

# Impact of 3D bookmarks on Navigation and Streaming in a Networked Virtual Environment

**Thomas Forgione**<sup>1</sup>  
thomas.forgione@irit.fr

Axel Carlier<sup>1</sup>  
axel.carlier@enseeiht.fr

Géraldine Morin<sup>1</sup>  
morin@enseeiht.fr

Wei Tsang Ooi<sup>2</sup>  
weitsang@nus.edu.sg

Vincent Charvillat<sup>1</sup>  
charvi@enseeiht.fr

<sup>1</sup>Université de Toulouse – IRIT

<sup>2</sup>National University of Singapore

May 11, 2016

# NVE

## Networked Virtual Environment

- 3D environment
- Streaming
- Free flying camera
- 5 degrees of freedom ( $x$ ,  $y$ ,  $z$ ,  $\theta$  and  $\phi$ )



Figure: An NVE from *Sketchfab*

# NVE

## Networked Virtual Environment

- 3D environment
- Streaming
- Free flying camera
- 5 degrees of freedom ( $x, y, z, \theta$  and  $\phi$ )

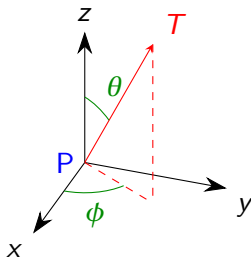


Figure: Spheric coordinate system

# Outline

- 1 Navigation in NVE
  - Controls
  - 3D Bookmarks
- 2 Quality of Experience
  - Experimental setup
  - Results
- 3 Impact on Quality of Service
  - Streaming policies
  - Comparison

# Outline

- 1 Navigation in NVE
  - Controls
  - 3D Bookmarks
- 2 Quality of Experience
  - Experimental setup
  - Results
- 3 Impact on Quality of Service
  - Streaming policies
  - Comparison

## State of the art : difficulty of navigation

### Ruddle et al. [2000]

- Some interactions can get the user lost in the 3D environment
- Hyperlinks as such are unusable

### Jankowski and Hachet [2015]

- Difficult with a 2D screen
- Strongly task-dependent

# Keyboard controls : difficulty of navigation

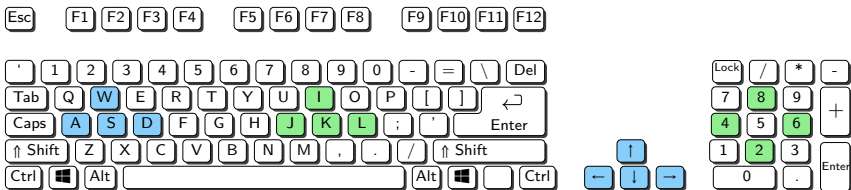


Figure: Keyboard controls

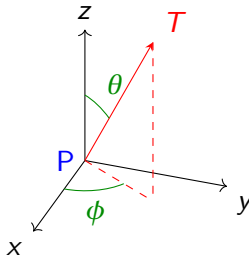
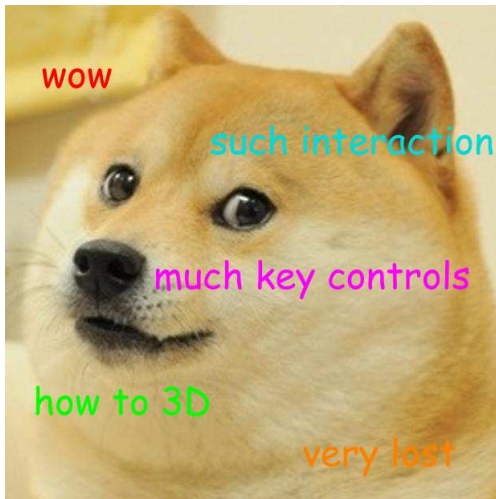


Figure: Spheric coordinate system

# Difficulty of navigation





# Arrow bookmarks



- Arrow curved according to the current position of the camera
- Shows path to targeted bookmark

3D data from McGuire [2011]

Figure: Bookmarks as arrows

# Fly-to transition



Figure: Smooth move from the current position to the bookmark

# Viewport bookmarks



Computer vision style bookmark

- Optical center
- Image plane

Figure: Bookmarks as viewports

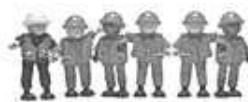
# Outline

- 1 Navigation in NVE
  - Controls
  - 3D Bookmarks
- 2 Quality of Experience
  - Experimental setup
  - Results
- 3 Impact on Quality of Service
  - Streaming policies
  - Comparison

