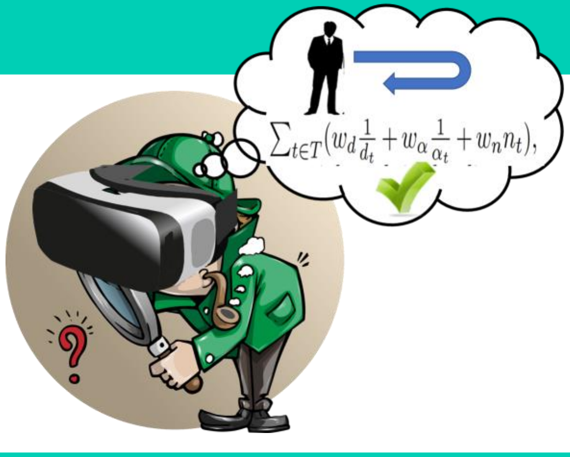
Toolset for Run-time Dataset Collection of Deep-scene Information

Gustav Aaro, Linköping University, Sweden

Daniel Roos, Linköping University, Sweden

Niklas Carlsson, Linköping University, Sweden



Proc. IEEE MASCOTS Workshop, Nov. 2020

First example scenario

Detective

Potential eye-witnesses

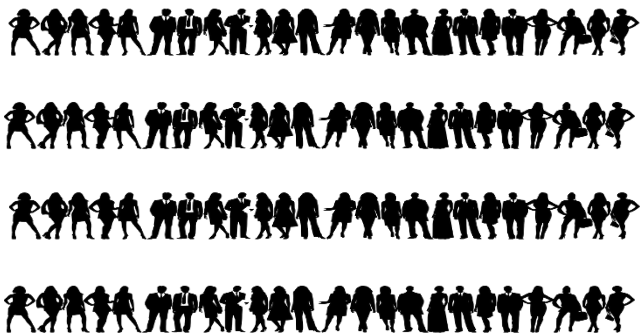
Second example scenario

Detective

Potential eye-witnesses

Second example scenario

Probability observed objects and things ??



Detective

Potential eye-witnesses

Contributions

- Methodology and software tool for generating run-time datasets capturing a user’s interactions with 3D environments

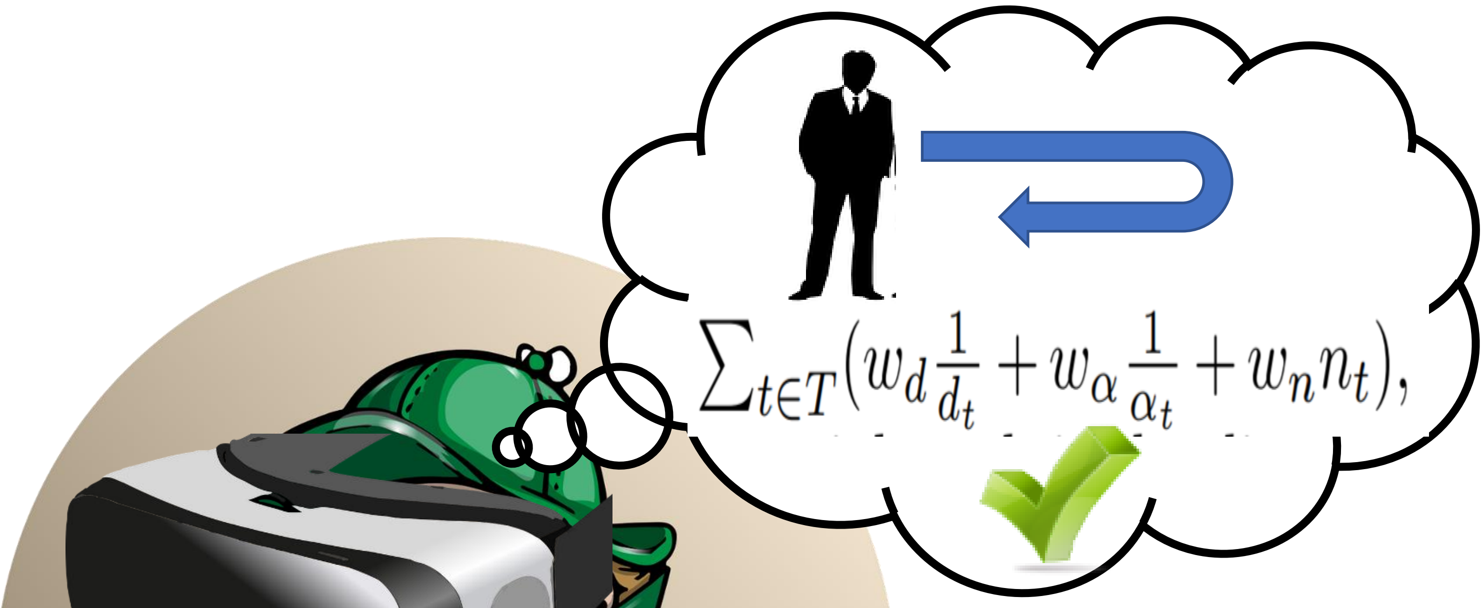
Contributions

- Methodology and software tool for generating run-time datasets capturing a user’s interactions with 3D environments
- Evaluate and compare different object identification methods that we implement within the tool

Contributions

- Methodology and software tool for generating run-time datasets capturing a user's interactions with 3D environments
- Evaluate and compare different object identification methods that we implement within the tool
- Use datasets collected with the tool to demonstrate example uses

Let’s start with desert first …



$$\sum_{t\in T}\left(w_d\frac{1}{d_t}+w_\alpha\frac{1}{\alpha_t}+w_n n_t\right),$$

Back to the main meal …

How do we effectively identify and get info about visible objects?

