Peer-Assisted Rendering for Networked Virtual Environments on Mobile Devices

黃瑋璨

新加坡國立大學

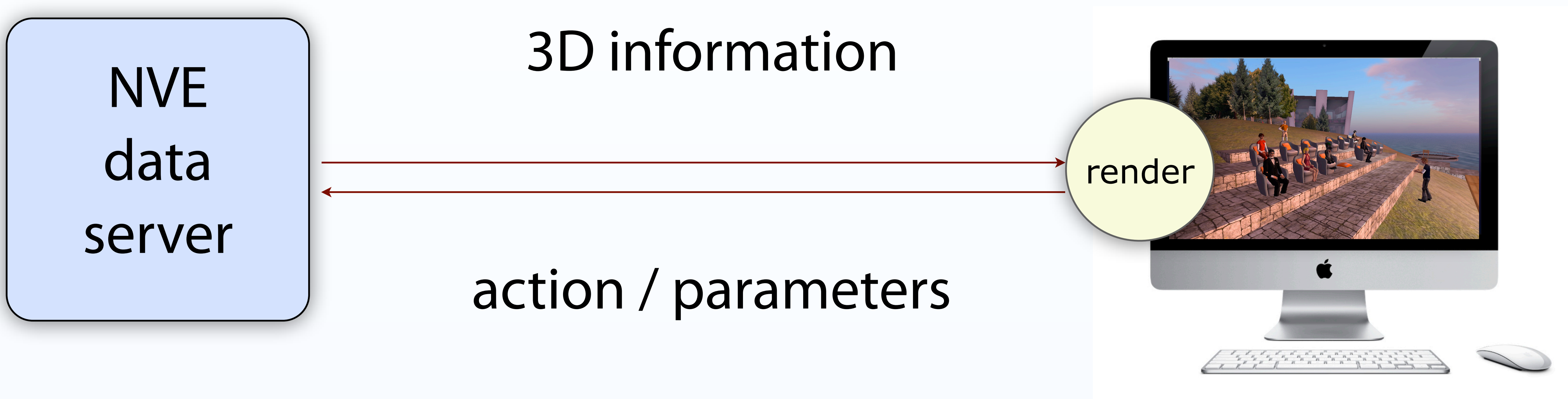
Wei Tsang Ooi NUS, Singapore
Minhui Zhu NUS, Singapore
Wei Cheng NUS / Manovega, Singapore

Geraldine Morin U of Toulouse, France
Sebastien Mondet U of Oslo, Norway / NYU, USA

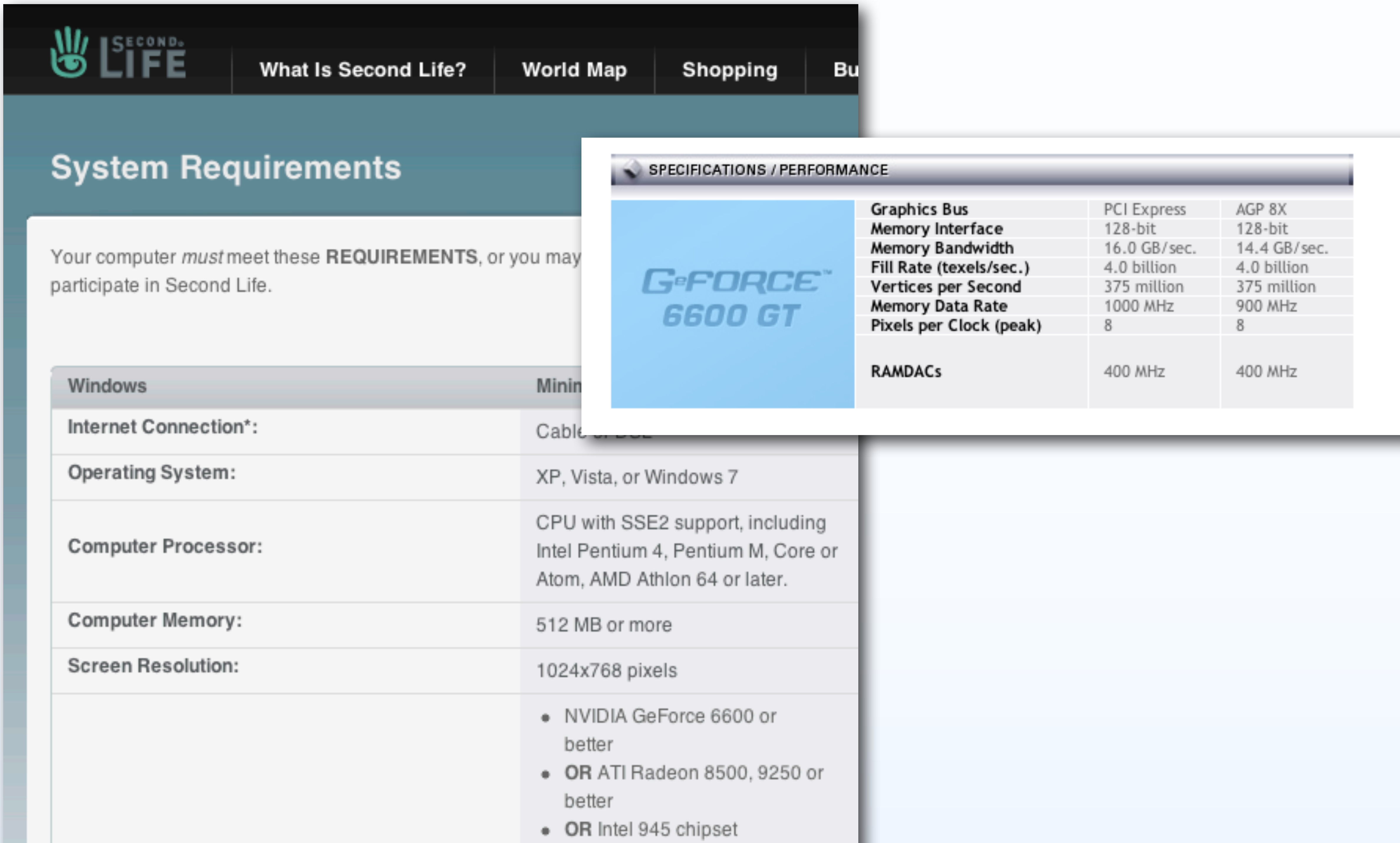
Peer-Assisted Rendering for **Networked Virtual Environments** on Mobile Devices


MMC

NetGames / MMVE 2010


NVE
data
server
3D information
action / parameters
render

Recommended Requirements for Second Life Client


What Is Second Life? | World Map | Shopping | Bu

System Requirements

Your computer *must* meet these **REQUIREMENTS**, or you may
participate in Second Life.

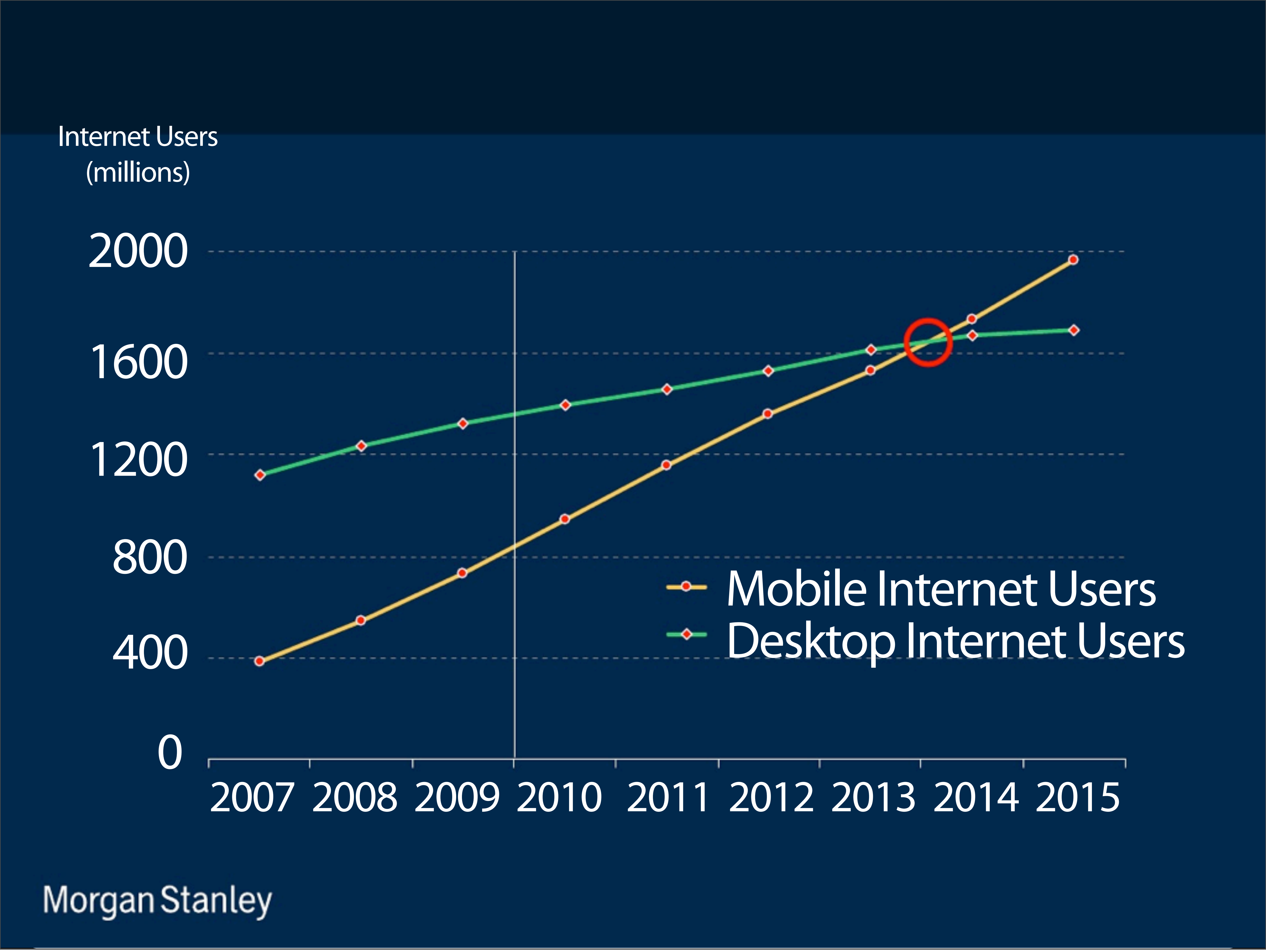
Windows	Minin
Internet Connection*:	Cable
Operating System:	XP, Vista, or Windows 7
Computer Processor:	CPU with SSE2 support, including Intel Pentium 4, Pentium M, Core or Atom, AMD Athlon 64 or later.
Computer Memory:	512 MB or more
Screen Resolution:	1024x768 pixels
	• NVIDIA GeForce 6600 or better • **OR** ATI Radeon 8500, 9250 or better • **OR** Intel 945 chipset