# Evaluating the impact of bookmarks with MicroWorkers

## Crowdsourcing

- People were recruited on [microworkers.com](http://microworkers.com)
- More than 100 started the experiment
- 51 of them completed it, and were paid
- Traces of all experiments were recorded in order to be replayed to simulate streaming



## Problem : task design

### Without task, people won't browse our 3D scenes

- They have nothing to do
- They are probably not interested in our data
- 3D browsing is difficult
  - ⇒ They will browse for a few seconds and leave
  - ⇒ We will not see the impact of bookmarks

### Our solution

- Ask people to perform a task
- Find hidden objects in the scene




Figure: A coin hidden behind a curtain

# Task design

## One micro-job

- Initial questionnaire (age, gender, etc...), tutorial
- 3 random tasks
- Final questionnaire to evaluate QoE

## One random task

- Select 1 3D scene out of 3 (loaded beforehand)
- Select 1 UI out of 3 (no bookmarks, 10 viewports, 10 arrows)
- Select 8 coins randomly out of 50 (positioned beforehand)
- **No correlation between bookmarks and coin positions** 

## Success

- At least 6 coins found among the 8 hidden coins in the scene



## Stats on 51 users

BM type	#Exp	Mean # coins	# completed	Mean time
No bookmarks	51	7.08	18	4:16 min
Arrows	51	7.39	27	2:33 min
Viewports	51	7.51	30	2:16 min

**Table:** Analysis of the sessions length and users success by type of bookmarks

BM type	Total distance	Fly-to distance	Ratio
No bookmarks	610.80	N/A	N/A
Arrows	586.30	369.77	63%
Viewports	546.96	332.72	61 %

**Table:** Analysis of the length of the paths by type of bookmarks

# Questionnaire

Questions	Answers
What was the difficulty level without bookmark?	3.04 / 5 $\pm$ 0.31 (99%)
What was the difficulty level with bookmark?	2.15 / 5 $\pm$ 0.30 (99%)
Did the bookmarks help you to find the coins?	42 Yes, 5 No
Did the bookmarks help you to browse the scene?	49 Yes, 2 No
Do you think bookmarks can be helpful?	49 Yes, 2 No
Which bookmark style do you prefer and why?	32 Arrows, 7 Viewports
Did you enjoy the experiment ?	36 Yes, 3 No

**Table:** List of questions in the questionnaire and summary of answers.

⇒ Quality of experience is better with bookmarks

## From users' paths to system queries

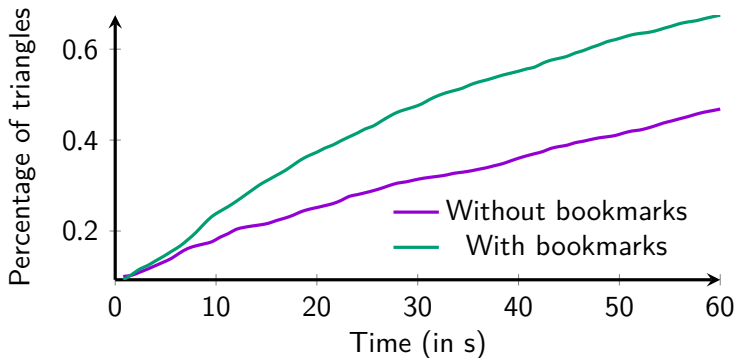


Figure: Percentage of polygons queried after a certain period of time

- Higher demand to the server with bookmarks
- Potential impact on QoS

## Impact of bookmarks on navigation

- Faster navigation
- Better performance w.r.t. the task
- Good QoE, task perceived as simpler
- Higher demand to the server

# Outline

- 1 Navigation in NVE
  - Controls
  - 3D Bookmarks
- 2 Quality of Experience
  - Experimental setup
  - Results
- 3 Impact on Quality of Service
  - Streaming policies
  - Comparison

# Streaming simulation

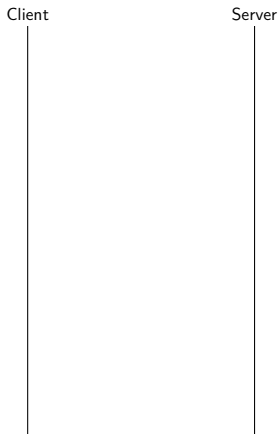


Figure: Streaming model

# Streaming simulation

Client info :