Per-object vs constant-ray …

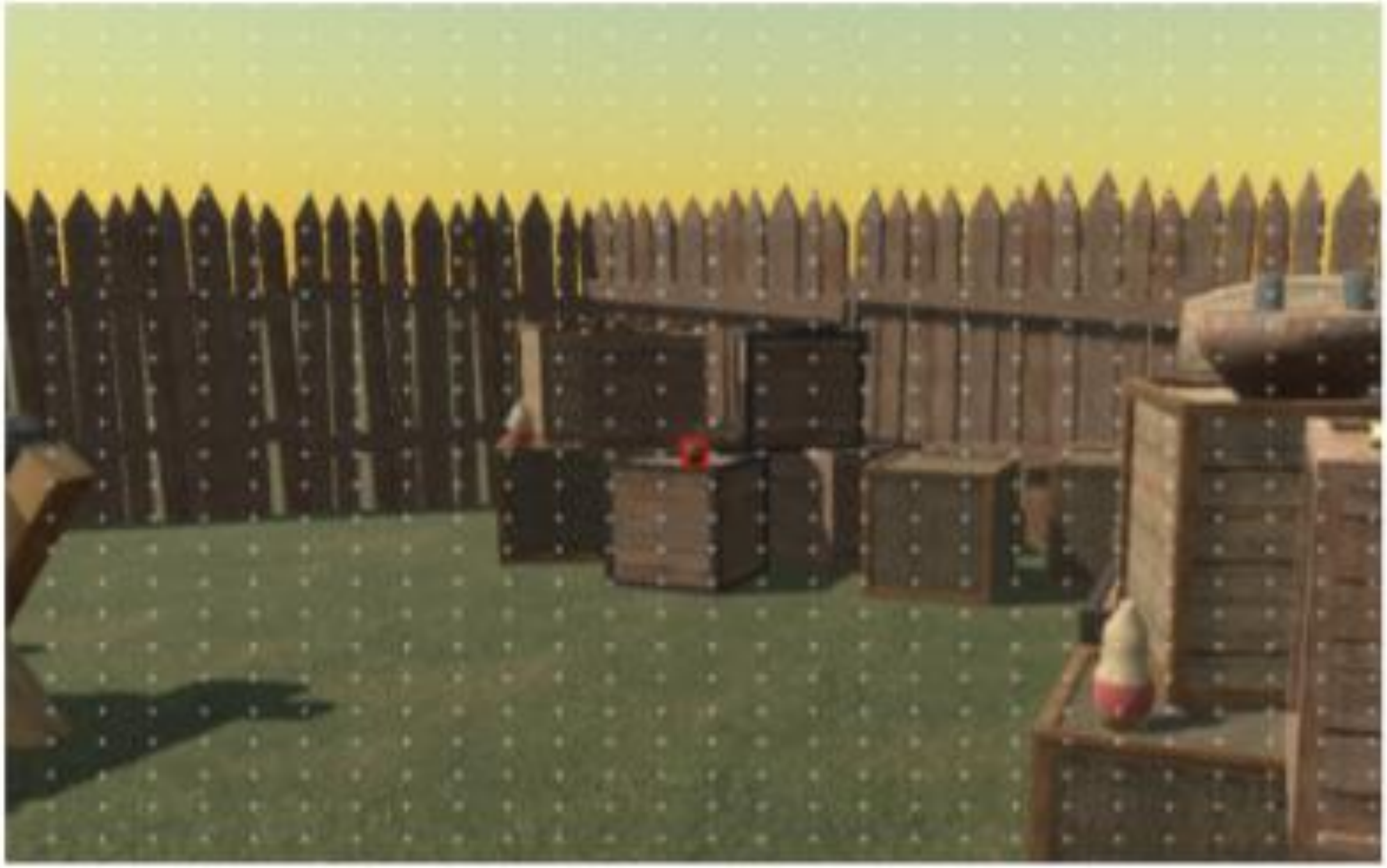
- Per-object does not scale well to large environments
  - Need to bound number of objects to consider

Per-object vs constant-ray ...

- Per-object approach does not scale well to large environments
  - Need to bound number of objects to consider

- Constant-ray approach considered next ...

Naïve ray-casting …

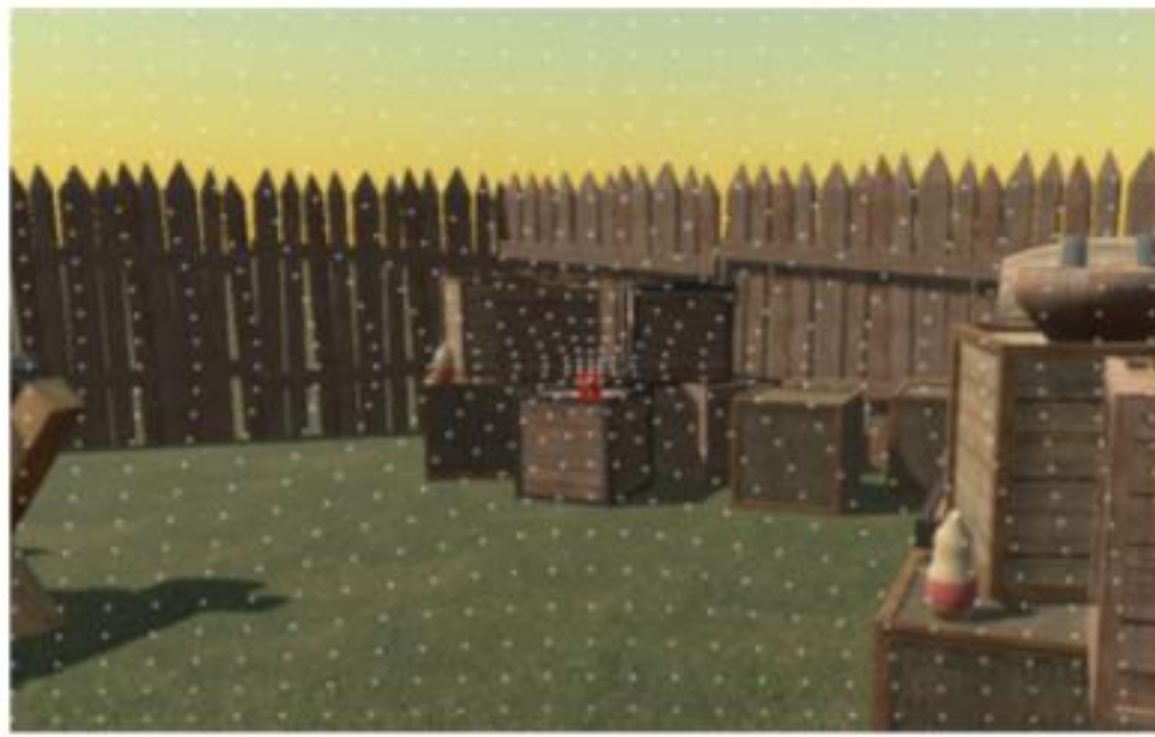


- Uniform grid …

Here, 1700 rays.

