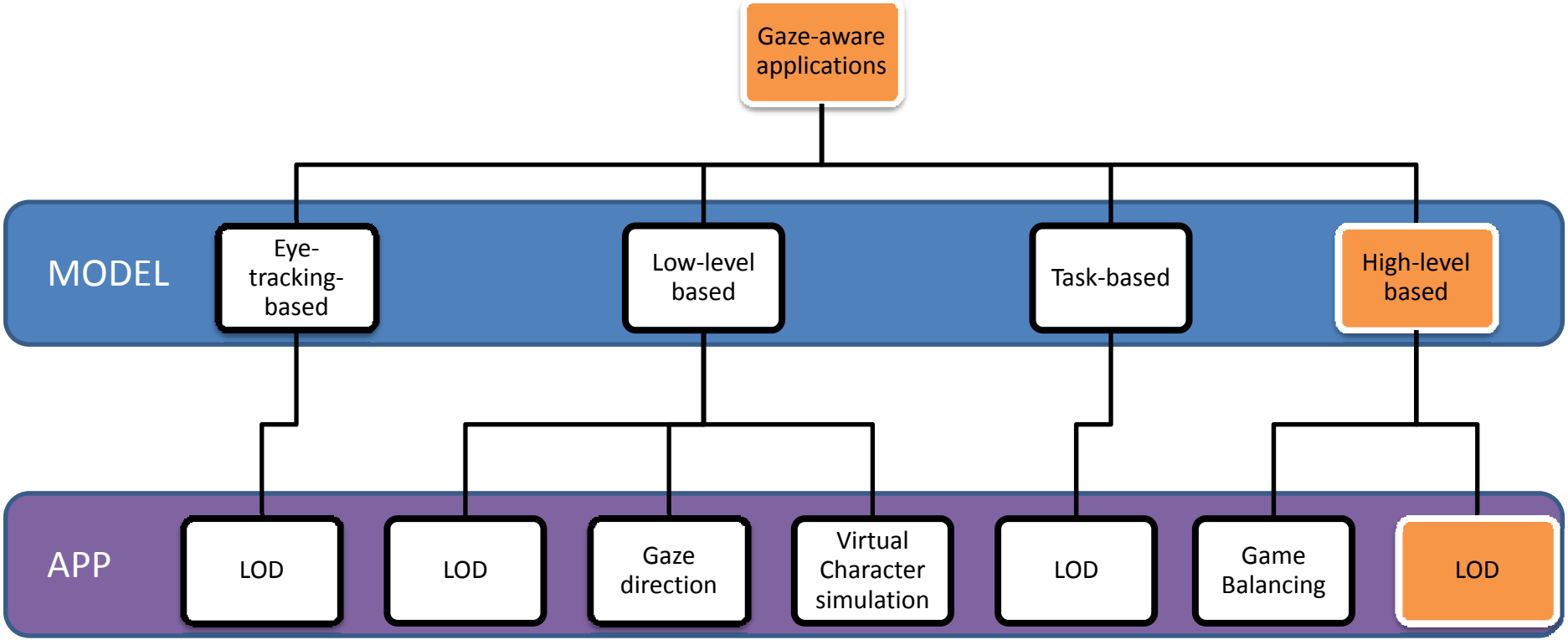
Outline


Gaze-aware applications
MODEL
Eye-tracking-based
Low-level based
Task-based
High-level based
APP
LOD
LOD
Gaze direction
Virtual Character simulation
LOD
Game Balancing
LOD

Extending the High Level Saliency Model
Introducing new High Level factors

Extending the High Level Saliency Model of Part 1

Extending our perceptual model
Introducing new High Level factors

Introducing new High Level factors to increase model accuracy

- Gaze allocation influenced by several other high level factors
- We set 3 criteria to be satisfied when introducing a factor
    1. The factor should affect attention intensely
    2. The factor should be measurable
    3. The factor should be observed in a video game