- Camera position
- Frustum of the camera
- Bookmark clicked if any

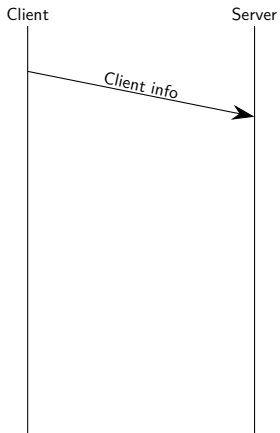


Figure: Streaming model

# Streaming simulation

Client info :

- Camera position
- Frustum of the camera
- Bookmark clicked if any

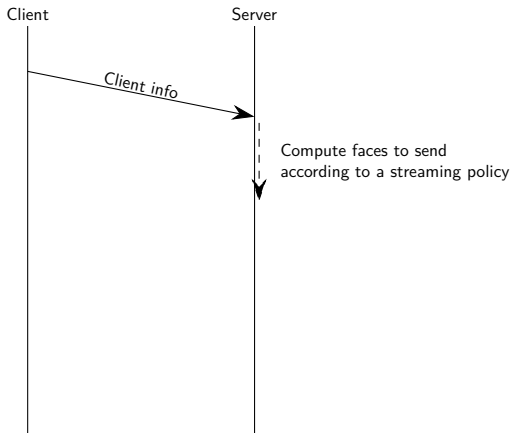


Figure: Streaming model



# Streaming simulation

Client info :

- Camera position
- Frustum of the camera
- Bookmark clicked if any

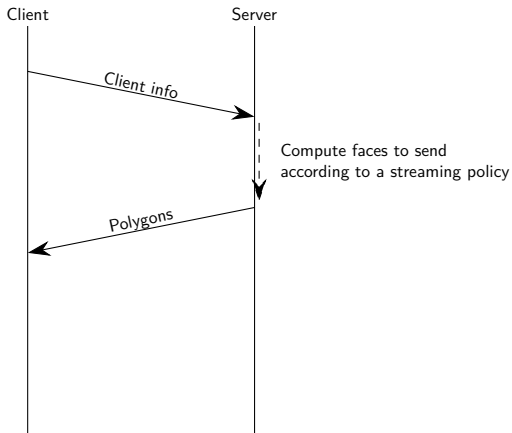


Figure: Streaming model

# Streaming simulation

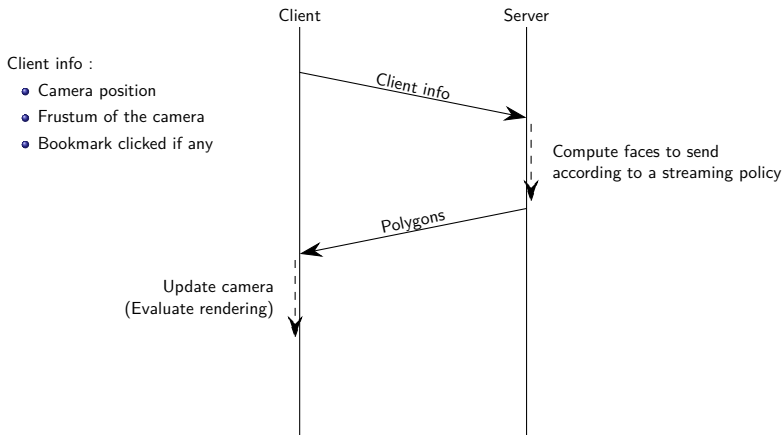


Figure: Streaming model

# Streaming simulation

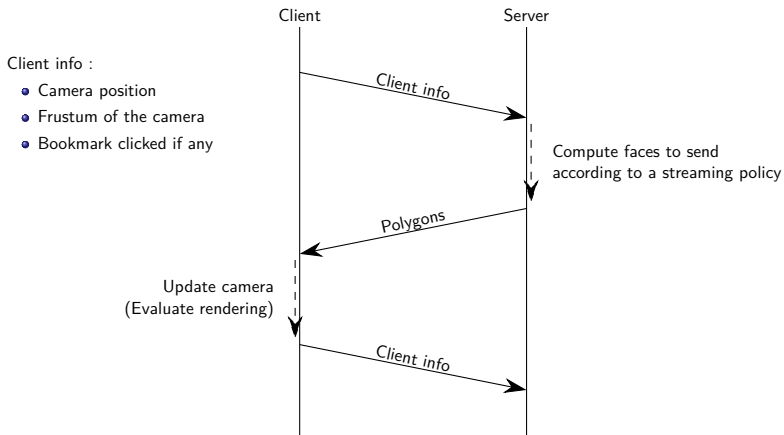


Figure: Streaming model

## Rendering evaluation

- One 3D model is fully loaded at the beginning
- One 3D model is loading progressively
- We compute the rendering on the fully loaded model
- $R = \frac{Pixels_{\text{already loaded face}}}{Pixels_{\text{all faces}}}$



