TECHNICAL UNIVERSITY OF CRETE

Context-aware Gaze Prediction applied to Game Level Design, Level-of-Detail and Stereo Manipulation

by

George Alex Koulieris

A thesis submitted in partial fulfillment for the degree of Doctor of Philosophy

in the

Computer Science Division

School of Electronic & Computer Engineering

September 2015

Declaration of Authorship

I, George Alex Koulieris, declare that this thesis titled, “Context-aware Gaze Prediction applied to Game Level Design, Level-of-Detail and Stereo Manipulation” and the work presented in it are my own. I confirm that:

- This work was done wholly while in candidature for a research degree at this University.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.

Jury

3-member Committee

Assoc. Prof. Katerina Mania	*Technical University of Crete, Greece*
Prof. Stavros Christodoulakis	*Technical University of Crete, Greece*
Prof. Douglas Cunningham	*Technical University of Cottbus, Germany*

Examiners

Prof. Kostas Balas	*Technical University of Crete, Greece*
Prof. Michael Zervakis	*Technical University of Crete, Greece*
Assoc. Prof. Michail G. Lagoudakis	*Technical University of Crete, Greece*
Assoc. Prof. Ann McNamara	*Texas A&M University, USA*

European Union
European Social Fund
OPERATIONAL PROGRAMME
EDUCATION AND LIFELONG LEARNING
investing in knowledge society
MINISTRY OF EDUCATION & RELIGIOUS AFFAIRS, CULTURE & SPORTS
MANAGING AUTHORITY
Co- financed by Greece and the European Union
NSRF
2007-2013
programme for development
EUROPEAN SOCIAL FUND

"It is impossible to enjoy idling unless there is plenty of work to do."

Jerome K. Jerome, Three Men In a Boat

TECHNICAL UNIVERSITY OF CRETE

Extended Abstract

Computer Science Division
School of Electronic & Computer Engineering

Doctor of Philosophy

by George Alex Koulieris

The prediction of visual attention can significantly improve many aspects of computer graphics and games. For example, image synthesis can be accelerated by reducing complex computations on non-attended scene regions and Level-of-Detail rendering improved. Current gaze prediction models often fail to accurately predict user fixations mostly due to the fact that they include limited or even no information about the context of the scene; they commonly rely on low level image features such as luminance, contrast and motion or pre-determined task restrictions on attention to predict user gaze. These features do not drive user attention reliably when interacting with an interactive synthetic scene, e.g. in a video game. In such cases the user is in control of the view-port often consciously ignoring low level salient features in order to navigate the scene or perform a task. This dissertation contributes two novel predictive scene context-based models of attention that yield more accurate attention predictions than those derived from state-of-the-art low level image saliency methods.

Both models presented take into account critical high level scene context features such as object topology and task-related object function that influence fixation guidance when gazing at interactive content. Developing the models was a challenging problem, since qualitative features such as object topology, inter-object relationships and tasks had to be quantified and formally considered in order to generate probabilities of object attendance based on subjective features. By acknowledging high level contextual features we were able to develop gaze predictors that accurately predict gaze in cases where low level image-based predictors fail.

The first model is an automated high level saliency predictor that incorporates six hypotheses/factors from perception and cognitive science which can be adapted to different tasks. The first hypothesis states that a scene is comprised of objects expected to be found in a specific context as well objects out of context which are

salient (scene schemata). The second claims that viewer's attention is captured by isolated objects (singletons). We employ an object-intrinsic factor accounting for canonical form of objects, an object-context factor for contextual isolation of objects, a feature uniqueness term that accounts for the number of salient features in an image and a temporal context that generates recurring fixations for objects inconsistent with the context.

We extended Eckstein's Differential Weighting Model by incorporating these six hypotheses. We then conducted a formal eye-tracking experiment which confirmed that object saliency guides attention to specific objects in a game scene and determined appropriate parameters for this model. We present a GPU based system architecture that estimates the probabilities of objects to be attended in real-time. We embedded this tool in a game level editor to automatically adjust game level difficulty based on object saliency, offering a novel way to facilitate game design. We perform a study confirming that game level completion time depends on object topology as predicted by our system. We then develop an attention-based Level-of-Detail manager that downgrades the quality of areas that are expected to go unnoticed by an observer to economize on computational resources. Our system (C-LOD) maintains a constant frame rate on mobile devices by dynamically re-adjusting material quality on secondary visual features (e.g. subsurface scattering) of non-attended objects. In a proof of concept study we establish that by incorporating C-LOD, complex effects such as parallax occlusion mapping usually omitted in mobile devices can now be employed, without overloading GPU capability and, at the same time, conserving battery power.

We then develop our second model, addressing the challenge of developing a gaze predictor in the demanding context of real-time, heavily task-oriented applications such as games. Our key observation is that player actions are highly correlated with the present state of a game, encoded by game variables. Based on this, we train a classifier to learn these correlations using an eye-tracker which provides the ground-truth object being looked at. The classifier is used at runtime to predict object category – and thus gaze – during game play, based on the current state of game variables. We evaluate the quality of our gaze predictor numerically and experimentally, showing that it predicts gaze more accurately than previous image-based approaches. Given that comfortable, high-quality 3D stereo viewing is becoming a requirement for interactive applications today, we use this prediction to propose a dynamic local disparity manipulation method, which provides rich and comfortable depth in sharp contrast to previous global disparity methods that suffer from extreme depth compression (cardboarding). A subjective rating study demonstrates that our localized disparity manipulation is preferred over previous methods.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Katerina Mania for her continuous motivation, patience, friendly encouragement and for pointing me to the exciting direction of working with gaze-based graphics. She shaped the way I approach research and provided me with direction and support becoming more of a mentor and life coach than a supervisor. Her continuous efforts to teach me how to properly write papers despite my inherent inclination for chattering will be eternally remembered. Hopefully :) the verbosity of my future papers will remind her of me from time to time. I will remember her as my friend with a great heart.

Besides my advisor, I would like to thank my two other collaborators. George Drettakis who truly made a difference in my life by trusting me early on in my career working together since my undergraduate thesis. I would not have considered a graduate career in Computer Graphics if it wasn't for him. His thorough involvement, energy and interest in this work paired with critical project ideas have been a source of motivation for me. I doubt that I will ever be able to convey my appreciation fully, but I owe him my eternal gratitude.

Douglas Cunningham for his critical advice in designing the psychophysics experiments that mattered. This thesis would certainly have been poorly presented and the models' performance would have never grown without his critical contribution in experimental data analysis. His insightful comments but mainly his decisive ideas on how to advance on this research are fully appreciated.

This research has been co-financed by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: Heracleitus II: Investing in knowledge society through the European Social Fund. I recognize that this research would not have been possible without this financial assistance for which I express my gratitude.

My sincere thanks also go to the HCSquared project that offered a research travel budget to visit the INRIA REVES group during a very critical point of this thesis research. That visit shaped the outcome of this work. Appreciation also goes out to

INRIA for providing me the opportunity to access their research facilities and software. During my stay at INRIA, discussions with Emmanuelle Chapoulie, Christian Richardt, Emmanuel Iarussi, Gaurav Chaurasia and Rachid Guerchouche inspired me a lot on how to advance this thesis. I must especially acknowledge Jorge Lopez Moreno for suggesting a cool idea in LOD management.

I would like to acknowledge people in conferences having an impact on this work with their insightful comments and discussions. I acknowledge the reviewers and members of the jury, for their time spent on my thesis. My studies also required the patience of many people who took part in experiments and I am thankful for their time.

I owe many thanks to my co-authors Christos Paraskeva and Nikos Sidorakis for all the funny moments working together in the lab. I thank all my fellow lab mates for the stimulating discussions, for the sleepless nights we were working together before deadlines, philosophical debates, venting of frustration during my graduate program and for all the fun we have had in the last 3.5 years. In particular, Valia, Vaggelio, Ioanna, Andreas, George, Kostas, Dimitris, Marios, Manolis and Giannis. I also thank our assistant Vicky Grigoraki who helped me a lot during this stay.

Completing my PhD degree was probably the most challenging thing that I have ever achieved. Without the help and support of my friends I would probably have never finished this project sane. In particular I would like to thank my friends in Chania: Giannis, Kostas, Giorgos^2, Markos, Maria, Stavros, Vyronas and Tzortzina. Also my friends outside of Crete, Elli, Stamos, Stamatia, Panagiotis, Apostolis, Anestis, Thanassis and Giota. I also thank the guys in the biking group "Podilatreis" for making my weekends more active, in addition to all the fast food restaurants for reversing the positive effects of the biking group.

I would like to thank my family for supporting me spiritually throughout working and writing this thesis and in my life in general.

Finally, I would like to dedicate this thesis to the memory of my late father who passed away during the first days of this project. I miss him everyday but I am glad to have lived with him for 25 years. His example, kindness and love will always be a cornerstone in my life.

Contents

List of Figures

List of Tables

Abbreviations

3D	**3 D**imensional
3DOF	**3 D**egrees **O**f **F**reedom
3DTV	**3 D**imensional **T**ele **V**ision
AI	**A**rtificial **I**ntelligence
ADB	**A**ndroid **D**ebug **B**ridge
ALU	**A**rithmetic **L**ogic **U**nit
C-LOD	**C**ontextual **L**evel **O**f **D**etail
CB	**C**hange **B**lindness
DF	**D**ecision **F**orest
DLL	**D**ynamic **L**ink **L**ibrary
DLP	**D**igital **L**ight **P**rocessing
DWM	**D**ifferential **W**eighting **M**odel
FIT	**F**eature **I**ntegration **T**heory
FOV	**F**ield **O**f **V**iew
FPS	**F**irst **P**erson **S**hooter
FSM	**F**inite **S**tate **M**achine
GPU	**G**raphics **P**rocessing **U**nit
GTOM	**G**aze **T**o **O**bject **M**apping
GUI	**G**raphical **U**ser **I**nterface
HLSM	**H**igh **L**evel **S**aliency **M**odeler
HMD	**H**ead **M**ounted **D**isplay
HMR	**H**ierarchical **M**ultiple **R**egression
IB	**I**nattentional **B**lindness
IPD	**I**nter **P**upillary **D**istance

LCD	**L**iquid **C**rystal **D**isplay
LCS	**L**iquid **C**rystal **S**hutters
LOD	**L**evel **O**f **D**etail
LR	**L**ikelihood **R**atio
MLR	**M**ultiple **L**inear **R**egression
MRT	**M**ultiple **R**ender **T**arget
NPC	**N**on **P**layable **C**haracter
OOB	**O**ut **Of** **B**ag
ROI	**R**egion **Of** **I**nterest
SDK	**S**oftware **D**evelopment **K**it
SVM	**S**upport **V**ector **M**achine
VE	**V**irtual **E**nvironment
VR	**V**irtual **R**eality

Symbols

C	near clipping distance	m
D	input data set	observations
D_{eye}	eye separation	m
d'_j	HLSM Gaussian distribution mean	1.0
d'_k	HLSM feature strength	1.0
d_{left}	left camera distance	m
d_{right}	right camera distance	m
f	HLSM frame number	1.0
F	HLSM previous frame count	1.0
λ_j	HLSM neuron response	1.0
M	training features	feature vectors
$mtry$	number of trees when branching	1.0
N	training samples count	1.0
$ntree$	number of trees grown	1.0
p	image disparity	pixels
P	probability of attendance	1.0
R	random baseline predictor	1.0
r_{aspect}	aspect ratio	rads
s	HLSM signal response	1.0
T	training data set	observations
w	vertex distance	m
w_j	HLSM factor weight	1.0
σ	HLSM Gaussian distribution standard deviation	1.0

Dedicated to the memory of my late father

Relevant Publications

Journal Papers

1. **Koulieris, G.A.**, Drettakis, G., Cunningham, D., Mania, K. (2015, submitted to IEEE VR special issue Transactions on Visualization and Computer Graphics). Gaze Prediction using Machine Learning for Dynamic Stereo Manipulation.

2. **Koulieris, G.A.**, Drettakis, G., Cunningham, D., Mania, K. (2014, September). An Automated High Level Saliency Predictor for Smart Game Balancing. ACM Transactions on Applied Perception (TAP) 11, 4, Article 17, 21 pages.

3. **Koulieris, G.A.**, Drettakis, G., Cunningham, D., Mania, K. (2014, July). C-LOD: Context-aware Material Level-of-Detail applied to Mobile Graphics. In Computer Graphics Forum Vol. 33, No. 4, p. 41-49.

Peer-Reviewed Conference Papers & Posters

4. Sidorakis, N., **Koulieris, G.A.**, Mania, K. (2015, March). Binocular Eye-Tracking for the Control of a 3D Immersive Multimedia User Interface. IEEE VR conference, 1st Workshop on Everyday Virtual Reality (WEVR), p. 15-18.

5. **Koulieris, G.A.**, Drettakis, G., Cunningham, D., Mania, K. (2014, July). High level saliency prediction for smart game balancing. In ACM SIGGRAPH 2014 Talks, p. 73.

6. McNamara, A., Mania, K., **Koulieris, G.A.**, Itti, L. (2014, July). Attention-aware rendering, mobile graphics and games. In ACM SIGGRAPH 2014 Courses, p. 85-112.

7. **Koulieris, G.A.**, Drettakis, G., Cunningham, D., Sidorakis, N., Mania, K. (2014, July). Context-aware material selective rendering for mobile graphics. In ACM SIGGRAPH 2014 Posters, p. 92.
   [**won 3rd place at the ACM Graduate Student Research Competition**]

8. Paraskeva, C., **Koulieris, G.A.**, Coxon, M., Mania, K. (2012, December). Gender differences in spatial awareness in immersive virtual environments: a preliminary investigation. In Proceedings of the 11th ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry, p. 95-98.

Chapter 1

Introduction

The prediction of visual attention can significantly improve many aspects of computer graphics and games. For example, image synthesis can be accelerated by reducing complex computations on non-attended scene regions [Cater et al., 2003] and Level-of-Detail (LOD) rendering improved [Lee et al., 2009]. Current gaze prediction models often fail to accurately predict user fixations. As discussed in Chapter 2, this is mostly due to the fact that they only take limited or no information about the context of the scene; they commonly rely on low level image features such as luminance, contrast and motion or pre-determined task restrictions on attention to predict user gaze. These features do not drive user attention reliably when interacting with an interactive synthetic scene, e.g in a video game. In such cases the user is *in control* of the view-port often consciously ignoring low level salient features in order to navigate the scene or perform a task.

This dissertation contributes two novel predictive scene context-based models of attention that yield more accurate attention predictions than those derived from state-of-the-art low level image saliency methods. Both models presented take into account critical high level scene context features such as object topology and task-related object function that influence fixation guidance when gazing at interactive content. Developing the models was a challenging problem, since qualitative features such as object topology, inter-object relationships and tasks had to be quantified and formally considered in order to generate probabilities of object attendance based on subjective features. By acknowledging high level contextual features we were able to develop gaze predictors that accurately predict gaze in cases where low level image-based predictors fail. We improve existing gaze-aware applications such as LOD management of complex effects and encompass gaze predictors in the novel areas of game balancing and stereo disparity manipulation.

This chapter provides detailed motivation in addition to a description of our novel contributions. We also describe the structure of this thesis.

1.1 Context

Predicting user gaze in computer generated imagery yields several exciting applications in game design, rendering and stereo manipulation.

Game Design. Many game genres levels rely on a search or target detection task to solve riddles, find game objects and advance game-play. However, designing game levels by placing objects in their respective locations is a tedious, manual operation. To make things worse, taking into account object placement in relation to game difficulty further complicates game level design. If gaze can be automatically and accurately predicted, several game level design tasks can be simplified. For example adjusting the difficulty of a game may be facilitated by automatically relocating objects estimated to attract attention [Feil and Scattergood, 2005].

Rendering. LOD algorithms render with higher visual fidelity those regions of a synthetic image that are expected to receive attention, allowing more efficient distribution of the limited resources of a graphics subsystem. LOD managers have been empowered with perceptual principles in the past to optimize the distribution of computational time and maximize the perceived quality of a rendered scene

[Luebke, 2003]. For example object eccentricity in relation to the centre of the display has been employed to degrade rendering quality without the lowered visual fidelity being perceived [Luebke, 2003]. By employing a LOD manager, computation time is minimized and the quality of an effect is downgraded away from the area of predicted focus, based on evidence determining that a user is not attending that scene area.

The interest in efficient LOD management has been recently renewed due to the explosive growth of the mobile market, which is extremely diverse in terms of computing power. Hardware restrictions of mobile devices prohibit the use of complex effects, such as subsurface scattering, that demand multiple texture fetches or intense Arithmetic Logic Unit (ALU) operations [Çapin et al., 2008]. An application's artistic feel is thus sacrificed in portable devices as content is displayed at degraded LOD or quality. A focused distribution of available resources only to *attended* areas is thus required in order to make it possible for complex rendering effects to be visualized on mobile platforms, achieving higher and more stable frame rates.

Stereo manipulation. Stereo 3D is expected to become ubiquitous; currently a multitude of companies receive huge revenue from 3D hardware, 3D software or 3D content production. Stereoscopic 3D movies are grossing 30% of the box office, often multiple times more than their 2D counterparts [Mendiburu, 2012]. Besides 3D for entertainment, 3D displays have become an invaluable tool for image-guided diagnosis, medical tissue visualization and surgical procedures. Remote guidance of robots carrying stereoscopic cameras for hazardous tasks has reduced task execution times and error rates.

Comfortable, high-quality stereo 3D is thus an important and timely requirement for real-time applications, especially given the recent popularity of commodity Head Mounted Displays (HMDs), such as the Oculus Rift that has the potential to transform Virtual Reality (VR) to a commodity for everyday use [Sidorakis et al., 2015]. It is well known that 3D stereo viewing often results in discomfort and eye fatigue, most commonly because of excessive disparities and the vergence-accommodation conflict [Hoffman et al., 2008]. To counter these problems, recent methods [Lang et al., 2010, Oskam et al., 2011] manipulate stereo content for more comfortable viewing, a process called *stereo grading*. However, they only apply a global compression of scene depths, often sacrificing depth, resulting in "flat" imagery (cardboarding). In contrast, focusing depth changes in the region of the user's attention has been shown to be particularly effective [Bernhard et al., 2014], corresponding to the fact that human stereo is based on where we look. Locally adapting stereo in predicted regions of attention is thus important for real-time applications, such as games.

In this work we enable attention-driven game design, gaze-aware LOD and stereo disparity management by addressing a significant challenge: effective real-time gaze prediction.

1.2 Problem Statement

Existing visual attention models, such as Feature Integration Theory (FIT), are predominantly driven by low level image features, such as contrast, luminance and motion that attract gaze [Treisman and Gelade, 1980]. FIT is a commonly used model of attention in computer graphics (Figure 1.1) [Itti and Koch, 2001, Longhurst et al., 2006].

However, it often fails to predict saccadic targets [Borji and Itti, 2013] because high-level properties, such as scene semantics and the performed task, strongly affect the planning and execution of user eye fixations [Borji and Itti, 2013, Einhäuser et al., 2008, Henderson and Hollingworth, 1999].

Previous work has shown that in interactive applications the *task* strongly influences gaze and more generally attention (e.g., [El-Nasr and Yan, 2006, Sundstedt

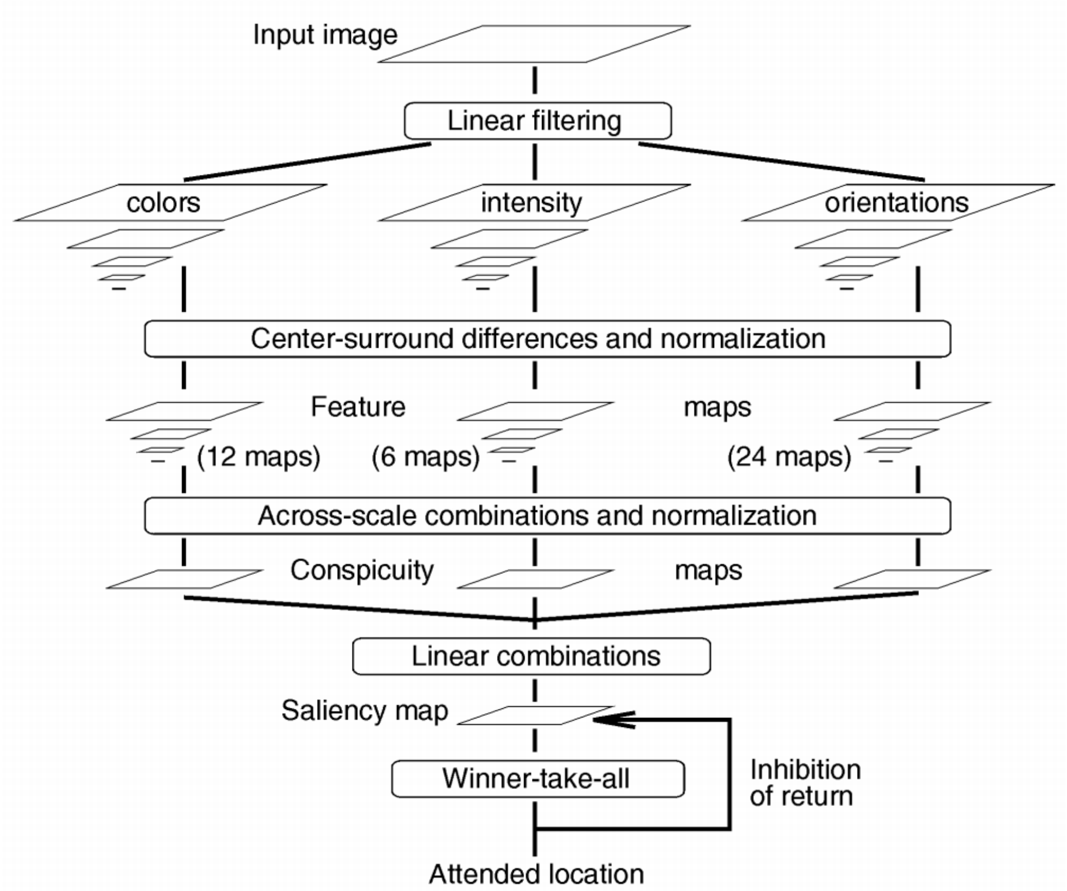


Figure 1.1: General architecture of a FIT-based attention model [Itti and Koch, 2001].

et al., 2008]). However, in the related literature when modelling goal-oriented attention the task has always been predetermined [Cater et al., 2003, Sundstedt et al., 2004, 2005]. In other approaches, the scene graph representation of scenes has been used to map gaze positions to objects and object semantics [Sundstedt et al., 2013]. Importance maps generated from off-line eye tracking data were used as a heuristic to predict user attention according to object properties present at runtime [Bernhard et al., 2010]. However, both approaches require manually pre-defined task-related objects and task objectives.

Furthermore, the contextual validity or appropriateness of an object's location affects visual search; when looking for a chimney, we usually direct our gaze first to the rooftops. Research in real environments [Einhäuser et al., 2008, Henderson and Hollingworth, 1999, Rensink, 2000] and interactive Virtual Environments

(VEs) [Mania et al., 2005, Mourkoussis et al., 2010, Zotos et al., 2009] has confirmed that attention is influenced by the semantic context of objects in the form of scene schemas. In other words, attention models based on low-level features fail to predict saccadic targets [Borji and Itti, 2013], partly because they do not consider critical high-level factors such as object topology when predicting attention [Einhäuser et al., 2008]. However, quantifying subjective, qualitative inter-object relationships is a real challenge.

Particularly in computer games where a task is constantly being executed, tracking luminance changes and moving objects is not sufficient to predict gaze. In addition, since players have full control of the view-port, it is hard to accurately guess where a player is looking at any given instant without knowing their current goal. Existing solutions for gaze predictors targeted to games require *manual categorization* of tasks and objects [Sundstedt et al., 2008], which is time consuming and impractical in our context.

1.3 Contributions

Our goal was to develop an automated context-aware saliency predictor which can be adapted to different tasks. We developed two novel gaze predictors. Each algorithm specializes to a different family of applications depending on the accessible scene context information and the availability or not of an eye tracker during the attention model formation. We demonstrate the success of the models in three gaze-aware applications (LOD management, Game balancing and Stereo manipulation).

We evaluate the quality of our gaze predictors by performing several confirmatory eye-tracking studies. These studies showed that our predictors are more accurate when compared to previous alternatives in the context of task-driven activities such as game-play. We used a modern game engine for our experiments and their successful validation. This choice underlines the relevance of our results for realistic use cases.

We present:

A physically plausible model of high level attention (Chapters 4, 5). To develop the first model we encode six hypotheses/factors from perception and cognitive science into mathematical equations that precisely describe *a.* semantic inter-object relationships (e.g. contextual validity), *b.* intra-object positional properties (e.g, object rotation) and *c.* object topology in terms of inter-object distances and placement (e.g. object isolation) allowing for the development of a computational model that can automatically estimate fixation guidance based on these hypotheses. Our gaze predictor incorporates these hypotheses/factors into the physiologically plausible Differential-Weighting Model (DWM) [Eckstein, 1998, Eckstein et al., 2006, 2002] that employs Bayesian priors to estimate the probability of a feature to be attended.

The hypotheses/factors are:

(i) The *scene schema hypothesis* stating that a scene is comprised of objects we expect to find in a specific context and salient objects that are not expected in a scene (see Figure 1.2) [Bartlett, 1932, Henderson et al., 1999, Hwang et al., 2011].
(ii) The *singleton hypothesis* stating that the viewer's attention is ordinarily captured by stimuli that are locally unique in a basic visual dimension such as orientation or depth i.e. isolated [Theeuwes and Godijn, 2002]. In our work, the singleton state is a context dependent measure not purely image-driven: Figure 1.3 shows that the spatially isolated vase attracts attention, though not salient in terms of color.
(iii) An object-intrinsic hypothesis accounting for the fact that an object pops out if it is rotated in a way that violates its expected posture. The expected posture is known as *canonical form* or canonical orientation [Becker et al., 2007].
(iv) We account for an *object-context hypothesis* for contextual isolation of objects,

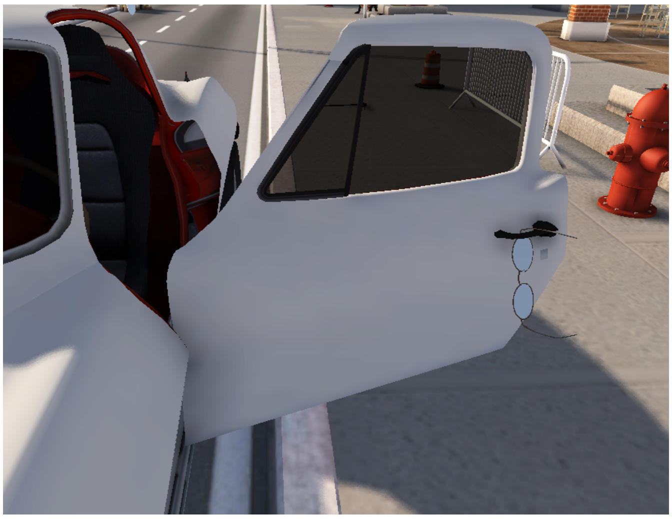


FIGURE 1.2: The spectacles attract attention as they are *inconsistent* with the car door context.

when objects belong to a group of similar objects but are dissimilar from those in the set.

(v) We employ a *feature uniqueness term* that accounts for the number of salient features in an image.

(vi) Finally we include a *temporal context* factor that generates recurring fixations for objects inconsistent with the context or in a non-canonical form as indicated in cognitive psychology literature.

Using this new model, we estimate the posterior probability that a viewer will fixate on an object based on the aforementioned high-level contextual features, independent of the viewer's task [Eckstein et al., 2006]. To find model parameters we perform several perceptual experiments, which also verify that high-level saliency guides attention. The experimental design controls for attentional effects from low level features such as luminance or contrast, allowing us to examine the unique contribution of context.

A game balancing paradigm based on attention (Chapter 4). We develop a tool based on the high level saliency model to automatically predict gaze in real-time. We then validate the tool's efficacy in adjusting game difficulty by altering

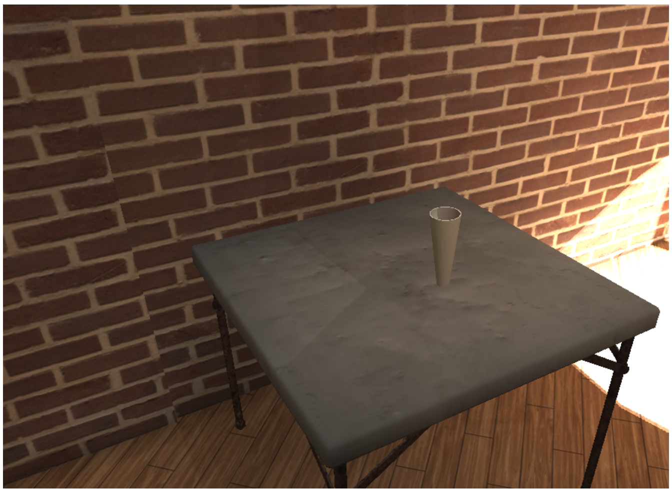


FIGURE 1.3: The spatially isolated vase attracts attention as it is a *singleton* object.

object placement based on saliency in a game-level editor. This facilitates game level balancing, offering a novel way to ease game design.

A Level-of-Detail method based on attention (Chapter 5). We incorporate the high level saliency predictor into a perceptually optimized renderer for mobile platforms. This saves computational time by automatically and seamlessly removing perceptually non-important details. Integration of a contextual attention model in a LOD manager enables the usage of – otherwise omitted – complex effects such as subsurface scattering, complex refraction and displacement mapping in low-power devices by applying them sparingly only in regions that are expected to be attended.

Our proof-of-concept implementation selects an appropriate LOD in real-time for subsurface scattering, complex refraction and bump mapping algorithms. We demonstrate the accuracy of our implementation by comparing its performance to actual eye-tracking data. We also acquire mobile Graphics Processing Unit (GPU) performance statistics in terms of frame time stability to ensure model effectiveness and quantify battery performance gain when limiting GPU utilization.

A machine learning based model of high level attention (Chapter 6). We propose a second model of high level gaze guidance that does not require extensive object-context information. This model predicts attention by learning user gaze behavior automatically via employing machine learning. The model is particularly effective in computer games, where our key observation is that a player's current goal is highly correlated with the present state of the game, as *encoded by game variables*. For instance, in a shooting game, the player's current goal will be related to the health and ammo of his/her character and of the enemies' movements. Based on this insight, we train a classifier to learn the correlation between game variables and object class the user looks at, using eye-tracking ground-truth data recorded during a training session. The resulting classifier can predict object category – and thus user gaze – for any subsequent game-play. Our approach is automatic since it uses machine learning to build the classifier, avoiding the need for manual object tagging and/or explicit definition of objects important to a task.

This model inherently accounts for high level features and task without any previous knowledge of high level hypotheses; however, it requires an eye tracker to learn gaze patterns which is not necessary by the first model.

Dynamic stereo disparity management based on attention (Chapter 6). We develop a stereo grading algorithm based on the second gaze predictor for dynamic disparity management in video games. Previous disparity mapping operators based on image-based saliency estimates are off-line [Lang et al., 2010] or when they are interactive, apply a global disparity transformation over the entire scene [Oskam et al., 2011]. Such manipulations often sacrifice depth, resulting in "flat" imagery (cardboarding). Using our machine learning gaze predictor we introduce dynamic and localized disparity manipulation, which provides high-quality depth information in a scene without sacrificing comfort. We validated that our stereo grading method is preferred over previous methods in subjective ratings.

1.4 Thesis Structure

The rest of the thesis is organized as follows:

- Chapter 2 discusses previous work in computer graphics, attention and perception that is relevant to the techniques described in this thesis.
- Chapter 3 gives a technical overview of the deployment test-bed and data acquisition framework for our saliency models.
- Chapter 4 describes our first approach to develop a physiologically plausible model of attention and its application in game balancing.
- Chapter 5 describes extending the saliency model of Chapter 4 with novel context-based factors and the application of fixation prediction in LOD.
- Chapter 6 discusses the second visual attention prediction model which is based on eye tracking data and machine learning over game state variables and its application to stereo disparity management.
- Chapter 7 concludes the thesis discussing current limitations and potential future applications.

Chapter 2

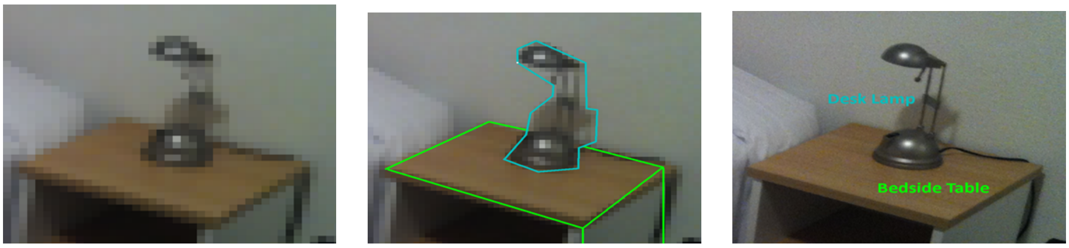
Previous Work

We present previous work on visual attention prediction in general and prediction of attention as employed in computer graphics and computer games. We present related work on game balancing and LOD. We investigate the internals of the Differential Weighting Model, the physiologically plausible model of attention employed in our predictor. Finally we investigate eye tracking, machine learning for games and stereo disparity manipulation algorithms. [1]

2.1 Visual Attention

Visual perception can be thought of as the active extraction and manipulation of environmental information. The visual perception pipeline starts with low-level processes which extract simple image regularities such as edges or color [Marr, 1982]. Subsequently, mid-level processes combine these properties to form higher-level features such as the shape of an object [Shipley and Kellman, 2001]. Finally, high-level processes map these mid-level features to meaning and semantics (Figure

[1] The literature review included in this Chapter has been presented by the author of this thesis as part of an ACM SIGGRAPH Course on attention-aware rendering, mobile graphics and games (co-presented by Laurent Itti, Katerina Mania and Ann McNamara) [McNamara et al., 2014].