SPECIFICATIONS / PERFORMANCE

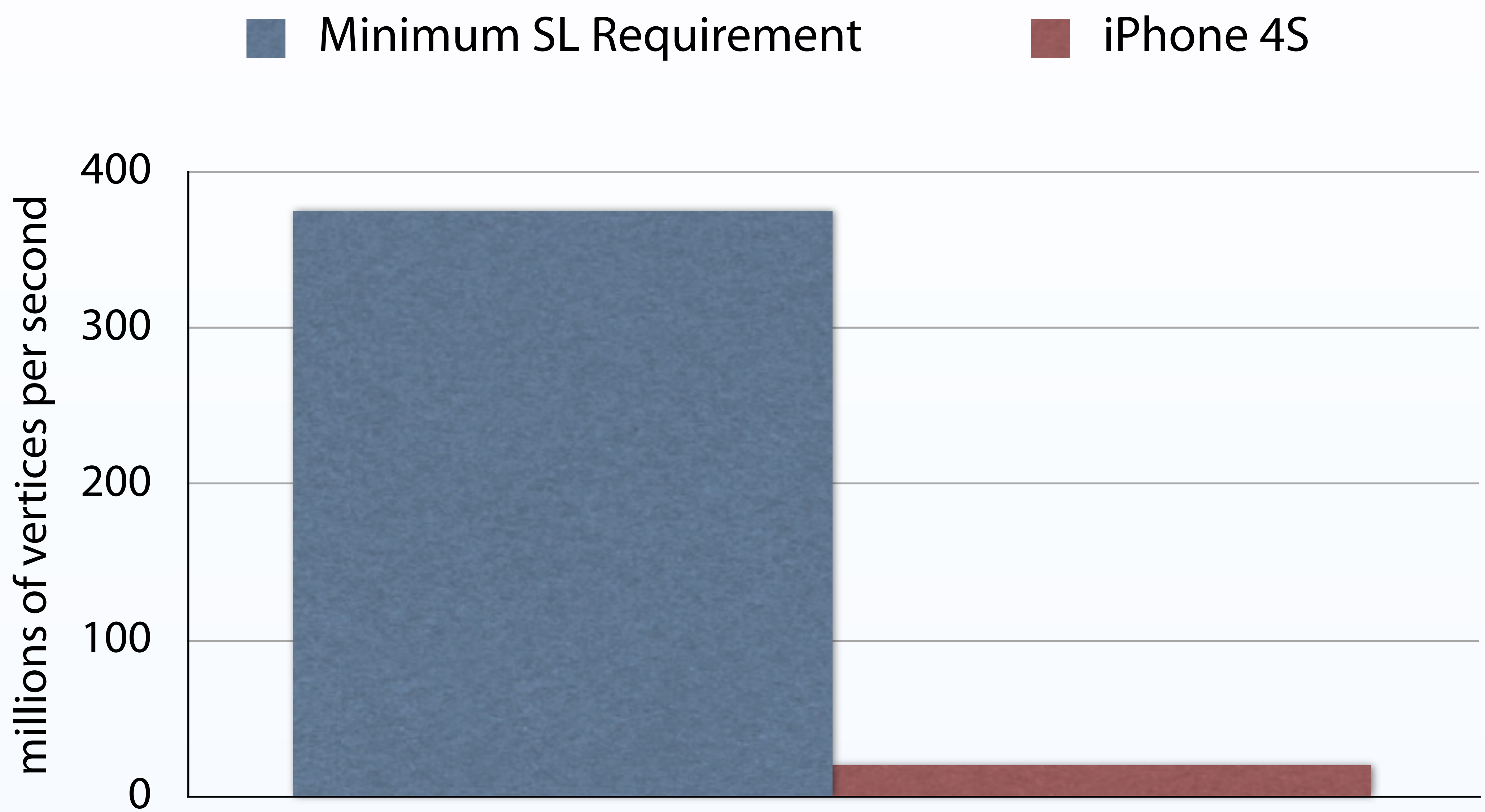
GeFORCE 6600 GT

Graphics Bus	PCI Express	AGP 8X
Memory Interface	128-bit	128-bit
Memory Bandwidth	16.0 GB/sec.	14.4 GB/sec.
Fill Rate (texels/sec.)	4.0 billion	4.0 billion
Vertices per Second	375 million	375 million
Memory Data Rate	1000 MHz	900 MHz
Pixels per Clock (peak)	8	8
RAMDACs	400 MHz	400 MHz

Peer-Assisted Rendering for Networked Virtual Environments on **Mobile Devices**


Internet Users
(millions)
2000
1600
1200
800
400
0
2007
2008
2009
2010
2011
2012
2013
2014
2015
Mobile Internet Users
Desktop Internet Users
Morgan Stanley

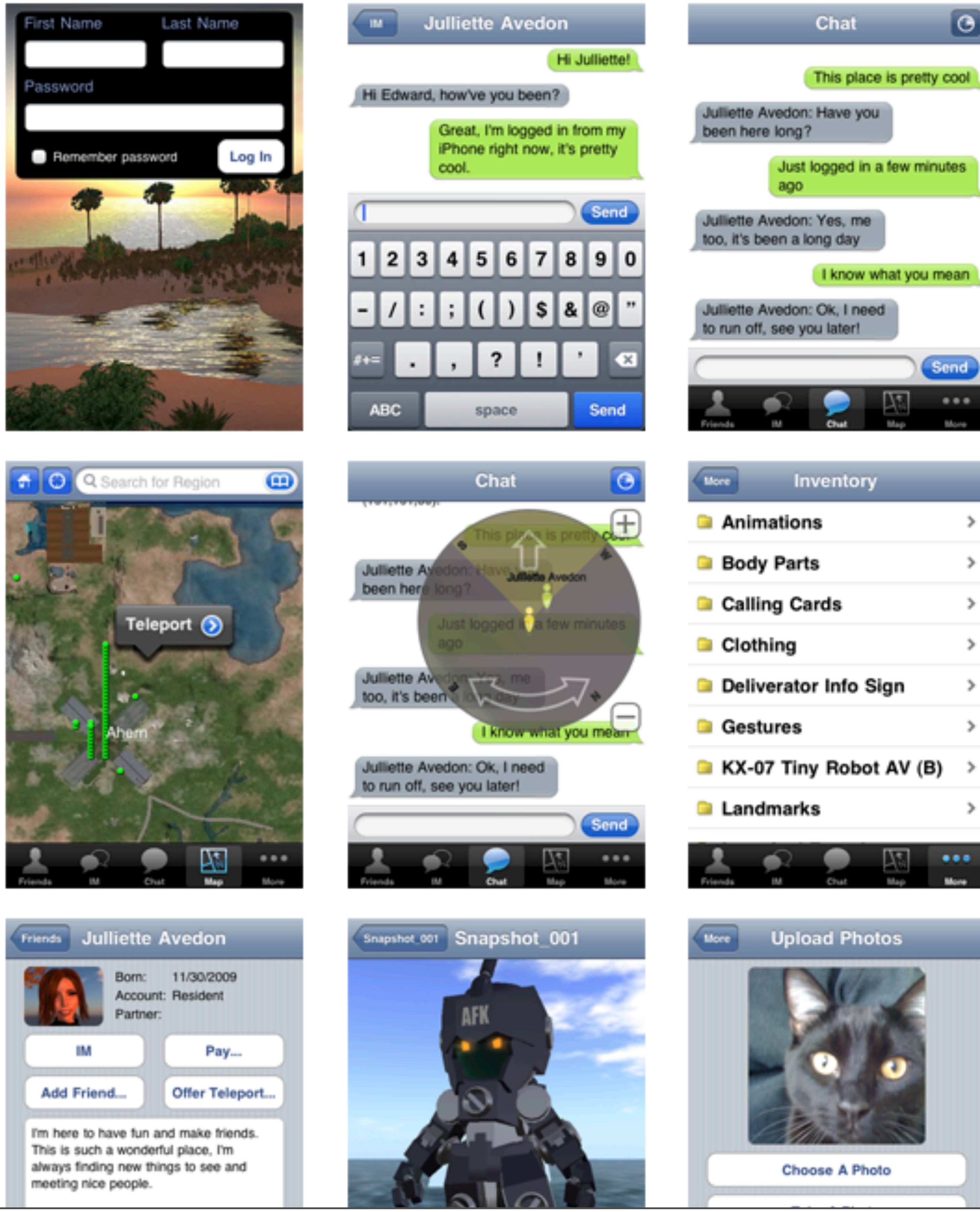
iPhone 4S: A5 Chip with **7x faster** graphics



(Based on benchmark of PowerVR SGX 543MP2 in the A5 in an iPad 2)