Extending our perceptual model
New High Level Factors

We introduce four additional components

1. Contextual isolation of objects

High Level Saliency
Contextual Isolation

VIDEO

Extending our perceptual model
New High Level Factors

We introduce four additional components

1. Contextual isolation of objects
2. Canonical form of objects

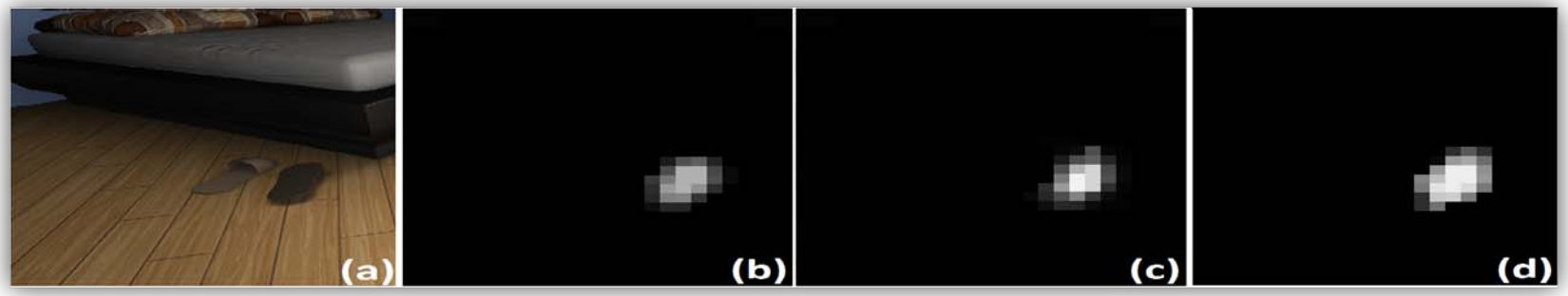
High Level Saliency
Canonical Form

VIDEO

Extending our perceptual model
New High Level Factors

We introduce four additional components

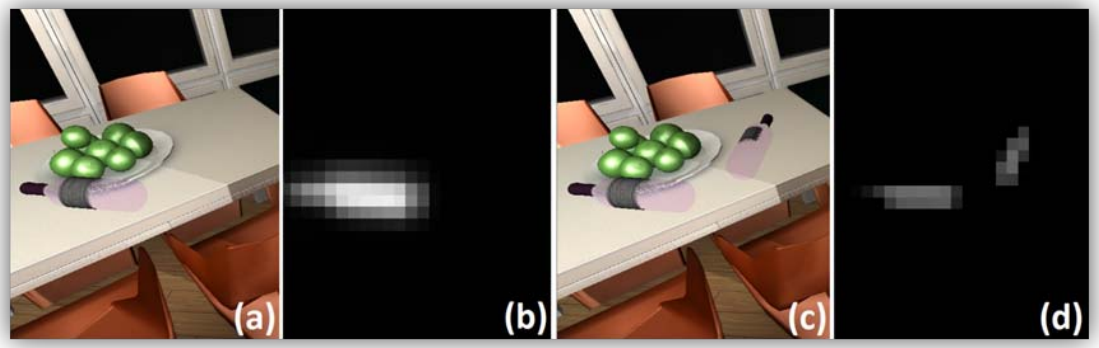
1. Contextual isolation of objects
2. Canonical form of objects
3. Temporal object coherence



