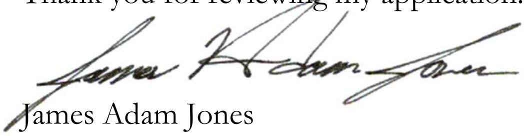
A Note to External Reviewers

Please find attached my dossier and application for promotion and tenure at Mississippi State University. To facilitate navigation through the digital version of this document, I have included a clickable table of contents and embedded bookmarks for each section. In most PDF viewers, these bookmarks will provide an outline-style view, allowing for easy navigation independent of the table of contents.

Please note that some sections listed in the table of contents are not included in the external review version of this document. These sections are either unavailable at this stage of the review process (e.g., Committee Report, External Reviews) or contain confidential information (e.g., Initial Offer Letter).

Thank you for reviewing my application.



James Adam Jones

J. Adam Jones – Promotion & Tenure Dossier

Table of Contents

Application Letter

BAGLEY COLLEGE OF ENGINEERING
Department of Computer Science and Engineering
Mississippi State University
Starkville, MS, USA, 39762
P. (662) 325-2756
office@cse.msstate.edu
www.cse.msstate.edu

Dear Dr. Shahram Rahimi,

Thank you for the opportunity to serve the Department of Computer Science and Engineering (CSE) and Mississippi State University (MSU). My service has led to numerous collaborations, mentoring of talented students, and teaching of relevant, impactful courses. As a result, I am pleased to submit my dossier and application for promotion and tenure. In this letter, I will highlight select accomplishments from my dossier.

Though I have been at MSU for a relatively short time, I believe I have made significant contributions to our missions of teaching, research, and service. Upon joining MSU, I was granted 2 years credit toward tenure for work conducted at the University of Mississippi (UM), so I will also discuss relevant accomplishments for my time there. I have put a great deal of thought, planning, and strategy into my long-term research goals as well as the transition of my lab from UM to MSU. As detailed further in the *Research Accomplishments* chapter, I enacted a transition plan immediately after receiving the offer to join the CSE faculty. This has led to the rapid establishment of external funding, new curricula, research collaborations, and a productive laboratory.

I believe that I bring a unique perspective to our department that translates well into my teaching and research. For instance, I have been fortunate to not only have observed the development of modern virtual reality (VR) but to have directly taken part in its advancement. A few of the experiences that I am particularly proud of include assisting in the development of the first cardboard and 3D-printed VR headsets, collaborating on prototypes that led to the Oculus Rift, and patenting VR techniques that are now considered industry standards. These experiences inform the manner and the content of my classes as well as the direction of my research. I saw early in my career that there are significant advancements to be made in VR research by incorporating practices from neuroscience and ophthalmology. My research focuses on combining these disciplines with computer science and human-centered computing. This research has resulted in 2 patents, 1 provisional patent, 8 journal articles, 16 conference proceedings, 1 book chapter, and numerous abstracts.

Since beginning a tenure-track appointment, I aided in securing $2,007,377 in funding across both MSU and UM. Of this funding, $1,435,427 has come to MSU, including an NSF CAREER award. Since joining MSU, I have submitted 21 proposals, 6 of which were funded (3 external, 3 internal). Projects funded at here include:

- *NSF CAREER Award: A Neuro-Ophthalmic Approach to Virtual Reality Research*. $624,999.
- *NSF Convergence Accelerator: Advancement of Driving Technology for Vocational Enablement*. $750,000.
- *MCCTR: Movement Disorders and Cognitive Impairments from SARS-CoV-2 Infection*. $53,928.
- *ORED Undergraduate Research - Creating and Testing a VR Tool Kit for Fall Prevention Training*. $2000.
- *ORED Undergraduate Research - AI2VR: Placing AI into VR to Evaluate Its Performance*. $1500.
- *Ottilie Schillig Teaching Projects: VXR - Virtual and Extended Reality Software Development*. $3000.

My teaching reviews are consistently strong across both institutions and are at least 10 percentage points above departmental averages. I have a record of developing new courses spanning topics from virtual reality to data science to computing fundamentals. I am also currently leading a team of colleagues from across campus in the development of a new, interdisciplinary neuroscience minor.

I have served my research community in a variety of capacities including work as an Associate Editor for two of my field's leading journals and serving on the organizing and program committees for my field's most impactful conferences for more than 10 years. I serve on multiple department and college-level committees covering areas ranging from graduate studies to engineering research directions.

Sincerely,

J. Adam Jones, PhD

Promotion and Tenure Form

Revised 07/25/2023

Mississippi State University
Application for Promotion and/or Tenure

Please check response(s) in both columns	
TENURE:	**PROMOTION:**
☒ Mandatory tenure decision ☐ Not applicable (early promotion or professional track position or already possess tenure)	☐ Promotion to Instructor II ☐ Promotion to Instructor III ☒ Promotion to Associate Professor ☐ Promotion to Full Professor ☐ Not applicable (only tenure decision)

Faculty members eligible for consideration for promotion or tenure must provide the department head or appropriate official with all pertinent available information by **October 1**. The department head or other appropriate official has the responsibility to assist the faculty member in preparing for tenure or promotion review.