Pocket Metaverse

Home About Product Support Buy

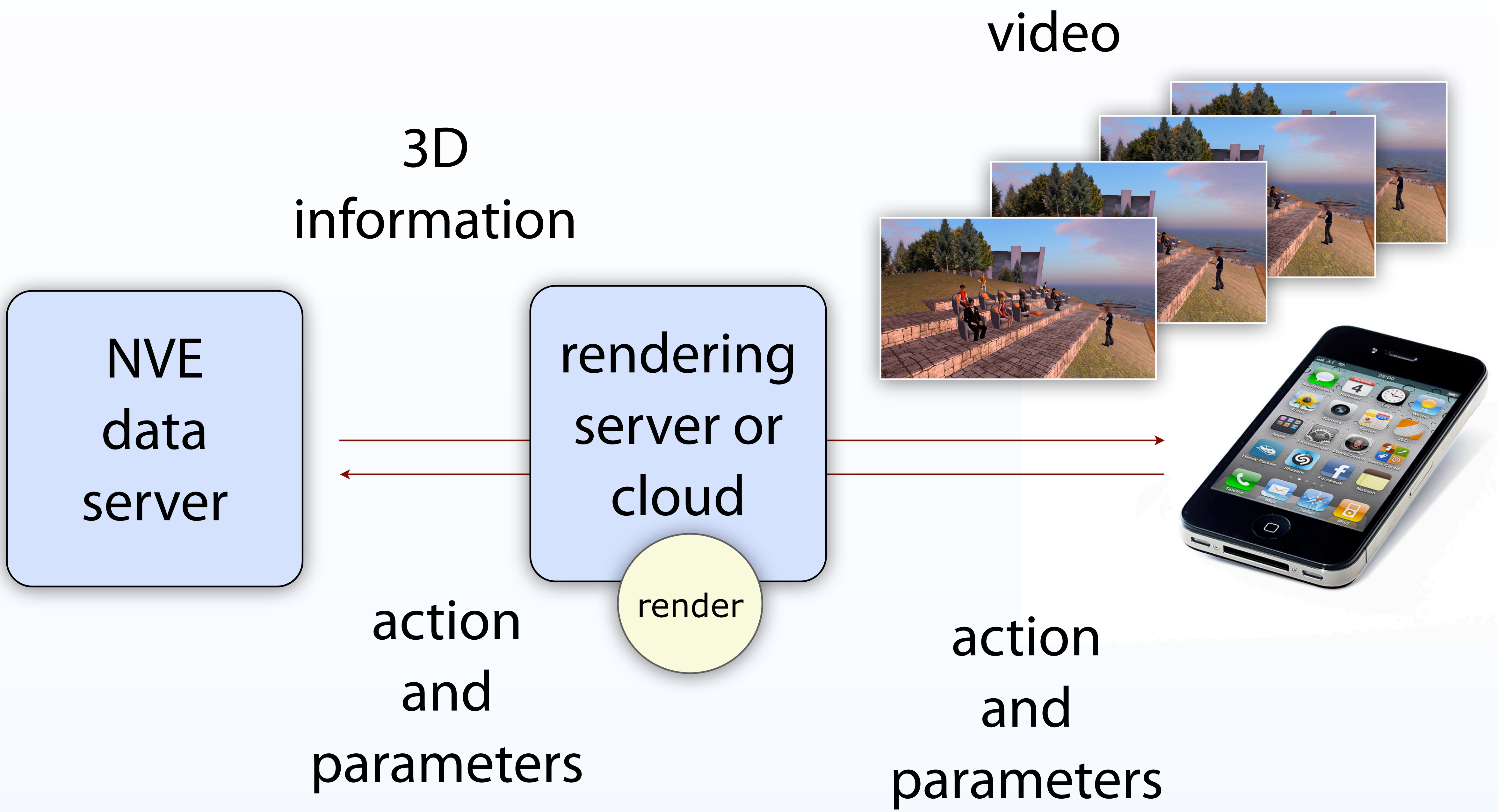


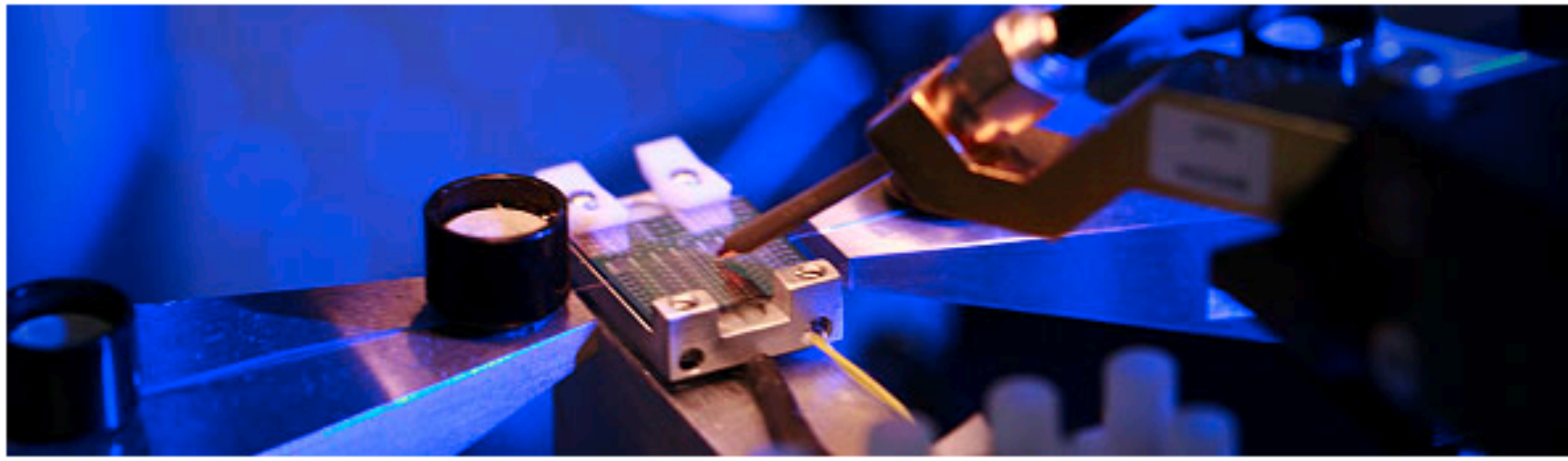
Features

- **Instant Messaging**
  Initiate or respond to individual or group IM's.
- **Chat**
  Converse with nearby avatars.
- **World Map**
  View sims, teleport, search or use landmarks.
- **MiniMap & Who's Nearby**
  Zoom in, move & turn, see who's nearby.
- **Inventory**
  View notecards & pictures, give & accept items.
- **Profiles**
  View profiles, make payments, teleport, befriend.
- **Groups**
  View groups, join & leave, send invites.
- **Search**
  Find people, groups, places, regions, and more.
- **Photos**
  Snap pictures and upload as textures.
- **Much More!**
  The most full-featured mobile app for SL.

how to render HQ 3D NVE on mobile devices?

Peer-**Assisted Rendering** for Networked Virtual Environments on Mobile Devices


video
3D information
NVE data server
rendering server or cloud
render
action and parameters
action and parameters



Research@Intel
Pushing the boundaries of possibility

Research@Intel | Recent Posts | Archives | About this Blog | Meet the Bloggers | Search

Interactive 3D Streaming by Alexander Sterkin

posted by **Guest Blogger** on **June 11, 2008**

Second Life® and World of Warcraft® are among the most prominent MMOGs. They demand lots of computing power – both from the CPU and Graphics. These demands overload any mobile device of today or near future, even including MIDs. By the time the mobile clients have caught up, the performance requirements for MMOGs will grow higher yet.

The 3D Streaming technology developed by Comverse® and Intel computes and renders the MMOG content on a powerful backend server, then smartly compresses and streams the graphics onto a client. A network gateway designed by Comverse allows streaming over both WiMAX and 3G cellular networks. With advanced software optimizations including SSE usage, a single Xeon 5400 backend system can serve simultaneously up to 14 clients.

What does this mean for users of Intel platforms? In fact, the Comverse 3D Streaming capability offers a great user experience across all Intel platforms. On the backend, it's the opportunity to offer the power of visual computing on high-end IA multicore platforms. On the client, it's a chance to drive the demand for MIDs over non-IA smartphones by offering content better suited for larger screens and more sophisticated UI offered by MIDs. Overall, it's a chance for telecom operators and content providers to offer a completely new service – running on the infrastructure that's

Recent Comments

"Yay, great! Congratulations. :) You Intel folks are awesome. Cheers.. http://www.kolkataways.com"

"I really miss the venue,thanks for post this videos... its really help..... Thanks "

"Joe...I agree with Brian. Also adds expense on the purchase side and then on the..."

"Hope McAfee research team will join the SCRUB, as they work on theses topics since..."

"this will be the best cpu ive ever

up to **14** clients


onLive®
SERVICE
GAMES
STORE
SUPPORT
CLOUD GAMING: A FASTER, EASIER WAY TO PLAY.
POWER
00:45
01:05
OnLive is cloud gaming, but what does that mean for you? It means instant access to the games you love on nearly any PC or Mac®, TV and many tablets and smartphones. Just like when you watch videos on demand, OnLive delivers the games you want—instantly—right over the Internet. And since our servers are always state-of-the-art, your gaming experience is, too.
Play FREE now!
Enter Email:
How will OnLive use my email?
Enter Password:
Weak
Medium
Strong
Create Player Tag:
Date of Birth:
Month
Day
Year
Email me about OnLive sales, deals and events.
I agree to the OnLive Privacy Policy.
Get Playing!
THE BUZZ
"...several minds around the CNET offices were officially blown. ..."
cnet
— Rich Brown & Dan Ackerman CNET

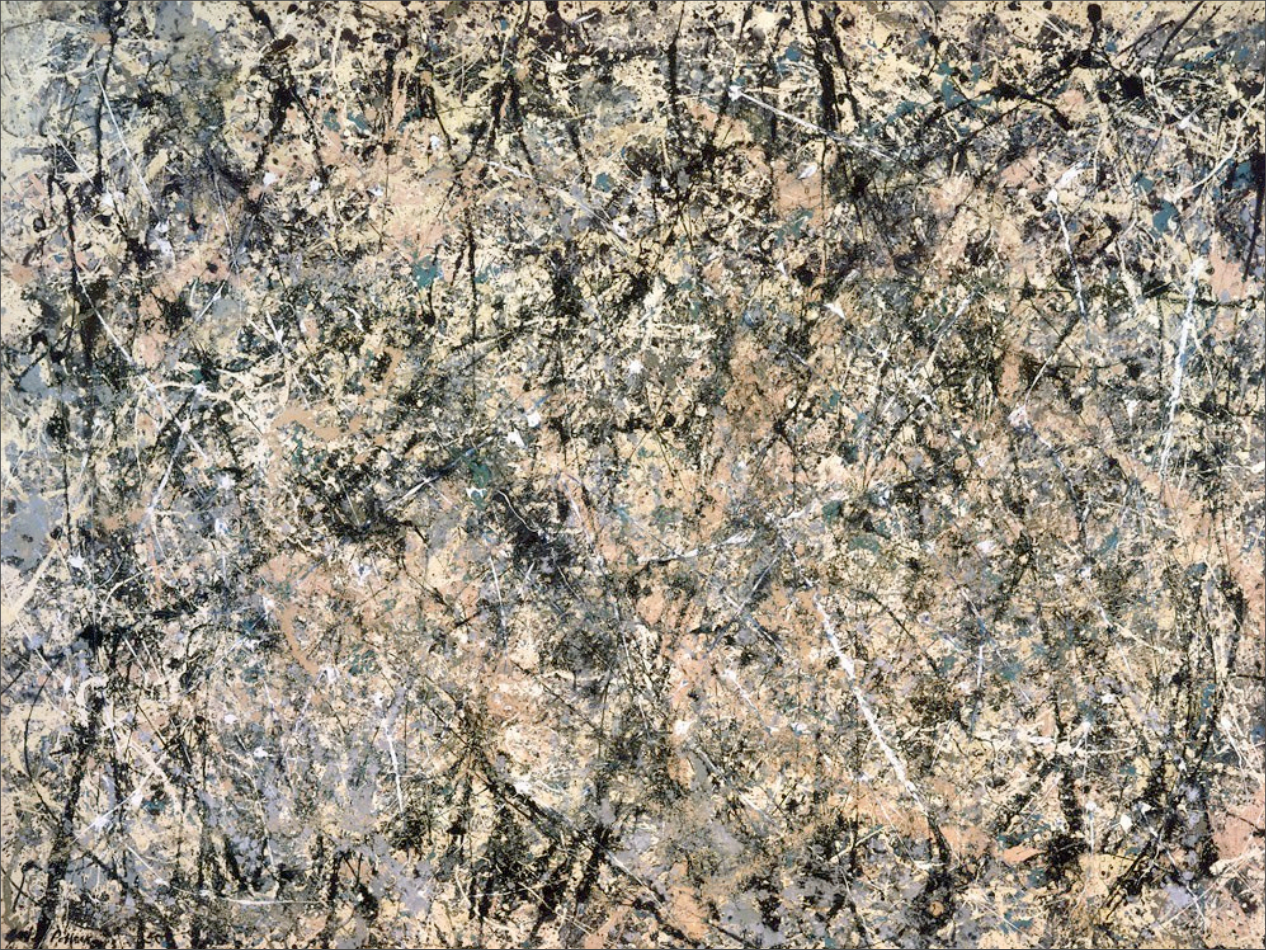
6.5

Mbps

average video data rate measured

There are network issues, the upload is too slow to read the controller inputs properly, and a tightened bandwidth mushes the 720p video into an abstract, YouTube-on-a-56k-modem, Jackson Pollock-style blur.

--- WIRED's review of OnLive



Measuring the Latency of Cloud Gaming Systems†

Kuan-Ta Chen[1], Yu-Chun Chang[12], Po-Han Tseng[1],
Chun-Ying Huang[3], and Chin-Laung Lei[2]

[1]Institute of Information Science, Academia Sinica
[2]Department of Electrical Engineering, National Taiwan University
[3]Department of Computer Science, National Taiwan Ocean University

ABSTRACT

Cloud gaming, i.e., *real-time game playing via thin clients*, relieves players from the need to constantly upgrade their

nology to build large-scale data centers. The massive computation and storage resources of data centers enable users to shift their workload to remote servers. As a result, thin clients are more convenient and also more powerful (with

135-240 ms for OnLive

400-500 ms for StreamMyGame

How to reduce **bandwidth** requirement and interaction **delay**?

key frames + depth

3D information

NVE data server

rendering server or cloud

render

action and parameters

action and parameters

Image-based Warping Example

(SIGGRAPH Newsletter 33(4), by L. McMillan and S. Gortler)