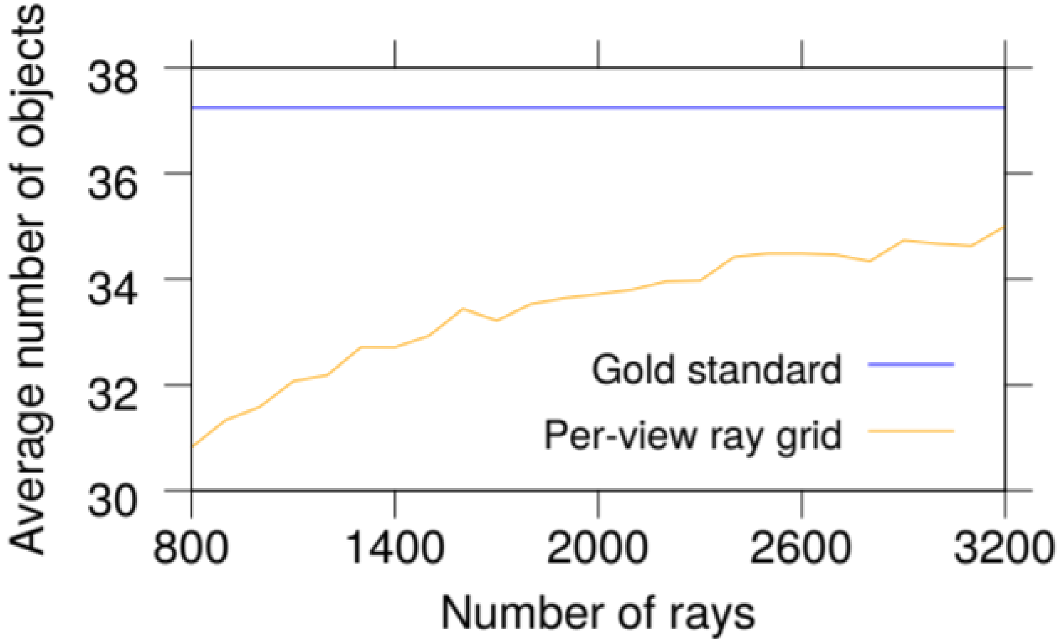
Baseline: Gaussian ray-casting …

$$f(x,y) = A \cdot e^{-\left(\frac{(x-x_0)^2}{2\cdot\sigma_x^2} + \frac{(y-y_0)^2}{2\cdot\sigma_y^2}\right)}$$



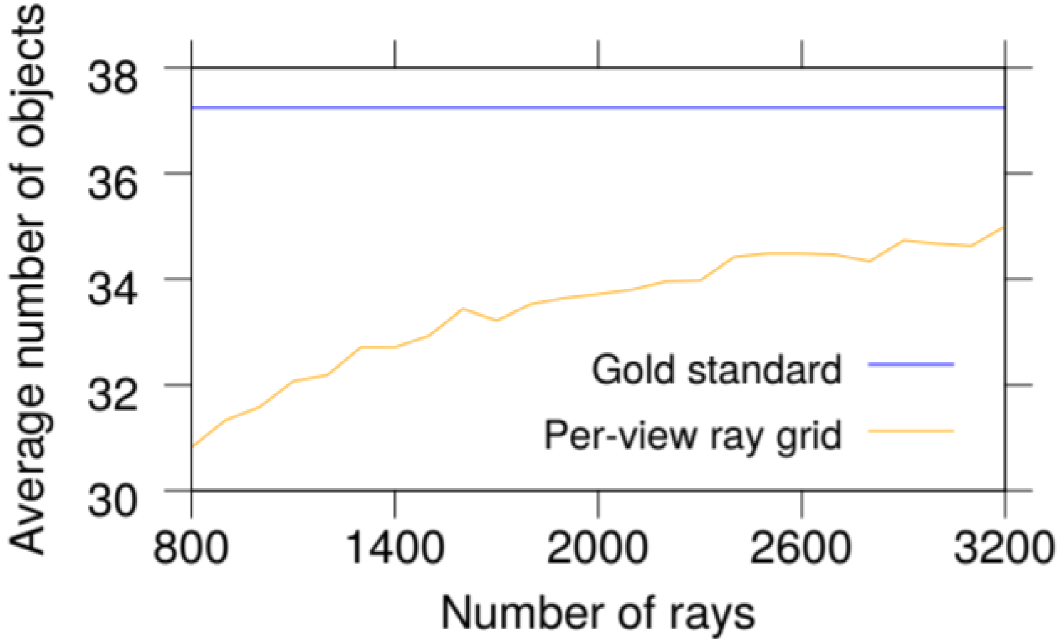
- More rays in center …

Evaluation: vs “gold standard”



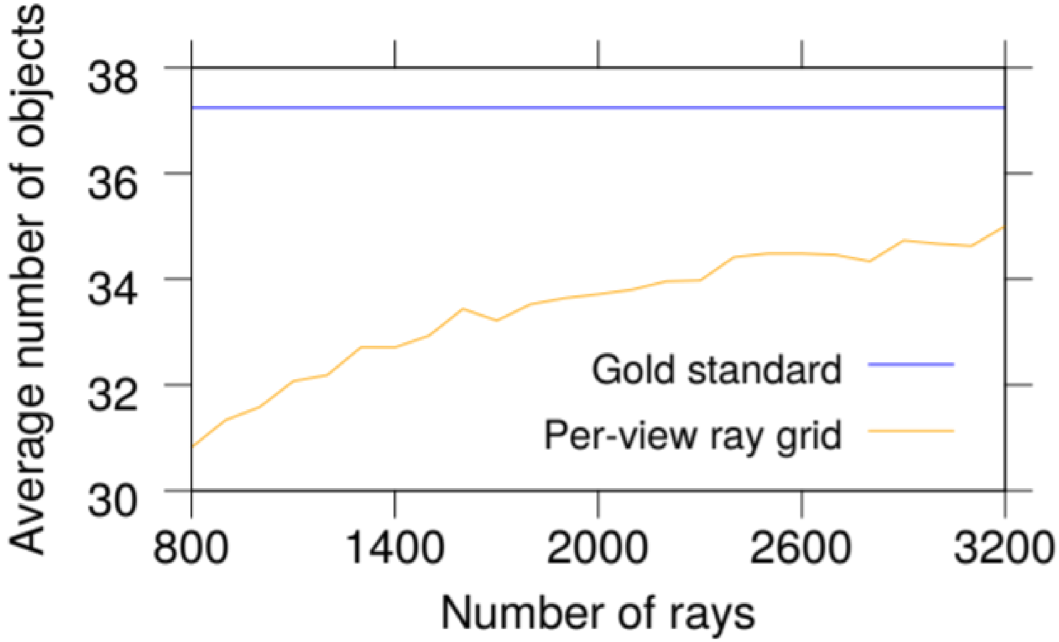
- Diminishing returns

Evaluation: vs “gold standard”