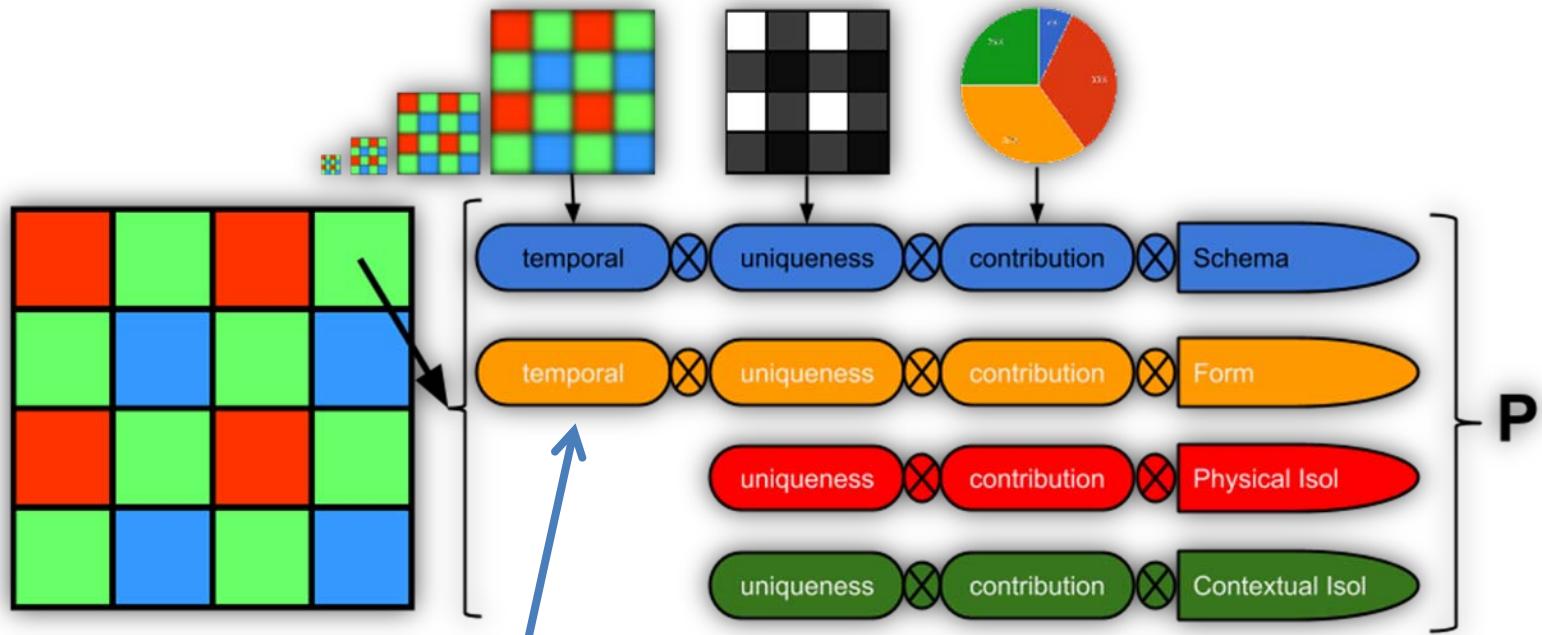
Extending our perceptual model
New High Level Factors

We introduce four additional components

1. Contextual isolation of objects
2. Canonical form of objects
3. Temporal object coherence
4. Feature uniqueness property



Extending our perceptual model
Estimating Posterior probability of attendance



$$w^{tmp}_{unit,x,y,f} = \prod_{f=1}^{F} P_{unit,x,y,f} e^{-af}$$

$$w^{unq}_{unit} = \frac{1}{|\vee|P_{unit,x,y,f}}$$

Perceptual Experiment
Obtaining experimental data (1)

Motivation

- Examine the effect of added factors on visual attention via
  - Task completion time
  - Eye tracking
- Obtain contribution weights of each factor

Perceptual Experiment
Obtaining experimental data (2)

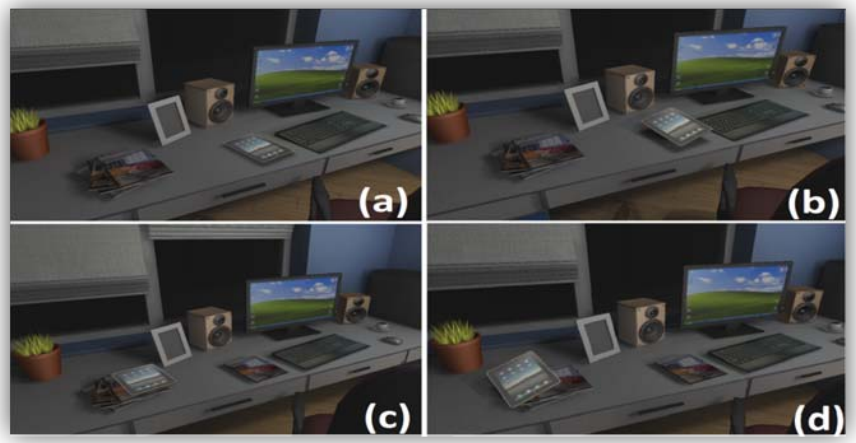
Stimuli used in order to obtain contribution weights