Reference

Ground Truth

Warped

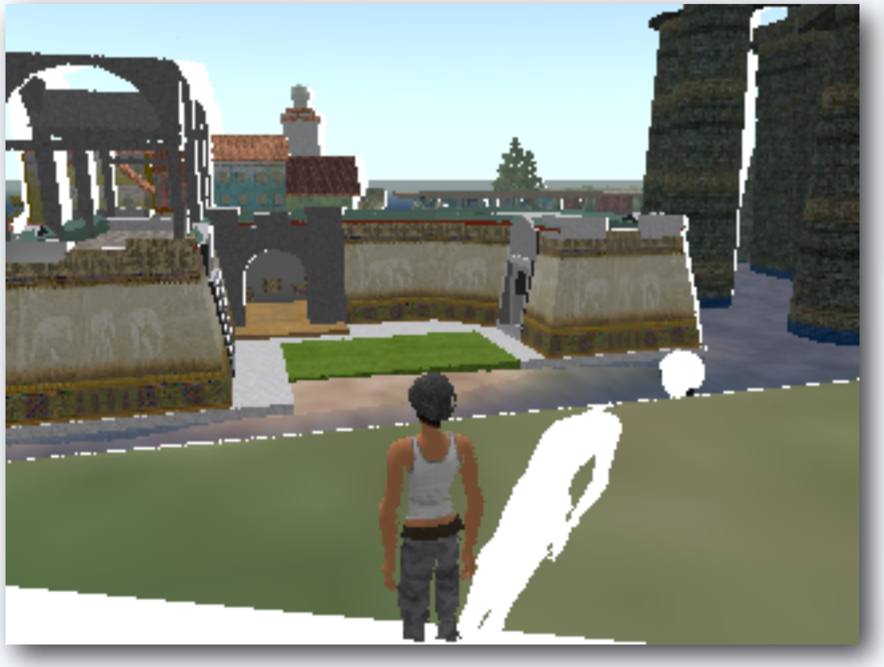
Reference

Ground Truth

Warped

Combined

Concealed

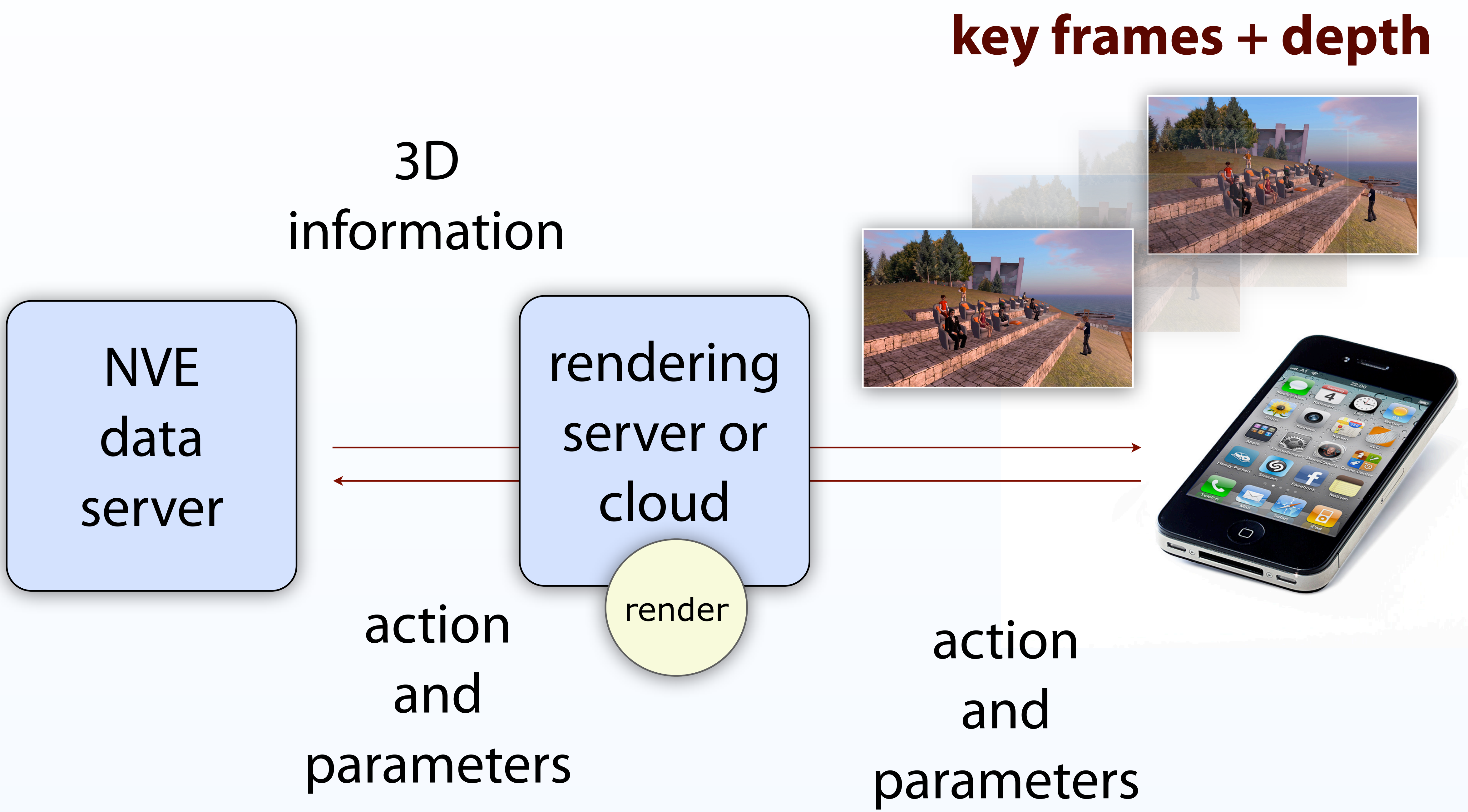


Warped

Warped

Combined

Concealed


key frames + depth
3D information
NVE data server
rendering server or cloud
render
action and parameters
action and parameters

Bandwidth reduced by **not sending every frame**

Latency reduced by **computing new scenes locally** works fine for static scene

(e.g., single user building walkthrough)

but NVE is **highly dynamic**

(other avatars are moving)

either send **more frequent key frames**

or

have **more holes in the rendered scene** relying on server or cloud is **not scalable** and could be **expensive**

Can we do better?

1. **more scalable solution**
2. **less data sent**
3. **fewer holes**
4. **support dynamic scenes**

Peer-Assisted Rendering for Networked Virtual Environments on Mobile Devices

Observation 1

NVE is accessed by multi-users simultaneously

Observation 2

Many clients use desktop with powerful GPU to access NVE

Observation 3

Many avatars move very little within the NVE

Observation 4

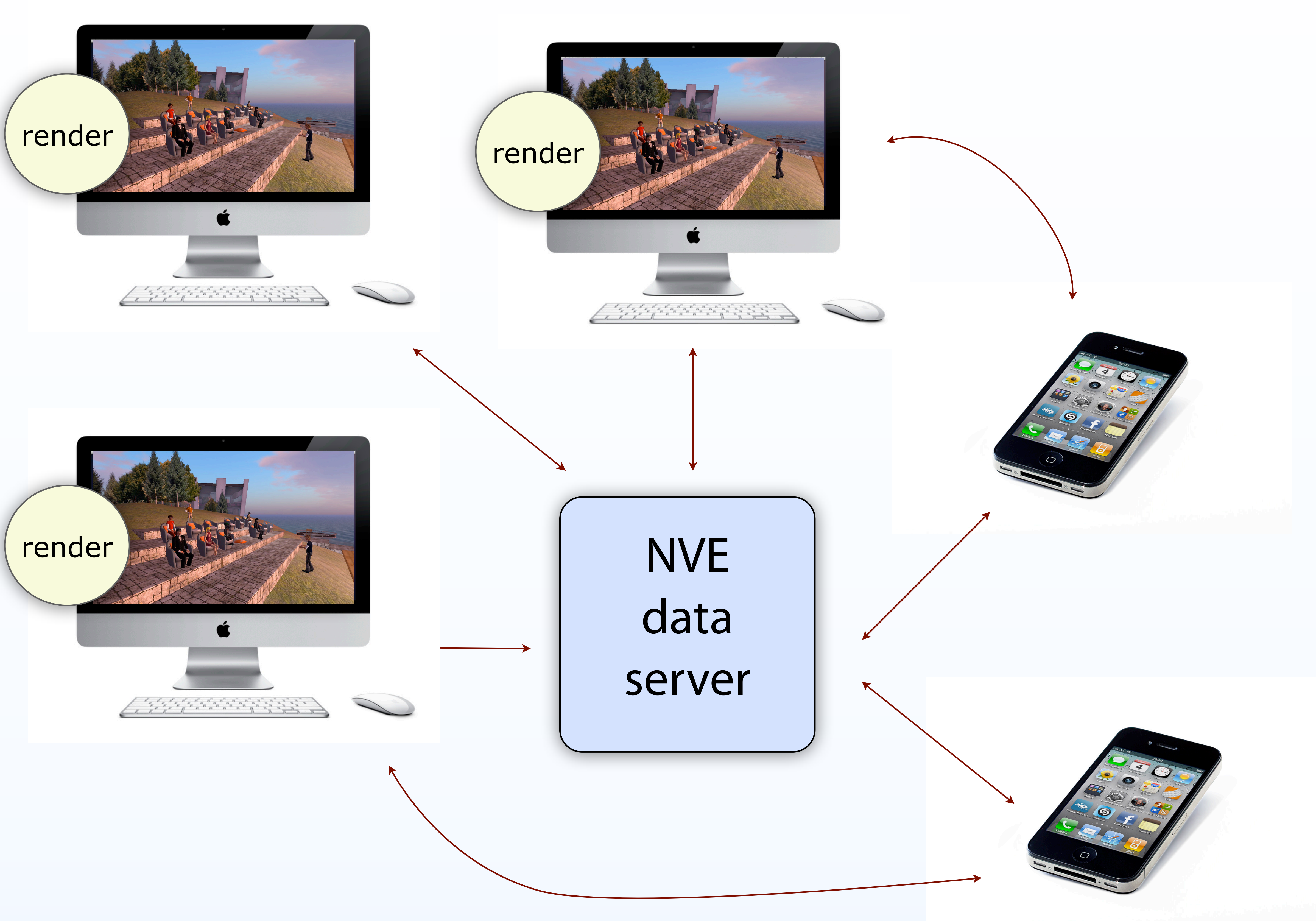
3D information is available

Observation 5

Many objects are static

Key Idea

Reuse rendering of objects at desktop clients for mobile clients


render
render
render
NVE
data
server

Can we do better?

1. **more scalable solution**
2. **less data sent**
3. **fewer holes**
4. **support dynamic scenes**

more scalable:

exploit peer resources instead
of relying on servers

exploit multiple peers

less data sent:

rendering of objects are not sent if they change very little

fewer holes:

objects rendered separately without occlusion and composed at client

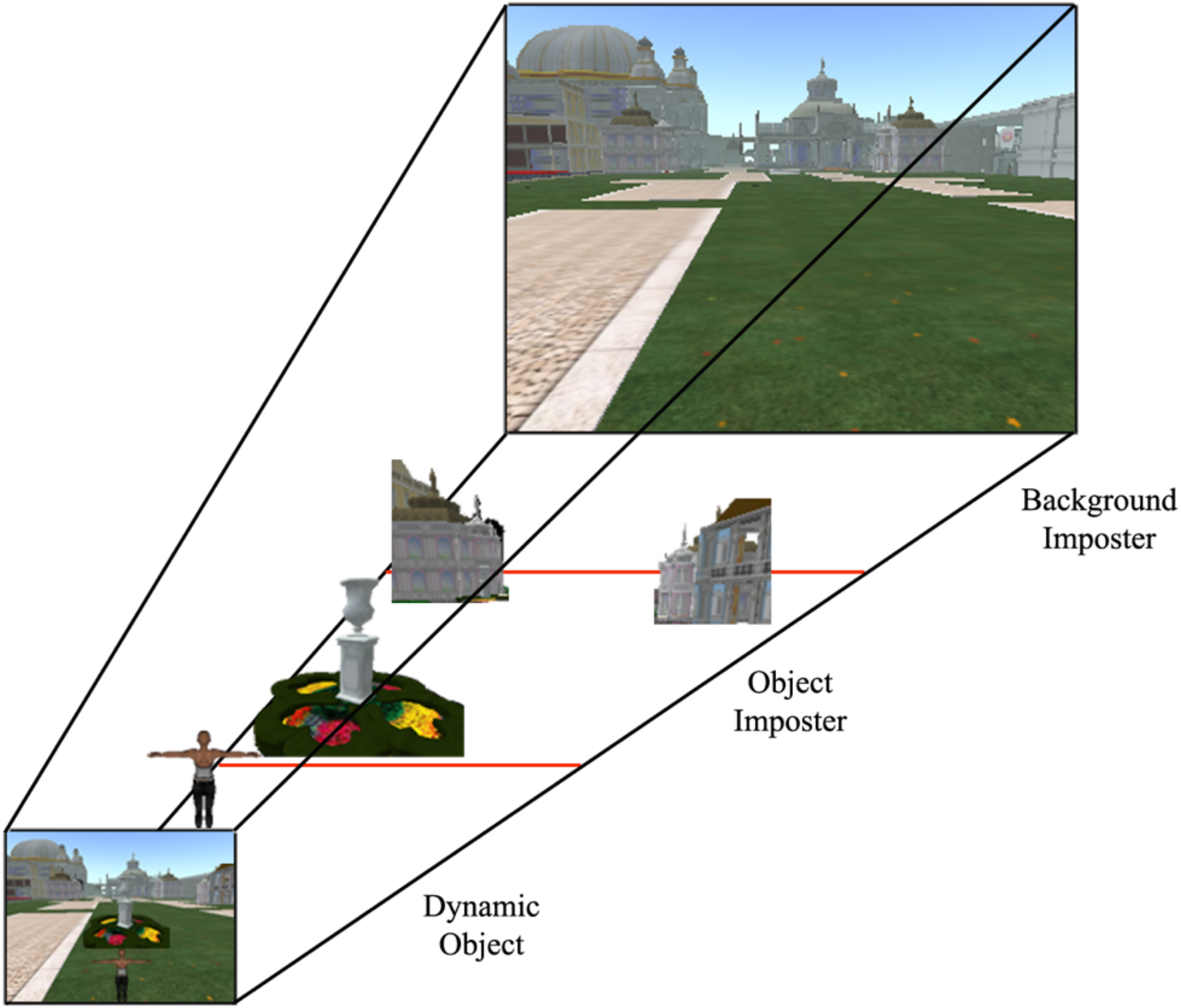
support dynamic scene:

dynamic objects can be rendered locally

how it works

3 types of rendering elements:

far static objects (background)
near static objects
dynamic objects (avatar)


