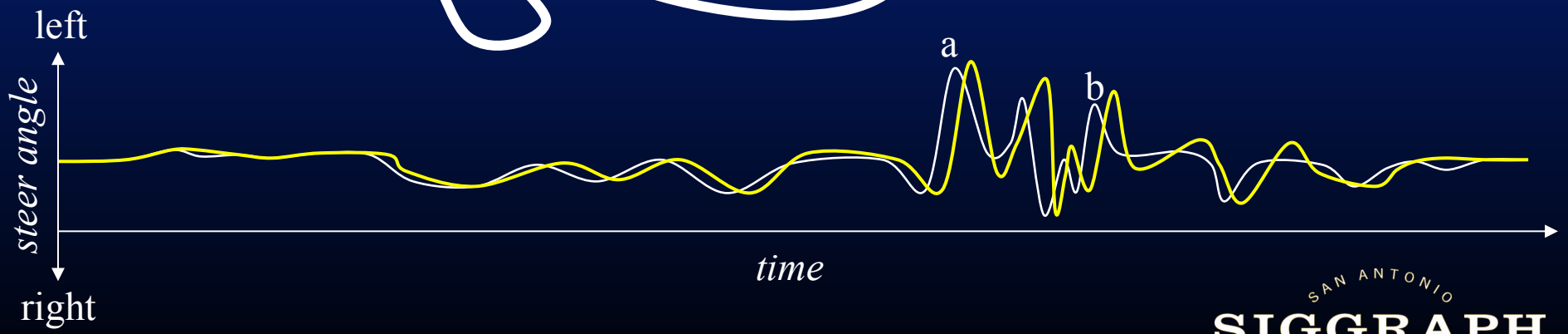
Closed loop tasks: a form of dynamic control

- *Minimizing error between system, target states*
- *States vary, w/ component freqs*

Usability: *dynamic control*



High freqs are more difficult



Usability: *dynamic control*

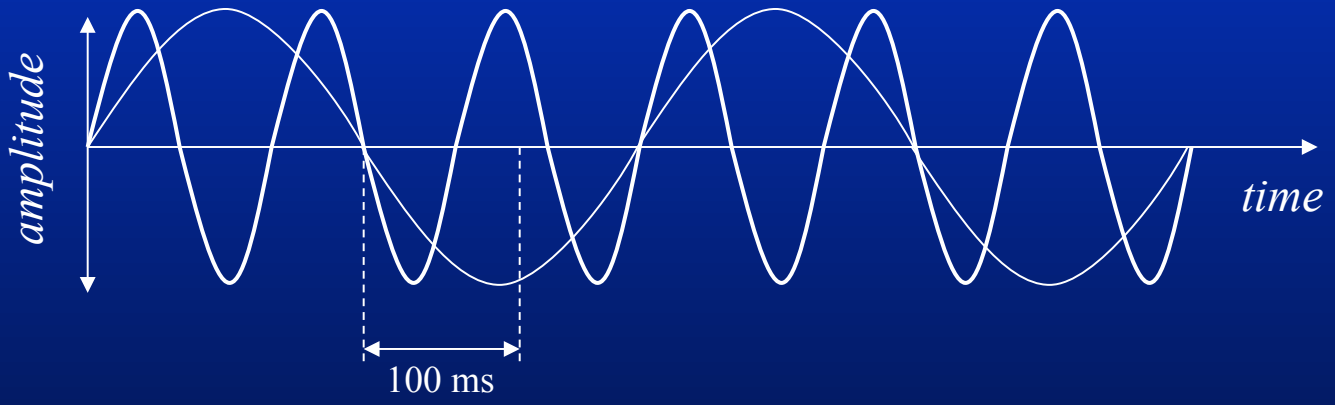
Successful dynamic manual control minimizes

- *Error*
- *System effort*
- *Human effort*

But most basic requirement:

- *Achieves stability - at least some tracking success*

Usability: *phase lag*



SR is measured in terms of phase lag

Degs of phase SR introduces between two signals

Note relation of temporal detail to task

Same SR can have diff't phase lags

Usability: *stability*

Interface controls are characterized by order

- *Zero order: in value = out value, e.g. mice, volume*
- *First order: in value = out velocity, e.g. joysticks*

And by gain

- *A ratio: value out / value in -- amplify or damp*

A system is stable when

- *Phase lag $\leq$ 180° and gain $\leq$ 1*

Usability: *better control*

Control alteration

Reduce order, reduce gain

Task alteration

Reduce difficulty and target frequencies

Display alteration

Prediction of target or previewing of system

E.g. in race game, better view of road

Usability: *complex tasks*

Control theory useful, but:

- *People aren't linear systems*
- *Tasks are partially open loop*
- *Tasks contain many target frequencies*
- *What about frame rates?*
- *What about visual detail?*

Must examine usability in applied settings

Usability: *complex tasks*

Research has studied three types of tasks:

- *Catching - primarily open loop*
- *Placement - primarily closed loop*
- *Tracking - very closed loop*

Studies in both 2D and 3D graphics env’ts

Usability: *complex tasks*

Catching

- *Ceiling in usability reached as frame rate improved*
- *Large (100 ms) std devs in frame rate harmful*

Placement

- *No ceiling reached; 100 ms std dev harmful*

Tracking

- *No ceiling reached; Hz also had effect*

Usability: *applied issues*

Hard to have “enough” temporal detail

- *Can have enough on open loop tasks*
- *Else 40 ms SR is harmful, 15 ms is perceived*