- Factorial combination of factors - 4 conditions generated
- 48 people, 12/condition

Perceptual Experiment
Conditions

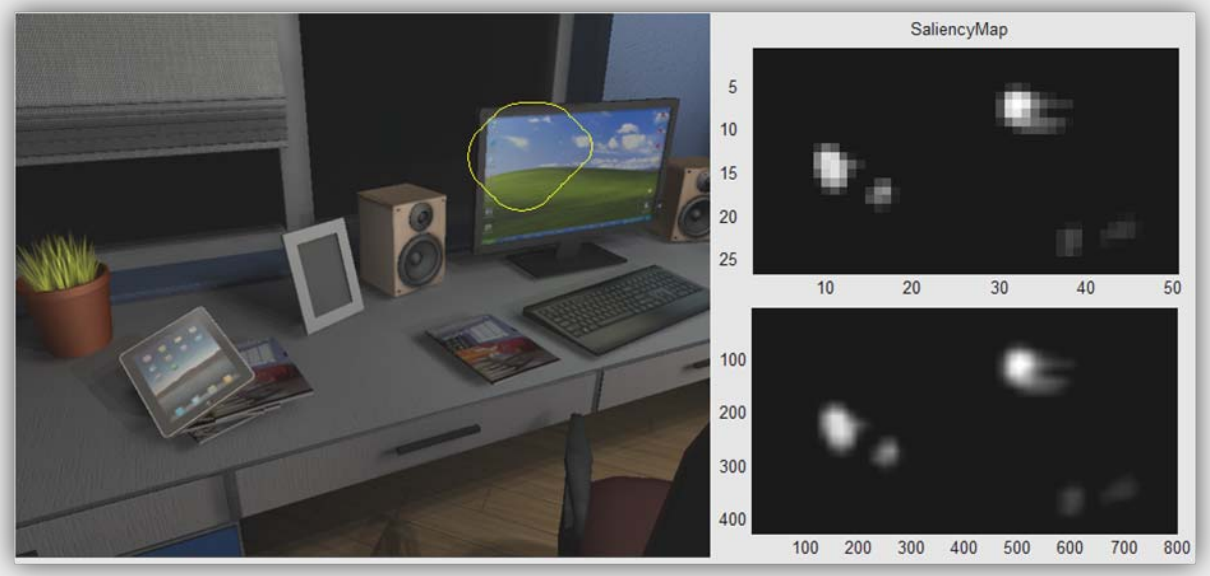
Conditions

1. Canonical/Compound
2. Non-Canonical/Compound
3. Canonical/Singleton
4. Non-Canonical/Singleton



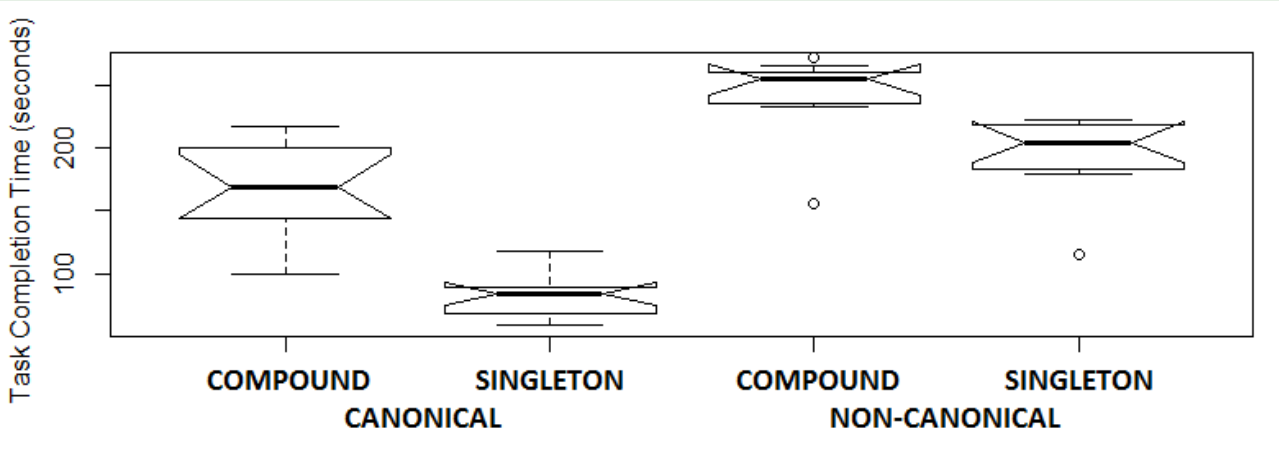
Perceptual Experiment
Saliency Toolbox