Low-level processes Mid-level processes High-level processes

FIGURE 2.1: Left to right: Low-level, Mid-level and High-level vision processes.

2.1)[Palmer, 1999]. A recent review of these theories can be found in [Borji and Itti, 2013].

To efficiently concentrate the limited brain resources of the mid- and high-level processes on those few low-level features that are likely to be important, the human brain is equipped with a selection mechanism known as focal attention. Some low-level features such as edges can automatically attract focal attention in an almost reflex-like fashion [Koch and Ullman, 1987]. Likewise, mid- and high-level features as well as goal-oriented properties can direct focal attention [Henderson et al., 1999, Yarbus et al., 1967]. For example, the contextual validity or appropriateness of an object's location will affect visual search; when looking for a chimney, usually we direct our gaze first to the rooftops. However, the fundamental question of how the visual system combines the influence of low-, mid-, and high-level components is a challenging research issue and remains largely unanswered due to the complexity of the human brain [Theeuwes, 2010].

The most common form of focal attention model is the two-stage model, such as FIT [Treisman and Gelade, 1980]. In two-stage models, a privileged set of low-level features are initially extracted everywhere in an image in parallel. The focal attention mechanism then selects a few locations in the image based on these features for further processing. In the second stage, the low level features at the selected locations are integrated and subjected to further processing in a slow, serial (i.e., one region at a time) fashion. A widely used saliency model inspired by FIT [Itti and Koch, 2001] employs low-level features such as contrast, luminance, and motion to determine which areas are likely to attract attention. Although FIT plausibly emulates many aspects of focal attention, it has been shown that: (i) complex stimuli such as surfaces are processed simultaneously and not in a serial fashion [Nakayama et al., 1986], (ii) visual attention is directed to objects in a scene rather than their low level visual attributes [O'Craven et al., 1999] and, (iii) observers may achieve multiple simultaneous foci of attention in the visual field, not supported by FIT [Awh and Pashler, 2000]. In other words, attention models based on low-level features often fail to predict saccadic targets [Borji and Itti, 2013], in part because they do not take into account high level factors such as scene context, task, or object topology [Einhäuser et al., 2008, Henderson and Hollingworth, 1999, Rensink, 2000].

2.1.1 Task-related Attention

What is a task? A task is formed as a sequence of clearly defined *actions* of an *actor* over objects; i.e. objects are conceptually dependent to actors via actions [Schank and Abelson, 2013]. The expected sequence of the actions when describing a task is represented by the script concept [Schank and Abelson, 2013, Tatler et al., 2011]. From the definition of the task, it becomes obvious that any series of actions is described both in the spatial and the temporal domain.

As indicated by previous work, it is apparent that attention deployment heavily depends on task. It is the task that defines which factors affect attention the most, also supported by psychophysical experiments indicating that the human vision is highly purposive and task specific [Triesch et al., 2003]. The task being conducted is important for fixation guidance since when allocating gaze, information gathered from the fixation point hold significant behavioral relevance; information satisfies the attempt to maximize reward by executing the task and reduces uncertainty about the environment [Tatler et al., 2011]. Since eye movements are used to gather information in order to accomplish tasks, a visual search produces consistent patterns of eye movements across observers when executing the same task [Peters and Itti, 2008] indicating a clear *spatial coupling* between the current fixation and a behavioral goal.

Fixation guidance is also extended to the *temporal domain*. Saccades are often proactive; they are directed to a location in a scene in advance of an expected event [Ehinger et al., 2009]. However, a complete understanding of eye movements during tasks and thus the way that low level and high level features coexist and guide gaze requires a clear understanding of how tasks are represented in the human mind [Tatler et al., 2011].

Regarding context-based attention in situations when a specific task has to be conducted, the estimation of the relative contribution of low and high level factors on fixation guidance instantly becomes a real challenge. For example, when free-viewing a scene, a low level motion signal attracts attention, since motion is one of the strongest cues affecting gaze deployment in dynamic scenes [Peters and Itti, 2008]. In such a case a low level attention predictor would successfully predict attention deployment. However, low level saliency itself, cannot predict fixations when there is an overt or covert task to be conducted in a real or virtual environment. In a VE experiment where participants needed to avoid obstacles while colliding with others, image-feature saliency could not yield accurate predictions since observers had to actively ignore low level salient features [Rothkopf et al., 2007]. In cases where a task is conducted and strong low level cues are absent, attention is mainly guided on the basis of high-level interest; fixations necessary to e.g. make a cup of tea, are consciously generated [Tatler et al., 2011].

A previous successful attempt to combine both low and task-based models of attention to increase prediction accuracy exists, but only for static photographs and a single pre-determined task [Ehinger et al., 2009]. Participants searched for faces in a set of 900 photographs, where a joint image saliency/task-based model attempted to predict the areas that participants were going to look to by averaging the relative contribution of both image saliency and task using predetermined, task-specific weights ("find the faces"). However, when dealing with dynamic scenes and multiple tasks, it still remains a great challenge to encode task information in order to select appropriate weights for each attentional factor depending on the task being conducted each time.

Employing Bayesian Priors to Predict Gaze More recently, a number of single stage models have been proposed, which are very effective at describing visual attention, however they have not been used to predict high-level saliency or gaze patterns in interactive VEs. For example, Eckstein has proposed DWM; a single-stage model of attention [Eckstein, 1998, Eckstein et al., 2006, 2002], which incorporates both low-level features as well as prior knowledge about scene context. The DWM models attentional processing using physiological noise in brain neurons and Gaussian combination rules. Contextual information in the DWM is embodied in the Bayesian priors provided to the model beforehand. For example, when searching for a chimney in a picture that contains a house, the visual elements depicting the roof of the house are given a higher prior probability than other scene elements.

2.1.2 High Level Saliency Factors

Gaze allocation is influenced by several context-related *high level factors* in cluttered environments. Not taking these factors into account deprives the model of important contextual information that would otherwise predict attention with higher accuracy. We set three criteria to be satisfied in order to classify a high level factor as fit to predict attention. A factor (i) should affect attention as documented in cognitive psychology literature, (ii) should be measurable (iii) should be observed in a video game or computer generated imagery.

However, quantifying qualitative features such as object topology, inter-object relationships and tasks is a real challenge. We competently address this challenge by formally expressing these features in a Bayesian framework in order to generate probabilities of object attendance based on subjective features. In this work, six phenomena within the perception literature pointing to specific roles that high-level information can play in focal attention have been considered:

Scene Schemata. The first – the *scene schema* effect – is based on the observation that a high proportion of objects in a scene can usually be expected to be found there. They are "consistent" with the scene. Sometimes, however, objects are in a scene or a location that is very atypical. Such "inconsistent" objects are potentially salient (see, e.g., Figure 2.2a) [Bartlett, 1932]. Research has shown that previously-acquired knowledge of stereotypical object placement in a scene combined with the on-going visual experience of a scene can attract focal attention [Bar et al., 1996, Brewer and Treyens, 1981, Henderson et al., 1999]. The ratio and location of consistent and inconsistent objects in a specific context can also influence whether the scene is perceived to be congruent overall [Einhäuser et al., 2008, Hwang et al., 2011, Rayner, 2009].

Physical Singletons. The second effect – the *singleton effect* – refers to the finding that stimuli that are locally unique in terms of color or topology capture attention (Figure 2.2b) [Theeuwes and Godijn, 2002]. Object perception is based on context-dependent processing of low-level variables i.e. pixels, therefore the singleton state is a high level semantic property of spatially isolated objects.

Contextual Singletons. The third effect is a subdivision of the physically compound state, described above, by introducing two sub-states based on findings from psychological research (e.g., [Koffka, 1935]). Specifically, we hypothesize that a physically compound object can either be *contextually compound* or *contextually isolated*. Objects belonging in a set are contextually compound. An object positioned in-between a set of similar objects, but dissimilar from those in the set, is hypothesized to pop out even when not salient in terms of e.g. color

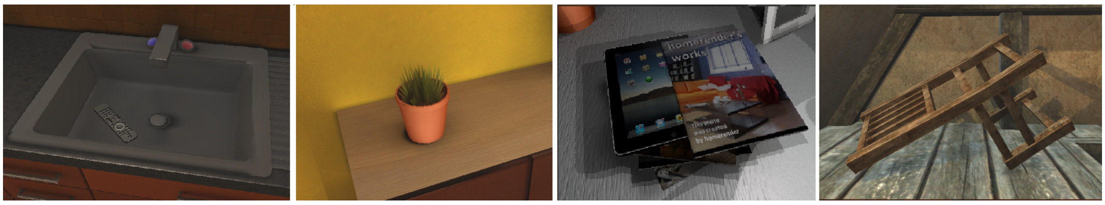


FIGURE 2.2: From left to right: A remote control is inconsistent with the sink context. The flowerpot is physically isolated. The tablet is contextually isolated. The chair is in a non-canonical form.

[Koffka, 1935]. For example, a tablet computer placed in-between magazines is salient (Figure 2.2c).

Canonical Form. The fourth effect is an object-intrinsic assumption. The three-quarters object view, that makes a large number of surfaces visible is considered to be an object's canonical form [Blanz et al., 1999, Secord et al., 2011]. The amount of angular deviation from this standard posture affects the object's saliency [Becker et al., 2007] (Figure 2.2d). Objects whose orientation is non-canonical are common in games e.g. dead characters or overturned vehicles.

Temporal Effects. Object coherence in time is also important. An attended location is usually prevented from being attended again [Posner and Cohen, 1984], an observation that has been used for LOD management [Longhurst et al., 2006]. However, there is strong evidence that recurring fixations are generated for objects that are inconsistent with the context or for objects that are in a non-canonical form [Becker et al., 2007, Henderson et al., 1999].

Feature Uniqueness. Finally, a single salient feature in an image pops-out more intensely than when several salient features exist [Frintrop et al., 2010, Itti et al., 1998], a biologically motivated feature uniqueness property.

2.1.3 Inattentional/Change Blindness and Eye-Tracking

Inattentional blindness (IB) is a psychological phenomenon which describes the act of failing to notice otherwise clearly visible and particularly salient objects in one's environment while engaged in and attending to a particular task [Pappas et al., 2005].

Change Blindness (CB) on the other hand gives name to the remarkable insensitivity to visual changes across saccades [Henderson and Hollingworth, 2003a]. In change blindness, observers fail to perceive large changes to a scene, as long as the change takes place during a brief interruption [Rensink, 2002], or is very gradual [Simons et al., 2000].

To study IB and CB various psychophysical experiments have been designed and performed. In an initial study [Neisser and Becklen, 1975], a display which presented two overlapping, simultaneous events was shown to participants. One of the events was a hand-slapping game in which the first player extended his hands with his palms up and the second player placed his hands on his opponent's hands with his palms down. The player with his palms up tries to slap the back of the other player's hands, and the other player tries to avoid the slap. On a second event three people were moving in irregular patterns and passing a basketball. Participants were asked to watch carefully one of the two events. If they monitored the hand-slapping game, they pressed a button with each attempted slap. If they monitored the ball game, they pressed the button for each pass. The results of this study are largely consistent with the findings of earlier research, as in most trials subjects had a great difficulty to simultaneously monitor both events.

Many subsequent studies (for a survey see [Rensink, 2002]) used this ball game task where observers attended to one team of players, pressing a key whenever one of them makes a pass, while ignoring the actions of the other team. After thirty seconds, a woman carrying an open umbrella ([Neisser, 1979]) or a black gorilla ([Simons and Chabris, 1999]) walk across the screen and are visible for approximately four seconds before walking off the far end of the screen. As expected, participants performed poorly in locating the unexpected stimuli. Surprisingly, only a handful of studies have incorporated eye tracking technology to their methodology.

Eye tracking data may indicate whether or not saccade targets relate to IB/CB. As far as IB is concerned useful conclusions were obtained by Cater et al. [2002]. Participants were asked to count the number of pencils in a cup placed in a computer generated scene while participants' eye movements were recorded. By varying the rendering quality of the background during the experiment they found that observers were usually incapable to distinguish the changes. A later extension to this experiment employed two still images with different rendering quality where participants were asked to count the teapots in a virtual scene (Figure 2.3)[Cater et al., 2003]. Eye tracking data confirmed that participants did indeed fixate on objects in the scene that were similar to the teapots and were of degraded quality, but failed to recognize that they were of lower rendering quality. This supports the idea that IB did indeed occur and this effect was not a degraded use of peripheral vision. Henderson and Hollingworth [2003b] studied IB when viewing complex real-world scenes. The pursuit of this study was to examine whether or not an individual can detect changes in a scene (such as rotated or removed objects) if he fixates on an area where the change takes place. Results indicated that participants were poor at noticing scene changes and that IB can occur even when individuals are fixating on the part of a scene that changes.

Moore [2001] and Mack [2003] did not find an answer to the most crucial question, whether or not individuals miss the stimulus completely or they do actually perceive it but memory fails to encode this information and thus it is forgotten. Similarly Pappas et al. [2005] fabricated a modern passing gorilla study [Simons and Chabris, 1999] that employed an eye tracker. They supported the initial claim that even when a stimulus crossed the fovea, not all individuals saw it. It was also discovered that some participants managed to notice the stimulus without fixating on it, which comes in contrast to the hypothesis stating that fixation is required to notice a stimulus.

Recently eye tracking and a lighter dropping magic trick has been employed to study IB [Kuhn and Findlay, 2010]. In this trick a magician picks up a lighter

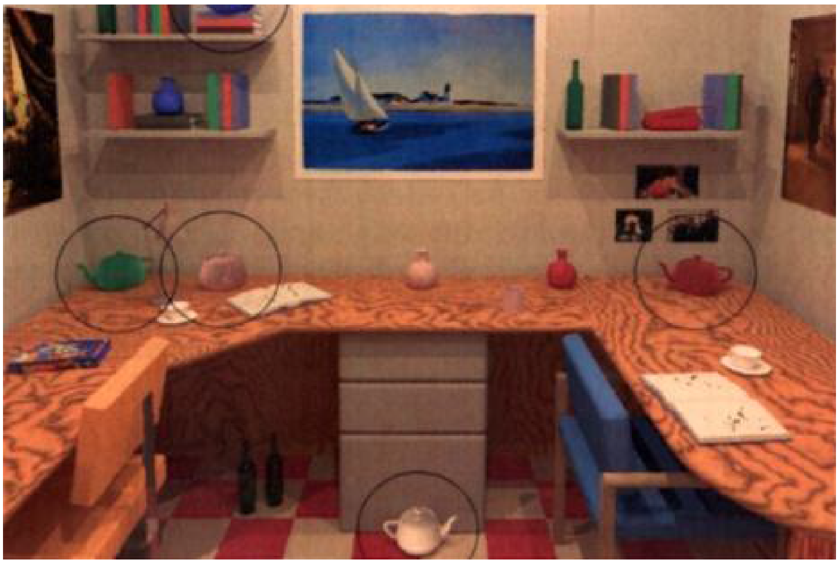


Figure 2.3: In this task, participants counted teapots. Rendering quality of the remaining virtual objects was degraded without participants perceiving the changes [Cater et al., 2003].

with his left hand and lights it. He then pretends to take the flame with his right hand and gradually moves it away from the other hand that is holding the lighter. During this move the magician is looking at his right hand. Once it has reached the other side, he snaps his fingers, waves his hand, and reveals that it is empty. At the same time the lighter is dropped into the lap which takes place in full view. Their results indicated that the point where observers were focusing at the time of the lighter drop was not affecting their ability to detect it. Covert attention probably was employed for the detection to occur.

Richards et al. [2012] conducted an experiment where a series of targets (white Ls and Ts) and distractors (black Ls and Ts) move around a computer screen bouncing off the sides of the display, and subjects monitor the number of bounces made by the targets but ignore the distractors. After a few seconds, an unexpected red cross traverses the screen. Their findings suggest that people that experienced IB (participants that did not notice the red cross) made more fixations and had longer gaze times on distractor stimuli, and were less likely to fixate on the unexpected stimulus. They thus had lower working memory capacity than those who did not experience IB i.e. they saw the unexpected stimuli. These findings are compatible to earlier research on working memory and IB. In addition, participants experiencing IB allocated their attention less efficiently than those who did not experience IB, as reflected in their eye movements tracked on irrelevant distractors.

The insensitivity to changes in a scene described by CB was initially studied by Rensink et al. [1997]. They wondered if CB is a general property of visual perception and for this reason they developed a flicker paradigm that simulated the visual events caused by moving the eyes; however not depending on eye movements to initiate scene changes. They accomplished this by inserting brief blank fields between alternating images of an original and a modified scene. Later eye movements were monitored while participants performed a change detection task with images of natural scenes [Hollingworth et al., 2001]. It was found that saccade targets hold a major role in the detection of changes to natural scenes.

We exploit the IB/CB effect when adjusting the rendering fidelity of complex shaders based on attention. We only alter shaders during player motion eliminating pop-out artifacts [Luebke, 2003] by exploiting the observer insensitivity to perceive changes occurring during brief interruptions.

2.1.4 Gaze Direction

An attempt to direct a viewer's gaze about a digital image has been presented (Figure 2.4) [Bailey et al., 2009]. Authors presented subtle luminance modulations to the peripheral regions of the viewer's field of view, in order to draw his attention over a modulated region. This modulation is automatically terminated before the viewer's foveal vision scrutinizes the stimuli that attracted his gaze. This new subtle gaze directing technique has a potential application in overriding IB and CB. According to the article this technique can be extended to motion pictures; in cases that this is useful or even important, certain areas or objects of natural and synthetic scenes that are expected to go unnoticed can be manipulated to attract viewer's attention.

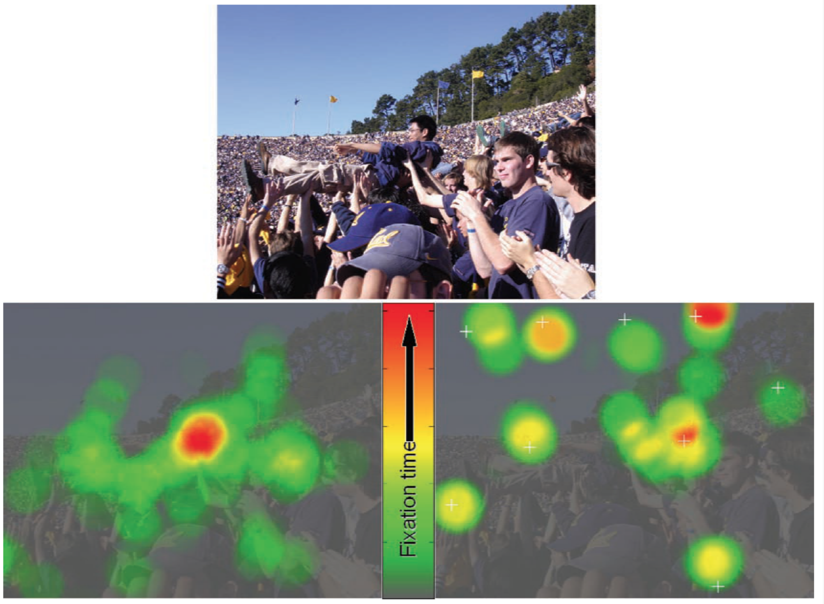


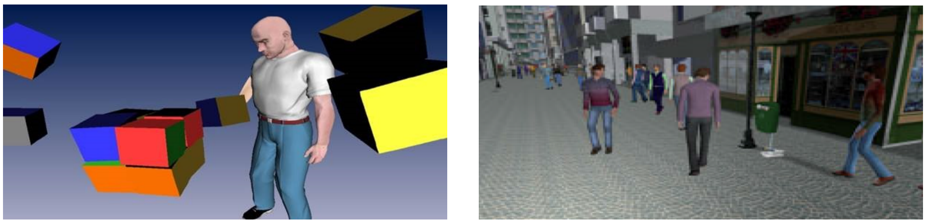
Figure 2.4: Fixation distributions for an image under static and modulated conditions. Input image (top). Gaze distribution for static image (bottom left). Gaze distribution for gaze-directed image (bottom right). White crosses indicate locations preselected by researchers for gaze direction [Bailey et al., 2009].

2.2 Attention in Computer Graphics

There is a large body of work on attention/gaze-based computer graphics and stereo viewing. Overviews can be found in corresponding surveys [Borji and Itti, 2013, Jacob and Karn, 2003, Mendiburu, 2012]; we review literature most relevant to our work.

2.2.1 Low/High Level Saliency & Tasks

In an effort to predict attention in pre-determined task areas, it has been shown that task importance maps may be used to accelerate rendering by reducing quality in regions that are unrelated to a given task [Cater et al., 2003]. Selective rendering guided by a FIT-based saliency model renders perceptually important parts of a scene in high quality while the remaining areas of the image are rendered at lower quality, thus saving in computational cost [Longhurst et al., 2006]. Other research has combined task maps with a low-level saliency map and validated the results



[Oyekoya et al., 2009] [Grillon and Thalmann, 2009]

FIGURE 2.5: Animating the gaze behavior of virtual characters and crowds.

using eye-tracking [Sundstedt et al., 2004, 2005]. However, FIT only uses low-level image characteristics often failing to predict fixations accurately.

Predicting gaze behavior in games may be used to optimize the distribution of computing resources [Sundstedt et al., 2008]. Saliency models have been employed to animate the gaze behavior of virtual characters [Oyekoya et al., 2009] and crowds [Grillon and Thalmann, 2009] (Figure 2.5).

Task relevant gaze behavior associated to first-person navigation in a virtual environment has been estimated by combining bottom-up and top-down components to compute user gaze point position on screen [Hillaire et al., 2010]. Saliency models and task related data have been linearly combined to track visually attended objects in a VE in task-specific areas [Lee et al., 2009], however, for a single pre-determined task.

Although task-based saliency estimations competently predict salient regions in pre-determined task-specific areas [Cater et al., 2003], the challenge is to estimate salient regions in all areas of a scene for different tasks via an integrated model. Research in interactive VEs has confirmed that attention is influenced by the semantic context of objects in the form of scene schemas [Mania et al., 2005, Mourkoussis et al., 2010, Zotos et al., 2009].

Mania and Robinson [2003] included a preliminary investigation of the effect of object consistency and illumination on object memory recognition in a VE (also

Mania et al. [2005]). Thirty-six participants across three conditions of varied rendering quality of the same space were exposed to a computer generated environment displayed on an HMD followed by completing a memory recognition task. The high quality and mid-quality conditions included a pre-computed radiosity simulation of an academic's office. The low-quality condition consisted of a flat-shaded version of the same office. They found that schema consistent objects of the scene were more likely to be recognized than inconsistent ones. Overall, higher confidence ratings were assigned to consistent rather than inconsistent items. Total object recognition was better for the scene including shadows compared to the flat-shaded scene. Even lower quality of rendering was adequate for better memory recognition of consistent objects.

More studies employed a more extreme set of rendering types: wireframe with added color and full radiosity [Mourkoussis et al., 2010, Troscianko et al., 2007] and polygon count [Zotos et al., 2009]. Their results showed a significant interaction between rendering type, object type, and consistent/inconsistent objects ratio. This suggests that inconsistent objects are only preferentially remembered if the scene looks "normal" or if there are many such objects in an "abnormal" scene such as in the wireframe condition. It was also shown that memory performance is better for the inconsistent objects in the radiosity rendering condition compared to the wireframe condition. They concluded that memory for objects can be used to assess the degree to which the context of a VE appears close to expectations, however, they did not propose a computational model for such an assessment.

In one step towards implicitly modelling high-level effects, machine learning techniques have been applied to eye tracking data in order to train a model to detect salient regions only in a pre-defined set of static photographs [Judd et al., 2009]. As an alternative to standard machine learning methods, a prototype self-refining fluid dynamics game that learns from crowd-sourced player data has been proposed [Stanton et al., 2014], concentrating computation in states the user will most likely encounter to improve simulation quality. A pipeline to derive gaze prediction heuristics from eye-tracking data for 3D Action Games has been proposed [Bernhard et al., 2010]. However, to date, a model that explicitly links in a physiologically plausible manner experimental outcomes on attention with object saliency is missing.

2.2.2 Attention based LOD

Gaze [Loschky and McConkie, 2000] and task [Cater et al., 2003] based LOD managers render the 2 degree fovea region in high quality (i.e. the high-resolution part of the visual field) and the periphery of vision with less detail. An eye tracker was employed by Luebke and Hallen [2001] to monitor fixations for gaze-directed rendering, allowing 3D model geometry to be simplified more aggressively in the periphery than at the center of user gaze. However, LOD management based on gaze encounters difficulties to maintain display updates without artifacts after fast eye saccades. Driving LOD based on pre-defined task areas is limited since it is impossible to quantify the nearly infinite number of potential tasks.

Since low level image features such as luminance, contrast and motion are known to attract attention [Itti et al., 1998], objects saliency models based on low-level features combined with task relevant information have been employed in order to drive LOD [Hillaire et al., 2010, Lee et al., 2009]. However, since high-level, cognitive phenomena also affect attention, low-level saliency models sometimes fail to predict fixations, especially when an observer manipulates interactive scenes [Sundstedt et al., 2008].

2.2.3 Modern LOD Approaches

Modern video games consist of various interconnected software components such as a graphics engine and an audio engine that share hardware resources. LOD methods are essential to improve the interactivity and responsiveness of graphics systems by distributing resources to the image regions that are expected to be attended [Luebke, 2003]. Traditional LOD approaches reduce polygon count by selecting an appropriate instance of polygonal complexity for each model depending on its importance [Luebke, 2003]. Object importance can be determined by attention deployment over the scene or perceptually motivated criteria such as the projected screen size of the object, eccentricity and velocity of objects [Clark, 1976].

However, polygonal counts are usually low in mobile devices and mobile GPUs are fill-rate bound deeming polygonal complexity LOD algorithms ineffective [Çapin et al., 2008]. Pixel shaders reproduce high quality visual details by exchanging polygonal complexity for additional ALU operations and heavy texture memory accesses. As computation power in mobile GPUs increases faster than memory bandwidth [Owens, 2005] a modern LOD manager should target significantly reduced texture fetches.

In this thesis, we develop and employ a sophisticated, multi-factor, context-based, attention predictor for interactive environments that takes into account contextual information about a scene to predict fixations more accurately when task-imposed restrictions exist compared to the state-of-the-art. We employ this predictor to optimize LOD for mobile platforms, balance game levels and manipulate stereo disparity.

2.3 Attention in Computer Games

Attention deployment greatly depends on game-play and vice versa [Sundstedt et al., 2013]. Eye tracking data has revealed that players playing First Person Shooter (FPS) games tend to concentrate on the center of the screen searching for enemies while in an Action-Adventure game players mostly explore the entire screen for game props to advance game-play [El-Nasr and Yan, 2006].

Attention in games may also get manipulated. A guiding principle and method based on the Guided Search theory [Wolfe, 1994] has been proposed to direct attention to target items that should be noticed by an observer in a video game e.g. an advertisement. In particular, when a frequently searched game object is modified to share perceptual features such as color or orientation with a target item, the item will attract attention [Bernhard et al., 2011].

2.3.1 Game Challenge

Player enjoyment is crucial for the success of a computer game. An enjoyable/optimal experience, also termed *flow*, is shown to be so satisfying that players take pleasure in the game with little concern for what they will get out of it [Czikszentmihalyi, 1990]. Sweetser and Wyeth [2005] suggested that flow experiences in games arise from eight core elements: concentration, challenge, skills, control, clear goals, feedback, immersion, and social interaction.

Challenge in particular, which is considered as the most important aspect of game design, refers to the ability of a game to be sufficiently intriguing and match the player's skill level. Both failure and success may become repetitive quickly. Successful games provide different levels of difficulty of their game play that adapt to player's increasing skills at an appropriate pace in order to maintain his interest [Desurvire et al., 2004, Pagulayan et al., 2003]. Thus, games should be designed with a proper balance of challenges and player skills. Improper balancing provokes anxiety (in a discouragingly hard game) or apathy (in a boringly easy game) [Johnson and Wiles, 2003].

2.3.2 Game Balancing and Search Tasks

Looking for an object is a common task in Adventure or Action-Adventure video games, often guiding level advances. The time spent searching for an object in a game should be in proportion to the advantage it conveys in game-play. Designers mostly rely on their experience and instinct while calculating cost/benefit ratios by manually placing objects and obstacles in their levels [Pagulayan et al., 2003]. Multiple rounds of play-testing and observation can stabilize choices in a level of a specific difficulty [Sweetser and Wyeth, 2005]. When using this approach, players themselves have to select a desired difficulty level via a menu. However a simple *"easy, medium, hard"* selection has very fuzzy borders between levels; easy and hard is not simple to define. A more sophisticated game difficulty management method is necessary.

Modern games of a vast variety of genres rely a lot to such manipulations. In a FPS special objects termed power-ups, offer an immediate means to replenish player's health levels or instil him with new capabilities [Lazzaro, 2004]. Game levels can be designed in a way to aid or burden the character to hide and protect himself. Role Playing Games and (Action-) Adventures employ special objects that can be found and collected in an inventory supporting the narrative and progressing the game when used in solving riddles or winning battles. Guidance in such games through portals or intuitive level design can assist roaming and feeling of immersion. The aim of these approaches is to solely prevent players from making errors and ultimately losing the game.

Since players' abilities vary and play-testers are not abundant to every game designer, a sophisticated approach such as the model we propose, that guides automatic object manipulation and game balancing based on high-level visual attention is crucial.

2.4 Quantifying Scene Semantics

Object perception in scenes relies on the integration of pre-existing knowledge with recently acquired knowledge from attentional processing [Henderson et al., 1999, Rensink, 2000]. This observation has been accounted for in a schema-based LOD framework where consistent with the context objects are rendered with lower quality without affecting information uptake [Zotos et al., 2009].

2.4.1 The Differential-Weighting Model

The DWM [Eckstein, 1998, Eckstein et al., 2006, 2002] estimates the interaction between visual evidence concerning a target in a scene and Bayesian prior probabilities indicating expectation and context of a scene. By combining sensory data with existing knowledge it calculates the posterior probability that a location will be fixated in a visual search task and thus predicts saccadic targeting.

DWM assumes that when searching for a target, each location in a scene elicits neuronal activity in relevant sensory units of each visual feature. This response is subject to Gaussian independent neutral noise, i.e. the outcome of the perceptual processing of this response is probabilistic. When a sensory unit is tuned to observe a specific feature, it responds at a higher rate when the observed feature is present. Neurons are subject to internal noise and have a response following a Gaussian distribution [Tolhurst et al., 1983]. After many trials, Figure 2.6 depicts the internal response probability density functions for noise-alone (left curve) and for signal-plus-noise trials (right curve). The model calculates the ratio of the joint likelihood of observing the feature's neural responses in each image region given that the target is present and the joint likelihood of observing the feature's responses given that the target is absent according to a selected probability. This noisy response is then weighted by context effects encoded in Bayesian priors relevant to specific stimuli. The Bayesian priors embody the probability of these stimuli to co-occur with other highly visible visual features of the image.

In this work we quantify for the first time critical high level factors and extend DWM to encode them in Bayesian priors (Chapter 4).

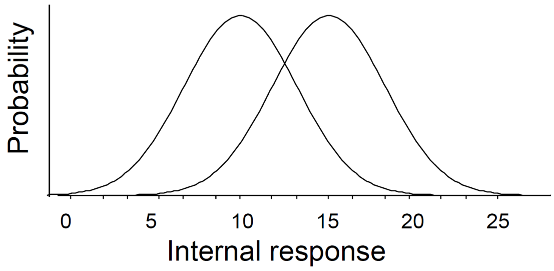


Figure 2.6: Internal response probability density functions for noise-alone (left curve) and for signal-plus-noise trials (right curve).

2.5 Eye Tracking

The gaze point of an observer can be directly determined via eye-tracking, however eye-tracking is rarely available in consumer applications.

Video-based eye-tracking (video-occulography) is the de facto standard in gaze estimation. Research in eye-tracking focuses in two major areas. *Eye detection* (locating/tracking eyes in an image) and *Gaze tracking* (determining the 3D line of sight & identifying attended location).

Eye detection techniques consist of *feature/shape-based* approaches, *appearance-based* approaches and *hybrid* approaches. Feature-based approaches rely on identifying local point features/contours of the eyes e.g. the iris center or limbus by fitting them to a rigid or deformable model of the eye. Appearance-based methods rely on an image template matching model built from the entire eye image. Hybrid approaches combine the benefits of feature- and appearance-based methods.

Gaze tracking techniques generate a gaze direction or Point-of-Regard from the image data via tracker calibration, saccade/fixation identification and by performing Gaze-to-Object-Mapping (GTOM). These techniques are either *2D-regression-based* or *3D-model-based.* 2D-regression-based methods assume the mapping of image features to gaze coordinates by parametrizing a polynomial. 3D-model-based methods compute the gaze direction by parametrizing a 3D geometric model of the eye.

Both feature-based [Ishimaru et al., 2013, Miluzzo et al., 2010] and appearance-based [Holland and Komogortsev, 2012] eye detection techniques can work on unmodified desktops and portable devices by using their embedded cameras. For gaze tracking both 2D [Holland and Komogortsev, 2012] and 3D [Wood and Bulling, 2014] models have been used. Feature-based methods on tablets are computationally intensive not allowing the device to perform any other task. Appearance-based methods on tablets require a lot of per-subject training and a lot of pre-processing to train a neural network for template matching. These methods have only been tested with the tablet performing eye tracking, and not with any other intensive application running simultaneously.

In this work, we propose two context-aware gaze prediction models that eliminate the need for real-time eye tracking allowing gaze-aware application deployment on most hardware platforms.

2.6 Machine Learning in Games

Machine learning algorithms can learn correlations of data from an existing dataset and make data predictions on novel data sets. Machine learning algorithms have been used in video games to accomplish a more believable, variable and challenging AI [Laird and VanLent, 2001]. In commercial video games machine learning has been employed both to learn at design-time, where its results are applied before publishing the game and to learn at runtime, for an individually customized game experience. For example, *LiveMove*™ is a machine learning tool recognizing motion and converting it to game-play actions to train a computer opponent. Another example is *Black and White*™ where the player's pet learns what to do in the game via reward and punishment.