Background
Imposter
Object
Imposter
Dynamic
Object

2 types of peers

assistant (desktop)
assistee (mobile)

repeat:

determine best set of assistants

check current list of static objects in view

if does not have the impostor for the object or

view of object has changed significantly

then

request for object impostors from assistants

render the impostors and dynamic objects

repeat:

determine best set of assistants

check current list of static objects in view

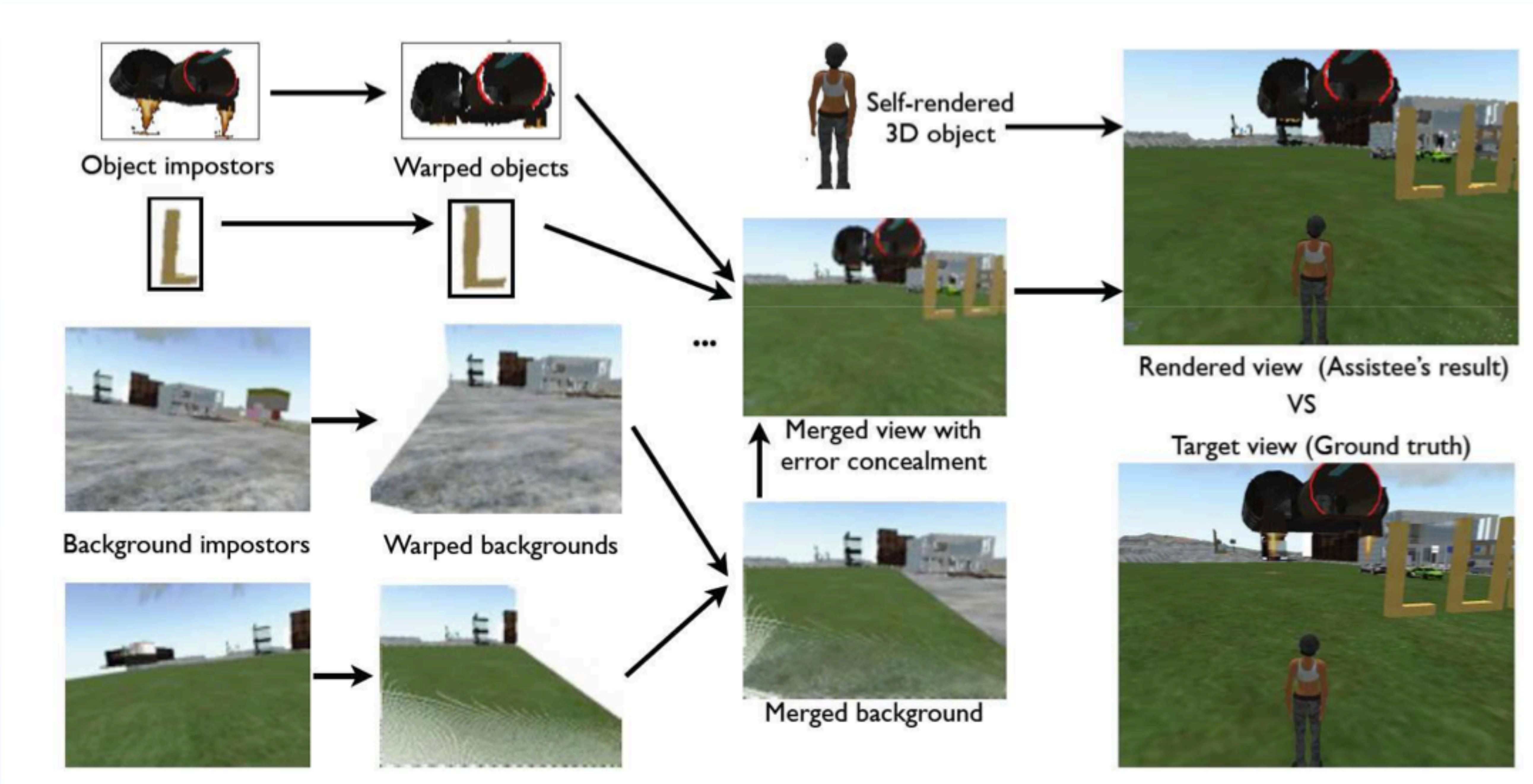
if does not have the impostor for the object or

view of object has changed significantly

then

request for object impostors from assistants

render the impostors and dynamic objects



Object impostors
Warped objects
Self-rendered 3D object
Background impostors
Warped backgrounds
Merged view with error concealment
Merged background
Rendered view (Assistee's result)
VS
Target view (Ground truth)

repeat:

determine best set of assistants

check current list of static objects in view

if does not have the impostor for the object or

view of object has changed significantly

then

request for object impostors from assistants

render the impostors and dynamic objects

questions

1. how to find assistants?
2. will there be enough of them in practice?

what makes a good assistant?

good assistants

1. have similar objects in view

   (so no need to download extra objects from server)

good assistants

2. have similar view of objects

   (so less errors when warping to assistee's view)

good assistants

3. have extra computational resources

good assistants

4. good network connectivity

   (low latency, loss rate, high throughput)

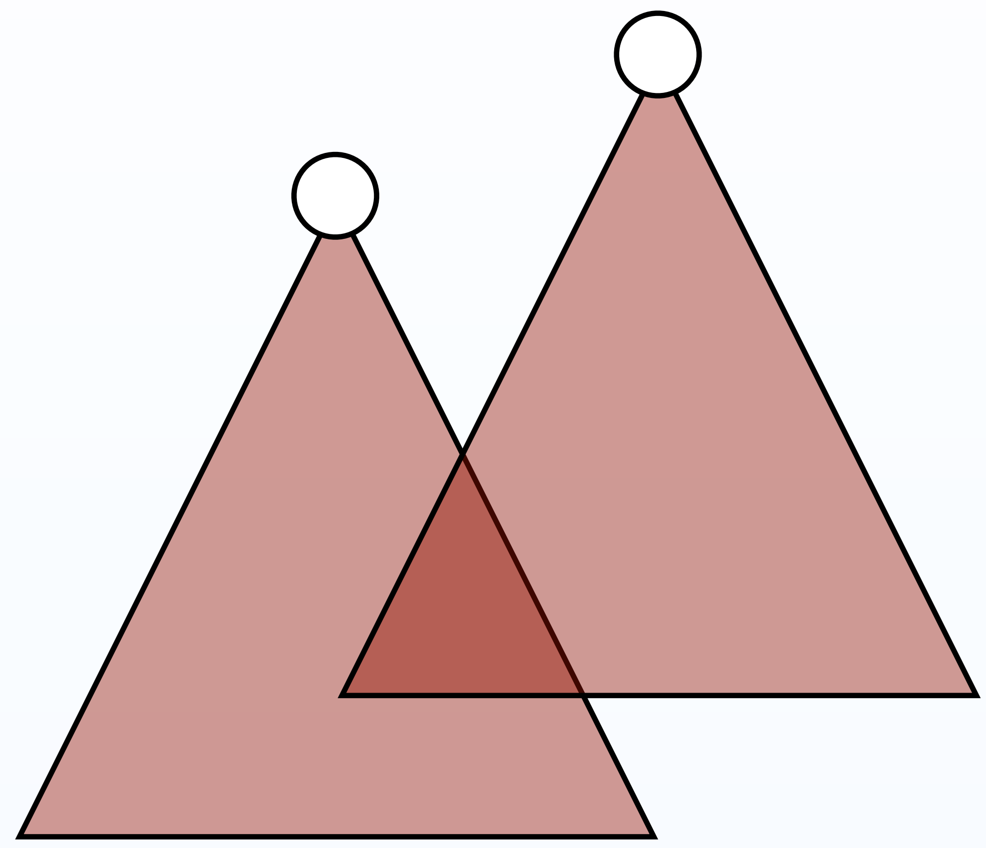
good assistants

5. trustworthy

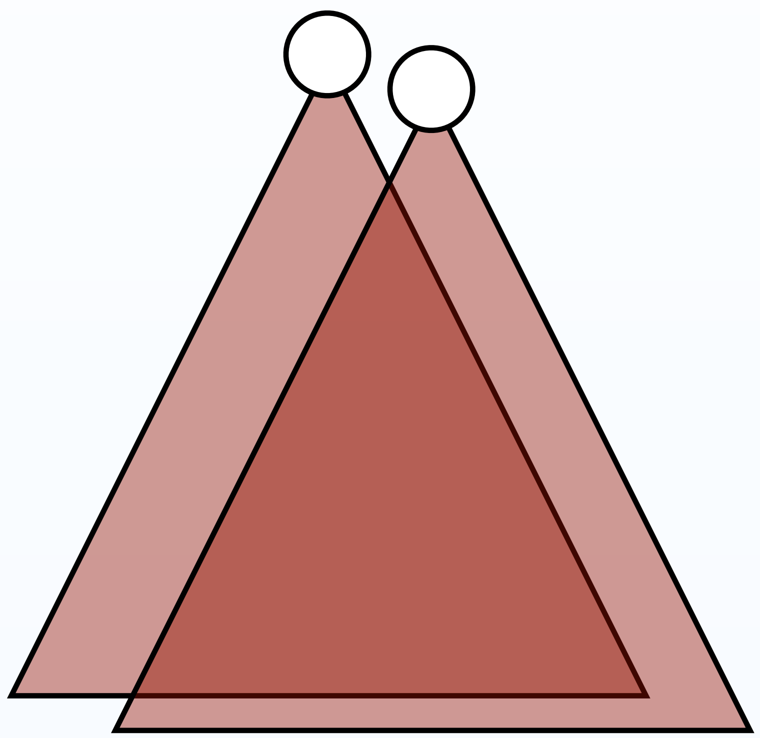
   (won’t send random stuff to assistee)

first step:

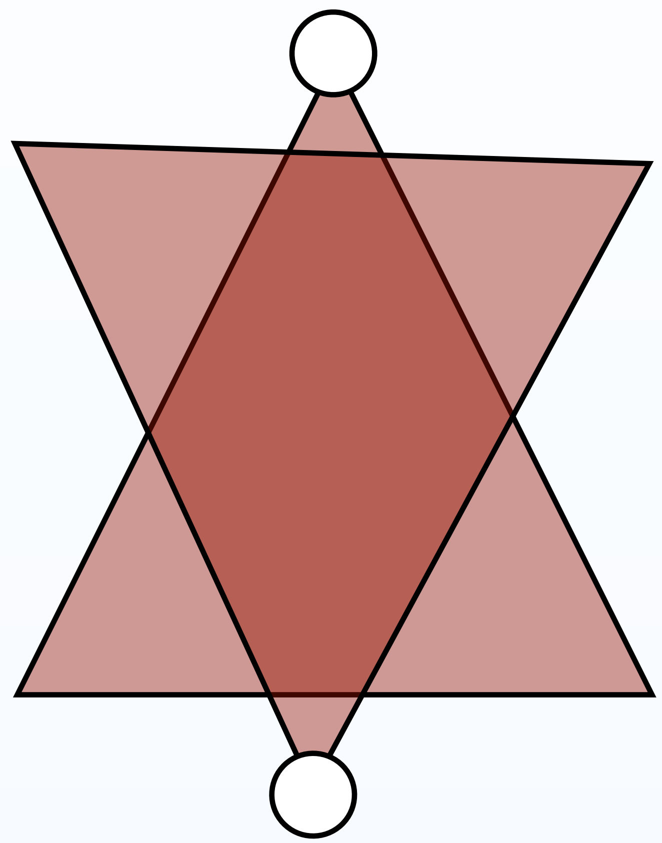
consider view similarity



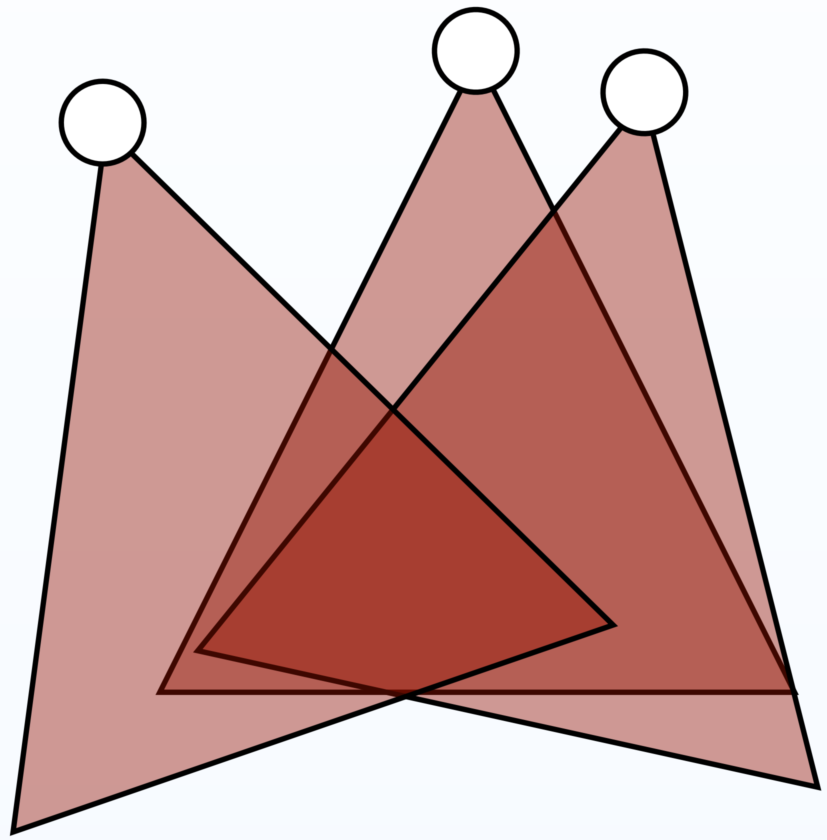
View similarity = ratio of overlapped area in the 2D viewing frustum of two avatars



High overlapped area



but viewing angle cannot be too different



for multiple assistants, take the
union of the overlapped regons.

VS*(*a*, *S*):** view similarity between assistee ***a and set of assistants ***S*.**

questions

1. how to find assistants?
2. **will there be enough of them in practice?**

Collected real traces from Second Life

1. insert bot into SL

2. log avatars seen (position, viewing directions) every 10s

Region	# Records	# Avatars
Freebies	5786	71
Japan Resort	5912	61
Sunland	2516	53

(one hour trace at noon, 23-01-11)

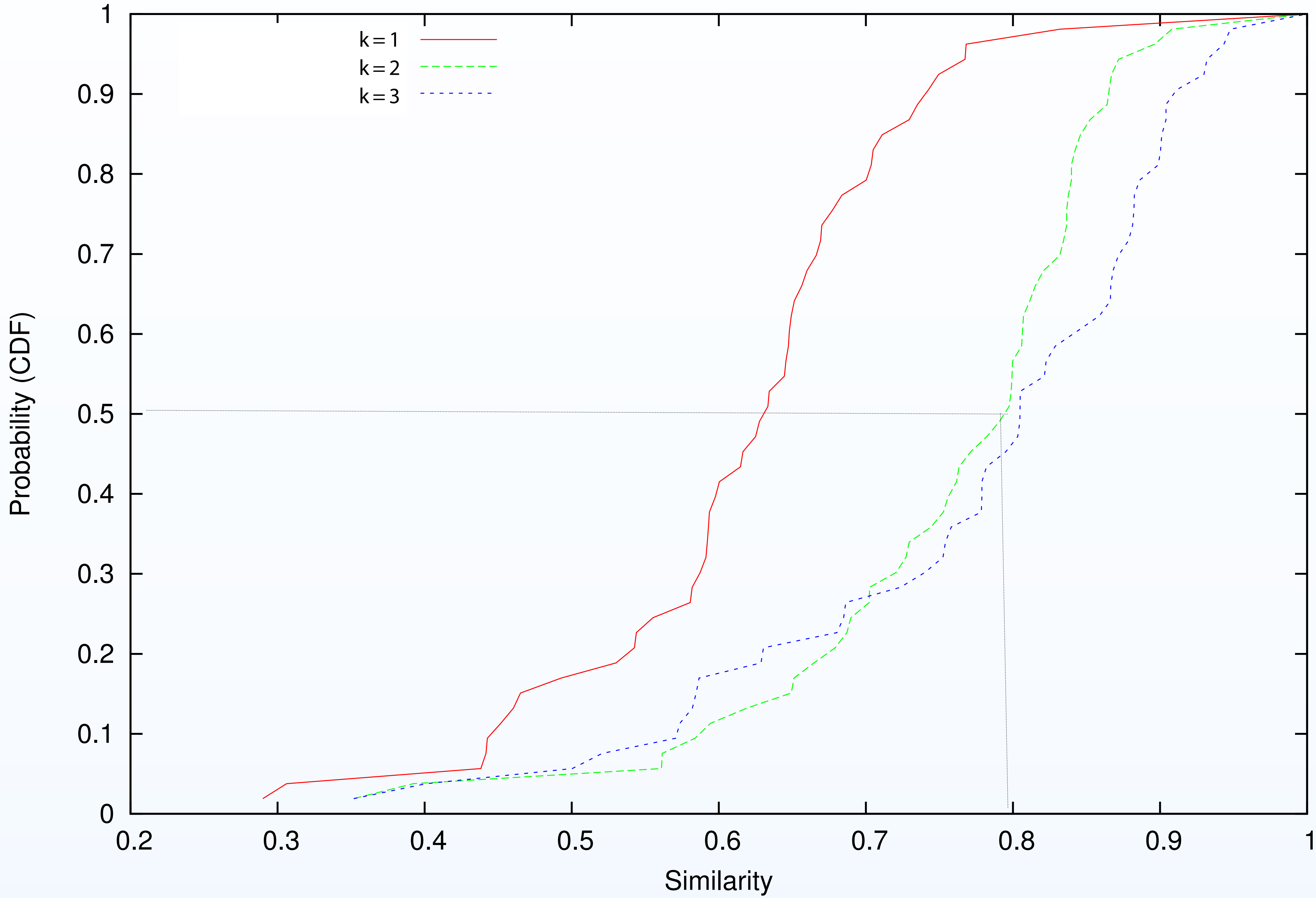
$VS_{max}(a, k, t)$:

maximum view similarity of
- assistee a
- over all subset of k assisstants
- at time t

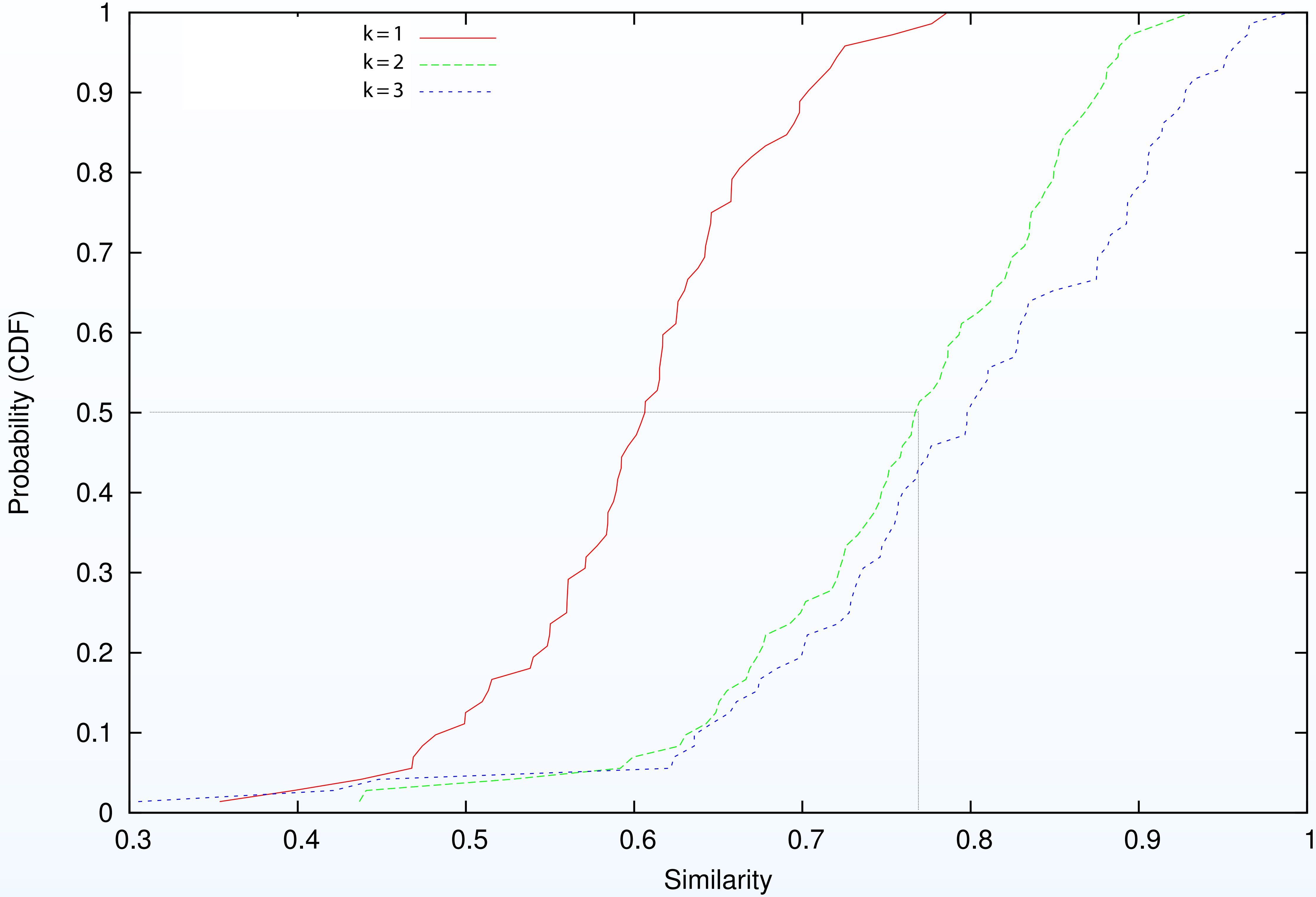
$\mathrm{VS}_{avg}(a, k)$:

average $\mathrm{VS}_{max}(a, k, t)$ over all t where a appears

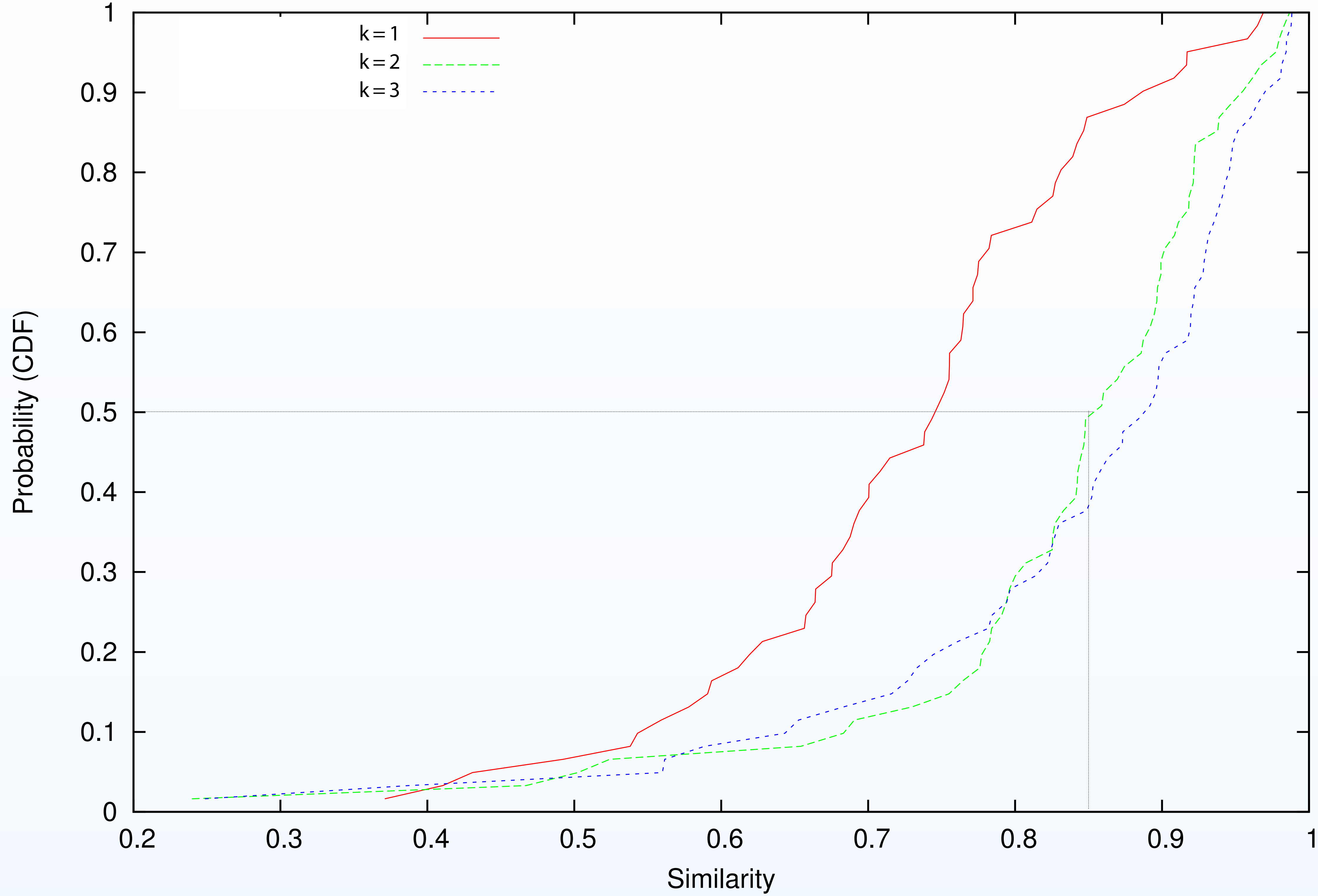
Sunland CDF of $VS_{avg}(a, k)$, for k = 1, 2, 3



Freebies CDF of $VS_{avg}(a, k)$, for $k = 1, 2, 3$



Japan Resort CDF of $VS_{avg}(a, k)$, for k = 1, 2, 3



2-3 assistants are sufficient.

More than half of users can find 2 assistants with >0.75 similarity on average (in the best case).

questions

1. **how to find assistants?**
2. will there be enough of them in practice?

Given:

***a*:** assistee
***k*:** limit on # of assistants
***A*:** set of candidate assistants

Find a subset $S \subseteq A$ such that $|S| \leq k$ and $VS(a, S)$ is maximum.

Greedy Heuristic

$S = \{\ \}$

repeat:

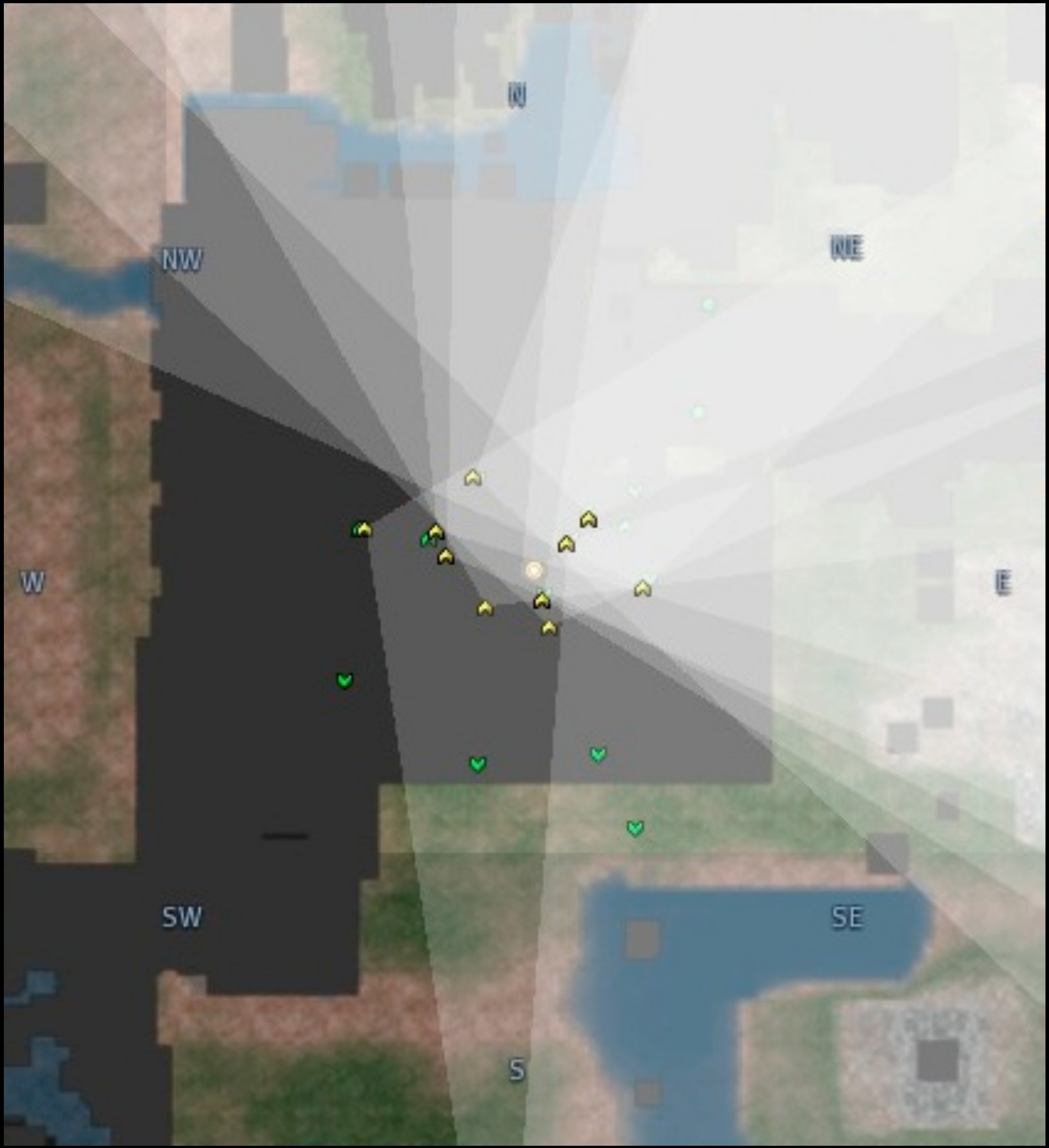
$$x_0 = \arg\max_{x \in A} VS(a, S \cup \{x\})$$

$$S = S \cup \{x_0\}$$

$$A = A - \{x_0\}$$

until $|S| > k$ or $VS(a,S) >$ threshold

Simulation Results


N
NW
NE
W
E
SW
SE
S

Rendering Results with Different Number of Assistants

k=1

k=3

k=5

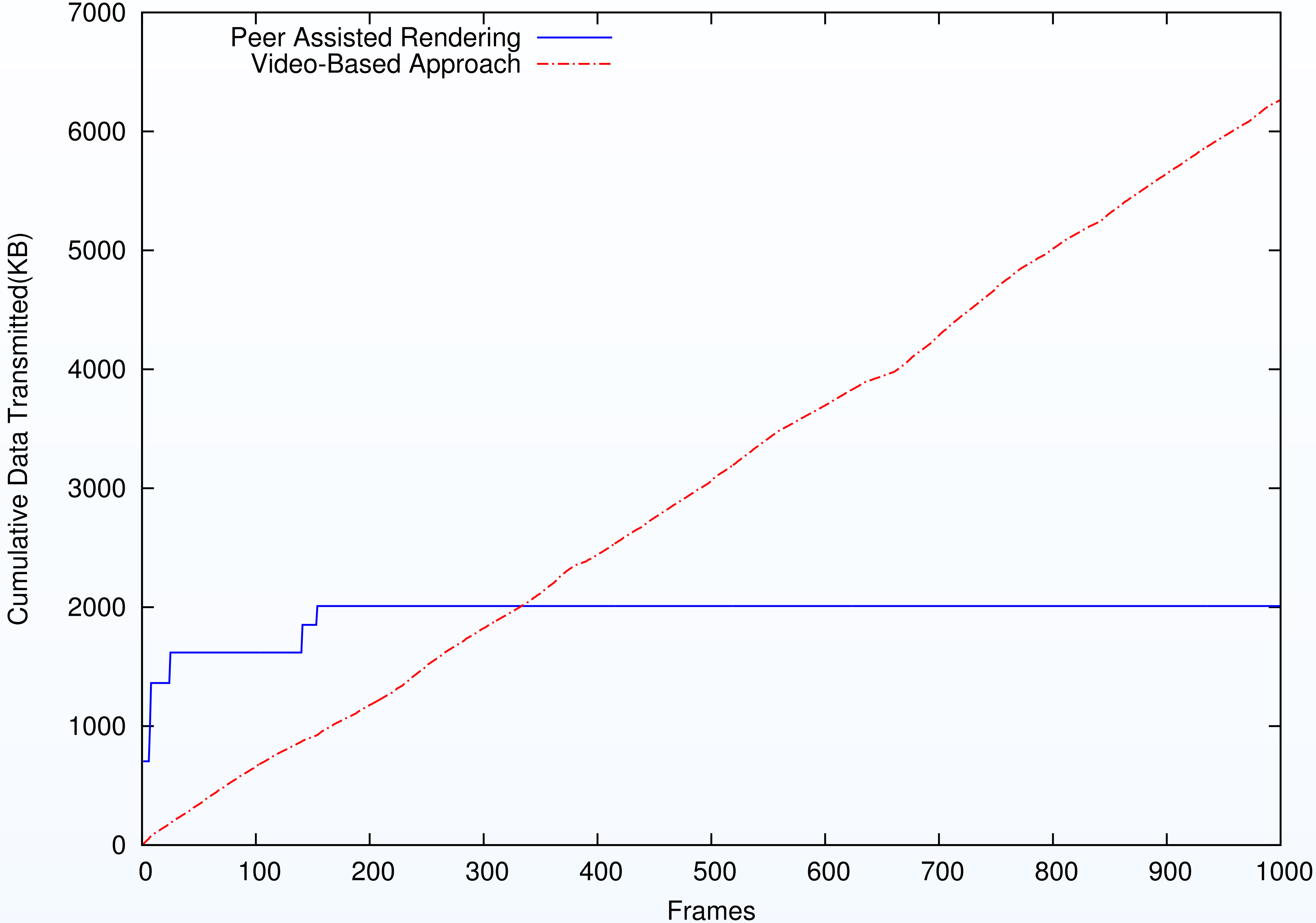
Rendering Results with Different Similarity Threshold

0.6

0.8

1.0

Transmission Overhead



On-going Research

Other considerations for choosing assistants

data completeness

CPU/GPU resources

network conditions

incentive

trust

Other work assistants can perform

download extra 3D objects?	render extra 3D objects?
X	X

Other work assistants can perform

download extra 3D objects?	render extra 3D objects?
X	X
X	✓

Other work assistants can perform

download extra 3D objects?	render extra 3D objects?
X	X
X	✓
✓	✓

Other ways of partitioning into rendering elements

given a frame rate threshold, maximize quality by adjusting what is rendered locally and what is rendered remotely.