Object placements had minimum low level saliency ***Walther & Koch, 2006***



Perceptual Experiment
Results

Task completion time distribution of the experimental conditions



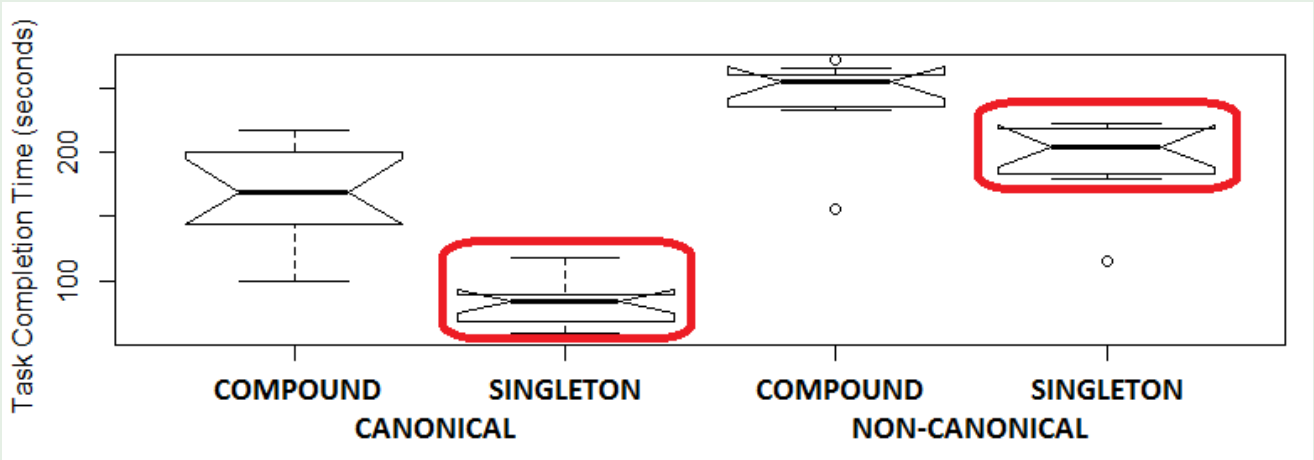
Perceptual Experiment
Results

Conclusions

- Introduced factors affect attention deployment
- Contextually isolated objects pop-out