Video Game State Variables. Game structure is defined during the design phase of a video game and is used to represent relationships between objects and player actions (obstacles to overcome etc.). Structure is represented in the source code via variables, which are used to represent commands and storage needs [Crawford, 1984]. For example, a player in a typical shooting game may be described by two vectors indicating location and rotation in the game scene, a value indicating health and a value indicating available ammo. The value of these variables in relation to the location or availability of ammo as well as to the location of an AI Non-Playable Character (NPC) influences the behavior of the player, e.g., whether he will run away from an enemy or engage in close combat.

A key idea of our work is that the players' behavior is related to the game state, and their gaze or attention will be related to their behavior. So, by automatically analyzing exposed variables of a video game, we can predict where they are looking. In our work we use Decision Forests (DFs), which provide powerful multi-label classification [Breiman, 2001] and support our goal of predicting the object class the user looks at, based on game state in real-time while a player is actively involved in game-play. DFs were selected since they use averaging to find a balance point between extremities in the samples, unlike single decision trees or Support Vector Machines (SVMs) that are likely to suffer from high variance or high bias depending on tuning parameters. DFs have very few parameters to tune and are effectively non-parametric. DFs do not require any knowledge about the underlying model of the data to yield predictions on novel data.

2.7 Gazing Stereoscopic 3D

Stereoscopic 3D displays create the illusion of depth by presenting a different image to each eye simulating natural vision. Stereo rendering significantly constrains attention modeling [Bruce and Tsotsos, 2005]. When attending a specific depth due to disparity, objects on other depth planes are not attended, and attention shifts faster to nearby objects than to objects deep into the scene [Han et al., 2005].

In this section we investigate stereoscopic rendering technologies and common issues during stereopsis.

2.7.1 The Rise of Stereo 3D

Stereoscopic 3D as a visualization medium for movies and games tells a story in the visual space both behind and in front of the screen, allowing for engaging content, stemming from the increased perceivable depth, enhancing artists' creativity. The addition of the third dimension in cinema is similar to the giant leap occurred with audio added to silent movies. Stereoscopic 3D aids character identification as it provides spatial detail missing from flat projection. From an information theoretic point of view, 3D essentially integrates two image views of the world in a single perceived scene, inherently providing extra information about the scene layout and character formation. Stereoscopic 3D also allows for increased sense of immersion since suspension of disbelief is effortless when simulating the sense of depth perceived in the real world. Depth aids the understanding of the current emotions experienced by the characters and allows the viewers to emotionally engage with them [Atkinson, 2011]. In close stereoscopic shots, the emotional charge increases because an actor's 3D volume now occupies the 3D visual space and human movement of bones and muscles is intensely visible [Mendiburu, 2012].

Stereoscopic 3D displays and content is soon to become ubiquitous. Visiting the movie theater is a popular social event and consumer HMDs are now becoming omnipresent with the extraordinary advent of the Oculus Rift, Samsung GearVR and HTC Vive. 3D displays have become an irreplaceable tool not only for entertainment but also for specialized applications including scientific visualization, image-guided surgical procedures, remote guidance of robots and battlefield reconnaissance.

In particular, 3D movie releases increase every year and a complete switch-over to 3D is expected to materialize soon since larger cinema screens yield intense

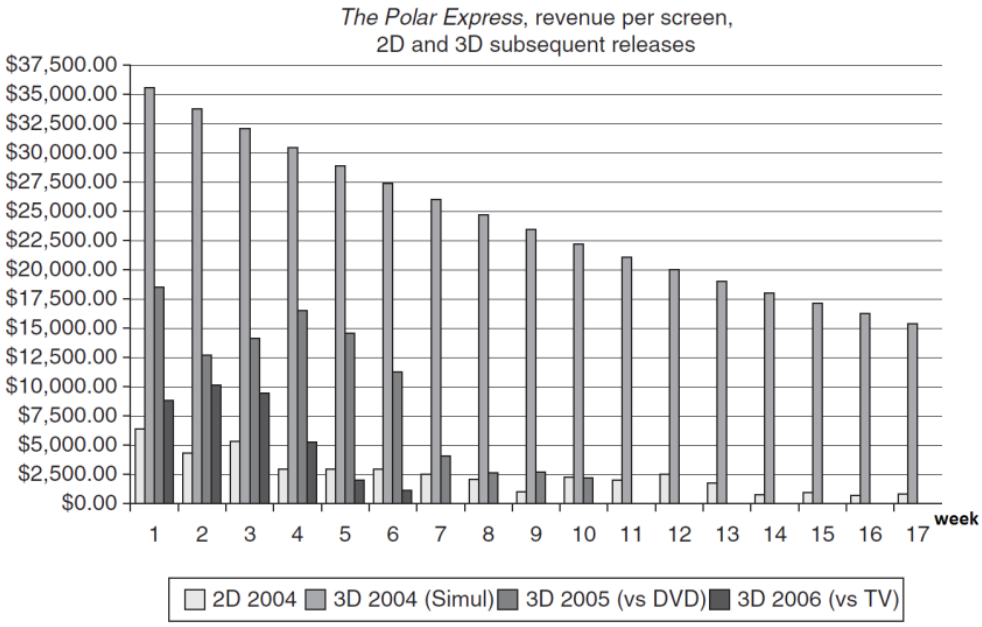


FIGURE 2.7: Graph showing that 3D movies earn one order of magnitude more revenue when compared to their 2D versions [Mendiburu, 2012]

stereo perception [Mendiburu, 2012]. Ultimately, 3D has the potential to become so ubiquitous but also critical for medical or simulation visualization applications, to the point that watching 2D content may occur only for the sake of nostalgia just like we watch black-and-white movies on TV (Figure 2.7) [Mendiburu, 2012].

Scientific visualization in stereoscopic 3D provides an additional visual axis displaying application-critical information. Perspective depth cues attract attention. Stereoscopic depth-of-focus techniques may be used to guide attention [Ware, 2012]. Stereoscopic 3D constitutes a valuable resource for the diagnosis and surgical treatment of pathologies [Udupa and Herman, 1999]. Image-guided surgical procedures decrease the mental effort of a doctor by guiding movements in three dimensions, since the necessary depth cues are provided by the 3D display itself.

Tasks hazardous to human life can be accomplished remotely through tele-robotic control benefiting from 3D displays. Immediate binocular coding of depth for tele-manipulation tasks critically requires operators to fully understand the relative locations of objects in the remote world. Experiments comparing 2D vs stereoscopic 3D interfaces indicated that 3D tele-operation reduces task execution time, error rates and time needed for training [Drascic, 1991].

2.7.2 Stereoscopic Viewing Details

In this section the basic geometry of stereoscopic vision relevant to this work is presented [Woods et al., 1993]. In stereoscopic rendering and in order to induce stereo perception, each eye obtains its own view rendered with a slightly offset camera location. The virtual screen is then perceived on the intersection of the left and right frusta. A stereo projection matrix is defined as a horizontally offset version of the regular monoscopic projection matrix, both offsetting for the left and right eyes along the x-axis. The projection axes should be parallel in order to avoid a converged configuration that introduces keystone distortions into the image, which can produce visual discomfort [Stelmach et al., 2003]. We use the standard asymmetric viewing frusta, as presented among others [Woods et al., 1993] shown in Figure 2.8.

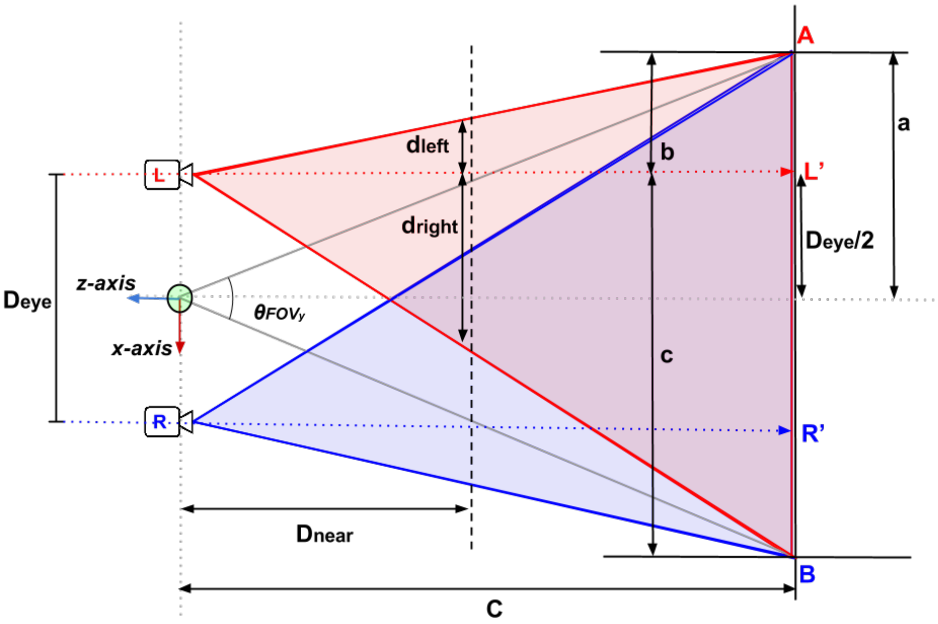


FIGURE 2.8: Asymmetric frustum stereo geometry.

The two cameras are symmetrically offset from the origin of x-axis at points L and R (Figure 2.8). The separation LR between them is D_{eye}. The cameras are directed parallel to one another, looking down $z-axis$. D_{near} is the near clipping distance and C is the distance between the camera and the perceived plane of focus, known as convergence distance. The left and right extremities of the virtual screen lie at points A and B respectively.

To generate an asymmetric viewing frustum the near clipping plane's top, bottom, left and right coordinates in addition to the near and far clipping planes distances are required [Woo et al., 1999]. To define a virtual screen a mono-frustum would be AOB. For this monoscopic frustum let us denote the Field-of-View (FOV) angle along the y-axis as θ_{FOVy} and the aspect ratio of the mono-frustum as r_{aspect}. We then estimate the top and bottom margins for both left and right frustums, in addition to D_{eye} as a system of simultaneous linear equations:

$$top = D_{near} \tan \frac{\theta_{FOVy}}{2} \qquad \& \qquad bottom = -top \tag{2.1}$$

The left frustum ALB, intersects the near clipping plane at d_{left} distance left of LL' and at d_{right} distance right of LL'. Given the triangles ALL' and BLL' we find that:

$$a = r_{aspect} C \tan \frac{\theta_{FOVy}}{2} \qquad \& \qquad \frac{d_{left}}{b} = \frac{d_{right}}{c} = \frac{D_{near}}{C} \tag{2.2}$$

$$b = a - \frac{D_{eye}}{2} \qquad \& \qquad c = a + \frac{D_{eye}}{2} \tag{2.3}$$

By interchanging b and c we estimate parameters for the right frustum ARB.

The image disparity p of a vertex with scene distance w is positive when the object is behind the virtual scene, and negative otherwise and is known as *parallax*.

Parallax depends both on interaxial separation D_{eye} and convergence distance C. We estimate D_{eye} for a predetermined maximum on-screen parallax $|p|$ (see [Jones et al., 2001, Shibata et al., 2011]) based on user-display distance and display size:

$$D_{eye} = \frac{w|p|}{w - c} \tag{2.4}$$

For the left eye frustum, parameters are:

$$left_L = -r_{aspect} \times a + (D_{eye} * b) \qquad right_L = r_{aspect} \times a + (D_{eye} * b) \tag{2.5}$$

For the right eye frustum, parameters are:

$$left_R = -r_{aspect} \times a - (D_{eye} * b) \qquad right_R = r_{aspect} \times a - (D_{eye} * b) \tag{2.6}$$

2.7.3 Stereo Technology and Common Issues

With the exception of HMDs that employ a dedicated display for each eye, stereoscopic 3D displays project the left and right views encoded together on a 2D screen and the display then relies on a decoding system which is commonly a pair of glasses, to selectively allow only one image to reach each eye. Based on the encoding/decoding domain of use, certain glasses work in the color spectrum such as Anaglyph and Infitec, others are time-driven such as active shutter glasses with Digital Light Processing (DLP), 3DTV and 3D projectors and others based on polarization (RealD) or space (Auto-Stereoscopic). Currently four fundamental stereoscopic technologies are adopted in the consumer market.

Passive filtered lenses. The classic red-blue "anaglyph" glasses. A modern version of filtered lenses uses polarization filters either linear or circular and is the de-facto standard for movie theaters. Color perception is negatively affected. Current generation polarized filters exhibit poor light output, essentially halving the light throughput of the projector. Visual discomfort is further increased due to the different spectral content presented to each eye, increasing the color rivalry. Optical crosstalk between channels stemming from poor spectral separation of colored lenses is a huge challenge [Konrad and Halle, 2007].

Active shutter glasses. The standard for 3DTV, also employed in some movie theaters (XpanD 3D). The media are displayed at a high frame rate and the glasses rapidly switch between black and clear using a pair of low-latency transparent Liquid Crystal Displays (LCD) (Liquid Crystal Shutters - LCS). One eye sees nothing while the other sees the correct image; a few milliseconds later, the viewing is reversed. Active shutter glasses suffer from [Konrad and Halle, 2007]: (i) A prismatic effect derived from the LCDs not being aligned correctly with the screen. (ii) Absolute precision is necessary for the glasses to produce accurate imagery; the error must be kept down to fractions of a microsecond. (iii) A transceiver device is often required to dispatch synchronization signals for the shutters. Despite these drawbacks, active shutter glasses are easy to employ with existing movie theater and television technology. Consumer electronics companies do not have to modify their screens in any way other than increasing the refresh rate.

VR goggles/HMDs. VR goggles employ a different display for each eye and are worn directly on the head of the user. There is no need for a decoding scheme. Commercial examples of HMDs include the recent excitement over the Oculus Rift (Figure 2.9), the Samsung GearVR and the HTC Vive. HMDs suffer from large disparities that cause fatigue in addition to the issues haunting the rest of the 3D display technologies. There are reports of people experiencing extreme binocular instability, poor depth perception and eye fatigue after being exposed to stereoscopic content [Williams and Wann, 1993]. 3D gaming in HMDs may even exacerbate this since one is not simply viewing a virtual 3D space but is also interacting with it.

Figure 2.9: The Oculus Rift HMD.

Auto-stereoscopic displays producing stereo without glasses. Commercially employed by the Nintendo 3DS. Two types of auto-stereoscopic displays exist. They can be either based on a lenticular lens, e.g. a saw-tooth prism in front of the screen that directs light in varying direction to separate each eye's view, or a parallax barrier, e.g. a series of slits in the display precisely placed to allow light from every other line of pixels to go one way or the other. These displays suffer from low resolution and need precise observer placement in front of the screen; otherwise the 3D illusion is destroyed. Effective resolution and brightness is halved since half of the lines are going one way and half the other way. In order to display 1080p content, a 4K display is needed. This method also requires modification of existing screens [Konrad and Halle, 2007].

2.7.4 Visual Discomfort, Fatigue & Stereo-Grading

Moving on from standard 2D/flat content to stereoscopic 3D triggers significantly more muscular and brain activity [Mun et al., 2012]. Viewer fatigue due to the vergence - accommodation conflict is common when viewing stereoscopic 3D content. The conflict is caused because the plane of focus (i.e., the screen) is fixed whereas eye vergence movements continuously occur when fusing stereoscopic content. Large stereoscopic disparities in video games further increase visual fatigue [Hoffman et al., 2008, Lambooij et al., 2011]. Symptoms range from an insensible overload of the visual system or slight discomfort that can cause major eye strain, provoke visually-induced headaches and lead to total loss of the depth perception [Lambooij et al., 2009]. The level of discomfort increases with the exposure time to 3D content not optimized for comfortable viewing.

High disparities force the eyes to rotate unnaturally in relation to each other. The standard tolerated disparity threshold is 24arcmin between the same point on two retinas [Jones et al., 2001] which however reproduces very small perceivable depths. Disparities can go well above this low threshold, however, visual discomfort may build up which should be avoided.

A solution to these issues is the stereo grading process, i.e. altering the depth structure of a scene by drawing objects in a user's comfortable disparity range, also known as the *comfort zone* of the observer [Shibata et al., 2011]. When objects are drawn in the comfort zone, clear and correctly fused binocular vision is achieved and discomfort is minimized [Shibata et al., 2011]. Such approaches have been developed for interactive stereoscopic applications or film, attempting to match the depths between cuts [Templin et al., 2014] or compress the depths of a scene [Lang et al., 2010, Oskam et al., 2011]. A perceptual disparity metric that can compare one stereo image to another to assess the magnitude of the perceived disparity change has been proposed [Didyk et al., 2011]. Universal depth compression may lead to limited depth perception or the cardboarding effect [Chapiro et al., 2014, Meesters et al., 2004].

Low quality stereo grading results to low quality stereo content and revenue losses because of decreased enthusiasm to play stereoscopic 3D games or watch stereoscopic 3D movies. Moreover, sales of 3DTVs, immersive headsets or autostereoscopic game consoles plummet and task execution times for tele-operation applications are increased together with reduced accuracy of tasks in the 3D space.

Visual Discomfort in Movies. In motion pictures, non-optimal stereo rig camera calibration leads to poor quality content acquisition for cinema. In stereoscopic movies it is common to adapt the range of disparities to a comfortable zone [Mendiburu, 2012] by adjusting the camera rig disparity during filming or via post-processing of the content. If this fails, novel views have to be generated from the start or composited from multiple stereo rigs of different disparities in order to alter the perceived depth of a scene.

Visual Discomfort in VEs. Interactive VEs such as video games accentuate the vergence - accommodation conflict since the viewer is not simply gazing at a virtual 3D space but is also interacting with it, altering the distance from objects in real time [Gateau and Neuman, 2010]. A virtual object closer to the observer than the in-focus depth plane exerts strong negative (crossed) disparity that may result in uncomfortable viewing, eye strain and diplopia. This discomfort in interactive scenes is due to both the selected disparity parameters and the lateral or in-depth motion of the objects [Jones et al., 2001].

Other Causes of Discomfort. The production of 3D imagery often results in image geometry defects such as image misalignment, keystone effects and colorimetric errors, i.e. color grading inconsistencies [Mendiburu, 2012]. In such cases, total loss of depth perception is common [Lambooij et al., 2009].

In this thesis we employ machine learning to *automatically* learn gaze patterns for different object categories and tasks without manual tagging, and accurately predict gaze accounting for complex task-dependent situations which would be very hard to encode explicitly. Our solution introduces *dynamic and localized* stereoscopic disparity management applied to 3D video games, for attended objects or areas based on the current task. Our approach smoothly relocates the perceived depth of attended objects/areas into the comfort zone of the observer, maintaining a rich sense of depth in sharp contrast to previous methods that suffer from severe depth compression (cardboarding).

2.8 Chapter Summary

We presented previous work on visual attention prediction and prediction of attention as employed in computer graphics and computer games. We presented related work on game balancing & LOD and investigated the internals of the DWM, eye tracking, machine learning for games and stereo disparity manipulation algorithms. In the following chapter we present a technical overview of the deployment test-bed and data acquisition framework for our saliency models.

Chapter 3

Overview of Data Acquisition Framework

Game engine pre-requisites in addition to eye- and head-tracking data acquisition and processing technical implementation details are presented in this Chapter. The data acquisition framework is employed in model formation and validation experiments of the following chapters.

3.1 Eye-tracker & Head-tracker Integration

3.1.1 Essential Unity3D Concepts

We employed the Unity 3D game engine. Each project in Unity 3D is composed of scenes. Employing several scenes distributes loading times and allows for better organization of the project in modules that can be tested individually. However, only a single scene can be loaded at one time for editing. The building blocks of each Unity project are called Assets. Assets include textures in the form of image files, 3D models in the form of meshes, audio files for sound effects etc. Every Asset (e.g. a mesh) when instantiated in a scene becomes a GameObject. Example

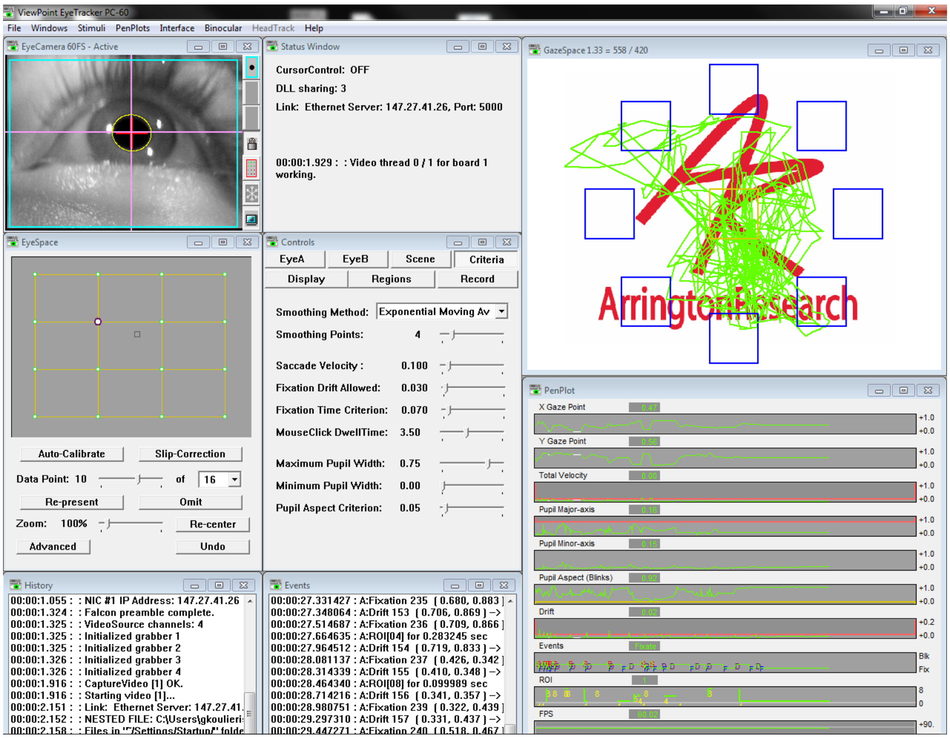
GameObjects include lights, cameras, particle emitters etc. GameObjects can be nested with each other to create parent-child relationships.

GameObjects are formed from Components; a Component is a script or module performing a function. Every GameObject must contain at least one Component, the Transform component holding the position, rotation and scale of the GameObject. More functionality can be introduced by adding more Components. Components introduce behavior, define appearance and interaction with the physics engine, etc. Example Components are Rigidbody, Collider, Animations, Audio sources, Scripts etc. In particular, a Script is a component that extends or modifies existing functionality of Unity3D. Scripts can be developed either in C-Sharp or JavaScript. A Prefab is a stored version of an object complete with its Components, Assets etc. that can be re-used in different parts of a project or even other projects. By employing Prefabs complex objects can be instantiated at any time.

3.1.2 Essential Eye-tracker Software Concepts

The eye-tracker of our HMD comes with Viewpoint; a software library by Arrington Research. The software provides a complete eye tracking Graphical User Interface (GUI) including stimulus presentation, eye movement and pupil monitoring paired with a Software Developer's Kit (SDK) for communication with other applications. Important Viewpoint eye-tracker concepts are presented below. For a complete reference please consult Viewpoint documentation [Arrington, 2015].

The Viewpoint main window includes the EyeCamera, GazeSpace, Status, and PenPlot windows (Figure 3.1). The EyeCamera window displays the video image of each eye providing controls to get more reliable eye tracking results by adjusting camera parameters. The GazeSpace window is the normalized coordinates window of the corresponding calibration geometry as estimated from EyeSpace coordinates in turn estimated from the EyeCamera image. GazeSpace displays fixation based on the relative location of the pupil, glint, and pupil-glint delta-vector as obtained



CursorControl: OFF

DLL sharing: 3

Link: Ethernet Server: 147.27.41.26, Port: 5000

00:00:1.929 : : Video thread 0 / 1 for board 1 working.

Smoothing Method: Exponential Moving Av

Smoothing Points: 4

Saccade Velocity : 0.100

Fixation Drift Allowed: 0.030

Fixation Time Criterion: 0.070

MouseClick DwellTime: 3.50

Maximum Pupil Width: 0.75

Minimum Pupil Width: 0.00

Pupil Aspect Criterion: 0.05

Data Point: 10 of 16

Zoom: 100%

```
00:00:1.055 : : NIC #1 IP Address: 147.27.41.26
00:00:1.324 : : Falcon preamble complete.
00:00:1.325 : : VideoSource channels: 4
00:00:1.325 : : Initialized grabber 1
00:00:1.325 : : Initialized grabber 2
00:00:1.326 : : Initialized grabber 3
00:00:1.326 : : Initialized grabber 4
00:00:1.916 : : CaptureVideo [1] OK.
00:00:1.916 : : Starting video [1]...
00:00:2.151 : : Link:  Ethernet Server: 147.27.41.
00:00:2.152 : : NESTED FILE: C:\Users\gkoulieri
00:00:2.158 : : Files in "~/Settings/Startup/" folde
```

```
00:00:27.331427 : A:Fixation 235  [ 0.680, 0.883 ]
00:00:27.348064 : A:Drift 153  [ 0.706, 0.869 ] ->
00:00:27.514687 : A:Fixation 236  [ 0.709, 0.866 ]
00:00:27.664635 : A:ROI[04] for 0.283245 sec
00:00:27.964512 : A:Drift 154  [ 0.719, 0.833 ] ->
00:00:28.081137 : A:Fixation 237  [ 0.426, 0.342 ]
00:00:28.314339 : A:Drift 155  [ 0.410, 0.348 ] ->
00:00:28.464340 : A:ROI[08] for 0.099989 sec
00:00:28.714216 : A:Drift 156  [ 0.341, 0.357 ] ->
00:00:28.980751 : A:Fixation 239  [ 0.322, 0.439 ]
00:00:29.297310 : A:Drift 157  [ 0.331, 0.437 ] ->
00:00:29.447271 : A:Fixation 240  [ 0.518, 0.467 ]
```

Figure 3.1: The Viewpoint Interface.

during calibration. EyeSpace provides information about calibration accuracy and allows identification and correction of individual calibration errors by allowing manual recalibration of individual points or the ability to omit problematic points.

The Controls window allows the user to adjust the image-analysis and gaze-mapping parameter settings and to specify the feedback information to be displayed in both the Stimulus window and the GazeSpace window. The user can select pupil segmentation and corneal reflection parameters to exclude erroneous reflections or shadows. Image quality adjustments can be made here and the tracking method specified. Smoothing parameters and segmentation criteria can also be set in this form. The Status window presents details about processing performance and measurements. The Stimulus window is a new window that pops up when calibration begins. The Pen Plot window displays plots of X and Y position of gaze, velocity, ocular torsion, pupil width, pupil aspect ratio and drift in real time.

3.1.3 Eye Tracking Pipeline

An infra-red light source both illuminates eyes and provides a specular reflection (glint) from the cornea. The video signal from the camera containing both the pupil and the glint is digitized by a video capture device. Image segmentation methods are applied to the eye image to identify the locations of the pupil and glint. Additional image processing operations locate the coordinates of these features and estimate the difference vector between these locations. A mapping function transforms eye position signals from EyeSpace coordinates to the subject's GazeSpace coordinates as mapped by a calibration procedure. A calibration component presents calibration stimuli at known locations to map points of the screen to specific intervals of the pupil-glint delta vector. A mapping function is then generated to map eye position in relation to display locations. Viewpoint data transfers are made possible via a Dynamic Link Library (DLL) for real-time access to all ViewPoint data. The VPX_InterApp.lib library file is imported in C-Sharp and its functions called.

3.1.4 Calibration

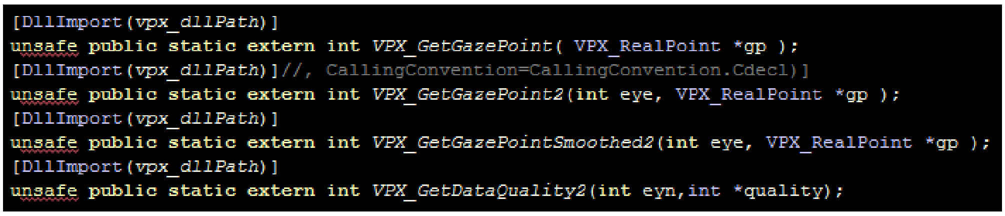
Prior to acquiring eye fixations the user must undergo a personal calibration process in order to match eye coordinates (EyeSpace) to gazed display coordinates (GazeSpace). Calibration takes approximately 1-2 minutes to complete during which several green square targets are displayed at different locations of the screen while the user is directed to fixate on their centers. During the calibration process it must be ensured that the pupil is accurately tracked at all times by paying attention to the camera window. Regarding calibration points at least 9 should be used; tests has shown that the best calibration results are yielded with around 16 to 20 points. Successful calibration is indicated by a rectilinear calibration point grid and well separated configuration points following calibration. Stray calibration points can be identified and re-calibrated or omitted. The EyeSpace window allows the user to select stray calibration points for re-calibration. If a point is selected to be re-calibrated it is then re-presented in the screen and the participant is asked to look at the center of it. If the calibration points grid is not rectilinear e.g. lines are crossing each other, a complete re-calibration should be done.

3.1.5 Head-Tracker Data Communication

The head tracker of our HMD is the InterSense InertiaCube3. It is an inertial Three Degrees of Freedom (3-DOF) orientation tracking sensor and software. It obtains motion sensing data using a miniature solid-state inertial measurement unit sensing angular rate of rotation, gravity and earth magnetic field along its three main axes. Angular rates are integrated to obtain the orientation (yaw, pitch, and roll) of the sensor. Gravimeter and compass measurements are used to prevent the accumulation of gyroscopic drift. The isense.dll SDK library provides a standard interface for the device that we integrate with Unity to receive head-tracking data. The InterSense Server Application, ISERVER provides communication services to applications requiring tracker data. It is the link between the head tracker's data output and third party applications such as Unity3D.

3.2 Data Processing

In order to receive data form the eye tracker the application needs to register with its DLL. Following registration, the Unity application obtains a unique message identifier used by ViewPoint for inter-process communication. Since the DLL is already pre-compiled and written in C++ which is an unmanaged programming language, we implemented a new class MyVPX in C-Sharp that binds the library with our application. C-Sharp allows calling unmanaged code from managed applications, through the DLLImport attribute. Using the DLLImport attribute we can tell the compiler to declare a function residing in the VPX_InterAPP.dll. The sample code below imports the necessary functions from the DLL.



```
[DllImport(vpx_dllPath)]
unsafe public static extern int VPX_GetGazePoint( VPX_RealPoint *gp );
[DllImport(vpx_dllPath)]//, CallingConvention=CallingConvention.Cdecl)]
unsafe public static extern int VPX_GetGazePoint2(int eye, VPX_RealPoint *gp );
[DllImport(vpx_dllPath)]
unsafe public static extern int VPX_GetGazePointSmoothed2(int eye, VPX_RealPoint *gp );
[DllImport(vpx_dllPath)]
unsafe public static extern int VPX_GetDataQuality2(int eyn,int *quality);
```

FIGURE 3.2: Sample function calls inside Unity3D.

C-Sharp allows using pointer variables in an unmanaged function code block only when the block is marked with the unsafe keyword. To be able to run unsafe code Unity needs two files to be included in the Unity project directory: smcs.rsp and gmcs.rsp file, containing only the command ”-unsafe”.

After registering with the Eye-tracking software, myVPX class defines a callback function which is managed by the library VPX_InterApp.dll. Inside the callback functions limited coding can be used. Since it interacts with unmanaged code written in C++ only standard data types such as integers or characters can be defined and used. In any other case the application crashes. For example, C-Sharp handles strings differently than C++. Inter-process communication fails for such data structures.

The *theCallBackFunction* function is responsible for data exchange between the library and the executable. When a connection is established, ViewPoint Status Window DLL Sharing counter denoting the number of third party applications that are currently registered increases by one.

Several times a second fresh data arrive from the frame grabber and the application sends them to all registered applications. The library VPX_InterApp.dll calls every function that was defined as a callback for each application in order to forward incoming data. Incoming data inform the application about possible new fixations, saccades, or error messages. For every new data entry quality checks are made, using the VPX_GetDataQuality2() function of the VPX_InterApp.dll.

5	Pupil scan threshold failed.
4	Pupil could not be fit with an ellipse.
3	Pupil bad: criteria limits exceeded.
2	Wanted glint, but it was bad, using the good pupil.
1	Wanted only the pupil and got a good one.
0	Glint and pupil OK.

Table 3.1: Quality Codes and information depending on quality code by Arrington. Data may be either fetched or discarded. Quality codes 0, 1 and 2 are considered as good data and fetched. Quality codes 3,4,5 are discarded.

Depending on quality check results codes (Table 3.1), gaze data are fetched using the VPX_GetGazePoint2() function.

Before exiting the application, the registration must be cancelled. This is mandatory since if not done probably, ViewPoint keeps sending data on a non-existent receiver, possibly crashing later application instances.

3.2.1 2D Gaze points De-projection

A 2D projection maps a 3D point from a 3D world space coordinate system to a 2D screen coordinate system. De-projection is the opposite. When receiving a 2D coordinate gaze point on screen we want to reconstruct the viewing ray emanating from the observers eyes that generated this point. This ray is described by a 3D world space origin and a direction. This is necessary for GTOM in order to detect fixated objects in the 3D scene. The raw data received form the ViewPoint application denote user's fixations in 2D normalized coordinates [0,1] in $x-y$ axes. We perform a ray reconstruction using this point. The ray origin is the virtual camera. The ray direction is such, that the ray passes through the gaze point.

First, the normalized 2D GazeSpace coordinates must be converted into 2D screen coordinates (pixels). Since the total resolution of our HMD display is 2560 x 1024 pixels every gaze data value has to be multiplied and scaled based on this custom resolution. In some experiments, only the dominant eye's data are considered as processable gaze data used and therefore, only the dominant half of the screen, 0 to 1280 pixels, horizontally is de-projected into the scene.

Having transformed the 2D gaze data into 2D screen coordinates the next step is to cast a ray into the scene. The ray must be cast in order to determine what the user is looking in the 3D world. The ray is cast in every frame having as the origin the dominant eye-camera location and direction passing through the GazePoint. The ray's depth is set to infinity.