- Diminishing returns
- With 3,200 rays, we miss only 7% of optimal

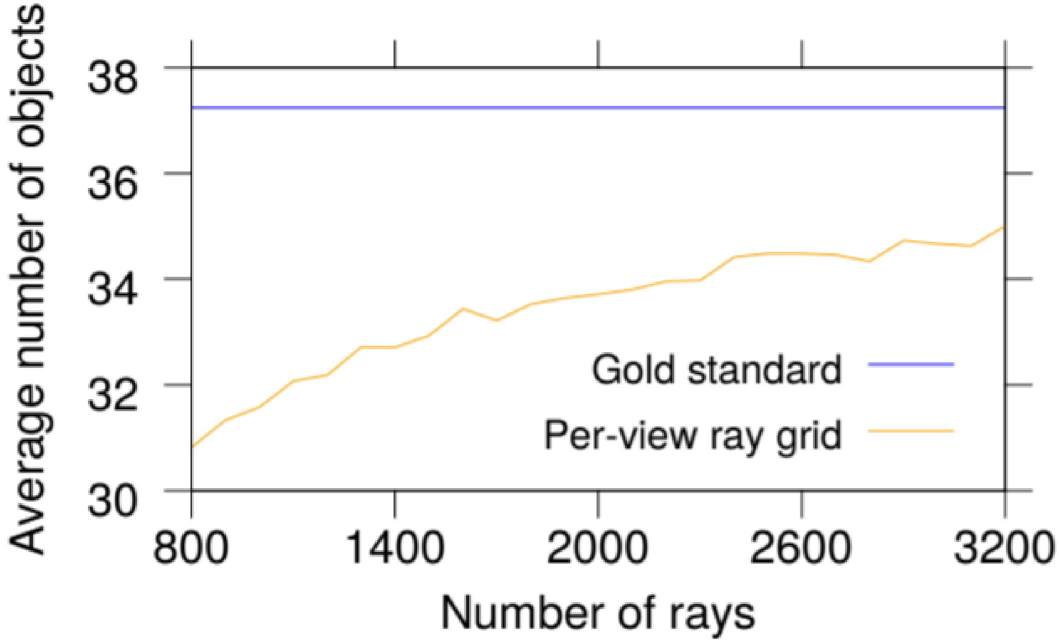
Evaluation: vs “gold standard”



- Diminishing returns
- With 3,200 rays, we miss only 7% of optimal

- Gold-standard obtained by casting 9,000 x 2,000 rays

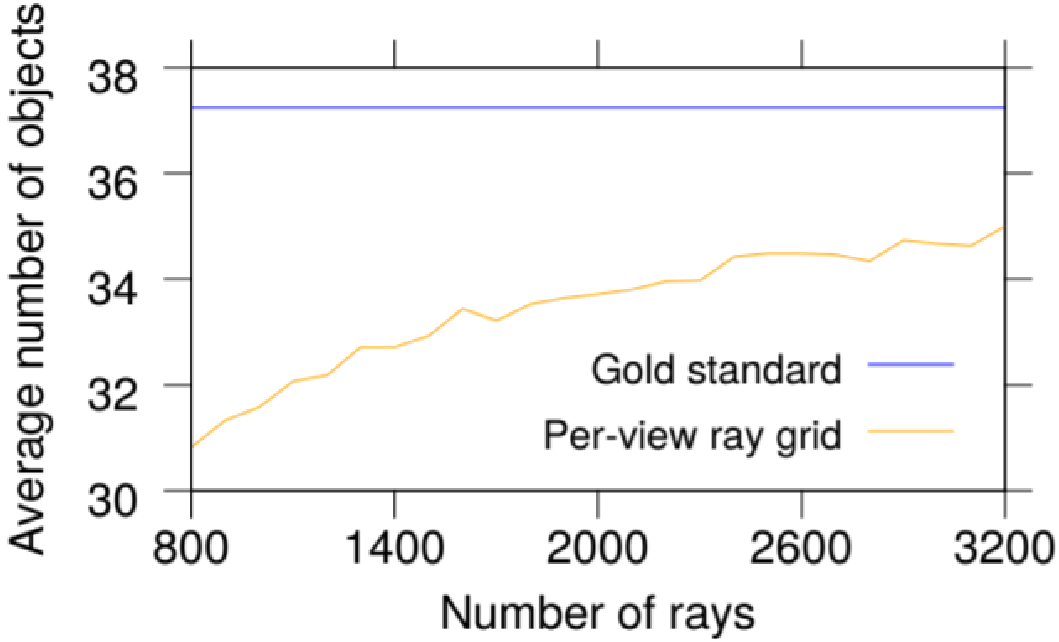
Evaluation: vs "gold standard"



- Diminishing returns
- With 3,200 rays, we miss only 7% of optimal

- Gold-standard obtained by casting 9,000 x 2,000 rays
  - Every pixel covered and have not found any object that other methods observe that "gold standard" does not.

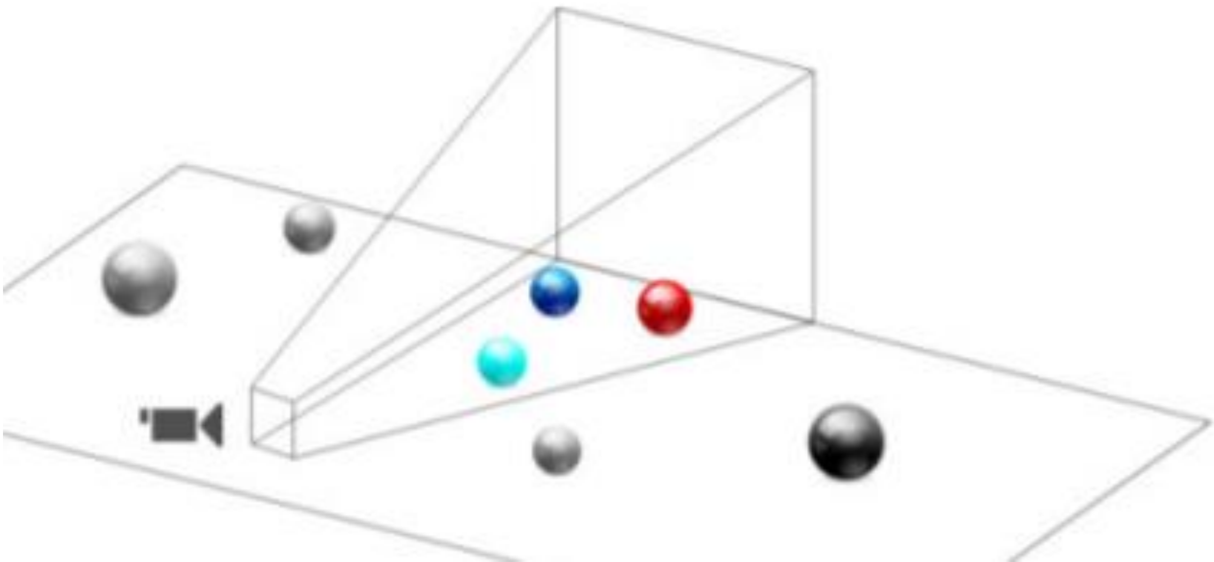
Evaluation: vs “gold standard”