Temporal detail variation a minor concern

- *Avoid large (100 ms), low freq or transient change*

Responsiveness more important than Hz

- *Prediction is sensitive to frame rate*

Progress

Visual fidelity

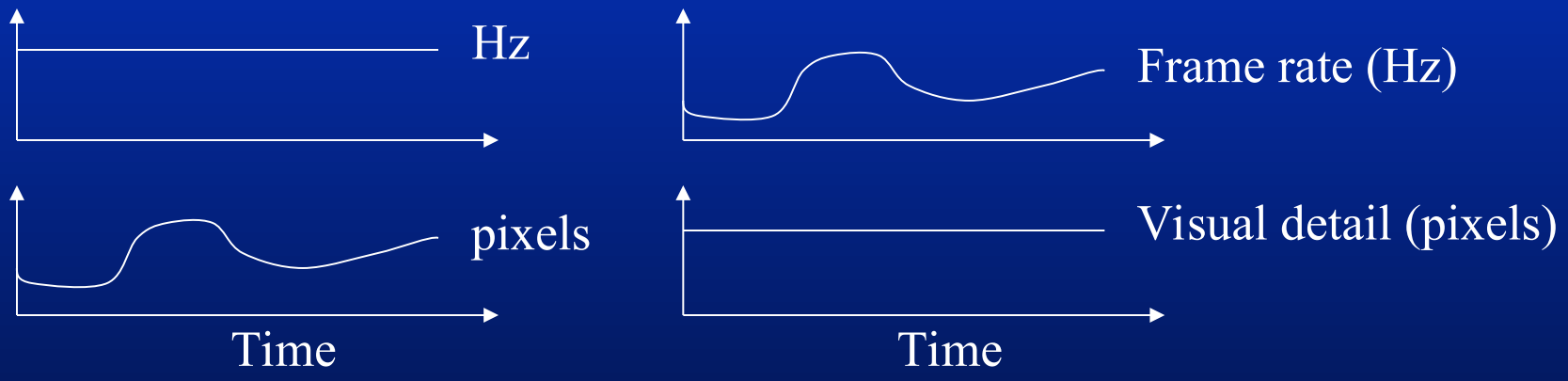
- *Experimental studies*
- *Semiautomatic simplification*

Temporal Fidelity

- *Measurement & control*
- *Importance to users*

Visual vs. Temporal Fidelity

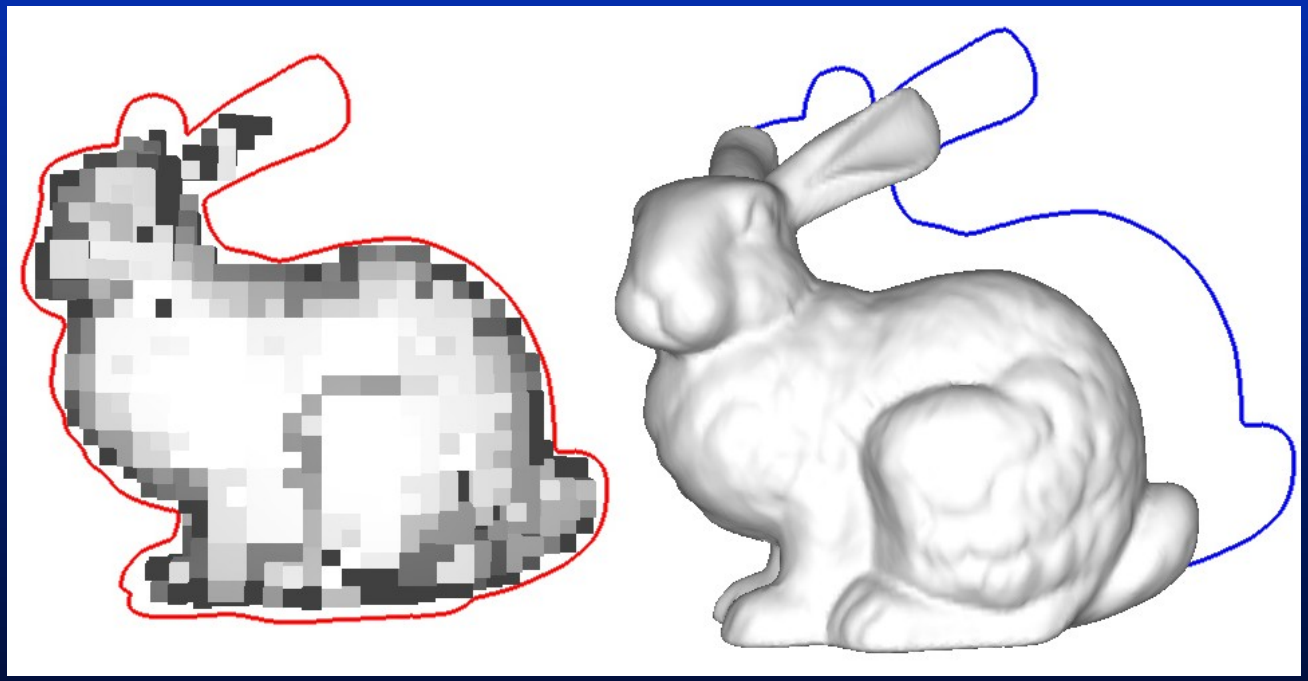
Temporal vs. visual



The right temporal/visual (TV) tradeoff?

Almost no prior research

TV tradeoff: *temporal error*



spatial error

temporal error

Delay results in temporal error

Distance between current, old locations

TV tradeoff: *proposal*

Render progressively

Monitor temporal and spatial error

If spatial exceeds temporal error

Continue improving the frame

Else start a new frame

TV tradeoff: *experiment*