If the ray cast does not hit anything then no action is taken but if a GameObject is hit, collider information are further processed. In our projects we usually consider the hit GameObject's name or tag; depending on the purpose of the ray cast the main controller class of the application decides on the action to be taken.

3.2.2 Head-Tracker Data acquisition

Data is passed from the head-tracking device through the InertiaCube3 software's SDK using a registration link with the tracker's isense.dll. The DLL is accessed with the C-Sharp DLLImport attribute, similarly to eye-tracking data acquisition. Using the ISD_OpenTracker function, the application searches for a tracker connected to the computer. If a tracker is detected then a timer is set to count the connection duration. ISD_ResetHeading synchronization is then performed. Following a successful connection, head-tracking data are passed from the device to our application by calling the ISD_GetTrackingData function once per frame. When the application exits, ISD_CloseTracker is called to terminate the connection.

The yaw, pitch and roll orientation data received from the head-tracker are used to adjust orientation. The ISD_CloseTracker function is responsible to terminate the connection when exciting the application.

3.2.3 SQLite integration

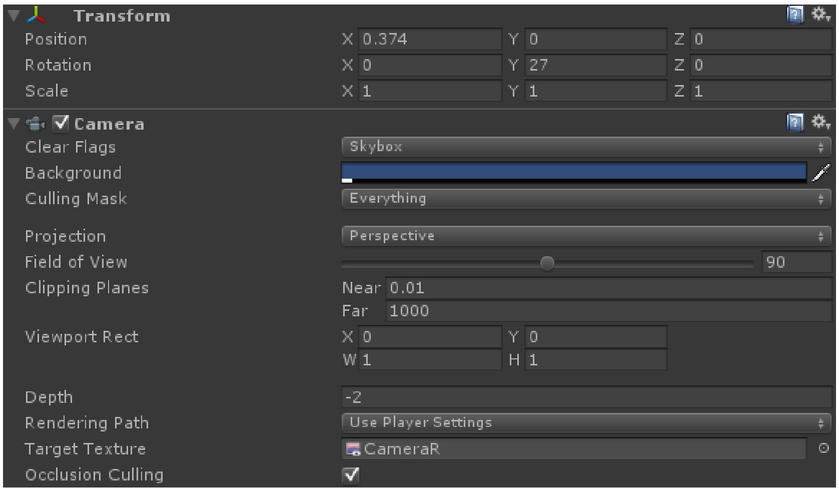
To record experimental data from the eye-tracker and head-tracker we employ an SQLite database. SQLite is a relational database management system contained in a programming library. SQLite is not a client-server database engine like common database systems. It is directly embedded to the end application. The SQLite engine has no standalone processes with which the application program communicates. The SQLite library is linked to the application and becomes part of it. SQLite's functionality can be called via function calls reducing latency in database access. The entire SQLite database is stored as a single self-contained cross-platform file on the host machine.

3.3 nVisor SX111 Stereo Camera

Binocular displays have a parameter called optical FOV overlap that is usually 100%. However, our nVisor SX111 HMD is designed with partial overlap, meaning that a portion of the FOV includes the same image in both eyes (binocular part) while separate, extended images are displayed in the left and right eyes displays (monocular part).

Partial overlap dramatically increases the FOV since the monocular portions of an image extend the viewing range of the field beyond the binocular part. However, the partial optical overlap configuration of the nVisor SX111 optical system requires more complex viewing frustum parameters.

The percentage of optical overlap is expressed as a fraction of the binocular FOV (50 degrees) over the monocular FOV (76 degrees). In the case of the SX111, this is 50 to 76 degrees = 65.7%. The decrease in optical overlap results in a decrease in the stereoscopic region of the display. Since most people perceive stereopsis in the center of their FOV, the reduced stereo angle remains acceptable.



Transform

Position X 0.374 Y 0 Z 0

Rotation X 0 Y 27 Z 0

Scale X 1 Y 1 Z 1

☑ Camera

Clear Flags Skybox

Background

Culling Mask Everything

Projection Perspective

Field of View 90

Clipping Planes Near 0.01 Far 1000

Viewport Rect X 0 Y 0 W 1 H 1

Depth -2

Rendering Path Use Player Settings

Target Texture CameraR

Occlusion Culling ☑

FIGURE 3.3: Parameters of the right virtual camera.

In order to display the left and right cameras outputs to each HMD screen we use both graphics card's outputs. The HMD requires SXGA resolution for each eye to work. Based on the SXGA format each camera of the HMD has a resolution of 1280 x 1024 pixels, amounting to a total of 2560 x 1024 pixels for the display of the HMD screens. The application view-port is expanded using the graphics card's option to consider both outputs as a single display of a total 2560 x 1024 pixels.

We designed a virtual head model with eyes and neck in Unity3D bearing the two virtual cameras. Both cameras were positioned at 2/3rds of head's height. We account for neck distance; the neck is the pivot point of rotation. The eye pair was displaced off-neck-axis.

The Inter-Pupillary Distance (IPD) between the two eyes/cameras was based on the average pupil distance measurement; 65mm (0.065 Unity3D world units). The IPD could be changed manually by using the keyboard.

Based on the HMD's specifications, the FOV of each camera-eye was set to 90 degrees and the cameras were rotated outwards by 13 degrees left and right respectively in relation to the virtual neck (Figure 3.3).

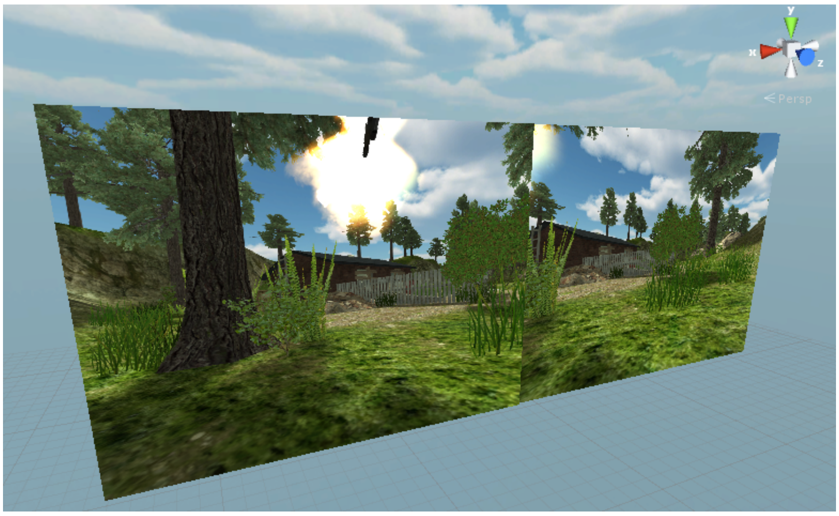


Figure 3.4: Twin render textures for the parallel camera rig.

In order to display the scene to the user, each camera's video output is projected on two render textures that correspond to the HMD screens. A third camera with a parallel projection matrix records the side-by-side render texture outputs and is used as the main output of the application (Figure 3.4).

3.4 Chapter Summary

This chapter included the technical overview of the deployment test-bed and data acquisition framework for our saliency models. In the following chapters we describe the employment of the aforementioned code bases for eye-tracking data acquisition and experimental validation of our results.

Chapter 4

High Level Saliency Modeling for Game Balancing

In this chapter, we present our new model of high-level attention. Before presenting our new model, we first describe how DWM handles Bayesian priors. We then explain how we extended DWM equations by encoding the interaction of novel factors affecting gaze deployment (Chapter 2) based on the Bayesian priors of the original model. This chapter represents our first attempt to identify the effect of the first two high level saliency factors on attention and also verifies our model validity on game balancing. The remaining four factors are integrated in Chapter 5. [1]

[1] The contributions in this chapter were published in the ACM Transactions on Applied Perception [Koulieris et al., 2014a] and presented at ACM SIGGRAPH [Koulieris et al., 2014c].

4.1 High Level Saliency Modeling

4.1.1 The Differential-Weighting Model

The DWM [Eckstein, 1998, Eckstein et al., 2006, 2002] estimates the interaction between visual evidence concerning a target in a scene and Bayesian prior probabilities indicating expectation and context of a scene. By combining sensory data with existing knowledge it calculates the posterior probability that a location will be fixated on in a visual search task and thus predicts saccadic targeting.

For each image frame f and each visual field location (x, y), each sensory unit responds in a noisy manner for each feature λ_j. DWM calculates the likelihood $l_{j,x,y,f}$ of observing the response λ_j given the presence of the target's j^{th} feature at that location and the likelihood of the response given the absence of the feature. The response has a Gaussian distribution [Tolhurst et al., 1983] with a mean of d'_j and a standard deviation σ. The likelihood $l_{j,x,y,f}$ that the j^{th} sensory unit takes a value $\lambda_{j,x,y,f}$ given the presence of the target's j^{th} feature at (x, y) on frame f is then

$$l_{j,x,y,f}(\lambda_{j,x,y,f}|s) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp{(-\left(\frac{(\lambda_{j,x,y,f} - d'_j)^2}{2\sigma^2}\right))} \tag{4.1}$$

s stands for signal and denotes the presence of the target.

The likelihood that the j^{th} sensory unit takes a value λ_j given the absence of the target's j^{th} feature is

$$l_{j,x,y,f}(\lambda_{j,x,y,f}|n) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp{(-\left(\frac{(\lambda_{j,x,y,f})^2}{2\sigma^2}\right))} \tag{4.2}$$

n stands for noise and denotes the absence of the target.

A likelihood ratio LR [Green et al., 1966] can be calculated as

$$LR_{j,x,y,f} = \frac{l_{j,x,y,f}(\lambda_{j,x,y,f}|s)}{l_{j,x,y,f}(\lambda_{j,x,y,f}|n)} = \exp\left(\frac{\lambda_{j,x,y,f}d'_j - 0.5d'^2_j}{\sigma^2}\right) \tag{4.3}$$

4.1.2 A New High-level Attention Model

As a first step, we propose a new two-factor model by integrating high-level information implied from semantic (schema inconsistency) or physical (singletoness) context represented by Bayesian priors in the DWM. We assume that (i) the internal response associated with high level saliency components is also subject to noise, (ii) dedicated, high-level sensory units are analogous to low-level sensory units. The high-level units fire at their highest rate when fed with the correct high-level feature, much as a low-level edge-detection unit reacts highest when an edge with the proper orientation is presented [Eckstein, 1998, Eckstein et al., 2006, 2002]. Whether the neural mechanism underlying a high-level sensory unit is a single neuron or a cluster of neurons does not matter. What matters is that there is an internal (neural) state reflecting whether this high-level feature is present or not. For example, we propose a sensory unit that monitors the degree to which an object is isolated. Such a unit would fire when a singleton object is in the FOV [Steinmetz et al., 2000].

We extended the original DWM equations to describe two high-level sensory units tuned to schema inconsistencies and the singleton state of objects. Equations 4.1 - 4.3 assume that the internal response generated by the presence of each visual feature is known a priori. Since neuronal response strength is unknown concerning scene schemata and singletons, we alter the DWM and instead calculate the posterior probability that the target is present at each pixel as a sum of K different feature strengths d'_k associated with scene schemata and singletoness.

$$P_{semantic,x,y,f}(s|\lambda) = \sum_{k=1}^{K} LR_{semantic,x,y,f,k} \tag{4.4}$$

$$P_{physical,x,y,f}(s|\lambda) = \sum_{k=1}^{K} LR_{physical,x,y,f,k} \tag{4.5}$$

We then average the components 4.4, 4.5 using weights $w_{semantic}$ and $w_{physical}$ that we obtain from perceptual studies (Section 4.2) to calculate the posterior probability that a location will be attended. A linear combination of components is a common practice in saliency detection algorithms [Frintrop et al., 2010].

$$P_{x,y,f} = w_{semantic}P_{semantic,x,y,f} + w_{physical}P_{physical,x,y,f} \tag{4.6}$$

Manipulating Attention for Adaptive Game Balance. Let us consider an example game balancing condition in a game. Suppose that a virtual object depicting a key is required to be collected in an adventure game to open a door. The player knows that he is searching for a key. Bayesian priors for the key as a target can be computed and weighted based on the location of the key. These priors could be evaluated in real time during game level loading and according to the desirable level of difficulty. Probabilities of existence in specific locations or as a singleton object can be inquired to anticipate the difficulty of a search task for this key and ultimately employed to modulate the cost/benefit curve. For example for an easy level of difficulty the key could be placed on a table (a consistent location for placing a key) or kept isolated. On the contrary, placing a key inside a set of other keys makes the task very hard and time consuming. Schema consistency can also be used inversely: When the player does not know exactly what he is looking for, an inconsistent placement of a searched-for object makes it salient and probably will attract user fixations.

Now let us consider an example execution of our model for a bar counter. A coffee mug which is consistent with the context and a medical kit which is inconsistent with the context are shown in Figure 4.1 (top). Consider a sensory unit that tracks schema inconsistencies. The $\lambda_{semantic}$ of the image regions corresponding to the medical kit is higher than the $\lambda_{semantic}$s associated to the mug and counter. The $\lambda_{semantic}$ communicates a subjective rating of consistency, e.g. the higher the number, the more inconsistent the object is in relation to the context (Figure 4.1

bottom left). Let us assume that $K = 1$, $d'_{semantic} = 0.6$ and $\sigma = 0.2$. Because the medical kit is inconsistent, we assume for this example that $\lambda_{semantic} = 1.0$, similarly for the mug $\lambda_{semantic} = 0.16$, for the bar counter $\lambda_{semantic} = 0.22$ because they are both highly consistent. The likelihoods ratios of observing the medical kit, mug and counter are $LR_{medikit} = 36315.5$, $LR_{mug} = 0.1$, $LR_{counter} = 0.3$ respectively as derived from equation 4.3. The schema inconsistency unit would then estimate the medical kit as the most salient (Figure 4.1 bottom right). Similarly, $\lambda_{physical}$ is used to calculate the likelihood ratios of observation based on whether an object is placed as singleton (not visualized in Figure 4.1).

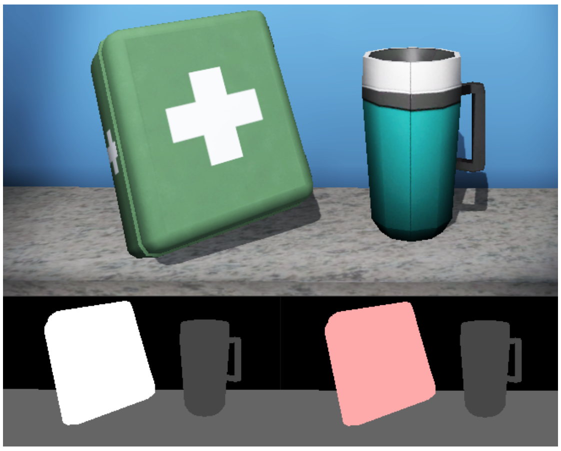


FIGURE 4.1: A bar counter context (top), $\lambda_{semantic}$ visualization (bottom left) and highest $LR_{semantic}$ highlighted (bottom right).

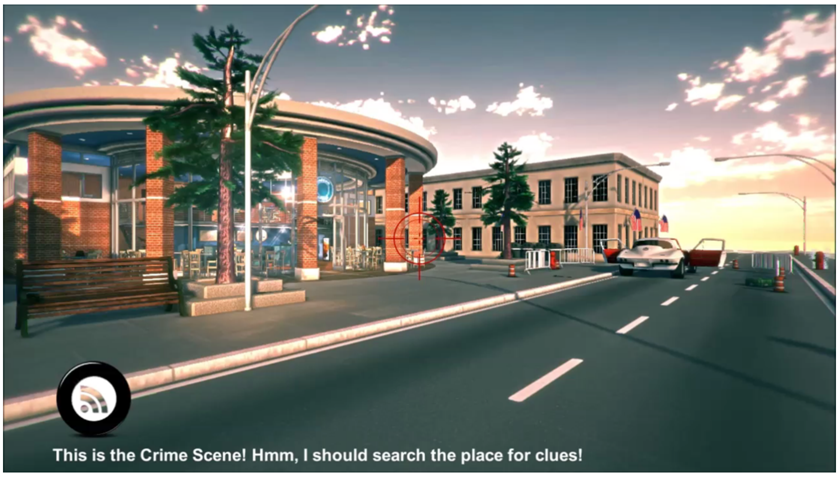
4.2 Real-time Evaluation of High Level Saliency Components

We examined the real-time effect of scene schemata and singleton on game-play for two reasons:

- The role of scene schemata and singletons in interactive, synthetic environments is unknown, even though their effects are well-documented for target detection displays or static real photographs [Einhäuser et al., 2008, Henderson et al., 1999, Rensink, 2000, Theeuwes and Godijn, 2002].

- Our extension of the DWM requires an empirical classification of objects in relation to scene schemata and the determination of the weighting factors w_j that signify the interaction between semantic (scene schemata) and physical context (singletons).

Inspired by Adventure games [Ju and Wagner, 1997], a suitable game genre to apply our method, we designed an environment that allows us to investigate the impact of high-level saliency on visual attention & game-play and recorded the time it took to search for plot-critical objects. The storyboard was based on the popular video game L.A. Noire$^{\text{TM}}$, a 2011 Action-Adventure neo-noir crime video game developed by Team Bondi$^{\text{TM}}$ and published by Rockstar Games$^{\text{TM}}$. A scene depicting a Coffee Shop (Figure 4.2) inspired by the "Driver's Seat" case of the game was heavily modified to include multiple areas representing a car schema and a cafeteria schema inclusive of sub-schemata representing a coffee shop counter and a lounge loft. We systematically controlled the semantic and physical states of plot-critical objects. Each object could be in a schema-consistent or a schema-inconsistent location, and could be in either a singleton state (positioned by itself) or a compound state (positioned in cluttered surroundings).



This is the Crime Scene! Hmm, I should search the place for clues!

Figure 4.2: The Coffee Shop scene.

4.2.1 Experiment 1: Defining Object Consistency

Here, we empirically classify scene objects as either consistent or inconsistent in relation to the context of each part of the scene. Specifically, a list of 50 objects (Appendix) was assembled and given to 21 graduate students (14 male, 7 female). Each participant used a 7-point Likert scale to rate how likely each item was to appear in a given scene. A rating of 7 meant that the object was very much expected to be in that location and 1 meaning the object was very much not expected. Half of the objects were tested in the Coffee Shop counter context and the other half were tested in the car context. We then selected a set of consistent objects from the high end of the scale and a set of inconsistent ones from the low end (based on the approach used in [Brewer and Treyens, 1981]). The classification of objects in relation to scene schemata is independent from a specific game scenario, i.e. a teapot is consistent with a kitchen context irrespectively of a background story. A taxonomy of common objects in relation to scene schemata that can be used in any game will be provided as part of the production level version of our system.

4.2.2 Experiments 2 & 3: Determining the Roles of Semantic and Physical Context

In Experiments 2 and 3 we examine the effect of physical (singletoness) and semantic (consistency effects) manipulations on game task completion time for two common tasks appearing in (Action-)Adventure games. In both tasks, the same general scenario was used: "*Adrian Black, a married man and a barista at the Coffee Shop decides to start a new life with his customer Nicole staging his own murder to cover a getaway with her*" (Appendix for the complete story). Participants were instructed in both tasks to find three decisive objects as quickly and as accurately as possible in order to solve the mystery (Figure 4.3). Experiment

2 used a *Search* task (participants knew exactly what they were searching for: a pair of spectacles, a pig purchase receipt and a wallet). Experiment 3 used a *Non-Search* task (participants did not know what they were searching for, and as such were exploring the environment with less of the specific purpose; the objects in quest were a photograph, a receipt, and a train ticket).

Our two main predictions are:

- Singleton objects will require less time to be recovered compared to objects in compound state because they capture attention no matter what the task is [Theeuwes and Godijn, 2002].
- When searching for an object, consistent locations will attract attention and therefore will require less time to be recovered than inconsistent locations. When not searching, on the other hand, objects at inconsistent locations should attract attention and therefore will require less time to be recovered compared to consistent locations [Eckstein et al., 2006].

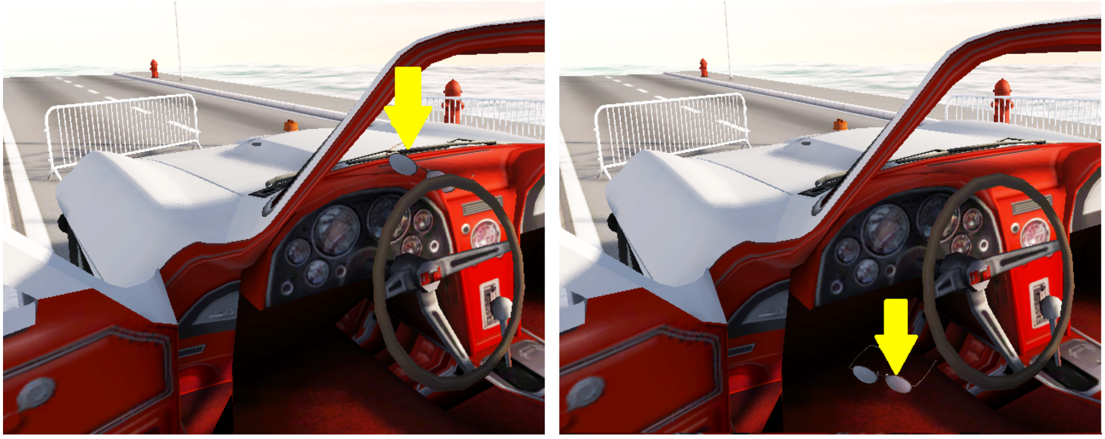


FIGURE 4.3: One of the desicive objects, the spectacles, as positioned in different conditions: Consistent/Compound (left) vs Inconsistent/Singleton (right).

4.2.2.1 Method

Each of the two main factors (Semantic Context and Physical Context) had two levels, which were factorially combined to produce four experimental conditions:

Consistent/Compound, Inconsistent/Compound, Consistent/Singleton and Inconsistent/Singleton object placement. The objects were positioned so as to maintain constant navigation time while reaching them across conditions and on similar visual angles within the VE. The four conditions above were the same for both experiments. A between-participants design was used, meaning that each person participated in only one experiment and in only one experimental condition.

Participants A total of 80 participants (56 male, 24 female; ages between 21 - 33) were recruited from the undergraduate and research population of our institution and were rewarded with pastry for their participation. All participants were familiar with first person perspective navigation and had normal to corrected vision. Upon arrival, each participant was randomly assigned to one of eight groups so that each group had 10 participants. Each group participated in only one of the experimental conditions.

Procedure and Apparatus Upon arrival, the participants signed a consent form and were then allowed to practice navigating in a training scene. The participants were then informed of the experimental scenario and positioned about 60cm from a 20” flat screen monitor (screen width of 44cm) at a resolution of 1680x1050. The game environment was rendered in real-time at a 60Hz constant refresh rate. First person viewing mode was used for navigation. The virtual camera was positioned at the level of the eyes of the subject’s avatar which was 1.80m in height. The avatar had three degrees of displacement freedom. Yaw and pitch angles of the camera were controlled with the mouse, while walking was controlled with the arrow keys of the keyboard. Task completion time as well as inspection start/end timings indicated by a mouse over a possible clue, collect attempts, collected (decisive or not) objects were stored in a database along participants’ age and gender.

4.2.2.2 Results

We subjected the completion times to a Multiple Linear Regression (MLR) analysis which, like the ANOVA, is a subclass of *general linear modelling*. Unlike the ANOVA, a linear regression also provides an explicit, quantitative model of how the different experimental factors affect performance along with the relative importance of the different factors [Cunningham and Wallraven, 2011]. This information is critical for deriving the DWM weights.

In MLR, the line $y = m_1 x_1 + \cdots + m_n x_n + b$ is fit to the data, with y being the participants' performance (e.g., task completion time) and each x_i being an experimental factor (e.g., physical or semantic context) and b being the intercept. Since our two factors are categorical, they must be *dummy coded*. We gave Compound a value of 0 and Singleton a value of 1. Likewise, Inconsistent and Consistent were set to 0 and 1, respectively. Each regression coefficient m_i indicates how many seconds faster a unit change (i.e., from 0 to 1) in the factor x_i will cause the completion time to be. Critically, the ratio of the mean squared prediction error of a model to the variance in completion time is directly related to the Pearson correlation coefficient [Cunningham and Wallraven, 2011] and indicates how much of the variance in completion time can be "explained" or predicted by the change in the independent variables. We will use this relative predictive values to derive the DWM weights.

Experiment 2: Search Task On average, participants needed 64.81, 72.10, 135.03 and 164.5 seconds to complete the Singleton/Consistent, Singleton/Inconsistent, Compound/Consistent, and Compound/Inconsistent conditions, respectively (see Figure 4.4). Regressing physical context onto completion time yields a model that explains 80.7% of the variation in completion time. This is a significant amount, $F_{1,38} = 159.1, p < .001$, showing the significant effect of physical context. There was also a significant effect of semantic context: a two predictor model regressing both physical and semantic context onto completion time explains 84.8%

of the variance. This increase in predictive power of 4.1% is statistically significant, $F_{1,37} = 10.068, p < 0.0031$. Finally, the interaction between physical and semantic context was marginally significant: adding a term to capture the variance jointly explained by semantic and physical context – while controlling for multicollinearity – explains an additional 1.5%, $F_{1,37} = 3.9621, p < 0.055$. The intercept, regression coefficients and statistical significance of each predictor in the two and three predictor models can be seen in Table 4.1. As can be seen in the table, the two predictor model predicts that performance in the Compound/Inconsistent condition should be 158.962 (the intercept) which is close to the actual value of 164.5. Changing from compound to singleton should speed up performance by 81.309 seconds (the regression coefficient for physical context), and changing from inconsistent to consistent should speed up performance by 18.381 seconds. Thus, performance in the Singleton/Consistent condition is predicted to be 59.272, which matches the actual value of 64.81 well.

Experiment 3: Non-Search Task On average, participants needed 89.74, 94, 173.05 and 144.90 seconds to complete the Singleton/Consistent, Singleton/Inconsistent, Compound/Consistent, and Compound/Inconsistent conditions, respectively (see Figure 4.5). The effect of physical context was again significant; a single predictor model explains 77.7% of the variance, a statistically significant amount, $F_{1,38} = 132.1, p < .001$. Semantic context was also significant; the two predictor model explained 80.2% of the variance, a statistically significant increase of 2.5%, $F_{1,37} = 4.578, p < 0.04$. The interaction was also significant; the three predictor model explains 84.7% of the variance, an increase of 4.5%, $F_{1,37} = 3.258, p < 0.003$. The intercepts and significance of the three predictors can be seen in Table 4.2.

Coefficients	Estimate Time	p-value
Intercept	158.962	< 0.0001
+Singleton placement	-81.309	< 0.0001
+Consistent placement	-18.381	0.003
+Joint Term	22.190	0.055

Table 4.1: The regression coefficients and their significance on the overall model, for the case of a Search task

Coefficients	Estimate Time	p-value
Intercept	153.008	< 0.0001
+Singleton placement	-67.111	< 0.0001
+Consistent placement	11.944	0.039
+Joint Term	-32.407	0.025

Table 4.2: The regression coefficients and their significance on the overall model, for the case of a Non-Search task

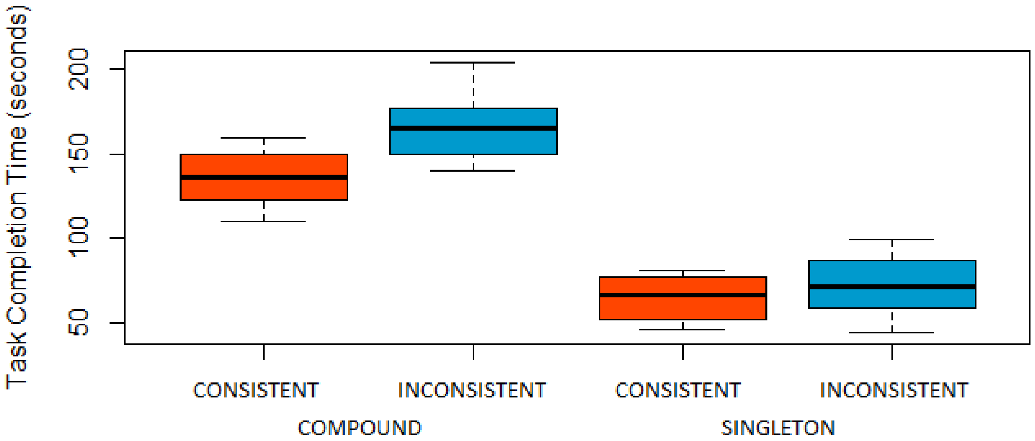


Figure 4.4: Task completion time distribution in a Search task. The thick, horizontal line in each box represents the median for that condition. The colored box around the median represents the middle quartiles and the outer bars represent the extremes.

4.2.3 Discussion

Both semantic and physical context play a statistically significant role in attention deployment, with physical context playing the dominant role. Moreover, an object is often inconsistent with its surroundings (and thus will probably grab attention) but neither in a singleton state nor salient in terms of low level features. In such cases, the scene schemata theory can predict its prominence. In agreement with our first prediction, placing an object in a singleton state decreased task completion time. The two predictor model indicates that performance in the singleton conditions is about 49% of that in the compound conditions for Search tasks, and about 59% for Non-Search tasks. In agreement with the first part of our second prediction, consistency decreases task completion time for a Search task. The significant interaction for Non-Search tasks, however, means that the effects

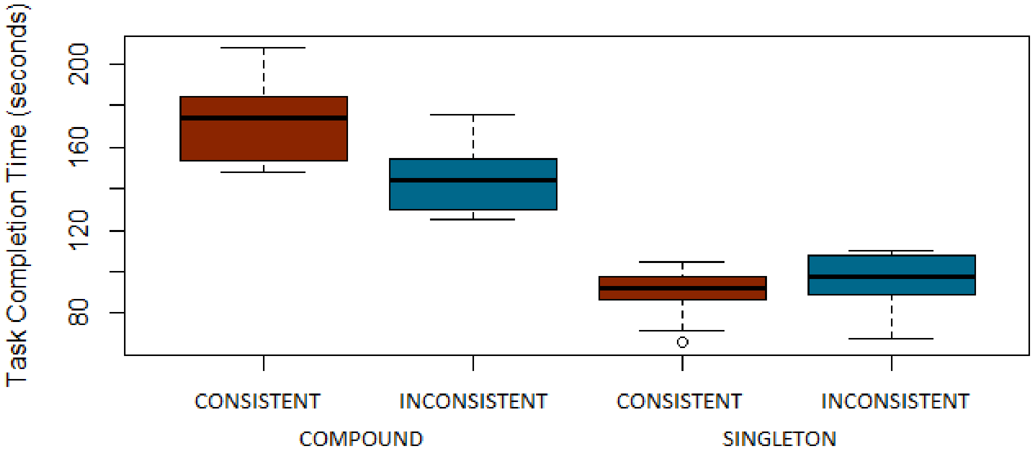


FIGURE 4.5: Task completion time distribution in a Non-Search task (the small circle visualizes an outlier's completion time).

of semantic was dependent upon physical consistency: inconsistent locations were only faster for compound objects. Contrary to prediction, inconsistency increased search time in a Non-Search task for singleton objects.

4.2.4 Model Initialization

We used the results of experiments 2 & 3 to derive weighting factors w_j for each dimension. In a Search task, a two predictor model explained 84.8% of the variance, with object singletoness explaining 80.7% and schema consistency 4.1%. Thus, $w_{physicalSEARCH} = 0.95$ (80.7% out of 84.8%) and $w_{semanticSEARCH} = 0.05$. In a Non-Search task, object singletoness explained 77.7% of the total 80.2%, giving us $w_{physicalSEARCH} = 0.97$ and $w_{semanticSEARCH} = 0.03$.

In order to calculate the likelihood values associated to the scene schema hypothesis we compare the associated scene schema of each examined object determined in Experiment 1 against the scene schemata associated with the objects that surround it. We define an object neighborhood of radius N as a multiple of the examined object's radius. We define c the count of all objects residing in this neighborhood and m the count of objects tagged with the same schema as the examined object inside the neighborhood.

We then define $\lambda_{semantic}$ as:

$$\lambda_{semantic} = \frac{c - m}{c} \tag{4.7}$$

Inconsistent objects signified by their varied schema relatively to their surroundings have greater $\lambda_{semantic}$ values than consistent objects.

In order to calculate the likelihood values associated to the singleton hypothesis, we both examine the number of neighbours for each examined object and employ the available image depth information. In particular, we can use the spatial derivatives to estimate the magnitude of the depth gradient. This operator indicates how distinct an object is from its environment and is a strong indication of whether it is a singleton.

We thus define $\lambda_{physical}$ as:

$$\lambda_{physical} = \frac{1}{|1 - c|} \times \sqrt{(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2} \tag{4.8}$$

Always $c > 1$.

4.3 Implementation and Game Balancing

In this section we describe a GPU implementation of our model and its integration in a game engine to assess game level difficulty. The efficiency of our model in predicting attention deployment is evaluated in Experiments 4 & 5.

4.3.1 GPU based Implementation

We developed a plug-in for Unity 3D$^{\text{TM}}$ game engine which we call High Level Saliency Modeler (HLSM). HLSM highlights objects expected to attract attention by estimating in real time the posterior probability term (Equation 4.6) of our new high level attention model in a pixel shader. Equations 4.1 - 4.5 are supplied with semantic consistency and object singletoness information in terms of the $\lambda_{physical}$ and $\lambda_{semantic}$ variables as determined by the experiments. The $\lambda_{semantic}$ (Equation 4.7) and $\lambda_{physical}$ (Equation 4.8) are calculated at runtime by both querying the scene graph and utilizing an edge detection kernel run over the depth buffer. The obtained likelihood ratio sums are then combined according to the w_j factors obtained from the regression analysis applied to the experimental task completion timings (Section 4.2.4). The d', σ and K values are user controlled via the system's user interface. Manipulating these parameters either increases or decreases the system's sensitivity to saliency resulting in more or fewer objects to be highlighted as salient respectively (Figure 4.7).