- Diminishing returns
- With 3,200 rays, we miss only 7% of optimal

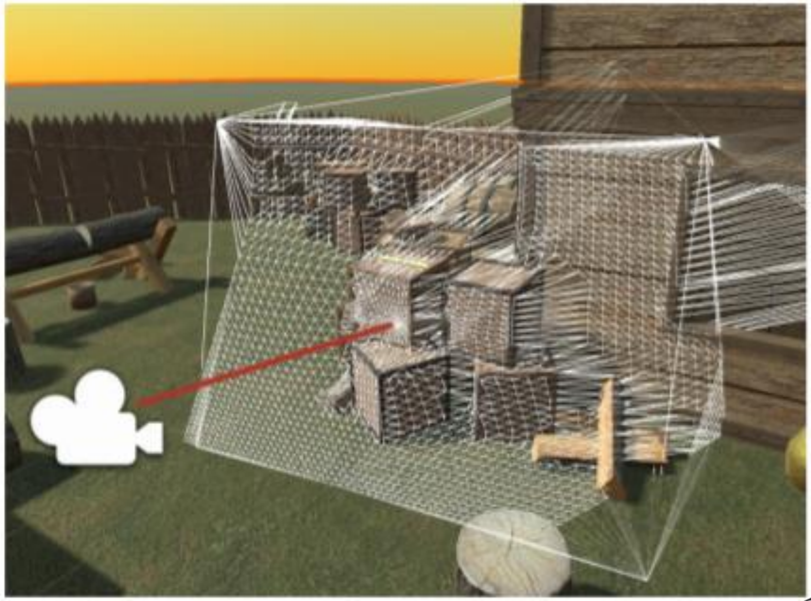
- Gold-standard obtained by casting 9,000 x 2,000 rays
  - Every pixel covered and have not found any object that other methods observe that “gold standard” does not.
  - For this reason, later we will only report recall. (Precision is always 100%.)

Refinement methods

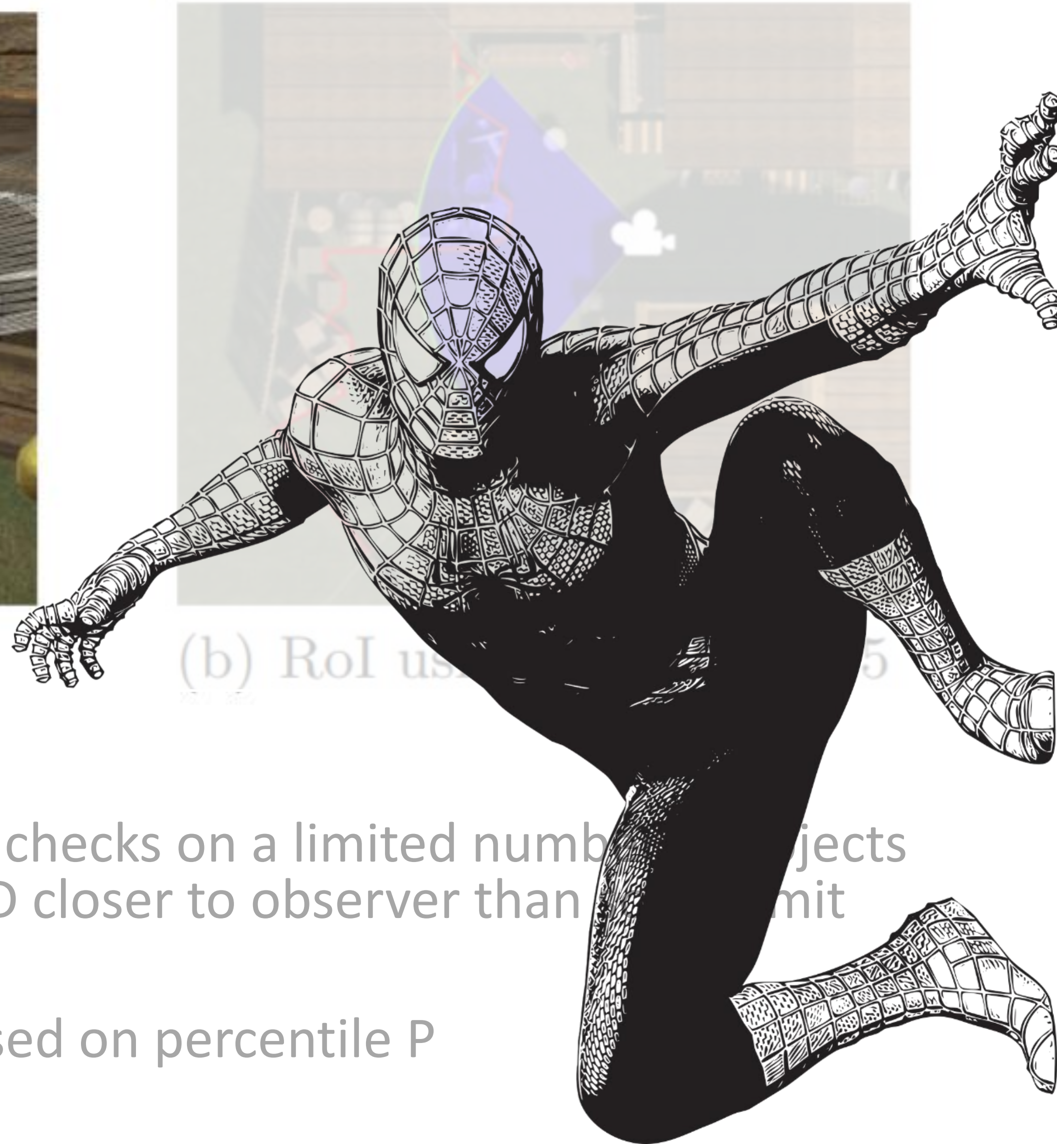


- Ideal: Perform extra per-object checks on a limited number of objects within the viewing frustum AND closer to observer than some limit