Figure: A loading 3D scene with  $R = 0.9$

## A naive streaming policy

### C : Frustum culling / Backface culling

- Send only polygons that may intersect the frustum of the camera
- Send only polygons that are oriented towards the camera
- Sort polygons by distance to the camera

## C : Frustum culling / Backface culling

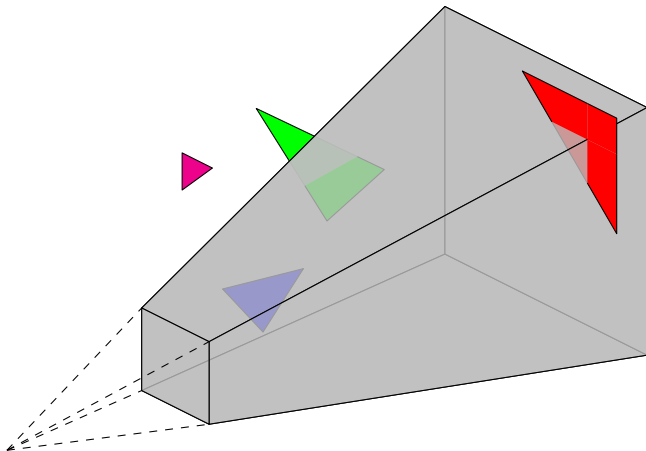


Figure: Frustum culling

## Two fold issue

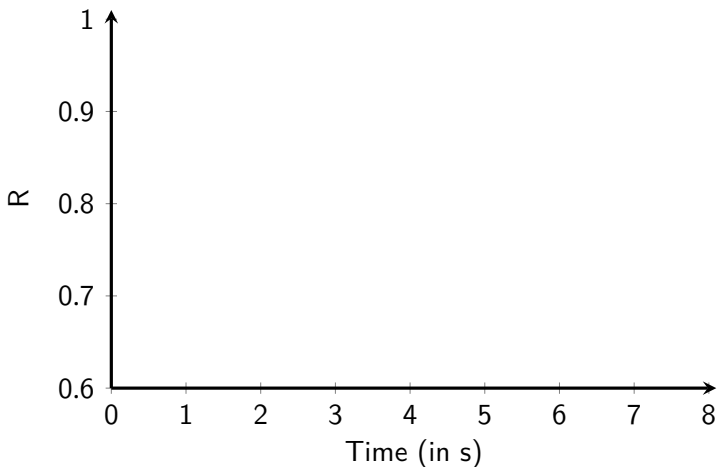


Figure: Evaluation of the naive streaming policy

## Two fold issue

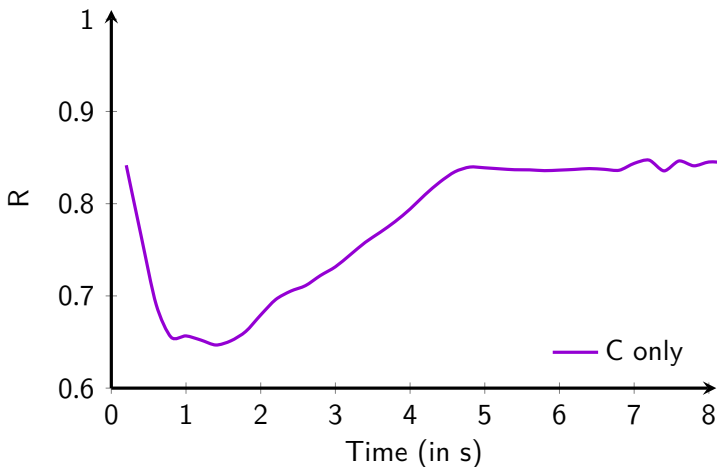


Figure: Evaluation of the naive streaming policy



## Server-side rendering

- Find visible triangles
- Sort them by projected area

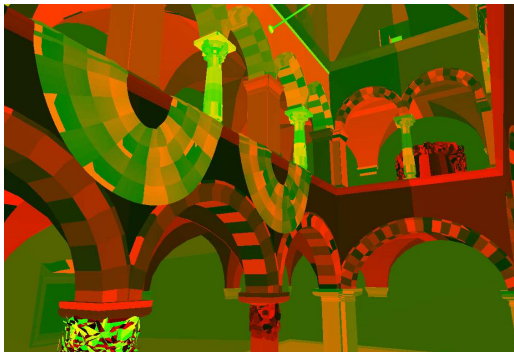
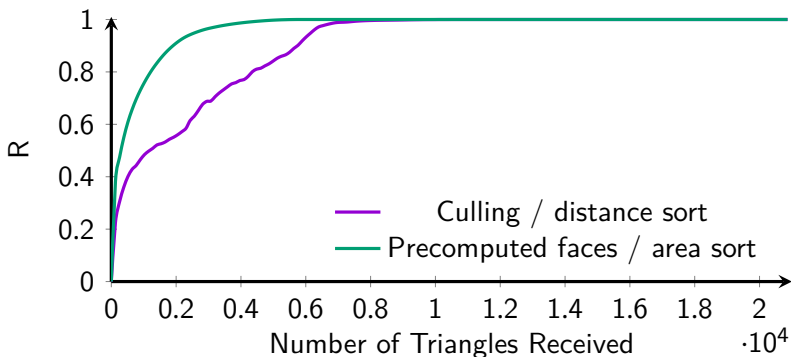


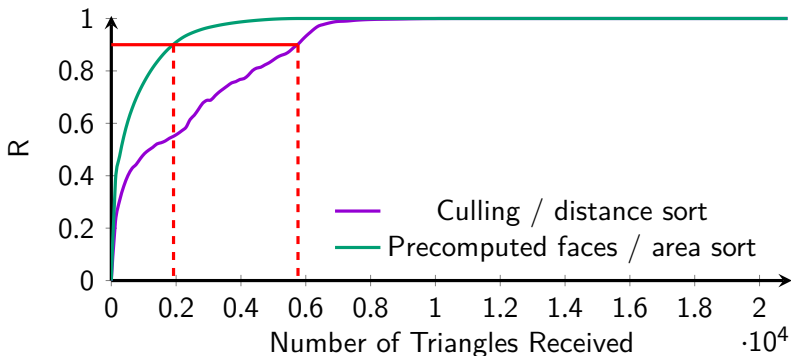
Figure: Server-side rendering

# Impact of server-side rendering



- Great impact on QoS
- Not scalable : the server cannot do rendering for everyone
- But we can do it on bookmarks  $\implies$  Precomputation

# Impact of server-side rendering



- Great impact on QoS
- Not scalable : the server cannot do rendering for everyone
- But we can do it on bookmarks  $\implies$  Precomputation

## A short review on prefetching

### Park et al. [2001]

- Predict the user's interest in an object
- Sort objects by interest and prefetch

### Hung and Liu [2006]

- Cluster the navigation path of users
- Predict future navigation path
- Prefetch objects near predicted path

### Zhou et al. [2015]

- Learn object access patterns
- Predict objects that are likely to be accessed and prefetch

### *C* : Frustum culling / Backface culling

- Send only polygons that may intersect the frustum of the camera
- Send only polygons that are oriented towards the camera

### *FD* : Fetch destination

Send the triangles that are seen by the bookmark clicked

### *PP* : Prefetch predicted

Send the triangles that are seen by the bookmarks that might be clicked

**C** : Frustum culling / Backface culling

- Send only polygons that may intersect the frustum of the camera
- Send only polygons that are oriented towards the camera