The pixel shader approach offers view-dependent estimations i.e. an object may or may not appear as singleton depending on the viewpoint. Additionally, the linearity of the likelihoods calculated allows for linear quantitative measurements. For instance, "an object x is more inconsistent than object z by a factor of q". This offers rich information about the semantic context of objects as opposed to the previously defined binary definition of an object being characterized as either consistent or inconsistent [Zotos et al., 2009].

4.3.2 Game Level Editing

Game balancing is a meaningful application of high level saliency modeling. Plot-critical objects are placed in their respective locations by game designers to achieve a purpose: Ease or make it difficult for the player when searching for them depending on the plot. Placing objects far from expected locations is standard in game balancing [Feil and Scattergood, 2005]. Integration of a high level saliency model in a game level editor can assist the level artists by highlighting salient objects. Designers using the proposed editor are able to reposition or tint props to make them less/more visible in real-time. This way, designers modulate the search-cost/benefit curve for easier or harder object recovery in Adventure or Action-Adventure games. When working with our tool, the game level designer proceeds as normal to place game objects as desired. The designer observes saliency visualization and examines the attention prediction for the current view. The current view or object placement may then be modified and high level saliency can be re-assessed in real-time. The overhead of investigating the attention predictions is minimal since the game level designer may save on time by not needing to elaborate on suitable locations for prop placement depending on the current game difficulty level that is developed. Our plug-in works in parallel with the editor, allowing the game designer to play-test the level while designing it.

4.3.3 Experiments 4 & 5: Evaluation of the Implementation

We designed an experiment to evaluate the efficiency of our model in predicting attention deployment by examining its effect on task completion time and by acquiring eye-tracking data. Since our tool is intended to be used by game level designers when creating game levels, the evaluation also indicates the model's potential as a means to adjust game level difficulty.

4.3.3.1 Design

We created four game levels corresponding to two experimental conditions (*Easy* / *Hard*) of a Search and a Non-Search Task. The placement of three critical objects was manipulated to systematically alter game difficulty. Our model implementation (HLSM) assisted object placement by highlighting objects that were expected to pop out in a Search task for the first two conditions (*Experiment 4*) and in a Non-Search task for the last two conditions (*Experiment 5*). Figure 4.6 shows a vase at a consistent/singleton layout expected to attract attention in a Search Task and thus marked as red by HLSM. When the vase is placed on the chair therefore being at an inconsistent/compound location it is not expected to pop-up in a Search Task. When objects pop out we expect a shorter task completion time, thus the easy level for the Search task was created by placing consistent objects at a singleton state in the scene. A hard game level expected to be completed slower was created by placing inconsistent objects at a compound state (Table 4.1). In relation to the Non-Search task the easy level was created by placing consistent objects at a singleton state and the hard level by placing consistent objects at a compound state expected to have the fastest/slowest recovery times respectively (Table 4.2). In all cases we use our saliency modeler, which indicates the appropriate configurations. We used the Saliency Toolbox [Walther and Koch, 2006] to ensure that the requested objects exhibited a minimum low level saliency (Figure 4.8). Constant navigation time to the individual objects was maintained regardless of location. Similar visual angles within the VE were maintained for all objects.

4.3.3.2 Participants and Apparatus

Forty participants (34 male, 6 female; mean age 23) were split in four groups; 10 played the easy Search task level, 10 played the hard Search task level, 10 played the easy Non-Search task level and the rest played the hard Non-Search task level. For the Search task participants were instructed to find three specific objects. For the Non-Search task participants observed the VE to identify three unknown objects that were indirectly described: “identify objects necessary for a car trip” (Figure 4.9). For both tasks participants were instructed to find the objects as quickly and as accurately as they could. Each subject participated in only one of the experimental conditions. The VEs were presented in stereo at SXGA resolution on an NVIS nVisor SX111 Head Mounted Display with a Field-of-View of 102 degrees horizontal. An InterSense InertiaCube3, three degrees of freedom head tracker was utilized for rotation and a game-pad for translation. Attached to the HMD was an eye-tracker by Arrington Research reconstructing the subject’s eye position through the Pupil-Center and Corneal-Reflection method at a rate of 30Hz. The eye tracking was performed to the dominant eye of each subject.

4.3.3.3 Completion Time Analysis

Experiment 4: Search Task An independent-samples t-test was conducted, revealing a significant difference between easy (M=42.83, SD=11.83) and hard (M=82.2, SD=21.88) level completion times, $t(9) = -4.54$, $p < 0.0001$. The easy task completion time was reduced to 52.1% of the hard task; 42.83 vs 82.2 seconds, that is consistent with the results of the regression analyses of Experiment 2: A consistent/singleton object placement is predicted to be reduced to 37% of an inconsistent/compound object placement completion time derived from 59.272 (intercept+singleton+consistency terms) vs 158.962 seconds (Table 4.1).

Experiment 5: Non-Search Task An independent-samples t-test was conducted, revealing a significant difference between easy (M=61.86, SD=17.57) and hard (M=138.35, SD=16.1) level completion times, $t(9) = -14.48$, $p < 0.0001$. The easy task completion time was reduced to 44.7% of the hard task; 61.86 vs 138.35 seconds, that is consistent with the results of the regression analyses of

Experiment 3: A consistent/singleton object placement is predicted to be reduced to 39.67% of a consistent/compound object placement completion time derived from 65.434 (intercept+singleton+consistent+joint terms) vs 164.952 seconds (Table 4.2).

The reduction of task completion time in the easy conditions when compared to the hard conditions for both the Search and Non-Search tasks validate our hypothesis that game level completion time depends on object topology as predicted by our system.

4.3.3.4 Eye-tracking Data Analysis

For every object in quest a Region-Of-Interest (ROI) was defined. Each ROI held meta-data indicating a consistent/inconsistent placement and a singleton/-compound placement of the object in relation to its surroundings. In total 9837 fixations to the ROIs were recorded. As a fixation we considered every spatially stable gaze lasting for at least 300 milliseconds [Salvucci and Goldberg, 2000].

For **Experiment 4** an independent-samples t-test was conducted on total object fixations per condition, revealing a significant difference between consistent/singleton (M=265.3, SD=15.41) and inconsistent/compound (M=182.6, SD=25.16) object placement, $t(9) = 7.45$, $p < 0.0001$.

For **Experiment 5** an independent-samples t-test was conducted on total object fixations per condition, revealing a significant difference between consistent/singleton (M=364.5, SD=44.92) and consistent/compound (M=171.3, SD=19.04) object placement, $t(9) = 15.6$, $p < 0.0001$.

The results indicate a clear influence of context consistency in attention deployment for the Search Task. Singleton objects attracted attention in both conditions since the total number of fixations for ROIs defined for objects in a singleton state was higher for both the Search and Non-Search tasks. We aggregated fixations collected over raw eye data from all participants and visual angles in multiple heat-maps (Figure 4.9). Observing the heat-maps indicated that in a Search task eye gaze is directed significantly more often to consistent locations in relation to the requested object (Figure 4.9). In a Non-Search task the eye scan pattern spans over the entire scene, which is consistent with previous literature stating that in an Action-Adventure game players mostly explore the entire screen for game props to advance the game-play [El-Nasr and Yan, 2006] (Figure 4.9).

Our model implementation successfully predicts the saliency of objects (Figure 4.6) that were identified as non-salient in terms of low level features (Figure 4.8) further validated by the eye-tracking study (Figure 4.9). Adjusting game level difficulty by manipulating object topology is thus feasible in Adventure or Action-Adventure games.

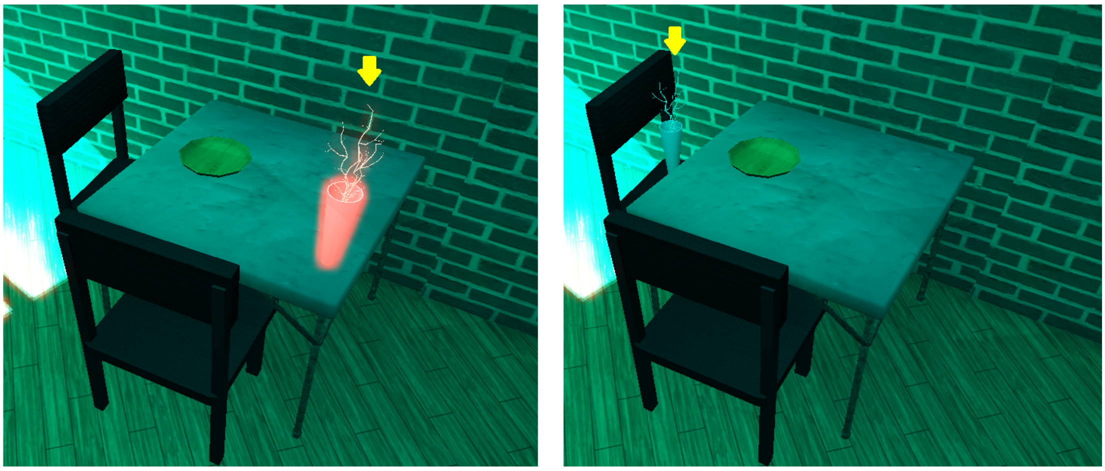


FIGURE 4.6: In a Search task, our tool highlights the vase at a consistent/singleton location signifying an easier recovery than at an inconsistent/compound location (on chair). The green hue indicates non-salient areas.

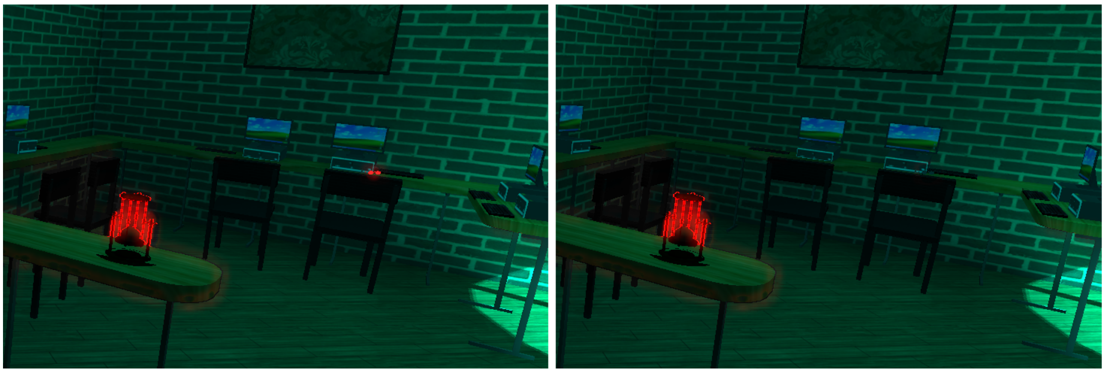


FIGURE 4.7: The system's sensitivity to saliency can be adjusted, resulting in more (left, water-clock & spectacles) or fewer (right, only water-clock) objects to be highlighted as salient.

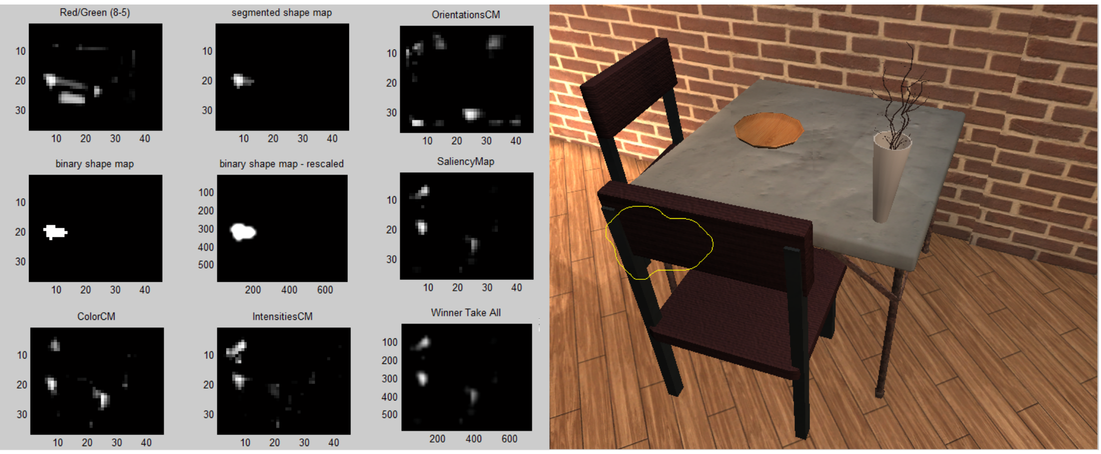


FIGURE 4.8: The Saliency Toolbox [Walther and Koch, 2006] indicates that the most salient area of the image is the dark area behind the chair.

4.4 Chapter Summary

This chapter presents a first attempt to devise a high level saliency predictor based on the topological relationships of objects with their surroundings and object-scene schema conformance for common tasks in (Action-)Adventure games. The framework automatically estimates attention deployment by identifying salient regions in the viewpoint. We conducted three experiments to verify that high level saliency of objects affects the time needed to find them in a VE and also obtained all the necessary weighting factors for our model [Koulieris et al., 2014a][Koulieris et al., 2014c].

Then we developed a GPU based computational model that implements our new

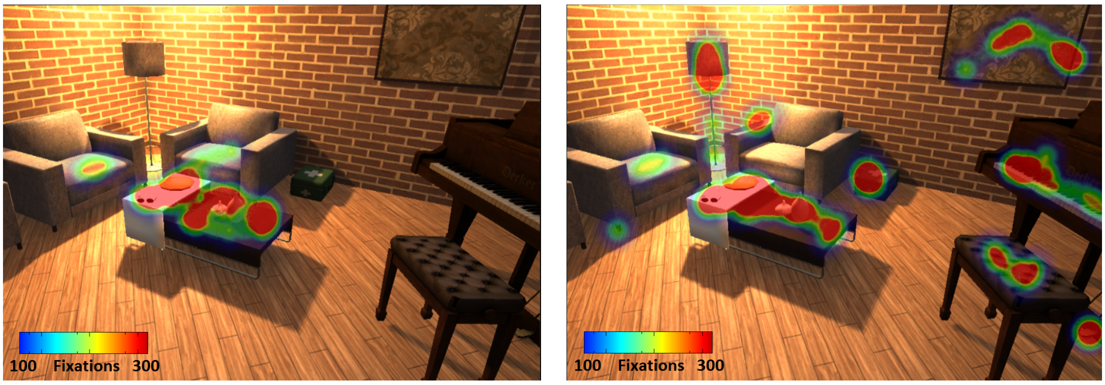


Figure 4.9: The left image indicates fixations for a Search task where subjects were requested to find a pair of spectacles. The right image indicates fixations for a Non-Search task where subjects were requested to identify objects necessary for a car trip. Areas receiving less than 100 fixations are excluded to eliminate noise.

model incorporating high level saliency components. The system estimates the probabilities of individual objects to be foveated in real time and can be used in an innovative game level editor automatically suggesting game objects' positioning in order to adjust the difficulty of the game. The system can be adapted to additional tasks, different than the ones presented here by acquiring the necessary parameters using the methodology we presented.

In the following Chapter we extend this model with additional high level saliency factors and develop a gaze-aware LOD manager based on the extended model.

Chapter 5

Context-Aware Level-of-Detail for Mobile Devices

In this chapter we present additional studies that extend the HLSM with four additional components: (i) an object-context singletoness factor that traces contextual object isolation, (ii) an object-intrinsic cognitive factor, termed canonical form of objects [Becker et al., 2007], (iii) a biologically motivated feature uniqueness factor [Frintrop et al., 2010] and (iv) a factor for temporal object coherence.

The first two new factors (contextual isolation and canonical form) are incorporated through two new high level sensory units. To account for feature uniqueness, equations determine the number of local maxima found in the probability output of a sensory unit. That is, the more maxima there are, the less unique a feature is. For example, if there is only a single violation of canonical form, its uniqueness weight is high. If several violations exist, all violations are less unique. Recurring fixations to areas containing canonical form violations or schema inconsistencies are generated by multiplying a unit's current output with a number of logarithmically attenuated previous outputs. We employ the model for LOD management on mobile devices. [1]

[1] The contributions in this chapter were published in the Eurographics' Association Computer Graphics Forum Journal [Koulieris et al., 2014b], and presented at Eurographics Symposium

5.1 Extending HLSM

For the basic equations of the HLSM please consult Chapter 4.

The probability output P of all units is multiplied with a feature uniqueness weight:

$$w_{unit}^{unq} = \frac{1}{|\vee|P_{unit,x,y,f}} \tag{5.1}$$

x, y denotes image location, f denotes frame number, $|\vee|$ the number of posterior probability local maxima estimated using the GPU (Section 5.4.2) (Figure 5.1).

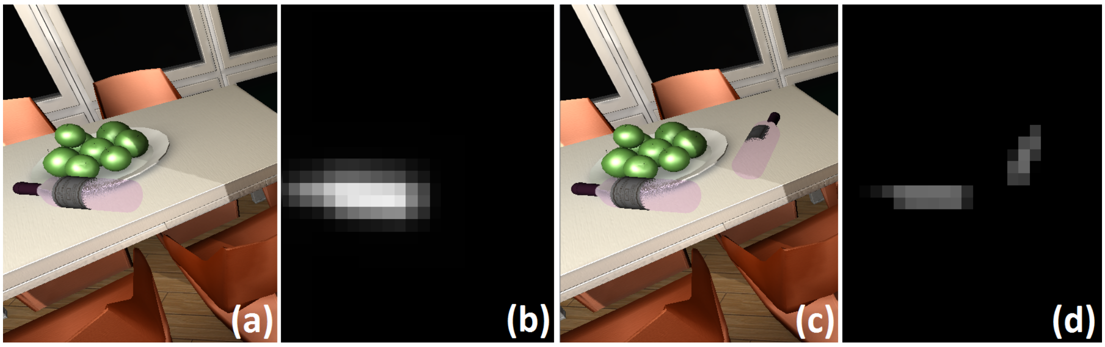


FIGURE 5.1: A single violation of canonical form in the FOV (a) provokes a response in the canonical form sensory unit (b). When more violations of canonical form exist (c) the sensory unit's output is attenuated (d).

The output of the schema consistency unit and the canonical form unit are also multiplied with a temporal context weight:

$$w_{unit,x,y,f}^{tmp} = \prod_{f=1}^{F} P_{unit,x,y,f}e^{-af} \tag{5.2}$$

F the number of previous frames examined, a is a user-defined attenuation factor (Figure 5.2).

The posterior probability $\mathbf{P}_{x,y,f}$ that an observer attends an image location, as part of our enhanced model, is linearly estimated [Frintrop et al., 2010] from both the semantic consistency *(sem)* and physical isolation *(phy)* units defined in Chapter 4

on Rendering 2014 and ACM SIGGRAPH winning 3rd place at the ACM Graduate Student Research Competition [Koulieris et al., 2014d].

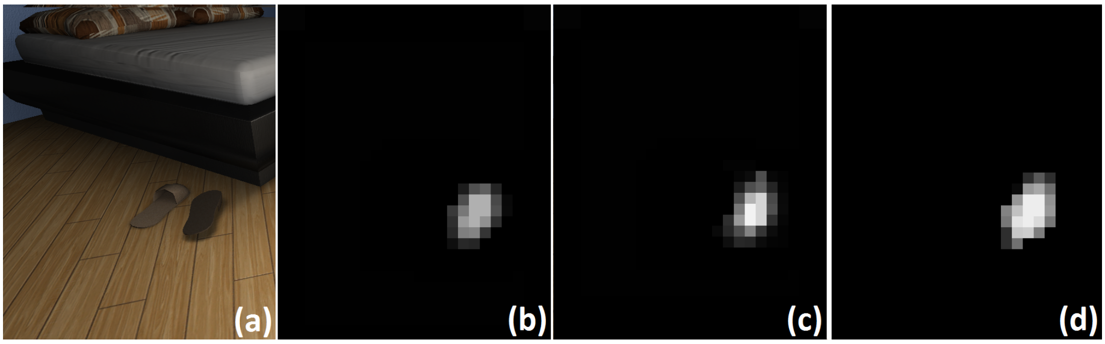


FIGURE 5.2: The slipper on the right is in a non-canonical form (a). The output of the canonical form unit is shown in the current frame (b), in a subsequent frame (c) and in a third frame after the first (d). The increasing probability will generate recurring fixations for our model.

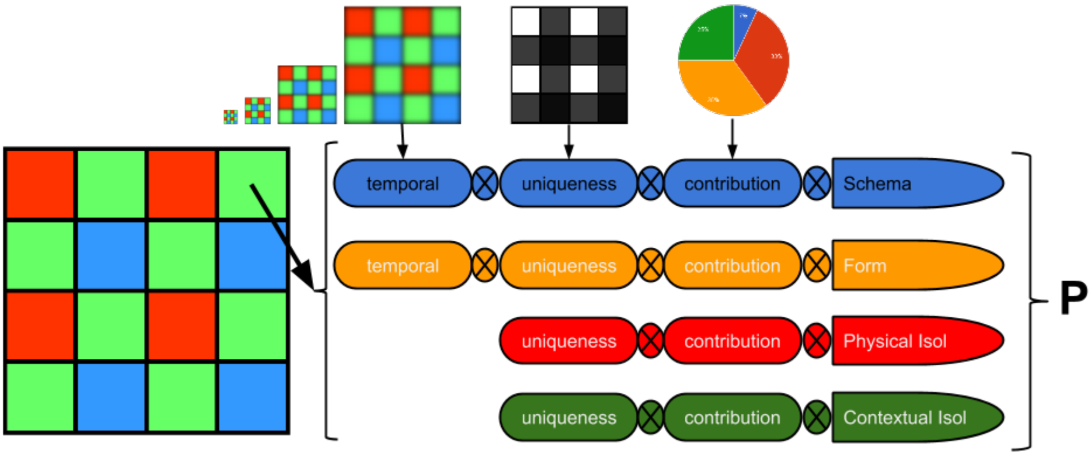


FIGURE 5.3: The posterior probability $\mathbf{P}_{x,y,f}$ term of the integrated model.

combined with the novel contextual isolation *(cnt)* and canonical form *(cfr)* units, updated for feature uniqueness and temporal context (Figure 5.3):

$$\begin{aligned}\mathbf{P}_{x,y,f} = {} & w_{sem} w_{sem}^{unq} w_{sem,x,y,f}^{tmp} \mathbf{P}_{\mathbf{sem},x,y,f} + w_{phy} w_{phy}^{unq} \mathbf{P}_{\mathbf{phy},x,y,f} \\ & + w_{cnt} w_{cnt}^{unq} \mathbf{P}_{\mathbf{cnt},x,y,f} + w_{cfr} w_{cfr}^{unq} w_{cfr,x,y,f}^{tmp} \mathbf{P}_{\mathbf{cfr},x,y,f} \quad (5.3)\end{aligned}$$

In Section 5.3 the contribution weights w_{sem} and w_{phy} that were estimated in Chapter 4 are adapted to our model and the weights w_{cnt} and w_{cfr} are estimated based on additional experimental data.

5.2 Perceptual Study

We conducted a perceptual experiment using a *Search* task to be comparable to the findings of Chapter 4. We thus: (i) examine the effect of violations of canonical form and contextual singletoness on visual attention and (ii) obtain contribution weights of each factor for our model.

Stimuli We factorially combined the two factors to control the spatial arrangement of three objects (a tablet computer, a pair of spectacles and a remote control; see Figure 5.4) in four VEs. The four scenes were contextually compound/canonical, contextually compound/non-canonical, contextually singleton/canonical, or contextually singleton/non-canonical (Figure 5.5). All objects were consistent with the scenes and were physically compound. The Saliency Toolbox [Walther and Koch, 2006] (Figure 5.6) was used to ensure that the three objects had a minimum low-level saliency.

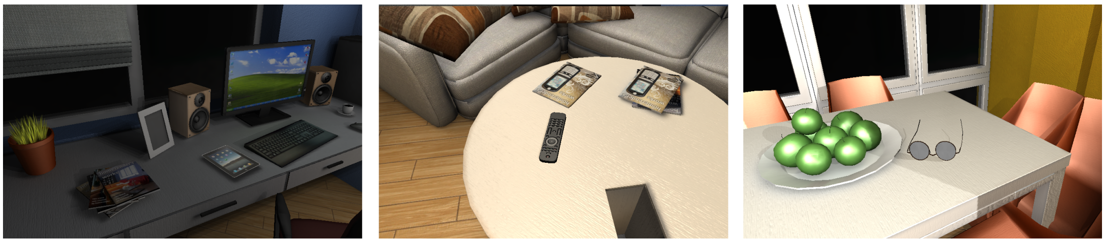


FIGURE 5.4: The subjects searched for three objects, a tablet computer, a remote control and a pair of spectacles.

Participants Forty-eight people participated (8 female, mean age 23) in the experiment, with 12 people being assigned to each of the 4 conditions.

Apparatus The stimuli were displayed on a nVisor^{TM} SX111 HMD, which has stereo SXGA resolution and a FOV of 102 degrees horizontal by 64 degrees vertical. Participants moved through the VE using a game-pad for translation and an InterSense^{TM} InertiaCube3^{TM} 3DoF head tracker for rotation. Navigation was restricted to -70/70 degrees vertically. Eye tracking information was recorded using a twin-CCD binocular eye-tracker by Arrington Research^{TM}, which was attached to the HMD. The eye tracker was updated at a frequency of 30Hz.

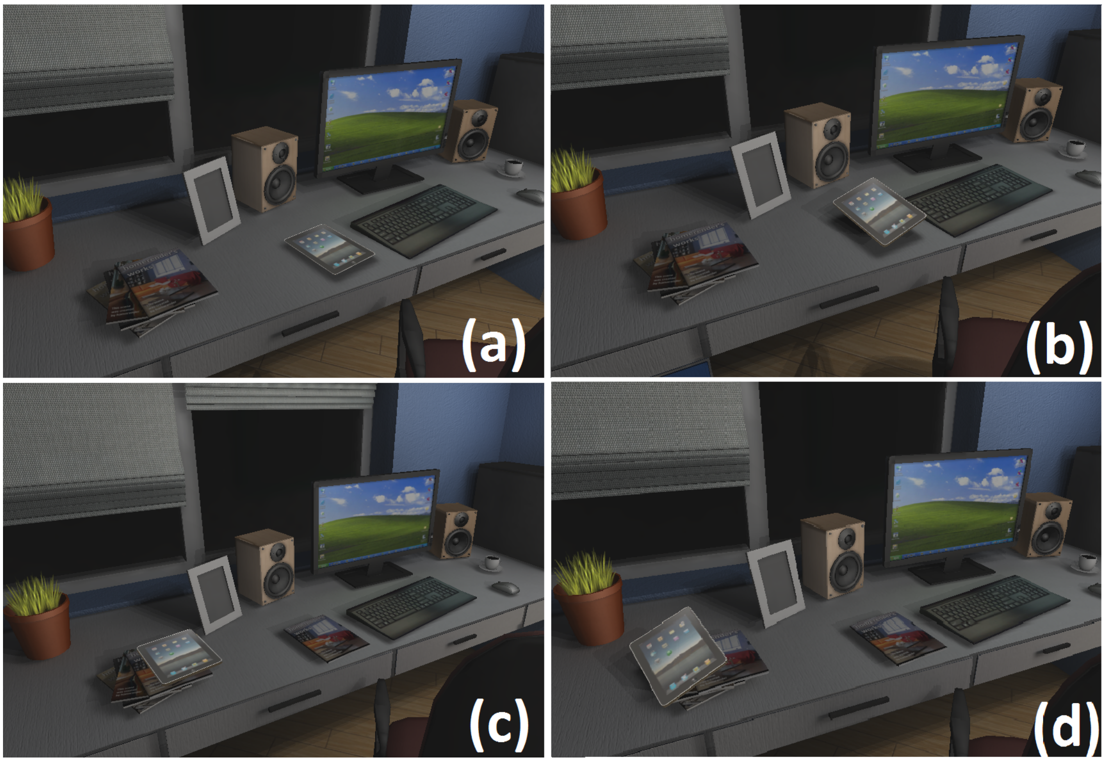


FIGURE 5.5: The tablet is in a (a) contextually compound canonical form (tablet and keyboard), (b) contextually compound non-canonical form (slanted), (c) contextually singleton canonical form and (d) contextually singleton non-canonical form.

Procedure Participants sat on a swivel chair and were familiarized with the setup in a training session. They were then asked to navigate around the scene in order to find and collect all three objects. Task accuracy, completion time, and eye-tracking data were recorded.

Results Task accuracy was always 100%. On average, participants needed 167.788, 255.386, 82.189, and 195.985 seconds for the compound/canonical, compound/non-canonical, singleton/canonical, and singleton/non-canonical conditions, respectively (Figure 5.7). Task completion times were analyzed with a linear Hierarchical Multiple Regression analysis (HMR) with contextual singletoness being entered at stage one and canonical form at stage two. HMR fits a linear model to the data, with one term for each factor. The weight associated with each term is related to the correlation coefficient between the dependent variable (here, completion time) and the different factors. This effectively describes how well

Coefficients	Estimate Time	p-value
Intercept	161.238	< 0.0001
+Non-Canonical Form term	100.697	< 0.0001
+Singleton Placement term	-72.500	< 0.0001

Table 5.1: The regressions coefficients for each factor.

changes in the measured data can be explained or predicted by changes in the factors. Contextual singletoness contributed significantly to the regression model, $F(1, 46) = 16.83, p < .001$ and accounted for 26.79% of the variation in task completion time. Introducing canonical form explained an additional 51.68%, $F(2, 45) = 82.03, p < .001$, for a total explained variance of 78.47%. The coefficients for the two factors can be seen in Table 5.1. Predictions for a condition can be obtained by combining the intercept (i.e., performance in the compound/-canonical condition) with the appropriate modifiers (i.e., the non-canonical form and/or singleton terms; see Table 5.1). The predictions of the model are consistent with the actual recorded completion times.

An analysis of the eye-tracking ROIs showed that attention is indeed attracted both to contextually singleton objects and to objects in a non-canonical form.

Discussion The canonical form and contextual isolation of objects play a significant role in attention deployment. In particular, in the non-canonical form conditions the objects were actively observed despite the fact that their recognition was extremely slow when compared to the canonical form condition. This is apparently in contradiction with the findings of Chapter 4 indicating that actively attended salient objects are easy to find. Thus, when managing LOD, an object in non-canonical form is salient and should always be rendered in high quality.

5.3 Weight Generation

In Chapter 4 the model weights were derived from the correlation coefficients by dividing the amount of variance that a factor explained by the total explained

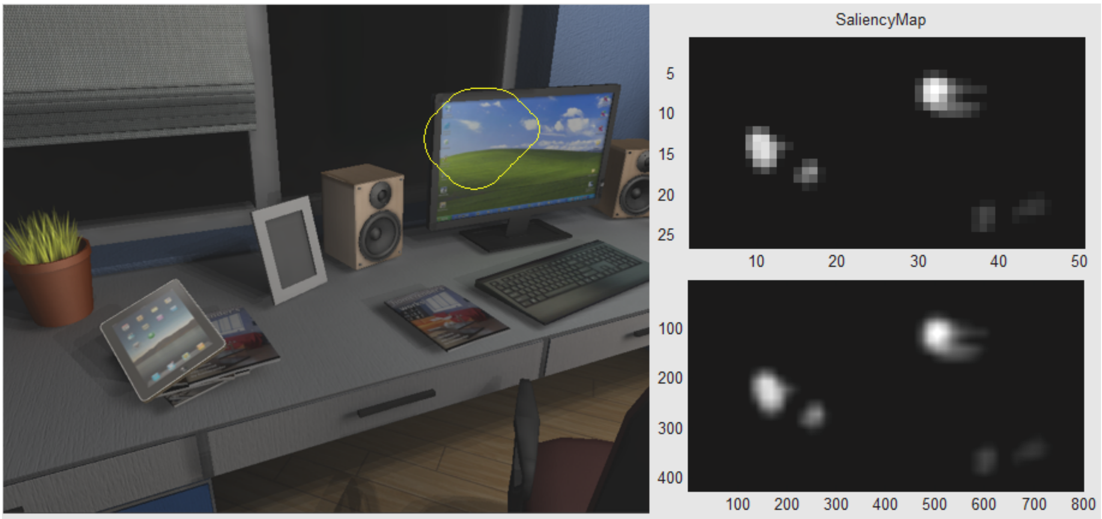


FIGURE 5.6: The yellow contour delineates the most salient region of the image as predicted by the Saliency Toolbox [Walther and Koch, 2006]. Our hypothesis is that the tablet in a non-canonical form is the most salient object in this image.

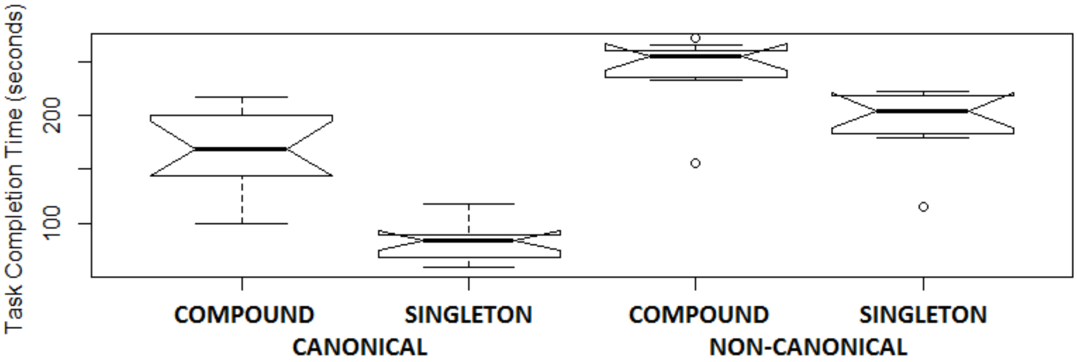


FIGURE 5.7: Task completion time distribution of the experimental conditions. The median value for each condition is depicted by the horizontal line. The notched boxes depict the middle quartiles. The outer bars represent the extremes for each case. The circles visualize outliers' completion times.

variance. Since a single, between-participants experiment using a factorial combination of all levels of all four factors does not exist (it would require a prohibitively large number of participants), it is not possible to directly determine the *relative* amount of variance each factor explains.