Conclusion

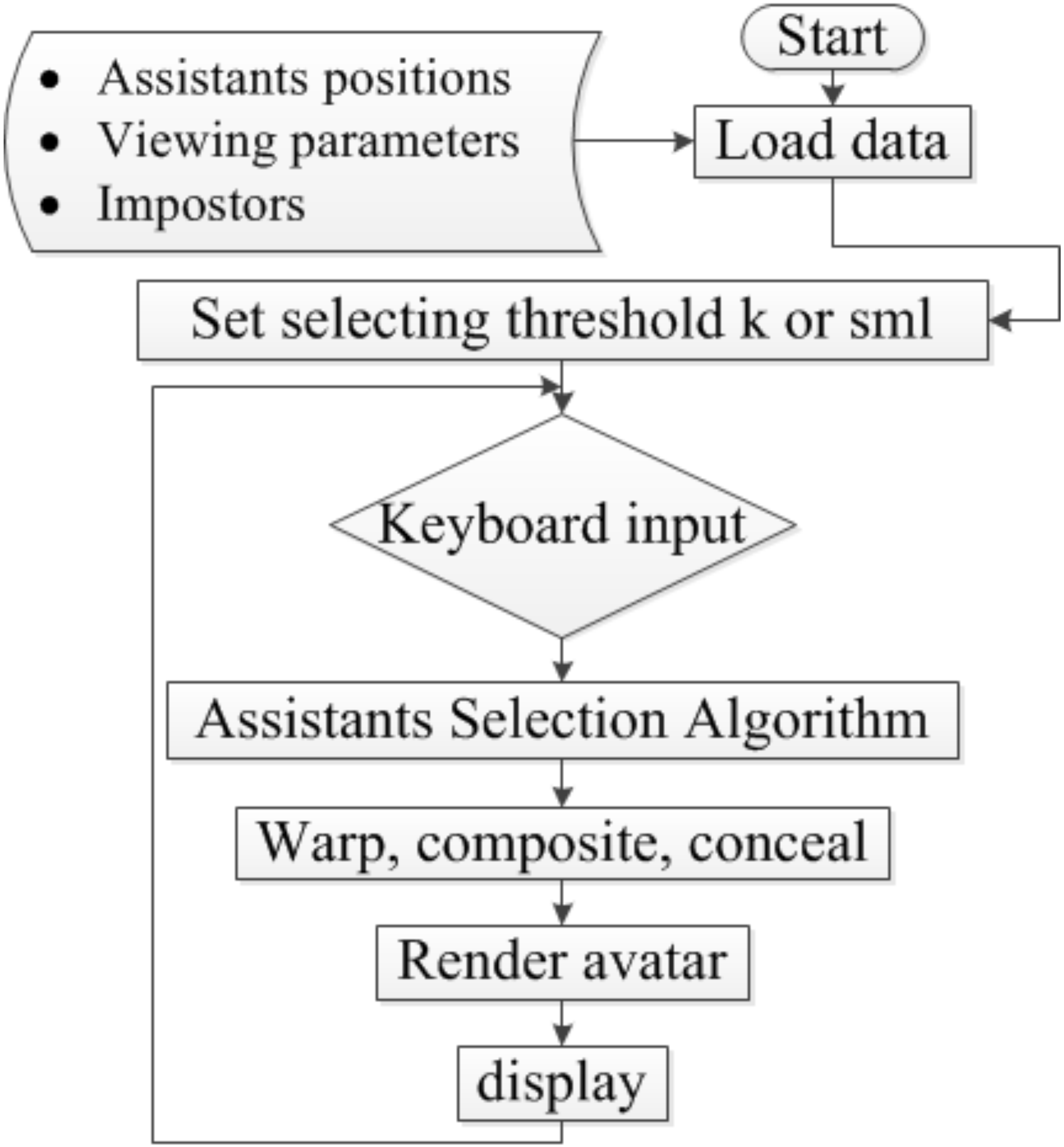
Two key ideas:

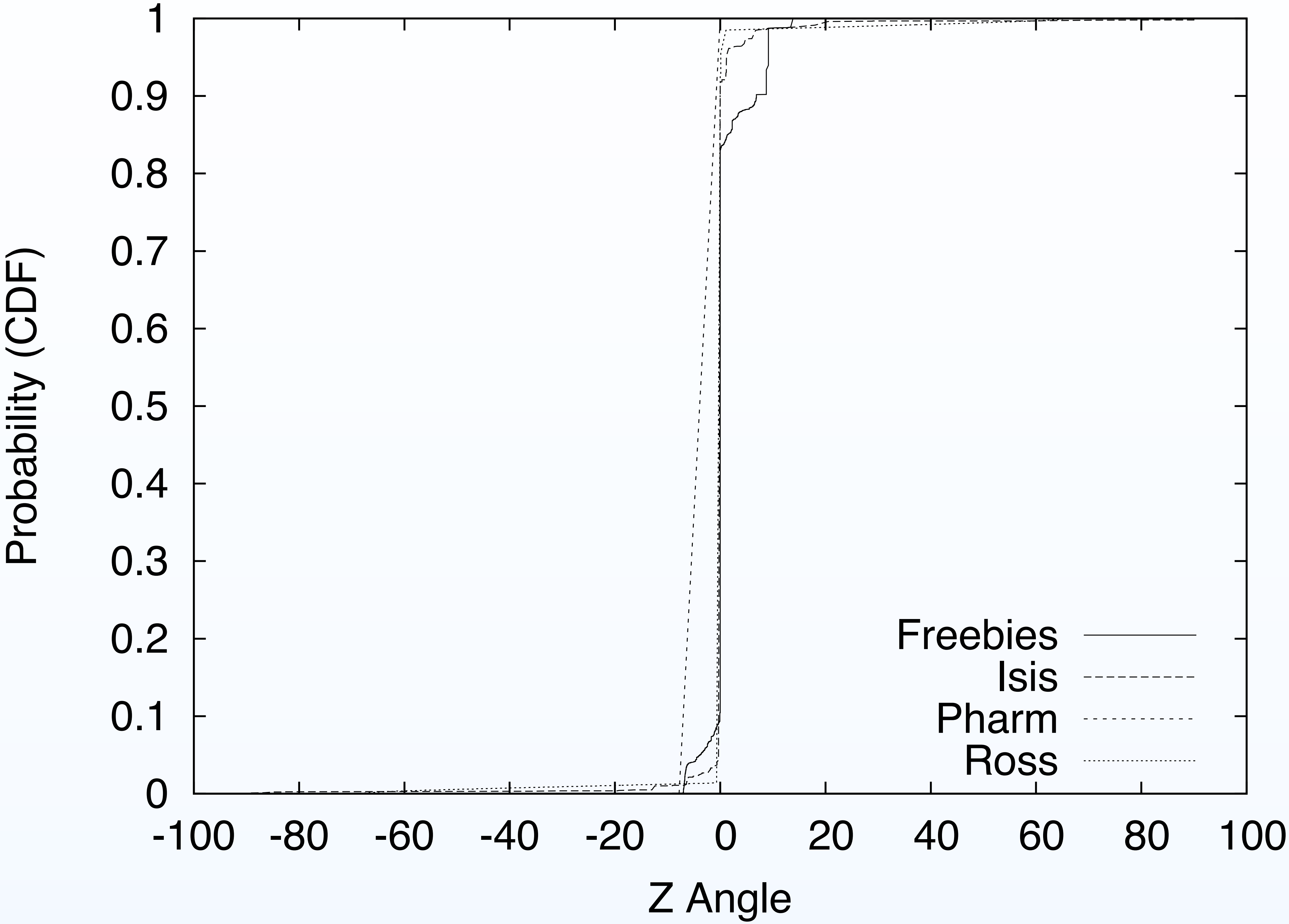
1. decompose into per-object impostors
2. exploit peers' rendering capability

謝謝

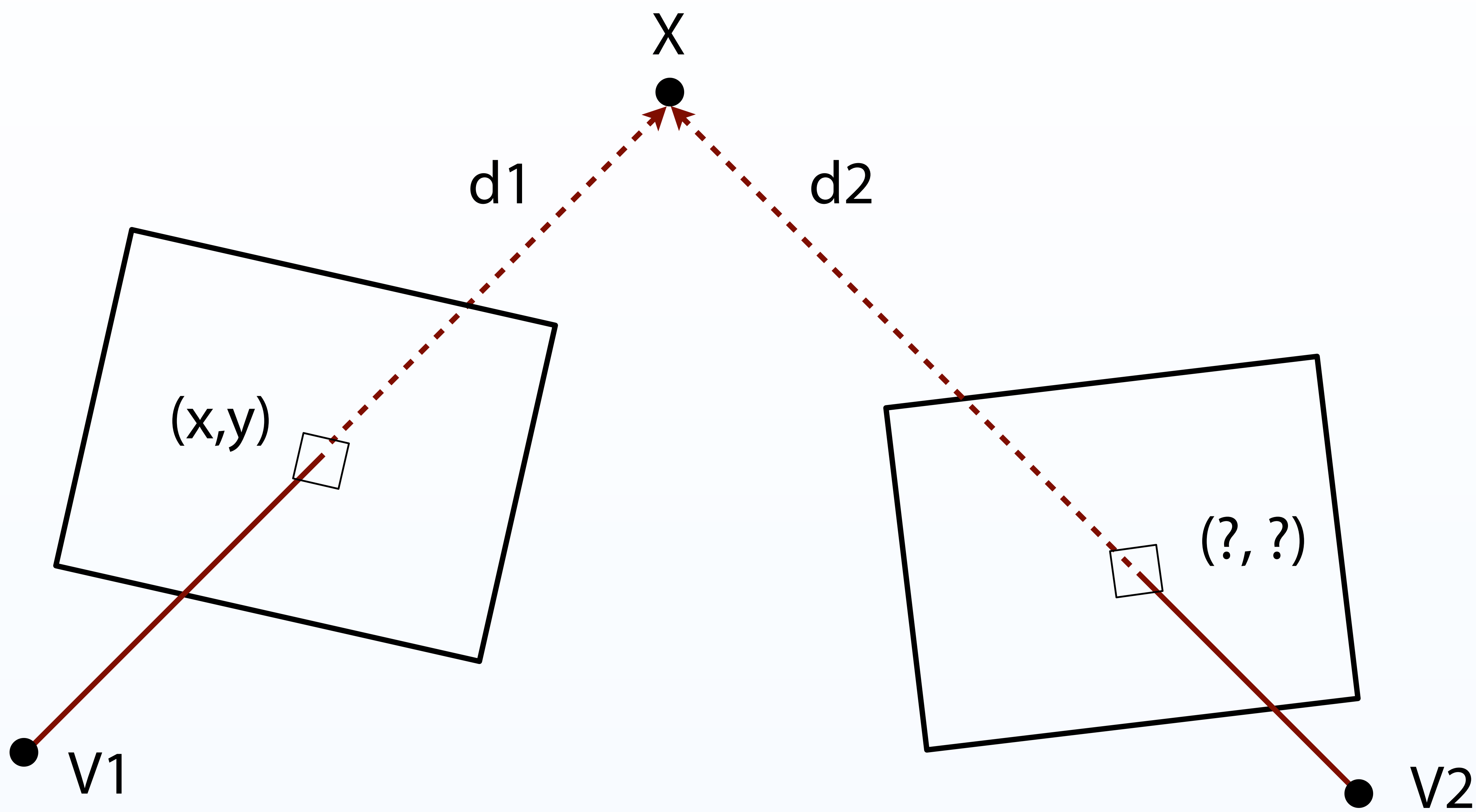
歡迎發問及指教

Backup Slides


Start
Assistants positions
Viewing parameters
Impostors
Load data
Set selecting threshold k or sml
Keyboard input
Assistants Selection Algorithm
Warp, composite, conceal
Render avatar
display



Probability (CDF)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-100
-80
-60
-40
-20
0
20
40
60
80
100
Z Angle
Freebies
Isis
Pharm
Ross


X
d1
d2
(x,y)
(?, ?)
V1
V2