**FD** : Fetch destination

Send the triangles that are seen by the bookmark clicked

**PP** : Prefetch predicted

Send the triangles that are seen by the bookmarks that might be clicked

*PP* : Prefetch Predicted

Try to guess the next bookmark clicked knowing the previous one

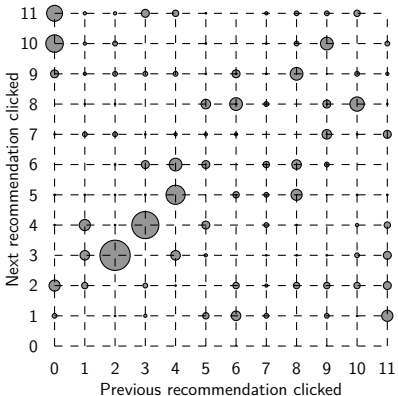


Figure: Patterns of access for a specific scene

**PP : Prefetch Predicted**

Try to guess the next bookmark clicked knowing the previous one

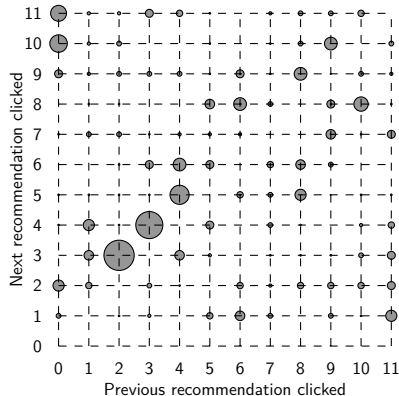


Figure: Patterns of access for a specific scene

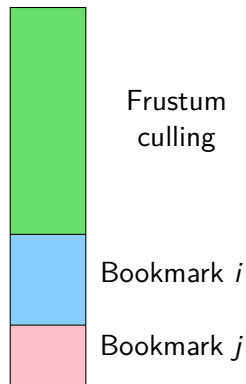


Figure: Prefetching packet



# Quality of rendering after a click on a bookmark

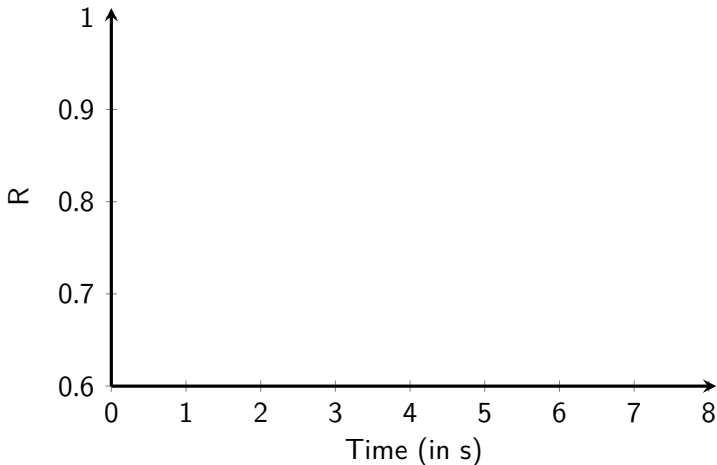


Figure: Comparison of streaming policies

# Quality of rendering after a click on a bookmark

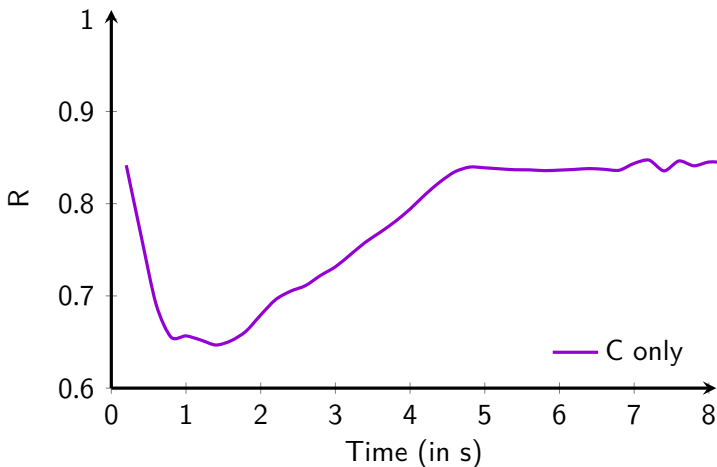


Figure: Comparison of streaming policies

# Quality of rendering after a click on a bookmark

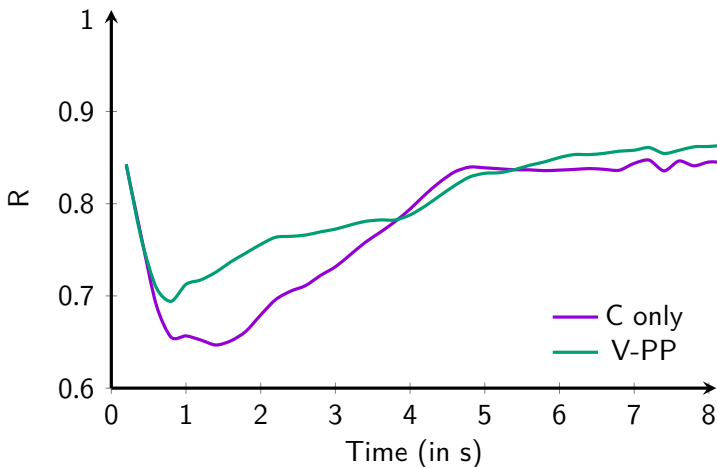


Figure: Comparison of streaming policies

# Quality of rendering after a click on a bookmark

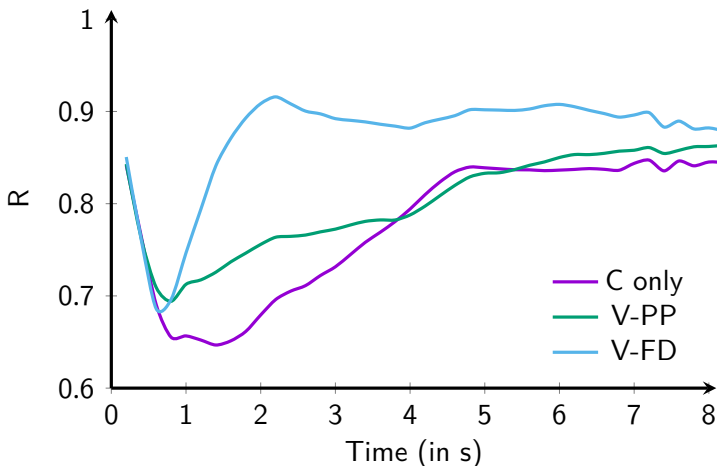


Figure: Comparison of streaming policies

# Quality of rendering after a click on a bookmark

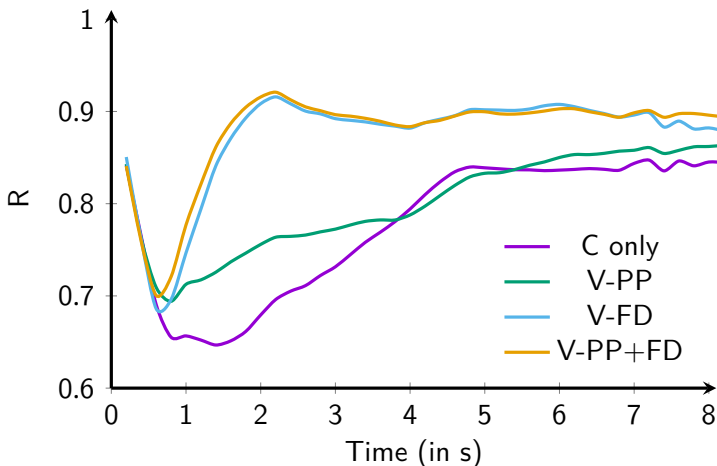


Figure: Comparison of streaming policies

## Improvement with prediction is not significant

### Conclusion on policies

- Frustum culling is not satisfying