Thus the w_{sem}, w_{phy}, w_{cnt} and w_{cfr} weights are estimated by re-calibrating and merging the data from Chapter 4 with the newly acquired experimental results [Cunningham and Wallraven, 2011]. The relative increase in task completion time for each condition will be used to generate the weights for each factor. Note that in our experiment all object placements were contextually consistent and physically compound. That is, the factors of Chapter 4 were held constant. Since these factors were the same in all conditions, the difference in completion time between conditions can not have been influenced by these factors. Likewise, in Chapter 4 all objects were in a canonical form. When an object was physically compound it was also always also contextually compound. Again, since these two factors did not differ between conditions, they can not have influenced the differences in completion times between conditions.

In addition, there is a baseline condition that is the same in both experiments. The physically compound/consistent condition in Chapter 4 is identical to our contextually compound/canonical condition. To re-calibrate the data, we divide the actual means of the conditions examined in Chapter 4 by the predicted value for this baseline condition (140.581). The means for the singleton/consistent, singleton/inconsistent, compound/consistent, and compound/inconsistent conditions, were 64.81, 72.10, 135.03 and 164.5, respectively. After normalization, these values become 0.46, 0.51, 0.96, and 1.17. By examining how changing from schema-inconsistent to schema-consistent for physically singleton objects $(0.51 - 0.46)$ and for physically compound objects $(1.17 - 0.96)$, we can determine the average effect of schema consistency (0.13). Likewise, the increase due to going from physically compound to physically singleton for consistent objects is $0.96 - 0.46$ and for inconsistent objects is $1.17 - 0.51$, given an average change of 0.58. The total time difference between slowest and fastest conditions $(1.117 - .46)$ is .71. Thus, physical isolation accounts for .58/.71 or 82% of the time change. Weights of .82 and .18 do not vary significantly from .95 and .05 found in Chapter 4.

The completion time for the baseline condition was predicted to be 161.238. Dividing the means in our four conditions (167.788, 255.386, 82.189 and 195.985 for the compound/canonical, compound/non- canonical, singleton/canonical, and singleton/non-canonical conditions, respectively) gives normalized completion times of 1.04, 1.58, 0.51 and 1.22. Thus, the increase due to canonical form for contextually compound objects is $1.58 - 1.04$ and for contextually isolated objects is $1.22 - 0.51$, giving us an average difference of 0.63. Likewise, the decrease due to contextual singletoness is the average of the differences for canonical $(1.04 - 0.51)$ and non-canonical objects $(1.58 - 1.22)$, which is 0.45. We then ensure that all weights sum to one by dividing them by the current sum (1.79).

The final weights are 0.07 for schema, 0.33 for physical isolation, 0.35 for canonical form and 0.25 for contextual isolation.

5.4 LOD for Mobile Graphics

We developed a generic material LOD manager based on attention for Unity 3D$^{\text{TM}}$ game engine that we call Contextual-LOD (C-LOD). C-LOD is a reactive fixed frame rate scheduler [Luebke, 2003] that constantly examines frame rate and attention deployment predictions using the criteria of our model. When frame rate drops below 30 frames per second on fill-rate bound mobile devices, C-LOD automatically lowers the rendering quality of objects predicted not to be attended until performance is restored (Figure 5.11). The highest quality possible is maintained for all attended objects. Our LOD manager adjusts LOD only during player motion. Pop-out artifacts [Luebke, 2003] are eliminated by exploiting the observer insensitivity to perceive changes occurring during a brief interruption known as the CB phenomenon [Simons and Levin, 1997].

5.4.1 C-LOD Effects

C-LOD can manage any effect that has at least two levels of detail. For this proof-of-concept implementation we selected three complex effects that are usually omitted in mobile devices as they require many texture fetches [Çapin et al., 2008].

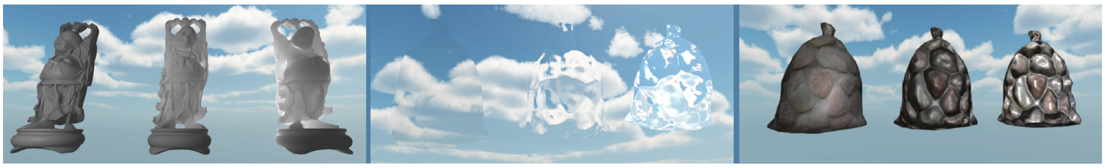


FIGURE 5.8: Left to right: Subsurface scattering, refraction and bump mapping low to high quality.

We used two LOD fall-backs for each effect, that require fewer texture fetches (Figure 5.8).

Subsurface light transport in translucent materials requires intense analytical calculations, making it impossible for mobile devices to render this effect [Jensen et al., 2001]. To simulate the high quality effect, we approximated light transport using a pre-computed map of local thickness for each model calculated by inverting the normals of the model and estimating ambient occlusion with the inverted normals [Barre-Brisebois, 2011]. The medium LOD level substitutes the thickness map with a standard distance-attenuated diffuse lighting combined with the distance-attenuated dot product of the view vector and the inverted light vector. The low quality fall-back is an opaque Blinn-Phong specular pixel shader.

Refraction is a computationally expensive effect for mobile devices. OpenGL ES2.0 devices do not support Multiple Render Targets (MRTs) thus existing methods that estimate refraction for both the front and back interfaces of an object are slow [Wyman, 2005]. Single interface refraction produces convincing results. Single interface refraction with chromatic aberration [Lindholm et al., 2001] was selected as the high level refraction effect. The medium effect removes chromatic aberration by exchanging the wavelength-dependent sampling of the RGB channels with a single lookup, significantly reducing texture fetches by a factor of three. The low quality effect is a uniformly distorted transparent pixel shader (Figure 5.9).

Bump Mapping via tessellation and displacement mapping is not available on OpenGL ES2.0 devices. For high quality bump mapping we incorporated the

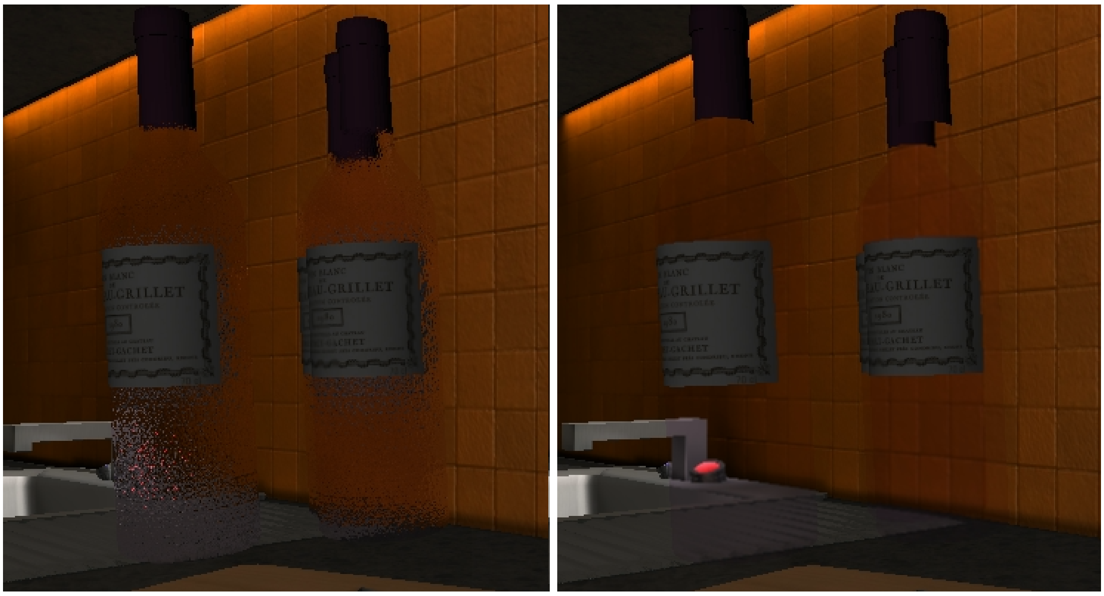


FIGURE 5.9: When rendered with C-LOD (right), the bottles in canonical form which are not expected to attract attention, receive a lower quality but faster refraction shader when compared to an all-high setting (left).

texture-heavy Parallax Occlusion Mapping method [Tatarchuk, 2006]. For the medium quality level effect, we employed simple parallax mapping that does not support self-shadowing [Kaneko et al., 2001]. The low quality is a standard normal mapped shader.

5.4.2 C-LOD Components

The Predictor We implemented our model in the GPU. Our system detects non-canonical object forms by examining object position in relation to the view vector. We utilize object IDs to locate contextually singleton objects. An analytical determination of feature uniqueness would require the calculation of the bi-variate partial derivative of each unit's output. Identifying local maxima in a Gaussian pyramid [Ziegler et al., 2006] is slow on mobile as it uses render buffer ping ponging. We count local maxima by employing an approximation that exploits hardware's linear interpolation capabilities. We render each unit's output in a 4x4 resolution frame buffer object only once each second. By thresholding 16 texel fetches per unit buffer we count up to 16 local maxima competently. We approximate temporal context calculations by storing up to F low resolution previous frame buffer objects and combine them using hardware blending and an 1D ramp texture storing the pre-calculated logarithmically attenuated function (Equation 5.2). We initialized our model equations using the the weights estimated in Section 5.3.

The Texel Engine C-LOD's Texel Engine constantly monitors object predictions derived from our attention model. A special 2D texture is updated that works as a material quality lookup table (Figures 5.10, 5.11). The columns of the texture correspond to all object/material combinations found in a scene and each row represents a LOD for all object/material combinations. A higher row number (bottom rows in the table) signifies a more aggressive simplification overall. Introducing a simplification for object/material combination x in row y imposes that all subsequent rows have the same or lower quality for x. This restriction maintains visual coherence between LODs and induces the smallest possible number of quality reductions. As a result, values over the diagonal of the texture are always the highest (white) signifying the highest quality possible. The system updates the texture once per second in synchronization with camera movement.

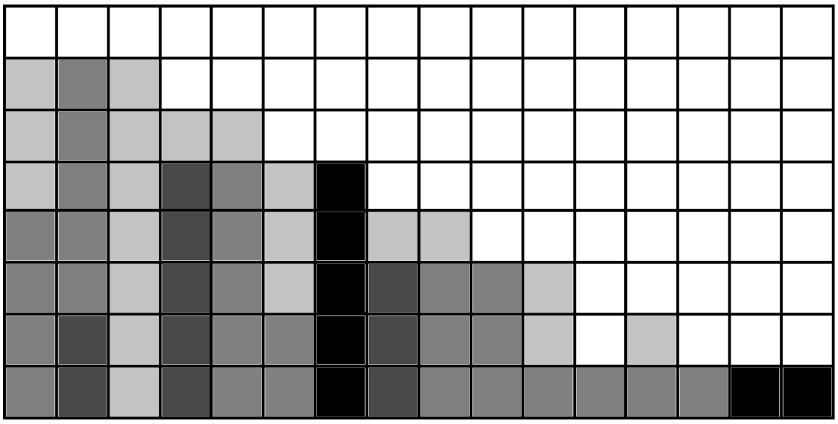


Figure 5.10: The Texel Engine precomputed texture used for communicating LOD selections. The columns of the texture correspond to all object/material combinations found in a scene and each row represents a LOD for all object/material combinations. A higher row number (bottom rows in the table) signifies a more aggressive simplification overall. Darker color denotes a lower visual fidelity LOD

The Bootstrapper The interaction between the graphics processor, CPU and memory of a mobile device is not trivial. When bootstrapping, C-LOD performs

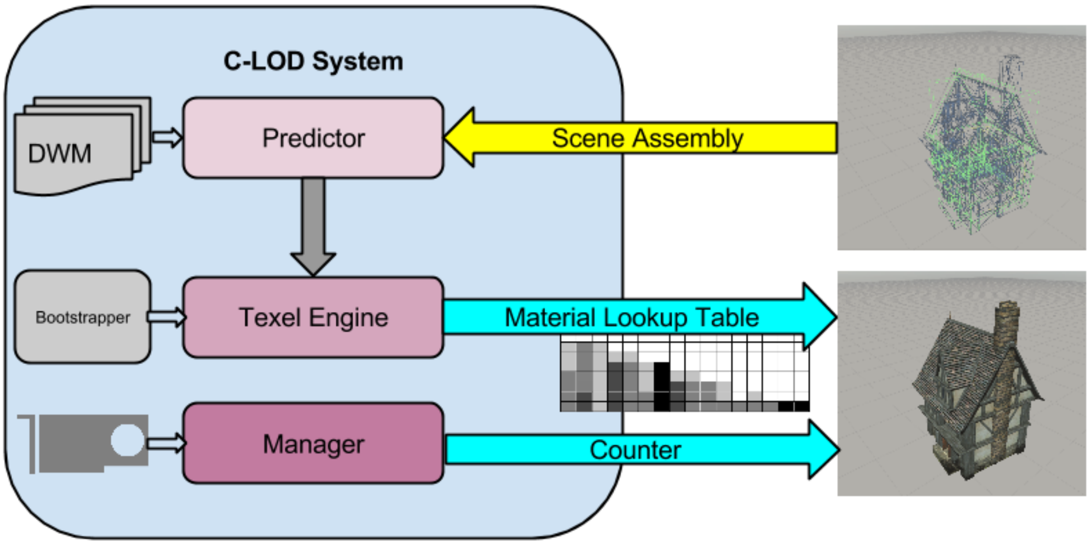


FIGURE 5.11: The C-LOD system architecture.

system profiling. The materials managed are initially rendered at their lowest quality. Then, in rapid succession, the quality level of each object's material is increased while frame rate is monitored. This procedure determines a scale factor that controls the aggressiveness of simplifications by the Texel Engine.

The Manager A Finite State Machine (FSM) monitors frame rate during execution. When frame rate drops and motion is detected, a counter is increased. This counter is communicated to all managed materials. A re-mapped object ID of each object is appointed as the u texture coordinate to sample the look-up table texture and the counter variable as the v texture coordinate. The sampled value is communicated to the fragment shader where it controls a conditional branch that selects the appropriate LOD for the shader or acts as an iteration counter, e.g. for ray marching in the parallax occlusion mapping shader. Updating the counter only when camera moves, reduces luminance offsets and flickering effects. Frame rate is constantly re-evaluated and the counter is increased/decreased to maintain the best LOD for the current conditions (Figure 5.11).

5.5 Evaluation of C-LOD

We evaluated C-LOD's efficacy both via eye tracking and by acquiring GPU performance data on a mobile device. We also measured battery performance.

Model Accuracy To measure the model's accuracy in predicting attention we performed an experiment on the eye-tracked HMD set-up of our lab.

Design To empirically verify that changes in LOD were not perceived and did not affect attention deployment, we rendered a scene consisting of 50k triangles and complex pixel shaders twice. In the first version of the scene (HQ), all effects were set in the highest quality possible. In the second condition (C-LOD) quality was managed by our system. The rendering was performed on a high-end desktop computer to eliminate fluctuations in the frame rate that would have occurred in a tablet device inadvertently affecting attention deployment. The FOV of the HMD was restricted to 40 degrees horizontally and 23 degrees vertically to simulate a 10.1" tablet held at a 30cm observer distance [Slater et al., 2010]. Participants were asked to find and collect seven objects placed in consistent, inconsistent, physically isolated, contextually compound, contextually isolated locations and in a canonical/non-canonical form. In total, 22 people participated (2 female, mean age 22), with 11 people in each of the two conditions.

Results In total, $88,404$ object fixations were recorded for all participants (Figure 5.12). Given that human attention may be directed at multiple foci [Awh and Pashler, 2000], we recorded the three most prominent objects predicted to be fixated by our system for each frame of the simulation. We defined three quantitative estimators to denote the ratio of frames that gaze was allocated in an increasingly larger subset of the predicted objects, to the total number of simulation frames. A baseline R estimator was defined that selects a random object in the FOV for each frame. Both conditions yielded similar results. We summarize the estimators and their results in Table 5.2. In short, the addition of the C-LOD changes

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%

TABLE 5.2: The ratio of frames that the attended object was predicted correctly for the high quality condition, the C-LOD managed condition and in total. E1 denotes that the gazed object matches the first prediction. E2 denotes that the gazed object matches either the first or the second predicted object. E3 denotes that the gazed object matches either the first, or the second or the third object.

did not alter gaze performance, and thus were most likely not perceived by the participants.

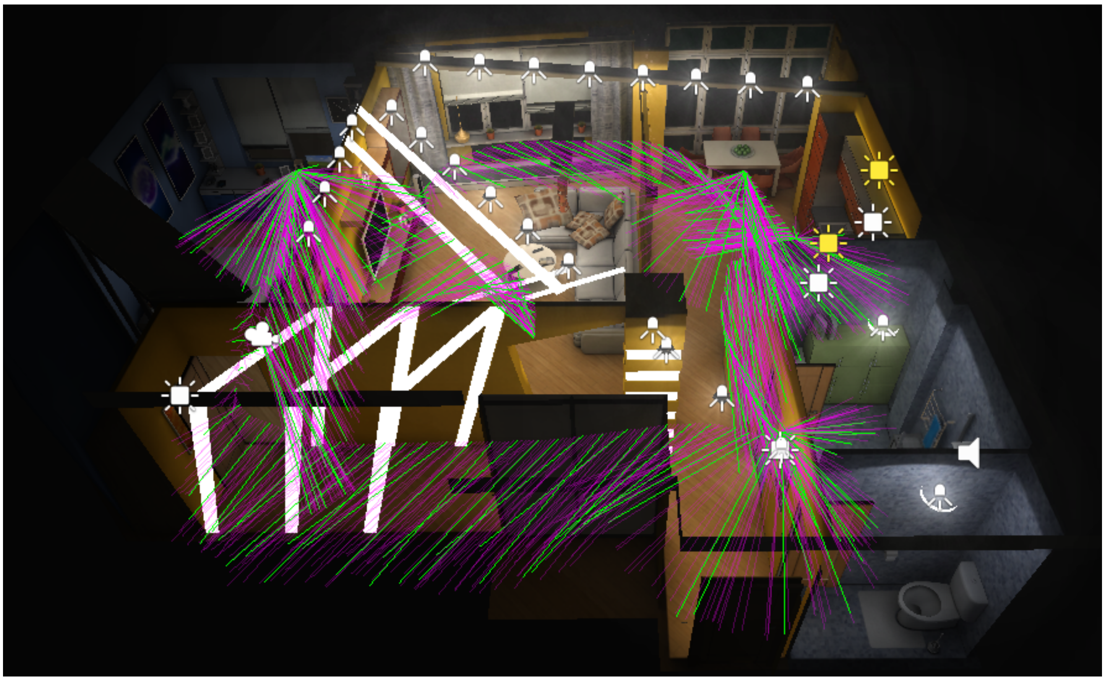


FIGURE 5.12: Our validation tool indicates the subject's gaze point with magenta colored beams. The green beams indicate predictions by our attention model.

Model Efficiency To assess the impact of C-LOD on GPU performance we reconstructed $2,947$ seconds of player motion of both experimental conditions on an Android quad-core Cortex A9 1.6GHz OpenGL ES2.0 mobile device and sampled the frame-rate at a 5Hz rate. A total of $17,681$ frame rate samples were collected. An independent-samples t-test was conducted, revealing a significant difference between the HQ ($M = 24.05, SD = 2.92$) and C-LOD ($M = 25.6, SD = 1.33$) conditions; $t(8, 418) = -44.16, p < 0.0001$. The C-LOD condition exhibits a consistently stabler frame rate and provides a slightly higher mean frame rate when compared to the HQ quality setting (Figure 5.13). The Android Debug Bridge (ADB) and Tracer for OpenGL tools were employed to conduct a deep frame inspection. C-LOD estimations run for 4ms on average per frame. Given the increase in mean frame rate between the two conditions it can be concluded that this cost is amortized between frames.

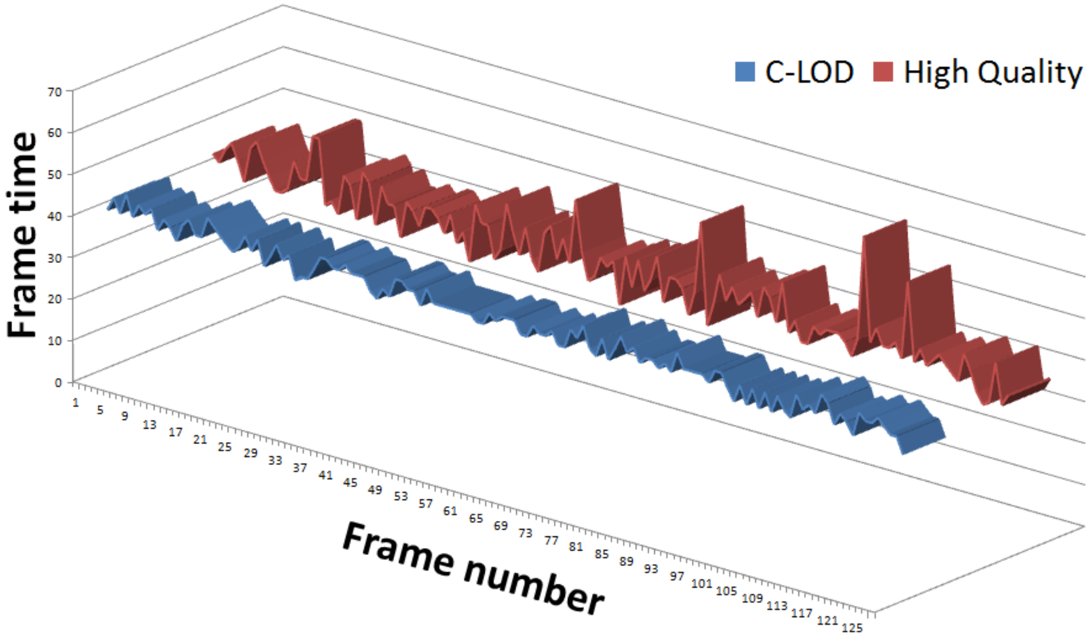


Figure 5.13: Frame time for 128 random sequential frames of the HQ and C-LOD conditions. Notice the intense fluctuation of the frame time in the HQ condition when compared to the C-LOD condition.

Battery life improvement Quering ADB indicated that the battery's average voltage drop was 21mVolts greater for the HQ condition versus the C-LOD managed condition. This indicates an increased discharge rate that was also portrayed in the total run time. Player motion data from the validation experiment were replayed in the HQ and C-LOD settings until battery run out. The C-LOD condition lasted 249 minutes; the HQ condition lasted for 233 minutes.

Discussion Results indicate that C-LOD identifies the observed object 8 times better than a random estimator in the worst case (Table 5.2). For three attended objects prediction rate approaches 90%. This also suggests that quality reductions go mostly unnoticed. Integrating C-LOD in a mobile 3D graphics application stabilizes frame rate without sacrificing perceived quality and boosts battery run time by 6.5% (Figure 5.13).

5.6 Chapter Summary

We presented an extension to the HLSM (presented in Chapter 4) by introducing four novel factors that affect attention deployment: object canonical form, contextual singletoness, feature uniqueness and temporal context. We acquired the parameters to re-initialize our model in a perceptual experiment [Koulieris et al., 2014b][Koulieris et al., 2014d].

We developed a LOD manager for mobile devices that maintains a constant framerate by selecting an appropriate LOD for materials based on attention. We evaluate the performance our algorithm via eye-tracking and by acquiring GPU performance data on mobile devices, confirming that complex effects such as parallax occlusion mapping that are usually omitted in mobile devices can now be employed without exhausting GPU capability. We verified an 6.5% increase in battery life due to less GPU utilization.

In the following Chapter we present our attempt to employ machine learning for gaze prediction, limiting the necessity to manually annotate objects with metadata. We also develop a gaze-aware local disparity manipulation algorithm.

Chapter 6

Gaze Prediction using Machine Learning for Dynamic Stereo Manipulation

Our work presented in previous chapters requires manual pre-processing in terms of tagging to define object semantics and/or high-level scene descriptions (schemas) and is restricted to scenes with static objects. In this section we present our latest method that does not suffer from these restrictions. This novel, machine learning method learns to predict gaze based on game state variables and ground truth eye-tracking data. [1]

6.1 Machine Learning-based Gaze Prediction

Our machine learning approach has three steps: identification of important game variables and object classes, data collection and classifier training. We used the *Realistic First Person Shooter Toolkit, RFPS*$^{\text{TM}}$ from the Unity3D$^{\text{TM}}$ Asset Store

[1] The contributions in this chapter are submitted for publication to the IEEE VR special issue Transactions on Visualization and Computer Graphics.

to demonstrate our approach. Screen-shots of the game are shown in Figures 6.1 and 6.3.

6.1.1 Identifying Important Variables and Object Classes

We investigated both variable range and employed a high pass filter on variable derivatives to measure their variation. We run the filter on all internal variables of the game as well as agent location/distance variables exposed by the game AI (Figure 6.1). In our example game, the total number of game variables was over 300. Similar to other machine learning algorithms that perform dimensionality reduction, we ignored the variables that exhibited little variability, focusing on the most informative 5% of the variables. The feature vector thus consists of 13 game variables. These can be seen in Table 6.1; e.g., the variables $\mathrm{Robot}_{\mathrm{dx,dy,dz}}$ encode the distance to the closest robot. All variables thus have valid values at any given time.

Prop_{dx}	Robot_{dx}	NPC_{dx}	Health	Ammo
Prop_{dy}	Robot_{dy}	NPC_{dy}	Hunger	
Prop_{dz}	Robot_{dz}	NPC_{dz}	Thirst	

Table 6.1: The most informative variables that were selected for data collection. dx, dy, dz variables denote distances from the object.

To determine object classes, we parsed the game scene hierarchy generating a set Λ of object categories or *class labels* used for training. Automatic parsing is possible since game objects are not randomly named and usually follow standard naming conventions [Gahan, 2013]. A common naming scheme is “Identificator - Modifier - Variant - Footprint - Optical Distinction”. For example “Tree broadLeaved 01 2x2 Green”. We exploit this scheme by employing a 3D model name parser that infers abstract object classes from object names, avoiding manual object tagging. Using this approach, 25 categories were found (Table 6.2).

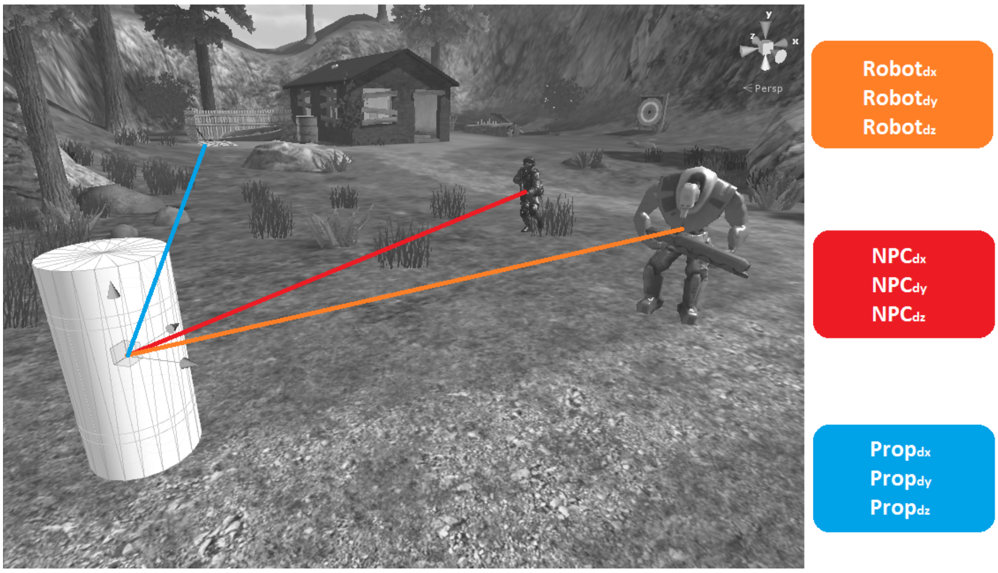


Figure 6.1: Distance vectors exposed by game AI were recorded.

FallenLog	Boat	WoodFence	Fence	Can
Ammo	Barrels	Brickhouse	Crate	Door
Rock	Tree	Water Pickable	Woodboard	Pond
Platform	Elevator	Robot	Soldier	Bush
Zombie	Mine	Food Pickable	Gun Pickable	

Table 6.2: Automatically extracted class labels

The data collection step uses eye-tracking to identify the correlation between the feature vector and the class labels, based on the object class being attended given the current state of the game.

6.1.2 Data Collection Setup Details

Apparatus. The stimuli were displayed on a nVisor$^{\text{TM}}$ SX111 HMD, which has stereo SXGA resolution and a FOV of 102 degrees horizontal by 64 degrees vertical. Participants navigated through the VE using a keyboard and mouse to simulate the FPS input paradigm; the HMD head tracker was disabled. Experienced gamers had no trouble controlling the game with standard *WASD keys* despite the keyboard been occluded by the HMD. Eye-tracking information was recorded using a twin-CCD binocular eye-tracker by Arrington Research$^{\text{TM}}$, which was attached to the HMD. The eye tracker was updated at a frequency of 30Hz.

The FOV of the HMD was restricted to 47.8 degrees horizontally and 23 degrees vertically to simulate a 24” display placed at a 60cm observer distance [Slater et al., 2010]. This was necessary since the eye-tracked HMD of our lab is a partial overlap HMD, which would otherwise force us to converge the virtual cameras. However, the central 50 degrees of its FOV are fully overlapped. By setting the horizontal FOV to be less then 50 degrees, no converging camera setup was necessary allowing us to correctly test our parallel camera stereo grading method. This is not a limitation of the method; employing a desktop eye tracker would yield similar results.

Procedure. All participants underwent a RANDOT stereo vision test (Figure 6.2) [Simons, 1981] and an eye dominance test to select the dominant eye for eye tracking before proceeding with the main experiment. Then the standard Arrington Research$^{\text{TM}}$ eye tracker calibration procedure was performed by each participant. We measured and set the correct IPD for each subject and selected very conservative parameters for the stereo pair in order to obtain a fail-safe and comfortable stereo for all participants, however, with minimal depth complexity. We collected 200 minutes of game-play data in total. During game-play, game state variables were recorded for every sample instance (e.g. $\vec{Robot}_{\mathrm{dx,dy,dz}} = (x, y, z)$). Eye-tracking fixations were used to identify the object via ray-casting and the object together with the game state were inserted into a database.

Eye-Tracking Data De-projection. The eye tracker yields time-coded [x,y] coordinates of fixations in the [0,1] range. To identify which object was fixated in the game FOV, we de-projected the eye-gaze space [x,y] coordinates in the frustum of the participant’s dominant eye. A ray was then reconstructed originating from the dominant eye camera center and passing through the de-projected eye coordinates. The ray was advanced through the scene. The first non-transparent 3D model bounding volume that the ray hit, was considered the attended object.

Stimuli. We modified the game level to have 60-90 seconds of game-play for data

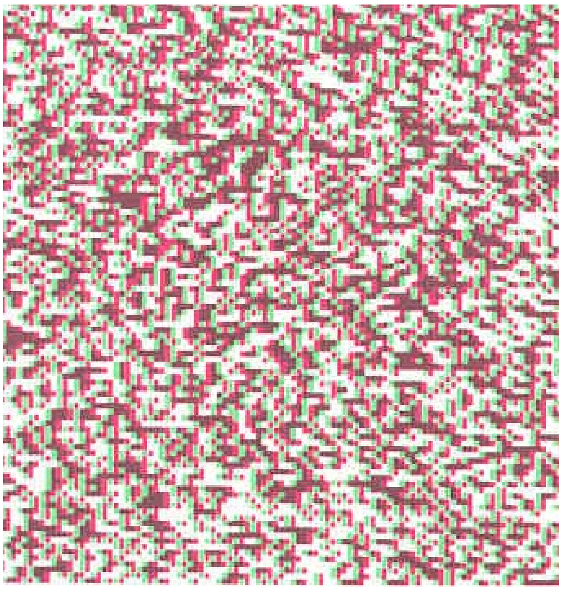


Figure 6.2: A RANDOT stereo test. Please use red/cyan anaglyph glasses; best viewed on a monitor.

collection. The players are required to reach a flaming spaceship while avoiding threatening robots, soldiers, zombies and mines (Figure 6.3). During data collection, the starting position of the player and the spaceship were all similar for every player and trial. An equal number of robots, soldiers, zombies and mines were spawned in random locations in the level during game-play, ensuring necessary variability in the stimuli, and a dense dataset of possible fixation patterns.

Pilot Study. A pilot study indicated that due to differences in individual performance, it was best to fix play time to at least 20 minutes rather than fix the number of trials. We use a speed based sample rate, with a low rate of 5 samples/second, which was increased linearly when the user moved faster through the environment. This allowed a reliable sampling of the obstacle configuration space.