Refinement methods



(a) 3D Delaunay surface



(b) RoI us 5

- Ideal: Perform extra per-object checks on a limited numb jects within the viewing frustum AND closer to observer than mit
  - Delaunay surface (DS)
  - Distance threshold (DT) based on percentile P

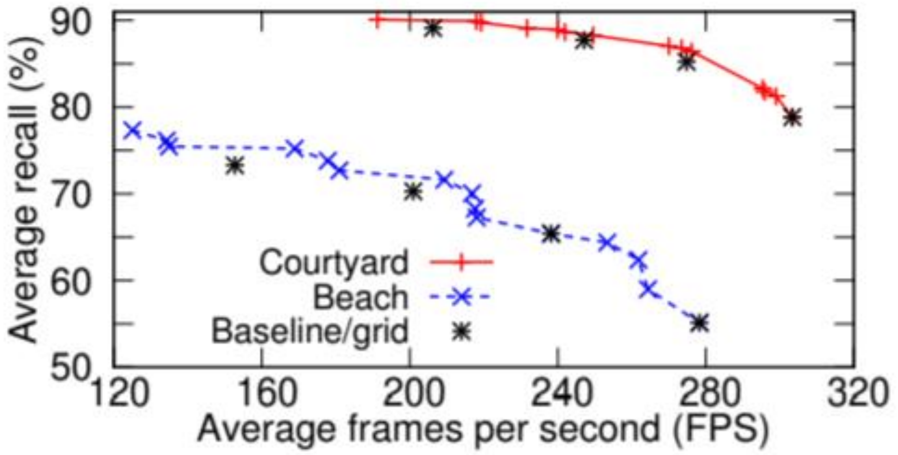
Refinement methods

(a) 3D Delaunay surface

(b) RoI using DT, $P = 0.5$

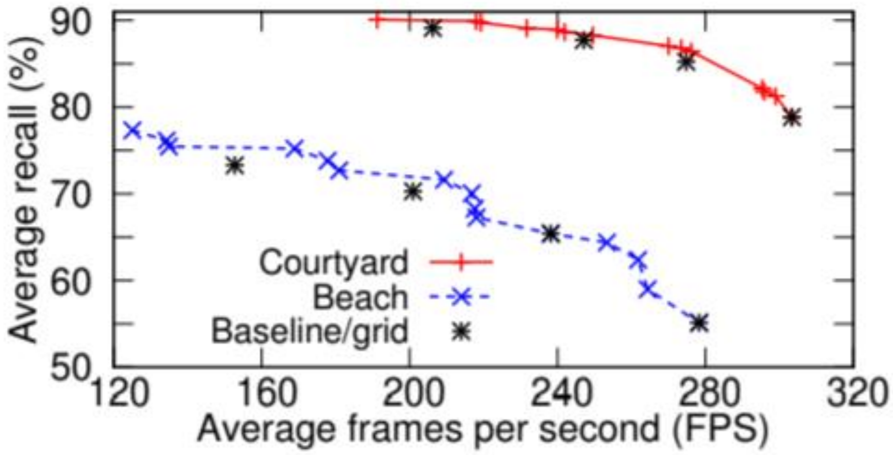
- Ideal: Perform extra per-object checks on a limited number of objects within the viewing frustum AND closer to observer than some limit
  - Delaunay surface (DS)
  - Distance threshold (DT) based on percentile P

Example results



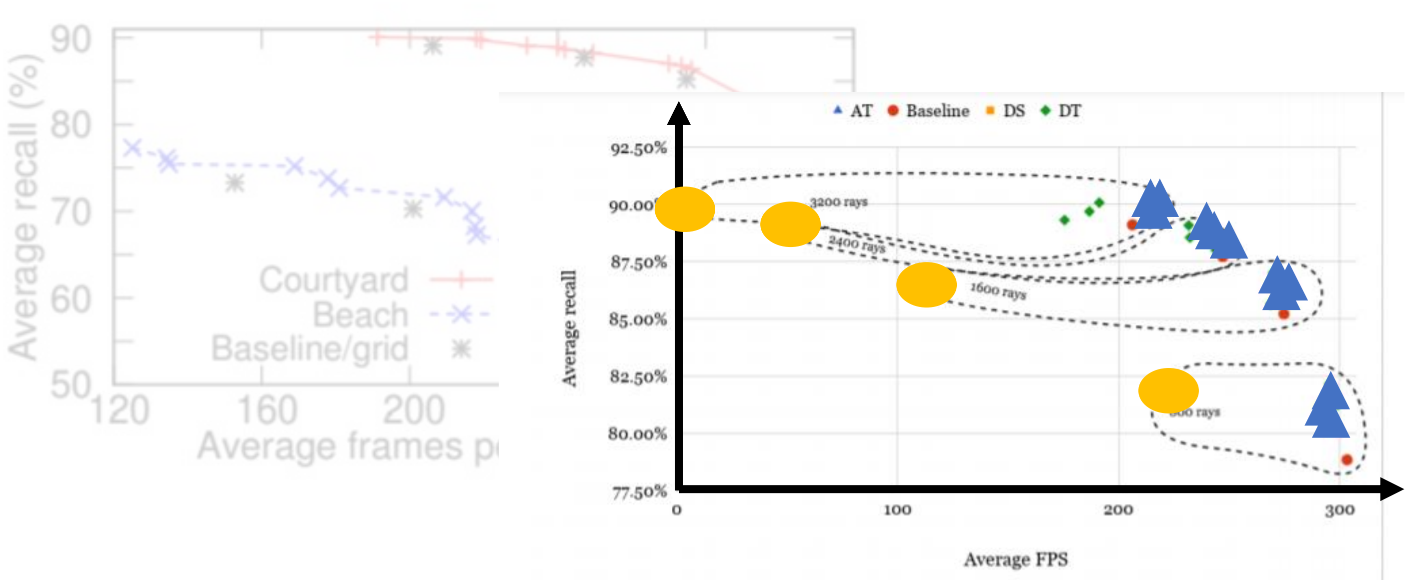
- Tradeoff between FPS and average recall

Example results



- Tradeoff between FPS and average recall
- Refinement methods can improve somewhat over baseline