- Prefetching is better than frustum culling but not significantly (due to low predictability in 3D)
- Fetch destination is really efficient
- Fetch destination + Prefetching is barely better

# Conclusion

## Bookmarks' impact on NVE

- Good for QoE
- Drawback on QoS
- Prerendering the bookmarks can limit this drawback

## Future work

- Optimize chunk size allocated for prefetching
- Adapt *fly-to* transition to bandwidth
- Study the influence of the bookmarks' positioning on user browsing pattern

Questions ?



- Shao-Shin Hung and Damon Shing-Min Liu. Using prefetching to improve walkthrough latency: Research articles. *Comput. Animat. Virtual Worlds*, 17(3-4):469–478, July 2006. ISSN 1546-4261. doi: 10.1002/cav.v17:3/4. URL <http://dx.doi.org/10.1002/cav.v17:3/4>.
- Jacek Jankowski and Martin Hachet. Advances in interaction with 3d environments. *Computer Graphics Forum*, 34(1):152–190, 2015.
- Morgan McGuire. Computer graphics archive, August 2011. URL <http://graphics.cs.williams.edu/data>.
- Sungju Park, Dongman Lee, Mingyu Lim, and Chansu Yu. Scalable data management using user-based caching and prefetching in distributed virtual environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 121–126, Banff, Canada, 11 2001.
- Roy A Ruddle, Andrew Howes, Stephen J Payne, and Dylan M Jones. The effects of hyperlinks on navigation in virtual