Participants. Ten people participated in the study (2 female, mean age 25). We selected only experienced FPS gamers since our goal is to provide a stereo optimizer for gamers, rather than general VE navigation. All participants played a training level to (i) subjectively verify that participants were indeed experienced gamers, and (ii) familiarize the participants with input controls and the VE. To avoid a

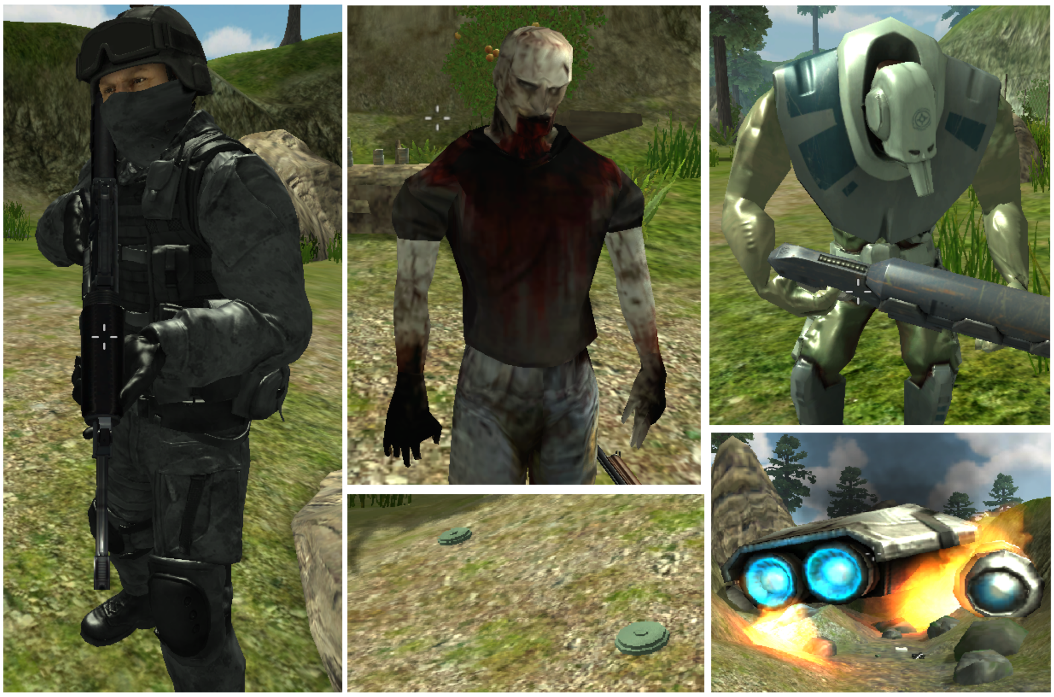


Figure 6.3: The player must reach the flaming spaceship while avoiding soldiers, zombies, robots and mines.

training effect for a search task, participants were instructed to locate a spacecraft during the training session. Participants were given candy as compensation.

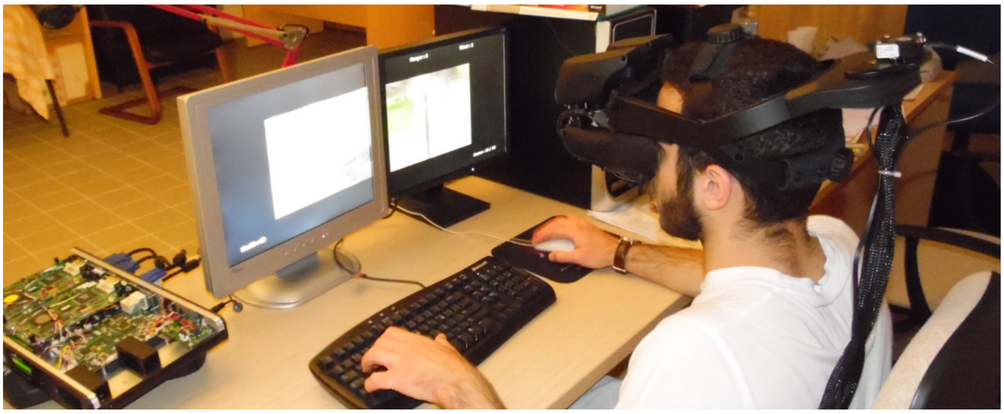


Figure 6.4: The experimental setup.

At the end of data collection, our database contains training data T, having N samples of $M = 13$ features (Table 6.1), $T = (X_1, y_1), (X_2, y_2), ..., (X_N, y_N)$ that will be used to train the DF. Each record $\in T$ includes an input feature vector, $X_i = x_{i1}, x_{i2}, ..., x_{iM}$ and the object class label $y_i \in \Lambda$ (Table 6.2) indicated by the eye tracker at the specific moment that sample X_i was taken.

6.1.3 DF Training and Tuning Details

To validate training accuracy, we use *Out-of-Bag* (OOB) estimates [Breiman, 1996, 2001], which have been shown to be as accurate as using a novel test set of the same size as the training set. Tuning is necessary for the forest to grow optimally in terms of OOB error. We validate test-time prediction accuracy experimentally in Sec. 6.3.1.

We employed a custom-made ID3 dichotomizer method in C-Sharp to generate a decision tree from the dataset. By employing a more sophisticated decision tree generation implementation, higher tree throughput could be achieved. As suggested in [Breiman, 2001], each tree was grown using $mtry = \left\lfloor \frac{\log M}{\log 2} \right\rfloor$ random features/game variables for each split of the tree and we have confirmed that this value is the optimal splitting parameter in terms of OOB error.

The procedure yields *ntree* datasets of the same size as T, grown from a random re-sampling of data in T with-replacement, N times for each dataset. 64% of the data in T were used for the generation of each tree [Breiman, 2001]. This results in $T_1, T_2, ..., T_{ntree}$ *bootstrap* datasets. For each T_i bootstrap dataset a tree is grown. To classify any new input data $D = x_1, x_2, \ldots, x_M$ we test them against each tree to produce *ntree* results $Y = y_1, y_2, ..., y_{ntree}$. For classification, the prediction for this data is the majority vote on this set (Figure 6.5).

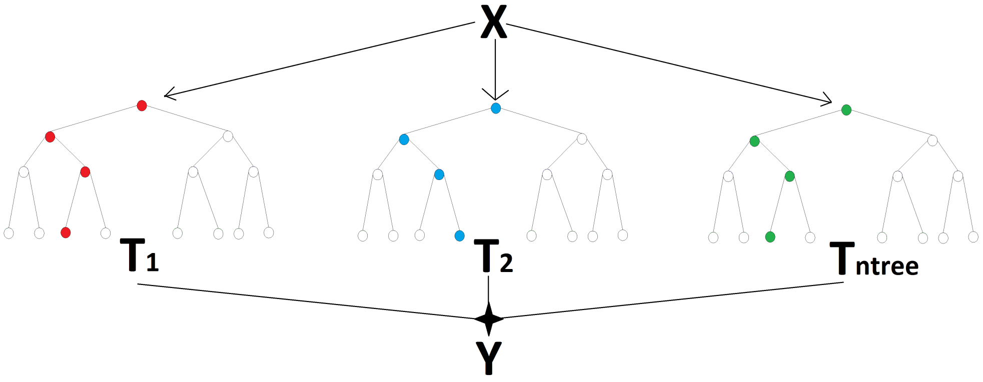


FIGURE 6.5: Each new vector instance is tested against each tree. The majority vote on this sample is the prediction of the DF.

Tuning. In DFs, there is no need for cross-validation or a separate test set to get an unbiased estimate of the test set error in contrast to other ML methods [Bishop et al., 2006]. The study of error estimates for bagged classifiers in [Breiman, 1996] indicates that the OOB estimate is as accurate as using a novel test set of the same size as the training set. Using the OOB error estimate removes the need for a set aside test set. Tuning is necessary for the forest to grow optimally in terms of OOB error.

To estimate the OOB error [Breiman, 2001], after creating the *ntree* classifier trees, we proceed for each (X_i, y_i) in the original training set T and select all T_k which do not include (X_i, y_i). This subset is a set of boostrap datasets which does not contain any record from the original dataset. This is known as the OOB set. There exist N such subsets, one for each data record in the original dataset T. The OOB classifier is the aggregation of votes only over T_k such that it does not contain (X_i, y_i). The OOB estimate for the generalization error is the error rate of the OOB classifier on the training set, compared to known y_i's. Simply put, the error rate for classification on the OOB portion of the data for each tree is recorded and the same is done after permuting all predictor variables. The difference between the two is then averaged over all trees, and then normalized by the standard deviation of the differences. The OOB error estimate is estimated internally, during forest generation (scripts in Appendix A.1).

Frequent Classes Under-sampling. When initially processing the data we encountered a class imbalance issue. Since the participants mostly attended moving objects (soldiers, robots, etc.) in the environment, more samples for these objects were recorded. When a subset of the classes accounts for the majority of the data, the classifier achieves high accuracy by erroneously classifying all the observations into these most frequent classes. This gives high accuracy for frequent classes, but poor predictions for the least frequent ones. To partially compensate we randomly under-sampled frequently sampled classes to balance the data and then trained the model with this balanced data [Breiman, 1996].

The final, balanced set T' spanned $N = 55151$ samples $\times$ $M = 13$ features & Class. We optimized the number of trees to make the OOB error rate converge in terms of a pre-selected error threshold. In the optimized dataset we only kept the object categories for which successful prediction rate $> 55\%$ was achieved. The prediction error rate for the 8 object categories that exceed this threshold can be seen in Table 6.3. The imbalance between sampling rates described previously explains why objects encountered less frequently have a higher prediction error rate in the DF structure, e.g. "Food" and "Explosives". The DF OOB estimate of error rate was found to converge to 16.26% for 100 trees (Figure 6.6). The class imbalance issue for complex games can be amended by increasing the DF training samples. This is expected to increase the number of successfully predicted object categories.

Robot	Soldier	Zombie	Health Pack
7.1	11.2	19.9	20.9
Gun Pickable	Explosives	Ammo	Food Pickable
25	36.3	37.8	40.6

TABLE 6.3: Prediction error rate for each object category.

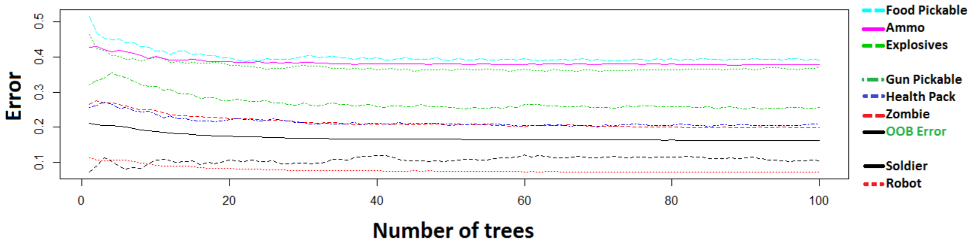


FIGURE 6.6: Prediction error rate for each object category in relation to forest growth.

6.2 Dynamic Stereo Grading based on Gaze

Now that we have a classifier that can predict gaze based on game state, we can place attended objects inside the comfort zone and as close to the plane of zero disparity possible, i.e. onto the virtual screen plane. We describe this dynamic stereo grading process next.

Our system linearly interpolates camera separation and asymmetric frustum parameters to avoid visual artifacts and observers becoming aware of the change. We use the standard asymmetric viewing frusta, as presented among others by Woods et al. [Woods et al., 1993] (details in Chapter 2).

Compared to previous stereo grading algorithms, e.g., [Oskam et al., 2011] our method performs *automatic localization* of the disparity manipulation.

The trained DF component of the stereo grader receives a game state variable vector as an input, and generates an object category prediction. The trained data structure is serialized and stored on the disk. This speeds up application loading time since tree generation only happens once. Our system is scalable; the number of queries to the DF structure, is a parameter that depends on the number of trees in the DF and system throughput. On our test setup (Intel Core I7@3.4GHz, 8Gb RAM) a DF structure based on 100 decision trees responds at a rate of 2 queries/second at runtime. However, this scales automatically: the faster the machine, the more times the DF can be queried.

After obtaining a prediction, the system searches for same-category objects in the viewing frustum, with three possible outcomes (Algorithm 1): A single, multiple or no objects of that category are found. If a single object is found, its distance to the camera is estimated via ray casting within the depth buffer. Then asymmetric frustum parameters are estimated [Woods et al., 1993] that shift the zero-parallax plane and thus the comfort zone close to the barycentre of that object. The distance of the object from the zero-parallax plane is a parameter that defines how deep or shallow these objects are perceived. To estimate this parameter we take into account the object's bounding volume radius. If multiple objects of that category are found, the combined barycentre of these objects is estimated. The zero-parallax plane is then brought close to the novel barycentre. If no object of that category is found this indicates that the predictor has failed. To achieve fail-safe stereo, the zero parallax plane is brought close to the largest object adjacent to the center of the view frustum.

Optimizations. The system is more aggressive when grading negative disparities that cause more strain than positive disparities [Mendiburu, 2012]. If the estimated negative disparity is larger than 3% of the distance of the virtual camera pair to the virtual plane, our method pushes the objects that are closest to the screen further back to minimize eye strain [Mendiburu, 2012]. By employing this approach, the system manages to keep all important objects in terms of gaze inside the comfort volume.

We linearly interpolate in time all camera transitions so that changes are not perceived by an observer, similarly to [Oskam et al., 2011]. However, OSCAM adjusts disparity in terms of the whole scene depth or by pre-determined *manually* selected depth ratios that are least-squares fit to the desired mapping. OSCAM's automatic fail-safe mode suffers from cardboarding when grading scenes with large depths. Our method generalizes OSCAM's by minimizing cardboarding *automatically.*

6.3 Evaluation and Results

We experimentally validated the accuracy of our predictor. We also compared its performance to a low-level saliency predictor and measured the perceived quality of our stereo grading compared to other approaches.

The validation experiment was split in 3 sessions with a 10-minute break between sessions to reduce eye strain. The 3 sessions were a pairwise comparison of Standard stereo (no disparity management), Ours and OSCAM: **a**: Standard <> Ours, **b**: Ours <> OSCAM, **c**: OSCAM <> Standard. We used our HMD's eye tracker

Algorithm 1 : Stereo Manipulation Process

```
procedure Trace
    for each second do
        Obtain game state variable vector
        Query DF to find object category
        if No. of objects $== 1$ then
            Estimate object barycentre
            Find its distance from camera & adjust parameters to bring it to the zero parallax plane
        end if
        if No. of objects $> 1$ then
            Estimate combined object barycentre State Find its distance from camera & adjust parameters to bring that to the zero parallax plane
        end if
        if No. of objects $== 0$ then
            Find the largest object in the centre in the FOV
            Estimate its barycentre
            Find its distance from camera and bring this to the zero parallax plane.
        end if
    end for
end procedure
```

to obtain gaze data only during session **a**. Ten participants not previously involved in any related experiment (2 female, mean age 23.5) completed the sessions successfully.

In each session players played 10 pairwise 10-second game rounds of predetermined game-play, lasting in total for 200 seconds (10 pairs × 2 conditions × 10 sec). The order in each pair was randomized. Session pairs intentionally imposed large disparity changes: objects moving in the view frustum, camera wildly panning, etc. Example conditions were designed having in mind common *disparity events* that cause strain and affect depth perception in a game. For example, a moving object (e.g., an enemy) is about to appear, and will be far away in depth. Depths should be compressed in time to prepare the observer for the moving object to avoid intense convergence motion. A second example is an object that is expected to appear due to a lateral movement and which will introduce an extreme disparity. Another example is an enemy that is shooting and threatening the -devoid of

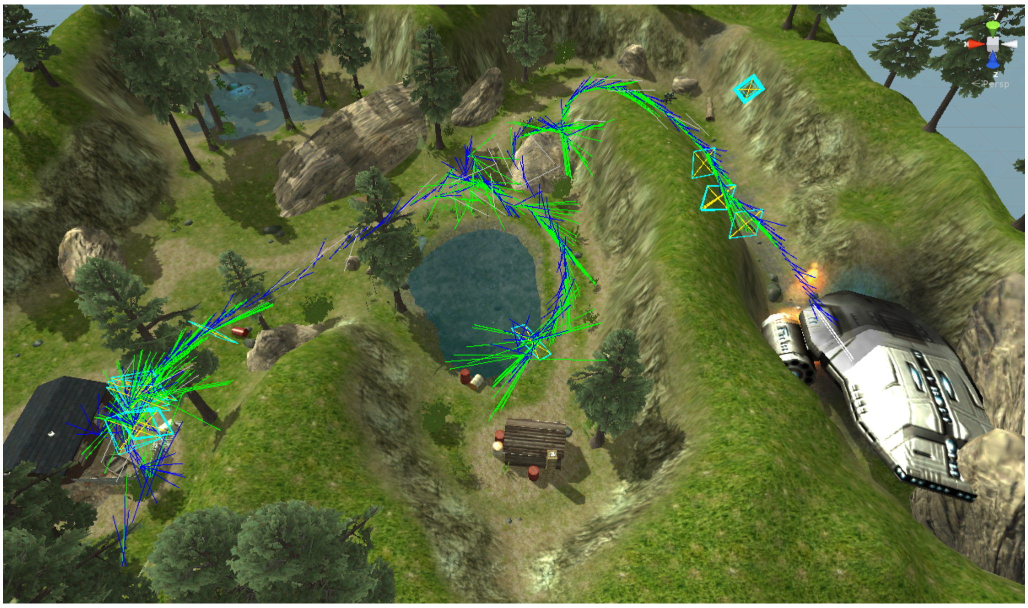


FIGURE 6.7: Visualization of successful attention predictions of our method with green beams. The blue beams indicate eye tracking data. The grey beams indicate failed predictions. The cyan colored squares indicate major frustum reconfigurations.

ammo- player but the player will not look at the enemy, instead the player will search for ammo lying on the floor.

6.3.1 DF-based Predictor Quality

We have already presented a validation of the training using OOB estimates in Section 6.1; here we perform test-time evaluation of our DF predictor using eye-tracking and compare to a state-of-the-art low-level predictor [Walther and Koch, 2006].

During session **a**, low level (x,y) predicted coordinates, DF predictions and eye tracking data of the view frustum were obtained at a rate of 1Hz, 1Hz and 30Hz respectively. An eye fixation was considered to be spatially stable if it lasted at least 300 milliseconds [Salvucci and Goldberg, 2000]. Thus for every second we obtain up to 3 possible fixation locations. We also define a baseline estimator R that selects a random object in the same view frustum at 1Hz.

Est.	Object gazed	Hits
R	random object	$< 5\%$
Low	x,y,radius	44.1%
DF	object category	76.2%

Table 6.4: The ratio of frames for which the attended object was predicted correctly during session **a**.

We perform a temporal window integration comparison for the 3 predictors (Low-level, DF and baseline). Since the sampling rate of all three compared predictors is 1Hz, if any of the user fixations within this one-second window are predicted correctly by a method, we consider that a hit. In particular, for the low level predictor, a prediction is considered a hit if the actual fixation lies inside a 128 pixel radius circle around the predicted x,y coordinates.

In our approach, if a fixation is on an object of a predicted category this is considered a prediction hit. We visualize our predictions compared to eye-tracking in Fig. 6.7. Table 6.4 shows the success rate of low-level and DF predictors. Our DF predictor outperforms the low level predictor when task-imposed constraints exist (Fig. 6.8).

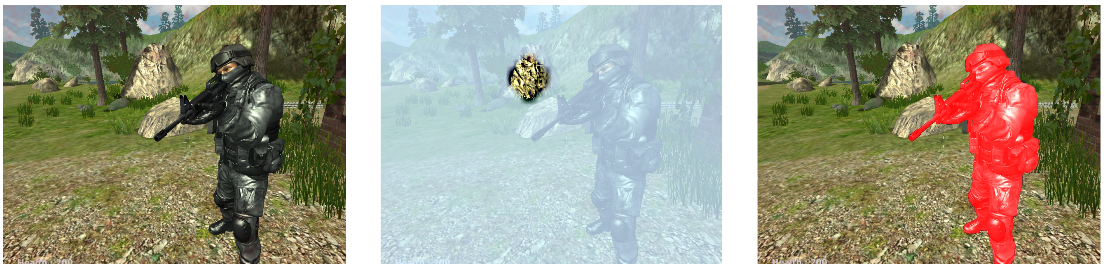
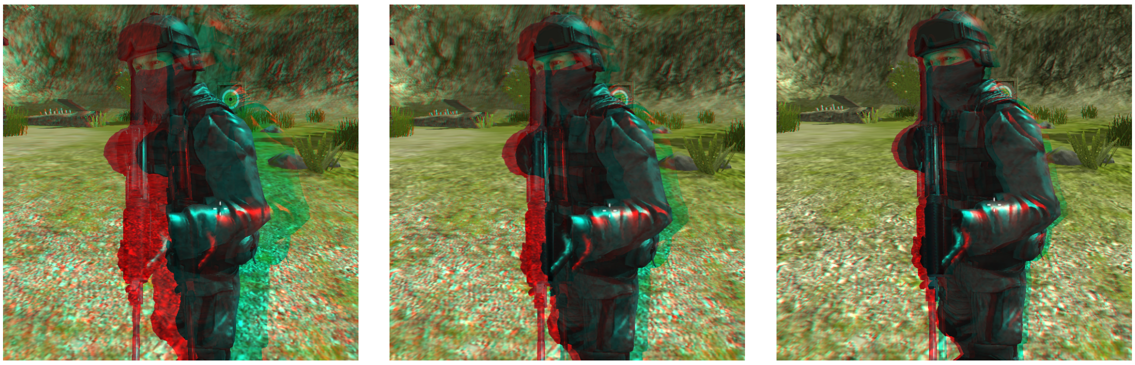


Figure 6.8: Comparison of low level gaze prediction (middle) and our DF predictor (right) for the same scene (left). The player is threatened by the soldier.

6.3.2 Dynamic Disparity Management

We used a protocol inspired by both [Lang et al., 2010, Oskam et al., 2011]. At the end of each pair in a session, participants were asked to choose between the



No display management Ours OSCAM

Figure 6.9: Left to right: No display management, Ours, OSCAM. Please use red/cyan anaglyph glasses; best viewed on a monitor.

two sessions of a given pair, to determine (i) which one had more depth and (ii) which was more comfortable in terms of diplopia and eye fatigue.

We received 600 answers in total (2 questions × 3 sessions × 10 conditions × 10 participants) (Table 6.5). For the first question about which condition had more depth, our method outperforms both OSCAM and Standard. Our method was preferred 71% of the time when compared to Standard, and 67% of the time when compared to OSCAM. We also confirmed that OSCAM is preferred over standard (68% of the time). All results are statistically significant (t-test, $p < 0.01$).

For question 2 on comfort our method and OSCAM outperform Standard being preferred 73% and 78% of the time respectively. These results are statistically significant (t-test, $p < 0.01$). However, when comparing our method to OSCAM in terms of eye strain, participants did not have a clear preference (52% vs 48% respectively, $p = 0.27$).

	Ours/Stan.	**Ours/OSCAM**	**OSCAM/Stan.**
Depth	Ours 71% stdev: 14.4	Ours 67% stdev: 24.5	OSCAM 68% stdev: 7.9
Strain	Ours 78% stdev: 13.3	*Ours 52 %* *stdev: 10.3*	OSCAM 78% stdev: 6.3

Table 6.5: Preferred method of stereo grading for each question and session. Ours<>OSCAM results for eye strain are non-significant.

6.4 Chapter Summary

We presented our latest generation predictor that yields high prediction success rates, is automatic, avoiding the need for manual tagging of objects and supports object motion in contrast to our previous high level approach (Chapters 4 + 5). However, it requires an eye-tracker in order to be trained.

The localized stereo grading approach presented provides better perceived depth than previous global methods, while maintaining similar levels of viewing comfort.

In the following chapter we conclude the thesis discussing current limitations and potential future applications

Chapter 7

Conclusion, Limitations and Future Work

We presented two gaze prediction models that account for task and high level saliency effects on visual attention. We developed three gaze-aware applications that exploit the prediction accuracy of our models to adjust game level difficulty, optimize GPU performance on mobile devices and reduce eye strain when watching stereoscopic 3D content.

7.1 Limitations and Intuitions

Gaze prediction. The current version of both models does not take into account low level image saliency which would otherwise further improve gaze prediction accuracy. We plan to extend our model with low-level factors for more accurate predictions when gross low level irregularities exist in an image.

The first object-semantics-based model requires object-class meta-data in order to predict attention. The production-level working system will provide a taxonomy of objects in relation to scene schemata as a library limiting the need for further experiments or manual input.

We expect that a trained instance of the machine learning-based model may be extended to different games with similar mechanics. Our learning based predictor could be used for additional applications, especially if the performance is improved. If prediction becomes faster, it could be used to adjust LOD, depth-of-field or game difficulty based on user gaze.

LOD and Game Balancing. The LOD manager only works with scenes where object placement is not altered during rendering; only the camera may freely move around the scene. We plan to investigate the performance of the proposed LOD manager in dynamic scenes. Regarding our game balancing editor we would like to evaluate our tool by presenting it to skilled experts in game level design.

Stereo manipulation. Although our experiments have focused on eye-glass-based stereo displays, similar problems appear for any single-screen display that supports stereo viewing [Masia et al., 2013]. Like all stereo grading methods, our approach modifies depth and speed of objects in a scene. Even though speed modification did not cause problems in our experiments, a more involved stereo-motion speed-preserving optimization strategy [Kellnhofer et al., 2013] could be required. Currently, the speed of the classifier is quite low (2 queries/second); a more optimized implementation of DF's would improve this speed. The training phase required an eye-tracked HMD; the typical cost of such a setup is probably a realistic option for a game-studio, however, a desktop eye tracker could be used as well.

7.2 Future Work

Dynamic Game Balancing and AI. We intend to integrate our model in a game engine for on-the-fly level difficulty adjustments and a smarter game AI. Objects could be repositioned dynamically resulting in an adjustable level of difficulty depending on user performance so far. Object placement could automatically shift after every re-spawn when a player comes back to life after being killed. A smarter AI could use high level saliency data to spawn opponents that pop-out or appear inconspicuously.

Cinematography. An attention based cinematography system could be developed that applies post-process effects based on attention. Simulating natural effects such as depth-of-field, camera path generation, context-aware replay, cut-scene generation, camera motion and dynamic lighting could benefit from a list of potentially gazed objects based on high level saliency. It has been shown that when these effects are dynamically adapted depending on gaze, users reproduce distances better in a VE [Moehring et al., 2009].

In-App Advertising. Identifying salient areas in a game to place geometry bearing corporate logos may improve in-app advertising since advertisements will be more visible during game-play.

Production level stereo grading. Regarding stereo disparity manipulation based on a machine learning predictor we demonstrated our ideas on a prototype game level. In a production context, our approach could serve as a basis for the development of a viable and attractive solution in game design. Training is easy and can be incorporated to the existing game testing pipelines by simply adding an eye tracker in the game testing rig. The trained model does not capture absolute eye tracking coordinates but learns vergence patterns based on game mechanics.

Gaze prediction to improve eye-tracking. Real-time low quality eye-tracking is feasible on mobile devices by acquiring images of the observer's eye using the front facing camera of a phone/tablet.

However, eye-tracking on mobiles suffers from all the standard issues of desktop eye-tracking (e.g. eye drifting) in addition to specific weaknesses inherent to mobile devices. In particular, eye-tracking on mobile devices by employing the front facing camera is hard for several reasons: (i) an infra-red light source is unavailable, thus a clear glint image on the cornea can not be acquired, (ii) the front-facing cameras are of low resolution, (iii) mobile devices have limited processing power, (iv) mobile devices are portable, thus a. the distance between the camera and the eyes is not standard, b. the head/device relation is not static and c. the environmental lighting is out of control.

An ambitious extension of this research would be to increase the accuracy of eye-tracking as estimated by the mobile device camera by probabilistically correlating fixations with gaze predictions generated from a mobile-friendly attention model which incorporates a high level saliency model based on semantics and a low level saliency based on image characteristics optimizing the accuracy of sampled eye data. The system proposed will generate attention predictions for the current view of a computer graphics scene.

Appendix A

Appendix

A.1 Analysis with R and Sample Scripts

The R language implements a variety of statistical and graphical techniques such as linear and nonlinear modeling, classical statistical tests, classification and clustering. R is extensible via functions and plug-ins. In this Appendix Sample R Scripts of Chapters 4-6 are presented.

Chapter 4, Experiment 2 R-Code:

```
#read data into variable
datavar <- read.csv("exp1.csv")

#attach data variable
attach(datavar)

#display all data
dataFrame <- data.frame(datavar)
dataFrame

#represent a categorical variable numerically using as.numeric(VAR)
#dummy code the CONSISTENCY variable into CONS = 1 and INCONS = 0
dCONSISTENCY <- as.numeric(CONSISTENCY) - 1
```

```
#represent a categorical variable numerically using as.numeric(VAR)
#dummy code the SINGLETONESS variable into COMP = 1 and SINGL = 0
dSINGLETONESS <- as.numeric(SINGLETONESS) - 1

#boxplot(TIME~SINGLETONESS*CONSISTENCY, data=dataFrame ,xlab =
"Semantic and Physical Context", ylab = "Time (seconds)", main = "")

boxplot(TIME~CONSISTENCY*SINGLETONESS, data=dataFrame ,
  col=(c("orangered1","deepskyblue3")),
  main="", ylab="Task Completion Time (seconds)")



#create a linear model using lm(FORMULA, DATAVAR)
onepredictor  <-lm(TIME ~ dCONSISTENCY , datavar)
twopredictors <-lm(TIME ~ dSINGLETONESS + dCONSISTENCY , datavar)
#generate model summary
summary(onepredictor)
summary(twopredictors)

anova(onepredictor,twopredictors)
```

Chapter 4, Experiment 3 R-Code:

```
#read data into variable
datavar <- read.csv("exp2.csv")

#attach data variable
attach(datavar)

#display all data
dataFrame <- data.frame(datavar)
dataFrame

#represent a categorical variable numerically using as.numeric(VAR)
#dummy code the CONSISTENCY variable into CONS = 1 and INCONS = 0
dCONSISTENCY <- as.numeric(CONSISTENCY) - 1

#represent a categorical variable numerically using as.numeric(VAR)
#dummy code the SINGLETONESS variable into COMP = 1 and SINGL = 0
dSINGLETONESS <- as.numeric(SINGLETONESS) - 1

#boxplot(TIME~SINGLETONESS*CONSISTENCY, data=dataFrame ,xlab = "Month",
ylab = "Maximum Temperature", main = "Temperature at Southampton Weather
Station (1950-1999)")

boxplot(TIME~CONSISTENCY*SINGLETONESS, data=dataFrame ,
  col=(c("orangered4","deepskyblue4")),
  main="", ylab="Task Completion Time (seconds)")

#create a linear model using lm(FORMULA, DATAVAR)
onepredictor  <-lm(TIME ~ dSINGLETONESS, datavar)
twopredictors <-lm(TIME ~ dCONSISTENCY + dSINGLETONESS, datavar)
#generate model summary
summary(onepredictor)
summary(twopredictors)

anova(onepredictor,twopredictors)
```

Chapter 4, Validation experiment R-Code:

```
#read data into variable
datavar <- read.csv("validation.csv")

#attach data variable
attach(datavar)

#display all data
dataFrame <- data.frame(datavar)
dataFrame

#represent a categorical variable numerically using as.numeric(VAR)
#dummy code the CONSISTENCY variable into CONS = 1 and INCONS = 0
dCONDITION <- as.numeric(CONDITION) - 1

#boxplot(TIME~SINGLETONESS*CONSISTENCY, data=dataFrame ,xlab = "Semantic
and Physical Context", ylab = "Time (seconds)", main = "")

boxplot(TIME~CONDITION, data=dataFrame ,
  col=(c("green1","red1")),
  main="", ylab="Task Completion Time (seconds)")


#create a linear model using lm(FORMULA, DATAVAR)
onepredictor  <-lm(TIME ~ dCONDITION , datavar)
#generate model summary
```

```
summary(onepredictor)

t.test(TIME~CONDITION)
```

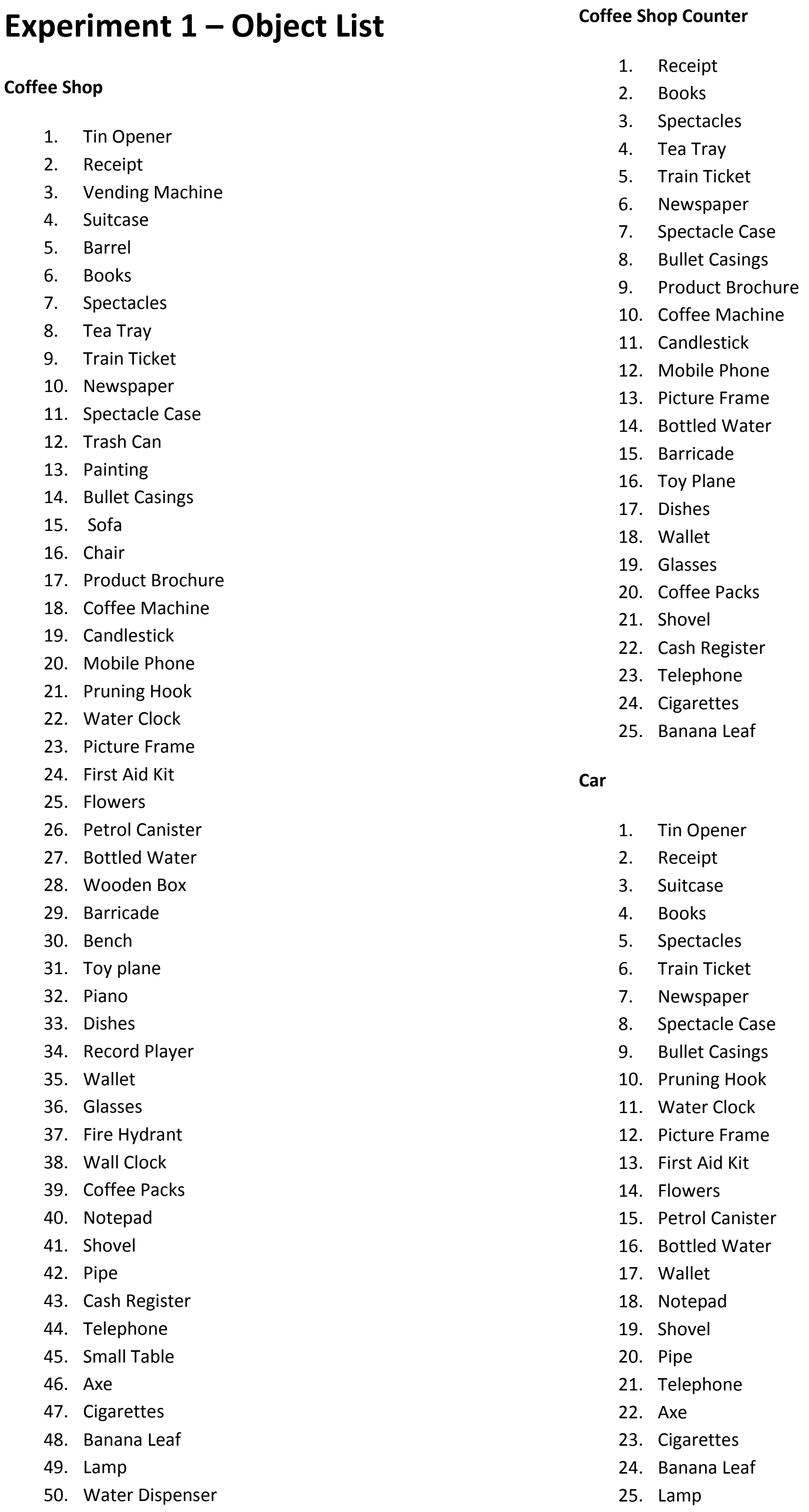
Chapter 5, Main experiment R-Code:

```
library(lattice)
#read data into variable
datavar <- read.csv("timings.csv")

#attach data variable
attach(datavar)

#display all data
dataFrame <- data.frame(datavar)
dataFrame

dSINGLETONESS <- as.numeric(SINGLETONESS) - 1

dFORM <- as.numeric(FORM) - 1

#boxplot(TIME~SINGLETONESS*CONSISTENCY, data=dataFrame ,xlab = "Semantic
and Physical Context", ylab = "Time (seconds)", main = "")

# Example of a Bagplot
#library(aplpack)
#attach(dataFrame)
#bagplot(FORM,TIME, xlab="Car Weight", ylab="Miles Per Gallon",
#  main="Bagplot Example")

#qplot(FORM, TIME, data=dataFrame , geom=c("boxplot", "jitter"),
#   fill=FORM, main="Mileage by Gear Number",
#   xlab="", ylab="Miles per Gallon")

#qplot(TIME, SINGLETONESS, data=dataFrame ,
#   facets=SINGLETONESS~FORM, size=I(2),
#   xlab="Horsepower", ylab="Miles per Gallon")

boxplot(TIME~SINGLETONESS*FORM, data=dataFrame , notch=TRUE,
  main="", ylab="Task Completion Time (seconds)")
#points(TIME~FORM, pch = 1)

#beanplot(TIME, TIME, horizontal=T,
 #        names=c("Business", "Law"),
  #       col=c("blue", "gold"))

## add some vertical jittering (use ‘factor=‘ to change its amount in both cases)
#dotplot(TIME~SINGLETONESS, data=dataFrame, jitter.x=TRUE, factor=0.5)
#stripplot(TIME~FORM, data=dataFrame, jitter.x=TRUE, factor=0.5)



#create a linear model using lm(FORMULA, DATAVAR)
onepredictor  <-lm(TIME ~ dSINGLETONESS , datavar)
singlepredictor  <-lm(TIME ~ dFORM , datavar)
twopredictors <-lm(TIME ~ dFORM + dSINGLETONESS , datavar)
#generate model summary
summary(onepredictor)
summary(singlepredictor)
summary(twopredictors)

anova(onepredictor,twopredictors)
```