Perceptual Experiment
Results

Task completion time distribution of the experimental conditions



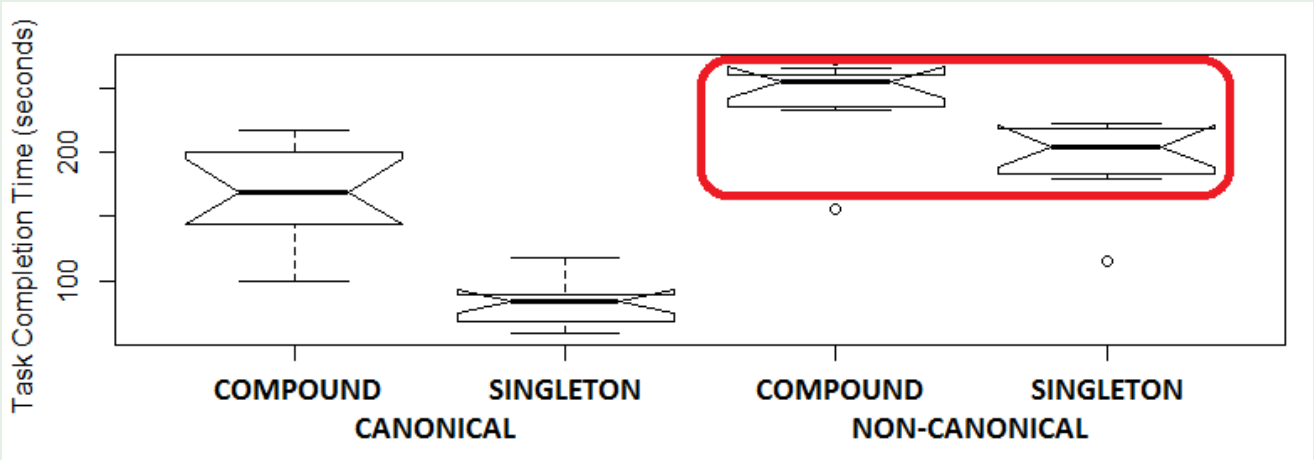
Perceptual Experiment
Results

Conclusions

- Introduced factors affect attention deployment
- Contextually isolated objects pop-out
- Objects in Non-Canonical form
  - Take longer to be recognized
  - Are actively observed

Perceptual Experiment
Results

Task completion time distribution of the experimental conditions



Perceptual Experiment
Results

Conclusions

- Introduced factors affect attention deployment
- Contextually isolated objects pop-out
- Objects in Non-Canonical form
  - Take longer to be recognized
  - Are actively observed
- Contribution weights for each factor were generated from timings and inserted into the model
  - Included weights of Part 1
- **Schema 7%, Physical Isolation 33%, Canonical Form 35% Contextual Isolation 25%**