Example results

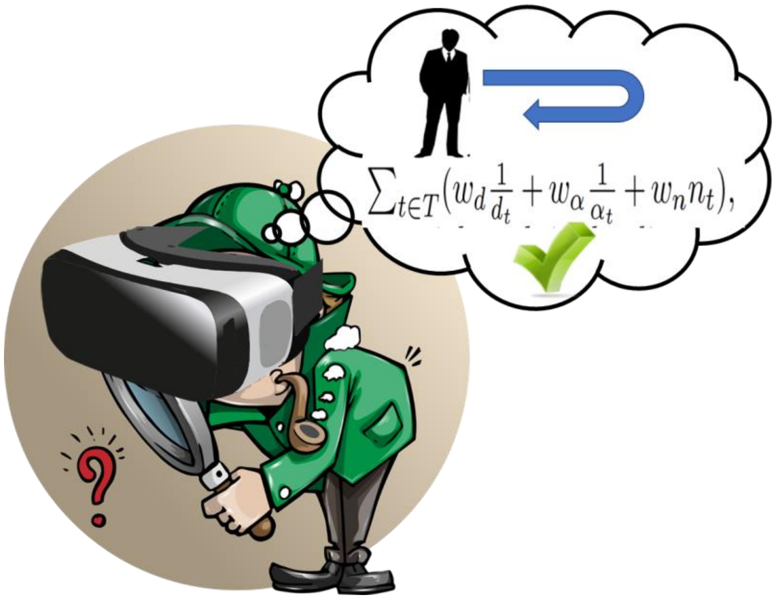


- Tradeoff between FPS and average recall
- Refinement methods can improve somewhat over baseline
  - Substantial overhead penalty to DS
  - DT typically provides the best tradeoff

Summary and conclusions

Methodology and software tool capturing

- **user movements** (position, rotation) and
- **visible objects** (object's identifier, distance, angle offset, volume, and how many rays hit the object at each time instance)
- at a **tunable time granularity** in immersive 3D environments.

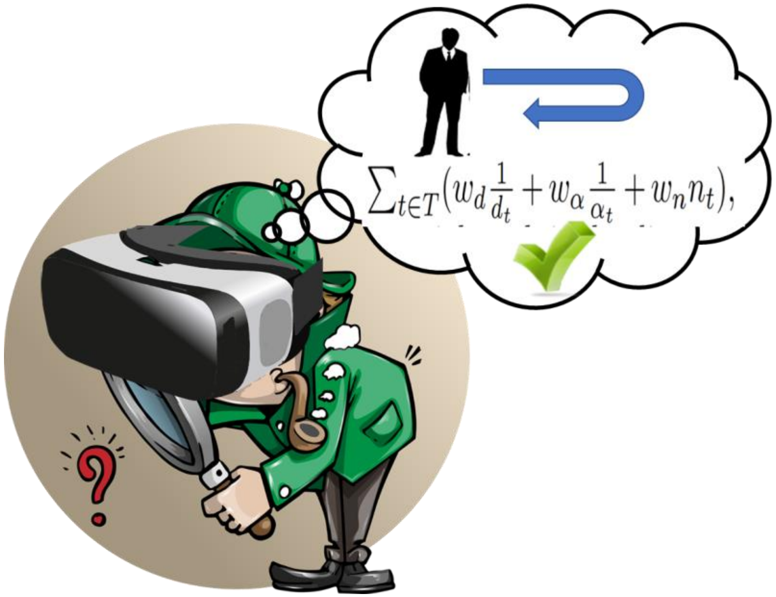


Summary and conclusions

Methodology and software tool capturing

- **user movements** (position, rotation) and
- **visible objects** (object's identifier, distance, angle offset, volume, and how many rays hit the object at each time instance)
- at a **tunable time granularity** in immersive 3D environments.

Lightweight object identification methods



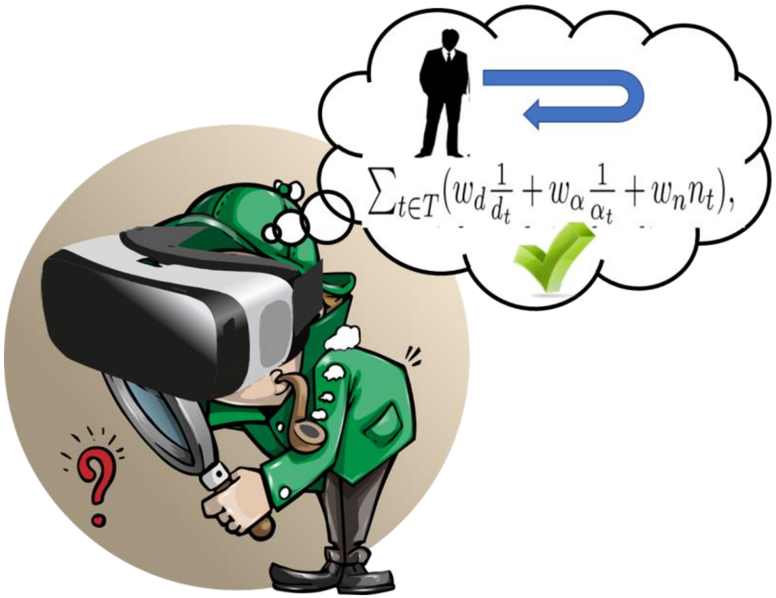
Summary and conclusions

Methodology and software tool capturing

- **user movements** (position, rotation) and
- **visible objects** (object's identifier, distance, angle offset, volume, and how many rays hit the object at each time instance)
- at a **tunable time granularity** in immersive 3D environments.