Chapter 6, Checking Random Forests R-Code:

```
library(randomForest)
D = read.csv ("8cats.csv", header = T)
rf = randomForest(as.factor(CATEGORY) ~ ., data=na.omit(D),
importance = TRUE, keep.forest=TRUE, mtry=4, ntree=100, do.trace=100)
plot(rf)
importance(rf)
varImpPlot(rf)
```

A.2 Object Lists, Consent Form, Storylines



Experiment 1 – Object List

Coffee Shop

1. Tin Opener
2. Receipt
3. Vending Machine
4. Suitcase
5. Barrel
6. Books
7. Spectacles
8. Tea Tray
9. Train Ticket
10. Newspaper
11. Spectacle Case
12. Trash Can
13. Painting
14. Bullet Casings
15. Sofa
16. Chair
17. Product Brochure
18. Coffee Machine
19. Candlestick
20. Mobile Phone
21. Pruning Hook
22. Water Clock
23. Picture Frame
24. First Aid Kit
25. Flowers
26. Petrol Canister
27. Bottled Water
28. Wooden Box
29. Barricade
30. Bench
31. Toy plane
32. Piano
33. Dishes
34. Record Player
35. Wallet
36. Glasses
37. Fire Hydrant
38. Wall Clock
39. Coffee Packs
40. Notepad
41. Shovel
42. Pipe
43. Cash Register
44. Telephone
45. Small Table
46. Axe
47. Cigarettes
48. Banana Leaf
49. Lamp
50. Water Dispenser

Coffee Shop Counter

1. Receipt
2. Books
3. Spectacles
4. Tea Tray
5. Train Ticket
6. Newspaper
7. Spectacle Case
8. Bullet Casings
9. Product Brochure
10. Coffee Machine
11. Candlestick
12. Mobile Phone
13. Picture Frame
14. Bottled Water
15. Barricade
16. Toy Plane
17. Dishes
18. Wallet
19. Glasses
20. Coffee Packs
21. Shovel
22. Cash Register
23. Telephone
24. Cigarettes
25. Banana Leaf

Car

1. Tin Opener
2. Receipt
3. Suitcase
4. Books
5. Spectacles
6. Train Ticket
7. Newspaper
8. Spectacle Case
9. Bullet Casings
10. Pruning Hook
11. Water Clock
12. Picture Frame
13. First Aid Kit
14. Flowers
15. Petrol Canister
16. Bottled Water
17. Wallet
18. Notepad
19. Shovel
20. Pipe
21. Telephone
22. Axe
23. Cigarettes
24. Banana Leaf
25. Lamp

Context-aware Gaze Prediction Evaluation Study

Consent Form

You are requested to fill out a questionnaire which will help the research and experimental stage of a doctoral thesis carried out at the Electronic and Computer Engineering Department, Technical University of Crete. The experiment is performed by the PhD candidate George-Alex Koulieris supervised by Associate Professor Katerina Mania. The experiment investigates the modeling of high-level saliency optical characteristics of a virtual environment, both semantically and in terms of object topology. The experiment is expected to last no more than 20 minutes. We will use your data anonymously along with the data of other participants.

Do you understand the consent form?	Yes	No
Do you grant permission to process your data?	Yes	No
Are you at least 18 years old?	Yes	No

Name/Surname	
E-mail	
Age Group	18-22 23-27 28-32 33-37 38-42 > 43
Gender	Male / Female
Date	___ / ___ / 201_
If you are a student please select	Undergraduate Graduate PhD Candidate
Current Status	Student Research Academic Other (report) _______________

Signature

Story Line

Search task

“Adrian Black works at the Coffee Shop that you will see. He is married, but very unhappy with his marriage. For this reason, Adrian engages in an affair with his customer Nicole and decides to start a new life with her. As he was too cowardly to file for divorce, he decides to fake his death to cover a getaway with her. He damps his car outside of his workplace, spoiling the car interior with blood from a live pig that he bought, but he was careless and left the receipt for the pig behind. He pretends that the murderer hides the body leaving the victim’s spectacles and wallet behind as a clue of a pretentious murder. You, in the role of detective Cole Phelps, have a hunch that the murder is staged. Find and collect as fast as you can, clues suggesting that Adrian is still alive, i.e. his spectacles, his wallet and the pig purchase receipt.”

Non-Search task

“Adrian Black works at the Coffee Shop that you will see. He is married, but very unhappy with his marriage. For this reason, Adrian engages in an affair with his customer Nicole and decides to start a new life with her. As he was too cowardly to file for divorce, he decides to fake his death to cover a getaway with her. He damps his car outside of his workplace, spoiling the car interior with blood from a live pig that he bought, but he was careless and left the receipt for the pig behind. You, in the role of detective Cole Phelps, have a hunch that the murder is staged. Find and collect as fast as you can, clues suggesting that Adrian is still alive, i.e. the pig purchase receipt, a clue that he was having an affair with Nicole, and a clue that he was planning to leave town with her.”

Bibliography

Arrington (2015). Viewpoint documentation. [Online; accessed 1-September-2015].

Atkinson, S. (2011). Stereoscopic-3d storytellingrethinking the conventions, grammar and aesthetics of a new medium. *Journal of Media Practice*, 12(2):139–156.

Awh, E. and Pashler, H. (2000). Evidence for split attentional foci. *Journal of Experimental Psychology: Human Perception and Performance*, 26(2):834.

Bailey, R., McNamara, A., Sudarsanam, N., and Grimm, C. (2009). Subtle gaze direction. *ACM Trans. on Graphics*, 28(4):100.

Bar, M., Ullman, S., et al. (1996). Spatial context in recognition. *PERCEPTION-LONDON*, 25:343–352.

Barre-Brisebois, C. (2011). Approximating translucency for a fast, cheap and convincing subsurface-scattering look. In *Game Developers Conference*.

Bartlett, F. C. (1932). Remembering: An experimental and social study. *Cambridge: Cambridge University*.

Becker, M. W., Pashler, H., and Lubin, J. (2007). Object-intrinsic oddities draw early saccades. *Journal of Experimental Psychology: Human Perception and Performance*, 33(1):20.

Bernhard, M., Dellmour, C., Hecher, M., Stavrakis, E., and Wimmer, M. (2014). The effects of fast disparity adjustment in gaze-controlled stereoscopic applications. In *Proc. of the Symp. on Eye Tracking Res. and Appl.*, pages 111–118. ACM.

Bernhard, M., Stavrakis, E., and Wimmer, M. (2010). An empirical pipeline to derive gaze prediction heuristics for 3D action games. *ACM Trans. on Applied Perception (TAP)*, 8(1):4.

Bernhard, M., Zhang, L., and Wimmer, M. (2011). Manipulating attention in computer games. In *IVMSP Workshop, 2011 IEEE 10th*, pages 153–158. IEEE.

Bishop, C. M. et al. (2006). *Pattern recognition and machine learning*, volume 4. Springer New York.

Blanz, V., Tarr, M. J., Bülthoff, H. H., and Vetter, T. (1999). What object attributes determine canonical views? *Perception-London*, 28(5):575–600.

Borji, A. and Itti, L. (2013). State-of-the-art in visual attention modeling. *IEEE Trans. on PAMI*, 35(1).

Breiman, L. (1996). Bagging predictors. *Mach. Learn.*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Mach. Learn.*, 45(1):5–32.

Brewer, W. F. and Treyens, J. C. (1981). Role of schemata in memory for places. *Cognitive Psychology*, 13(2):207–230.

Bruce, N. D. and Tsotsos, J. K. (2005). An attentional framework for stereo vision. In *Computer and Robot Vision, 2005. Proc.. The 2nd Canadian Conference on*, pages 88–95. IEEE.

Çapin, T. K., Pulli, K., and Akenine-Möller, T. (2008). The state of the art in mobile graphics research. *IEEE Computer Graphics and Applications*, 28(4):74–84.

Cater, K., Chalmers, A., and Ledda, P. (2002). Selective quality rendering by exploiting human inattentional blindness: looking but not seeing. In *Proceedings of the ACM symposium on Virtual reality software and technology*, pages 17–24. ACM.

Cater, K., Chalmers, A., and Ward, G. (2003). Detail to attention: exploiting visual tasks for selective rendering. In *Proc. Eurographics workshop on Rendering*, pages 270–280.

Chapiro, A., Diamanti, O., Poulakos, S., OSullivan, C., Smolic, A., and Gross, M. (2014). Perceptual evaluation of cardboarding in 3D content visualization. *Proc. of ACM SAP*.

Clark, J. H. (1976). Hierarchical geometric models for visible surface algorithms. *Communications of the ACM*, 19(10):547–554.

Crawford, C. (1984). *The art of computer game design*. Osborne/McGraw-Hill Berkeley, CA.

Cunningham, D. W. and Wallraven, C. (2011). *Experimental Design: From user studies to psychophysics*. A.K. Peters.

Czikszentmihalyi, M. (1990). Flow: The psychology of optimal experience. *Praha: Lidové Noviny*.

Desurvire, H., Caplan, M., and Toth, J. A. (2004). Using heuristics to evaluate the playability of games. In *CHI'04 extended abstracts on Human factors in computing systems*, pages 1509–1512. ACM.

Didyk, P., Ritschel, T., Eisemann, E., Myszkowski, K., and Seidel, H.-P. (2011). A perceptual model for disparity. In *ACM Trans. on Graphics*, volume 30, page 96. ACM.

Drascic, D. (1991). Skill acquisition and task performance in teleoperation using monoscopic and stereoscopic video remote viewing. In *Proc. of the human factors* *and ergonomics society annual meeting*, volume 35, pages 1367–1371. SAGE Publications.

Eckstein, M. P. (1998). The lower visual search efficiency for conjunctions is due to noise and not serial attentional processing. *Psychological Science*, 9(2):111–118.

Eckstein, M. P., Drescher, B. A., and Shimozaki, S. S. (2006). Attentional cues in real scenes, saccadic targeting, and bayesian priors. *Psychological Science*, 17(11):973–980.

Eckstein, M. P., Shimozaki, S. S., and Abbey, C. K. (2002). The footprints of visual attention in the posner cueing paradigm revealed by classification images. *Journal of Vision*, 2(1).

Ehinger, K. A., Hidalgo-Sotelo, B., Torralba, A., and Oliva, A. (2009). Modelling search for people in 900 scenes: A combined source model of eye guidance. *Visual cognition*, 17(6-7):945–978.

Einhäuser, W., Spain, M., and Perona, P. (2008). Objects predict fixations better than early saliency. *Journal of Vision*, 8(14).

El-Nasr, M. S. and Yan, S. (2006). Visual attention in 3D video games. In *Proc. of the 2006 ACM SIGCHI Intl. Conf. on Advances in computer entertainment technology*. ACM.

Feil, J. and Scattergood, M. (2005). *Beginning game level design.* Course Technology PTR.

Frintrop, S., Rome, E., and Christensen, H. I. (2010). Computational visual attention systems and their cognitive foundations: A survey. *ACM Trans. on Applied Perception (TAP)*, 7(1):6.

Gahan, A. (2013). *3ds Max Modeling for Games: Insider's guide to game character, vehicle, and environment modeling.* CRC Press.

Gateau, S. and Neuman, R. (2010). Stereoscopy from xy to z. *Courses of SIGGRAPH Asia*, 63.

Green, D. M., Swets, J. A., et al. (1966). *Signal detection theory and psychophysics*, volume 1. Wiley New York.

Grillon, H. and Thalmann, D. (2009). Simulating gaze attention behaviors for crowds. *Computer Animation and Virtual Worlds*, 20(2-3):111–119.

Han, S., Wan, X., and Humphreys, G. W. (2005). Shifts of spatial attention in perceived 3-d space. *The Quarterly Journal of Experimental Psychology*, 58(4):753–764.

Henderson, J. M. and Hollingworth, A. (1999). High-level scene perception. *Annual review of psychology*, 50(1):243–271.

Henderson, J. M. and Hollingworth, A. (2003a). Eye movements and visual memory: Detecting changes to saccade targets in scenes. *Perception & Psychophysics*, 65(1):58–71.

Henderson, J. M. and Hollingworth, A. (2003b). Global transsaccadic change blindness during scene perception. *Psychological Science*, 14(5):493–497.

Henderson, J. M., Weeks Jr, P. A., and Hollingworth, A. (1999). The effects of semantic consistency on eye movements during complex scene viewing. *Journal of experimental psychology: Human perception and performance*, 25(1):210.

Hillaire, S., Lécuyer, A., Regia-Corte, T., Cozot, R., Royan, J., and Breton, G. (2010). A real-time visual attention model for predicting gaze point during first-person exploration of virtual environments. In *Proc. of the 17th ACM Symp. on Virtual Reality Software and Technology*, pages 191–198. ACM.

Hoffman, D. M., Girshick, A. R., Akeley, K., and Banks, M. S. (2008). Vergence–accommodation conflicts hinder visual performance and cause visual fatigue. *Journal of vision*, 8(3):33.

Holland, C. and Komogortsev, O. (2012). Eye tracking on unmodified common tablets: challenges and solutions. In *Proc. of the Symp. on Eye Tracking Res. and Appl.*, pages 277–280. ACM.

Hollingworth, A., Schrock, G., and Henderson, J. M. (2001). Change detection in the flicker paradigm: The role of fixation position within the scene. *Memory & Cognition*, 29(2):296–304.

Hwang, A. D., Wang, H.-C., and Pomplun, M. (2011). Semantic guidance of eye movements in real-world scenes. *Vision research*, 51(10):1192–1205.

Ishimaru, S., Kunze, K., Utsumi, Y., Iwamura, M., and Kise, K. (2013). Where are you looking at? feature-based eye tracking on unmodified tablets. In *Pattern Recognition (ACPR), 2013 2nd IAPR Asian Conference on*, pages 738–739. IEEE.

Itti, L. and Koch, C. (2001). Computational modelling of visual attention. *Nature reviews neuroscience*, 2(3):194–203.

Itti, L., Koch, C., Niebur, E., et al. (1998). A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. on PAMI*, 20(11):1254–1259.

Jacob, R. J. and Karn, K. S. (2003). Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. *Mind*, 2(3):4.

Jensen, H. W., Marschner, S. R., Levoy, M., and Hanrahan, P. (2001). A practical model for subsurface light transport. In *Proc. of the 28th annual conference on Computer graphics and interactive techniques*, pages 511–518. ACM.

Johnson, D. and Wiles, J. (2003). Effective affective user interface design in games. *Ergonomics*, 46(13-14):1332–1345.

Jones, G. R., Lee, D., Holliman, N. S., and Ezra, D. (2001). Controlling perceived depth in stereoscopic images. In *Photonics West 2001-Electronic Imaging*, pages 42–53. International Society for Optics and Photonics.

Ju, E. and Wagner, C. (1997). Personal computer adventure games: their structure, principles, and applicability for training. *ACM SIGMIS Database*, 28(2):78–92.

Judd, T., Ehinger, K., Durand, F., and Torralba, A. (2009). Learning to predict where humans look. In *ICCV*, pages 2106–2113. IEEE.

Kaneko, T., Takahei, T., Inami, M., Kawakami, N., Yanagida, Y., Maeda, T., and Tachi, S. (2001). Detailed shape representation with parallax mapping. In *Proc. of ICAT*, volume 2001, pages 205–208.

Kellnhofer, P., Ritschel, T., Myszkowski, K., and Seidel, H.-P. (2013). Optimizing disparity for motion in depth. In *Computer Graphics Forum*, volume 32, pages 143–152.

Koch, C. and Ullman, S. (1987). Shifts in selective visual attention: towards the underlying neural circuitry. In *Matters of Intelligence*, pages 115–141. Springer.

Koffka, K. (1935). *Principles of Gestalt psychology.* Harcourt, Brace and World, New York, NY.

Konrad, J. and Halle, M. (2007). 3-d displays and signal processing. *IEEE Signal Processing Magazine*, 6(24):97–111.

Koulieris, G. A., Drettakis, G., Cunningham, D., and Mania, K. (2014a). An automated high level saliency predictor for smart game balancing. *ACM Trans. on Applied Perception (TAP)*, 11(4).

Koulieris, G. A., Drettakis, G., Cunningham, D., and Mania, K. (2014b). C-LOD: Context-aware material level-of-detail applied to mobile graphics. *Computer Graphics Forum (Proc. EGSR)*, 33(4):41–49.

Koulieris, G. A., Drettakis, G., Cunningham, D., and Mania, K. (2014c). High level saliency prediction for smart game balancing. In *ACM SIGGRAPH 2014 Talks*, page 73. ACM.

Koulieris, G. A., Drettakis, G., Cunningham, D., Sidorakis, N., and Mania, K. (2014d). Context-aware material selective rendering for mobile graphics. In *ACM SIGGRAPH 2014 Posters*, page 92. ACM.

Kuhn, G. and Findlay, J. M. (2010). Misdirection, attention and awareness: inattentional blindness reveals temporal relationship between eye movements and visual awareness. *The Quarterly Journal of Experimental Psychology*, 63(1):136–146.

Laird, J. and VanLent, M. (2001). Human-level AI's killer application: Interactive computer games. *AI magazine*, 22(2):15.

Lambooij, M., Fortuin, M., Heynderickx, I., and IJsselsteijn, W. (2009). Visual discomfort and visual fatigue of stereoscopic displays: a review. *J. of Imaging Sci. and Tech.*, 53(3).

Lambooij, M., IJsselsteijn, W., and Heynderickx, I. (2011). Visual discomfort of 3DTV: Assessment methods and modeling. *Displays*, 32(4):209–218.

Lang, M., Hornung, A., Wang, O., Poulakos, S., Smolic, A., and Gross, M. (2010). Nonlinear disparity mapping for stereoscopic 3D. *ACM Trans. on Graphics*, 29(4):75.

Lazzaro, N. (2004). Why we play games: Four keys to more emotion without story. *URL: http://www.xeodesign.com.*

Lee, S., Kim, G. J., and Choi, S. (2009). Real-time tracking of visually attended objects in virtual environments and its application to lod. *Visualization and Computer Graphics, IEEE Trans. on*, 15(1):6–19.

Lindholm, E., Kilgard, M. J., and Moreton, H. (2001). A user-programmable vertex engine. In *Proc. of the 28th annual conference on Computer graphics and interactive techniques*, pages 149–158. ACM.

Longhurst, P., Debattista, K., and Chalmers, A. (2006). A GPU based saliency map for high-fidelity selective rendering. In *Proc. of the 4th international conference on Computer graphics, virtual reality, visualisation and interaction in Africa*, pages 21–29. ACM.

Loschky, L. C. and McConkie, G. W. (2000). User performance with gaze contingent multiresolutional displays. In *Proc. of the 2000 Symp. on Eye tracking research & applications*, pages 97–103. ACM.

Luebke, D. and Hallen, B. (2001). *Perceptually driven simplification for interactive rendering*. Springer.

Luebke, D. P. (2003). *Level of detail for 3D graphics*. Morgan Kaufmann.

Mack, A. (2003). Inattentional blindness. looking without seeing. *Current Directions in Psychological Science*, 12(5):180–184.

Mania, K. and Robinson, A. (2003). Simulating spatial assumptions. In *ACM SIGGRAPH 2003 Sketches & Appl.*, pages 1–1. ACM.

Mania, K., Robinson, A., and Brandt, K. R. (2005). The effect of memory schemas on object recognition in virtual environments. *Presence: Teleoperators and Virtual Environments*, 14(5):606–615.

Marr, D. (1982). Vision: A computational investigation into the human representation and processing of visual information. *Inc., New York, NY*.

Masia, B., Wetzstein, G., Didyk, P., and Gutierrez, D. (2013). A survey on computational displays: Pushing the boundaries of optics, computation, and perception. *Computers & Graphics*, 37(8):1012–1038.

McNamara, A., Mania, K., Koulieris, G., and Itti, L. (2014). Attention-aware rendering, mobile graphics and games. In *ACM SIGGRAPH 2014 Courses*, page 6. ACM.

Meesters, L. M., IJsselsteijn, W. A., and Seuntiens, P. J. (2004). A survey of perceptual evaluations and requirements of three-dimensional TV. *Circuits and Systems for Video Technology, IEEE Trans. on*, 14(3):381–391.

Mendiburu, B. (2012). *3D movie making: stereoscopic digital cinema from script to screen*. CRC Press.

Miluzzo, E., Wang, T., and Campbell, A. T. (2010). Eyephone: activating mobile phones with your eyes. In *Proc. of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds*, pages 15–20. ACM.

Moehring, M., Gloystein, A., and Doerner, R. (2009). Issues with virtual space perception within reaching distance: Mitigating adverse effects on applications using hmds in the automotive industry. In *Virtual Reality Conference, 2009. VR 2009. IEEE*, pages 223–226. IEEE.

Moore, C. M. (2001). Inattentional blindness: Perception or memory and what does it matter. *Psyche*, 7(2).

Mourkoussis, N., Rivera, F. M., Troscianko, T., Dixon, T., Hawkes, R., and Mania, K. (2010). Quantifying fidelity for virtual environment simulations employing memory schema assumptions. *ACM Trans. on Applied Perception (TAP)*, 8(1):2.

Mun, S., Park, M.-C., Park, S., and Whang, M. (2012). Ssvep and erp measurement of cognitive fatigue caused by stereoscopic 3D. *Neuroscience letters*, 525(2):89–94.

Nakayama, K., Silverman, G. H., et al. (1986). Serial and parallel processing of visual feature conjunctions. *Nature*, 320(6059):264–265.

Neisser, U. (1979). The control of information pickup in selective looking. *Perception and its development: A tribute to Eleanor J. Gibson*, pages 201–219.

Neisser, U. and Becklen, R. (1975). Selective looking: Attending to visually specified events. *Cognitive psychology*, 7(4):480–494.

O'Craven, K. M., Downing, P. E., and Kanwisher, N. (1999). fMRI evidence for objects as the units of attentional selection. *Nature*, 401(6753):584–587.

Oskam, T., Hornung, A., Bowles, H., Mitchell, K., and Gross, M. H. (2011). Oscam-optimized stereoscopic camera control for interactive 3D. *ACM Trans. on Graphics*, 30(6):189.

Owens, J. (2005). Streaming architectures and technology trends. In *ACM SIGGRAPH 2005 Courses*, page 9. ACM.

Oyekoya, O., Steptoe, W., and Steed, A. (2009). A saliency-based method of simulating visual attention in virtual scenes. In *Proc. of the 16th ACM Symp. on Virtual Reality Software and Technology*, pages 199–206. ACM.

Pagulayan, R. J., Keeker, K., Wixon, D., Romero, R. L., and Fuller, T. (2003). User-centered design in games. *The human-computer interaction handbook: fundamentals, evolving technologies and emerging applications*, pages 883–906.

Palmer, S. E. (1999). *Vision science: Photons to phenomenology.* The MIT press.

Pappas, J. M., Fishel, S. R., Moss, J. D., Hicks, J. M., and Leech, T. D. (2005). An eye-tracking approach to inattentional blindness. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 49, pages 1658–1662. SAGE Publications.

Peters, R. J. and Itti, L. (2008). Applying computational tools to predict gaze direction in interactive visual environments. *ACM Trans. on Applied Perception (TAP)*, 5(2):9.

Posner, M. I. and Cohen, Y. (1984). Components of visual orienting. *Attention and performance X: Control of language processes*, 32:531–556.

Rayner, K. (2009). Eye movements and attention in reading, scene perception, and visual search. *The quarterly journal of experimental psychology*, 62(8):1457–1506.

Rensink, R. A. (2000). The dynamic representation of scenes. *Visual cognition*, 7(1-3):17–42.

Rensink, R. A. (2002). Change detection. *Annual review of psychology*, 53(1):245–277.

Rensink, R. A., O'Regan, J. K., and Clark, J. J. (1997). To see or not to see: The need for attention to perceive changes in scenes. *Psychological science*, 8(5):368–373.

Richards, A., Hannon, E. M., and Vitkovitch, M. (2012). Distracted by distractors: Eye movements in a dynamic inattentional blindness task. *Consciousness and cognition*, 21(1):170–176.

Rothkopf, C. A., Ballard, D. H., and Hayhoe, M. M. (2007). Task and context determine where you look. *Journal of vision*, 7(14):16.

Salvucci, D. D. and Goldberg, J. H. (2000). Identifying fixations and saccades in eye-tracking protocols. In *Proc. of the 2000 Symp. on Eye Tracking Res. & Appl.*, pages 71–78. ACM.

Schank, R. C. and Abelson, R. P. (2013). *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures.* Psychology Press.

Secord, A., Lu, J., Finkelstein, A., Singh, M., and Nealen, A. (2011). Perceptual models of viewpoint preference. *ACM Trans. on Graphics*, 30(5):109.

Shibata, T., Kim, J., Hoffman, D. M., and Banks, M. S. (2011). The zone of comfort: Predicting visual discomfort with stereo displays. *Journal of vision*, 11(8):11.

Shipley, T. F. and Kellman, P. J. (2001). *From fragments to objects: Segmentation and grouping in vision*, volume 130. Elsevier.

Sidorakis, N., Koulieris, G., and Mania, K. (2015). Binocular eye-tracking for the control of a 3d immersive multimedia user interface. In *Everyday Virtual Reality (WEVR), 2015 IEEE 1st Workshop on*, pages 15–18.

Simons, D. J. and Chabris, C. F. (1999). Gorillas in our midst: Sustained inattentional blindness for dynamic events. *Perception-London*, 28(9):1059–1074.

Simons, D. J., Franconeri, S. L., and Reimer, R. L. (2000). Change blindness in the absence of a visual disruption. *Perception-London*, 29(10):1143–1154.

Simons, D. J. and Levin, D. T. (1997). Change blindness. *Trends in cognitive sciences*, 1(7):261–267.

Simons, K. (1981). A comparison of the frisby, random-dot e, tno, and randot circles stereotests in screening and office use. *Archives of ophthalmology*, 99(3):446–452.

Slater, M., Spanlang, B., and Corominas, D. (2010). Simulating virtual environments within virtual environments as the basis for a psychophysics of presence. *ACM Trans. on Graphics*, 29(4):92.

Stanton, M., Humberston, B., Kase, B., O'Brien, J. F., Fatahalian, K., and Treuille, A. (2014). Self-refining games using player analytics. *ACM Trans. on Graphics*, 33(4).

Steinmetz, P. N., Roy, A., Fitzgerald, P., Hsiao, S., Johnson, K., and Niebur, E. (2000). Attention modulates synchronized neuronal firing in primate somatosensory cortex. *Nature*, 404(6774):187–190.

Stelmach, L. B., Tam, W. J., Speranza, F., Renaud, R., and Martin, T. (2003). Improving the visual comfort of stereoscopic images. In *Electronic Imaging 2003*, pages 269–282. International Society for Optics and Photonics.

Sundstedt, V., Bernhard, M., Stavrakis, E., Reinhard, E., and Wimmer, M. (2013). Visual attention and gaze behavior in games: An object-based approach. In *Game Analytics*, pages 543–583. Springer.

Sundstedt, V., Chalmers, A., Cater, K., and Debattista, K. (2004). Top-down visual attention for efficient rendering of task related scenes. In *International Workshop on Vision, Modeling and Visualization*, pages 209–216.

Sundstedt, V., Debattista, K., Longhurst, P., Chalmers, A., and Troscianko, T. (2005). Visual attention for efficient high-fidelity graphics. In *Proc. of the 21st spring conference on Computer graphics*, pages 169–175. ACM.

Sundstedt, V., Stavrakis, E., Wimmer, M., and Reinhard, E. (2008). A psychophysical study of fixation behavior in a computer game. In *Proc. of the 5th Symp. on Applied perception in graphics and visualization*, pages 43–50. ACM.

Sweetser, P. and Wyeth, P. (2005). Gameflow: a model for evaluating player enjoyment in games. *Computers in Entertainment (CIE)*, 3(3):3–3.

Tatarchuk, N. (2006). Dynamic parallax occlusion mapping with approximate soft shadows. In *Proc. of the 2006 Symp. on Interactive 3D graphics and games*, pages 63–69. ACM.

Tatler, B. W., Hayhoe, M. M., Land, M. F., and Ballard, D. H. (2011). Eye guidance in natural vision: Reinterpreting salience. *Journal of vision*, 11(5):5.

Templin, K., Didyk, P., Myszkowski, K., Hefeeda, M. M., Seidel, H.-P., and Matusik, W. (2014). Modeling and optimizing eye vergence response to stereoscopic cuts. *ACM Trans. on Graphics*, 33(4):145.

Theeuwes, J. (2010). Top–down and bottom–up control of visual selection. *Acta psychologica*, 135(2):77–99.

Theeuwes, J. and Godijn, R. (2002). Irrelevant singletons capture attention: Evidence from inhibition of return. *Perception & Psychophysics*, 64(5):764–770.

Tolhurst, D. J., Movshon, J. A., and Dean, A. (1983). The statistical reliability of signals in single neurons in cat and monkey visual cortex. *Vision research*, 23(8):775–785.

Treisman, A. M. and Gelade, G. (1980). A feature-integration theory of attention. *Cognitive psychology*, 12(1):97–136.

Triesch, J., Ballard, D. H., Hayhoe, M. M., and Sullivan, B. T. (2003). What you see is what you need. *Journal of Vision*, 3(1):9.

Troscianko, T., Mourkoussis, N., Rivera, F., Mania, K., Dixon, T., and Hawkes, R. (2007). Memory for objects in virtual environments. *Journal of Vision*, 7(9):763–763.

Udupa, J. K. and Herman, G. T. (1999). *3D imaging in medicine.* CRC press.

Walther, D. and Koch, C. (2006). Modeling attention to salient proto-objects. *Neural Networks*, 19(9):1395–1407.

Ware, C. (2012). *Information visualization: perception for design.* Elsevier.

Williams, M. and Wann, J. (1993). Binocular vision in a virtual world. *Ophthalmic Physiol Opt*, 13:387.

Wolfe, J. M. (1994). Guided search 2.0 a revised model of visual search. *Psychonomic bulletin & review*, 1(2):202–238.

Woo, M., Neider, J., Davis, T., and Shreiner, D. (1999). *OpenGL programming guide: the official guide to learning OpenGL, version 1.2.* Addison-Wesley Longman Publishing Co., Inc.

Wood, E. and Bulling, A. (2014). Eyetab: model-based gaze estimation on unmodified tablet computers. In *Proc. of the Symp. on Eye Tracking Res. and Appl.*, pages 207–210. ACM.

Woods, A. J., Docherty, T., and Koch, R. (1993). Image distortions in stereoscopic video systems. In *IS&T/SPIE's Symp. on Electronic Imaging: Science and Technology*, pages 36–48. International Society for Optics and Photonics.

Wyman, C. (2005). An approximate image-space approach for interactive refraction. *ACM Trans. on Graphics*, 24(3):1050–1053.

Yarbus, A. L., Haigh, B., and Rigss, L. A. (1967). *Eye movements and vision*, volume 2. Plenum press New York.

Ziegler, G., Tevs, A., Theobalt, C., and Seidel, H.-P. (2006). On-the-fly point clouds through histogram pyramids. In *Workshop on Vision, Modeling, and Visualization (VMV 2006)*, pages 137–144.

Zotos, A., Mania, K., and Mourkoussis, N. (2009). A schema-based selective rendering framework. In *Proc. Applied Perception in Graphics and Visualization*, pages 85–92. ACM.