LOD for Mobile Graphics
C-LOD for Unity 3D™

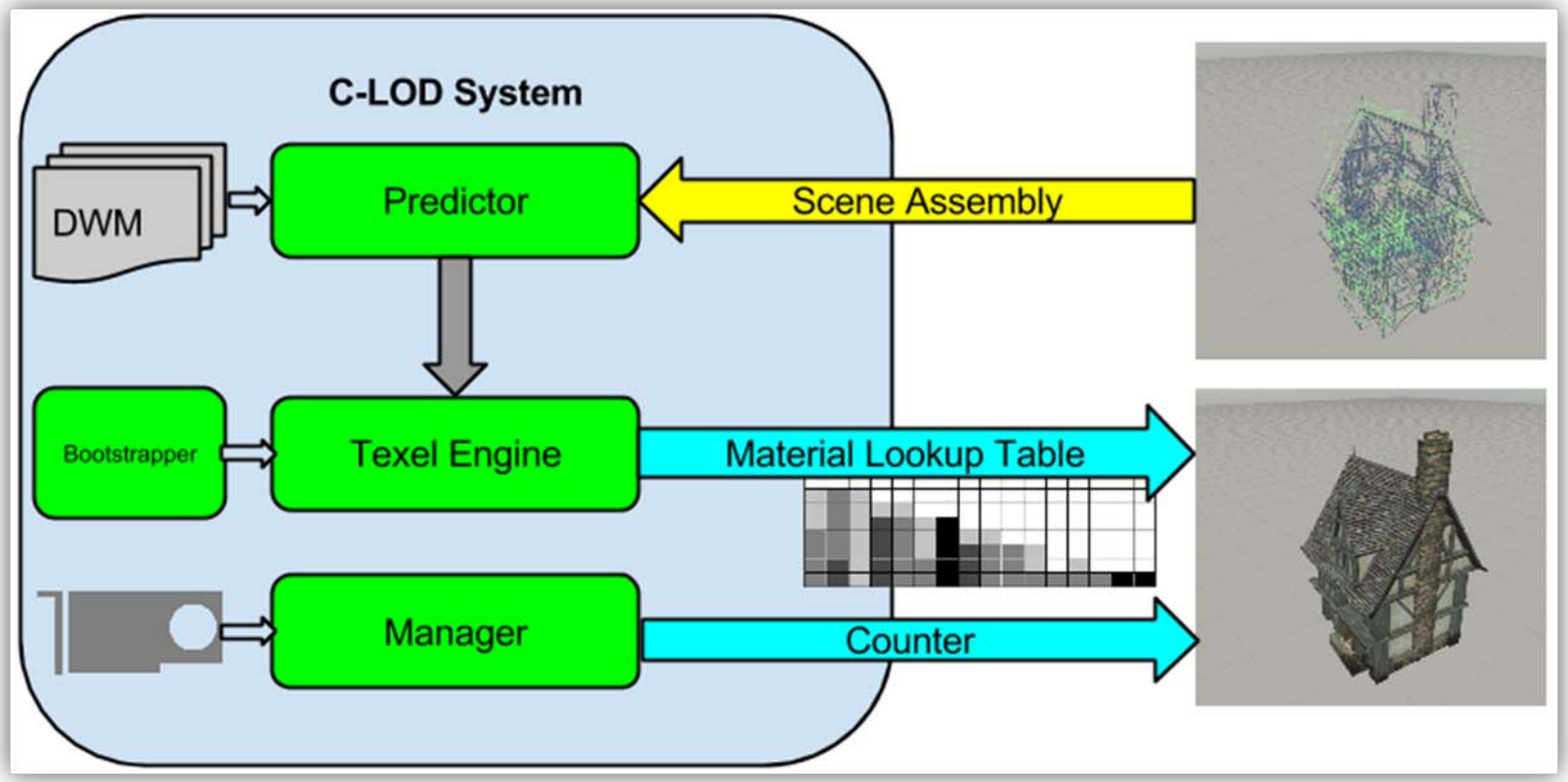
Introduction

- Reactive fixed frame rate scheduler based on attention
- C-LOD lowers the rendering quality of non-attended objects
- The highest quality is maintained for all attended objects

Three complex effects usually omitted in mobile devices were selected:

1. Subsurface scattering
2. Refraction
3. Bump mapping

C-LOD for Unity 3D™
C-LOD Components (1)



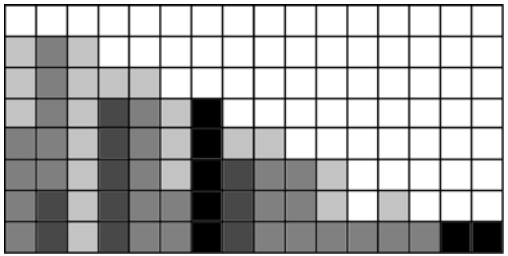
C-LOD for Unity 3D™
C-LOD Components (2)

The Predictor

- GPU implementation of the model – pixel shader
- Examines object placement: view vector, object IDs
- Identified units' local maxima, generates repetitive firings **– exploit hardware linear interpolation**

The Texel Engine

- Material look-up table in a 2D texture
- Texels contain LOD parameters e.g. iteration counter



C-LOD for Unity 3D™
C-LOD Components (3)

The Bootstrapper

- System profiling at bootstrapping (GPU-Memory interaction)
- Scalable LOD simplification

The Manager

- Finite State Machine
- Continuous frame rate evaluation
- LOD selection based on frame rate trends

Evaluation of C-LOD
Model Accuracy

Evaluation of C-LOD manager via eye-tracking evaluation

1. High Quality (HQ) condition
2. C-LOD managed condition

- 7 objects in quest
- 22 people participated, 11 in each condition
- 88404 object fixations were recorded