Lightweight object identification methods

Relatively simple methods to illustrate example use cases



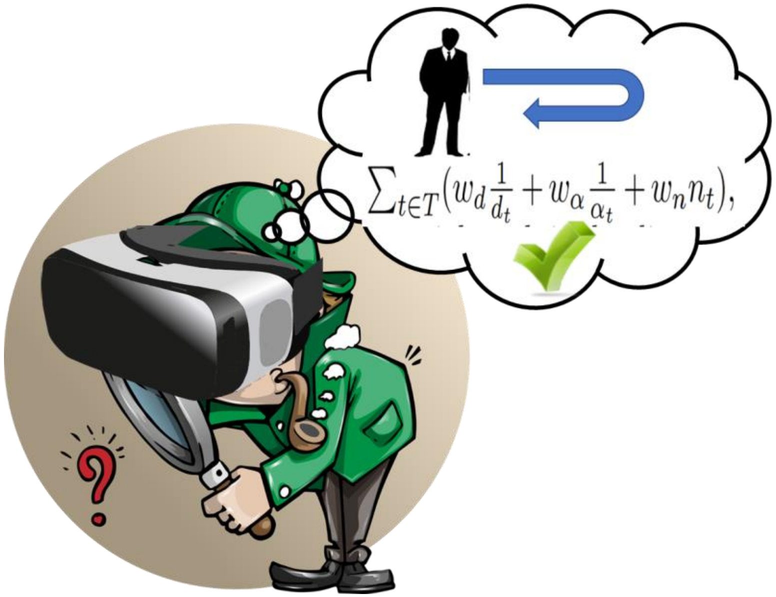
Summary and conclusions

Methodology and software tool capturing

- **user movements** (position, rotation) and
- **visible objects** (object's identifier, distance, angle offset, volume, and how many rays hit the object at each time instance)
- at a **tunable time granularity** in immersive 3D environments.

Lightweight object identification methods

Relatively simple methods to illustrate example use cases



Future work include user studies

Summary and conclusions

Methodology and software tool capturing

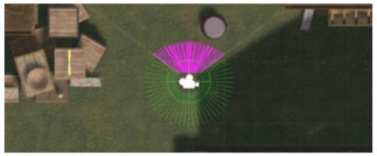
- **user movements** (position, rotation) and
- **visible objects** (object's identifier, distance, angle offset, volume, and how many rays hit the object at each time instance)
- at a **tunable time granularity** in immersive 3D environments.

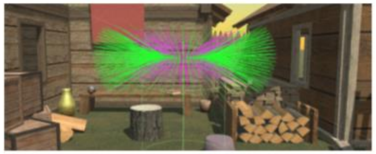
Lightweight object identification methods

Relatively simple methods to illustrate example use cases

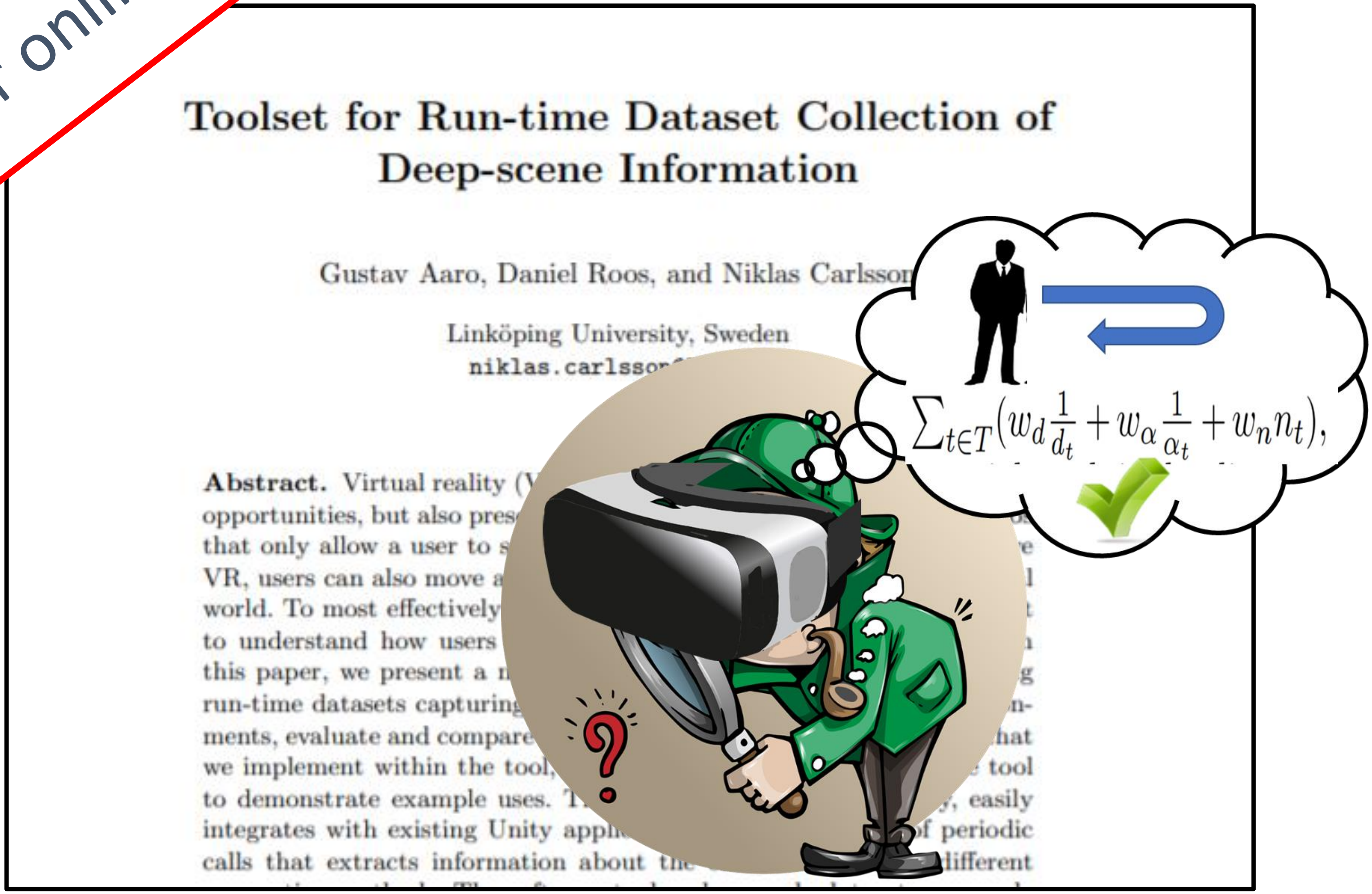
Future work include user studies

Have also extended tool to captures objects in other directions





Paper online!


Toolset for Run-time Dataset Collection of Deep-scene Information

Gustav Aaro, Daniel Roos, and Niklas Carlsson

Linköping University, Sweden
niklas.carlsson

Abstract. Virtual reality (V
opportunities, but also pres
that only allow a user to s
VR, users can also move a
world. To most effectively
to understand how users
this paper, we present a n
run-time datasets capturing
ments, evaluate and compare
we implement within the tool,
to demonstrate example uses. T
integrates with existing Unity appl
calls that extracts information about the

LINKÖPING UNIVERSITY

Niklas Carlsson (niklas.carlsson@liu.se)

* Illustrative images created using images from pixabay.com