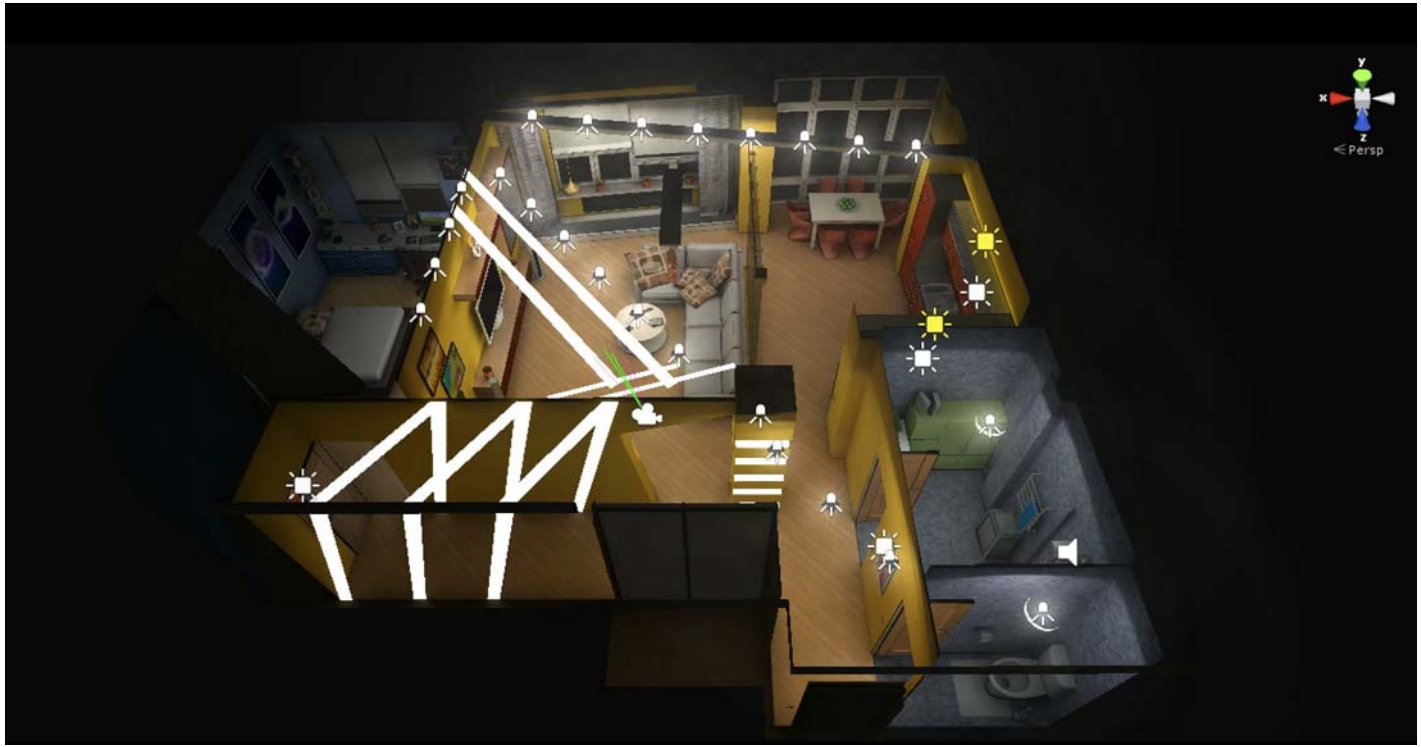
Evaluation of C-LOD
Results

Percentage of frames that the attended object was predicted correctly for each condition

Est.	Object gazed	HQ	C-LOD	Total
R	random object	$< 5\%$	$< 5\%$	$< 5\%$
$E1$	1st prediction	40%	42.3%	41.1%
$E2$	1st or 2nd	69.9%	74.8%	72.3%
$E3$	1st or 2nd or 3d	86.9%	92.7%	89.7%

The addition of C-LOD changes did not alter gaze performance

Evaluation of C-LOD
Validation Tool



VIDEO

Magenta beams: Gaze - Green beams: Correct predictions

Evaluation of C-LOD
Model Efficiency

Acquiring GPU performance data on a mobile device to estimate C-LOD gain

- 2 conditions: High Quality - C-LOD
- Sampled frame rate

Evaluation of C-LOD
Model Efficiency

Results of GPU performance evaluation

- t-test revealed a significant difference between conditions
- C-LOD: consistently more stable frame rate than HQ
- 4ms/frame cost of C-LOD amortized