Materials to be provided in the applicant's dossier include:

*1. Cover letter from the candidate requesting promotion and/or tenure.
*2. Completed University Promotion and Tenure application form (this cover page and attached pages) with appropriate responses and associated documentation. This must include a summary sheet of teaching evaluations.
*3. Complete up-to-date vita.
*4. Copy of the initial offer letter and, if necessary, an additional letter detailing significant changes.
*5. Letters from external reviewers (to be added by the department head). The department head should include a sample letter sent to external reviewers and biographical information about reviewers as appropriate.
6. All materials required by the academic unit's procedural guidelines.
7. All supporting documentation desired by the candidate.

*Only these items will be reviewed routinely above the college level. Items 1-7 must go to dept. head and dept. committee. Deans, college committees, and the Provost require items 1-5 but may also request items 6 and 7. Department heads and deans can use their discretion in sending forward any important information included in items 6 and 7. All department head, dean, and committee recommendations should be included in the package to the Provost.

Note: Please refer to the Faculty Handbook for information pertaining to the Promotion & Tenure process.
To apply and be considered for tenure requires that you be a citizen of the United States or be a permanent resident or have begun the permanent residence process (verification required) in order to be eligible for permanent employment in this country.

Are you a citizen or permanent resident of the United States: ☒ Yes ☐ No

If No, have you applied for permanent residency: ☐ Yes ☐ No (Date process initiated (if Yes): _____________)

Name of Applicant: James Adam Jones

Present rank: Assistant Professor Date of appointment at current rank: August 2021

College/School: Bagley College of Engineering Department: Computer Science and Engineering

Department Head: Shahram Rahimi

Preferred Mailing Address (Include City and Zip Code): 208 Oakmont Rd., Starkville, MS, 39759

Initial rank at MSU with date of appointment: Assistant Professor, August 2021

Tenure track date of appointment: August 2021 Years of transferred service (if applicable): 2

Advanced Degrees with Dates: PhD (12/2011), MS (08/2007), BS (12/2004), AA (05/2002)

Salary Funding (%): E&G: 100% MSU Research Unit: _______ Extension: ______ Other: ______

All other information contained in the attached application is correct to the best of my knowledge.

Date: August 7, 2024 Signed: ______________________
Faculty Member

I. **Current Fall semester responsibilities**:

A. Current instruction

	Course number	Title	Credit hours	Number of students
1. Undergraduate:	CSE 4433	Virtual and Extended Reality Development	3	29
	CSE 4000	Directed Individual Study	3	1
	CSE 4800	Undergraduate Research	0	4
2. Graduate:	CSE 6433	Virtual and Extended Reality Development	3	5
	CSE 9000	Dissertation or Thesis	3+	5

3. Advisees: Undergraduate: 0 Master's/Specialist: 4 Doctoral: 2 Postdoctoral: 0

4. Non-credit educational programs (documented, non-credit instruction/teaching with student assessment).

   a. Adult audiences:

   b. Youth audiences:

      - 2024, Starkville-Oktibbeha School District – Summer Enrichment Program, Virtual Reality World Building (Grades 5-8), 20 students – *NSF CAREER Award sponsored outreach*

   c. Professional and technical audiences:

   d. In-service trainings/assistance:

B. Current or on-going research/creative/performance activities

My current or on-going research is closely related to the work the led to my NSF CAREER award. These are research directions that I am directly engaged with, but it is also important to note that I do plan on continuing to support colleagues in their research efforts as well. Further details about these areas as well as current research supporting them are provided in the Research Accomplishments section of my dossier.

Ophthalmic Schematic Eye Models for VR: My lab is incorporating VR rendering techniques and ophthalmic schematic eye models, used in clinical practice to closely approximate the anatomy and optics of the human eye, to improve the accuracy of virtual environment registration. This addresses a critical issue that hinders the real-world applicability of VR and AR. Current research suggests that achieving registration below 5mm may be challenging with existing pupillary distance (PD) based eye models. I suggest that this limitation is due to inadequate ocular modeling. While using pupils as a convenient approximation for rendering realistic VR is appealing, it becomes problematic due to the eye's complex nature. The eye has separate optical, rotational, and physical centers that are not coincident. Furthermore, the eye's nodal point, which serves as the optical center of projection, moves along multiple axes during eye rotation. It is important to note that the nodal point is an internal feature of the eye and cannot be directly observed. This makes using an ophthalmic modeling approach a promising way of approximating its position.

Misalignment of Display Optics and the Eye: My preliminary studies suggest that achieving a high degree of geometric fidelity is possible in VR when accurately modeling the eye, but this scenario is not realistic for most immersive environments. In practice, misalignment between the observer's eyes and headset optics is common, especially in headsets with limited or no PD adjustment options. This misalignment can lead to image distortions due to disparities between the display optics and ocular optics. As such, we must ask how much error is introduced to the eye's retinal projection when there is a misalignment between these two optical systems? Preliminary data shows image distortions of up to 3° for a reasonable range of eye separations (64±5mm) when the observer's PD did not match that of the headset. This misalignment can potentially cause disorientation, misperceived motion, or simulator sickness as observers shift their gaze or move within a virtual scene.

Localizing Neural Activity While in VR: At the heart of VR perception research lies a pivotal question: Does the human brain process VR in the same way as it does the real, physical world? If not, what factors contribute to this disparity? These are questions that can only be answered from the neural perspective. Studying neural activity patterns during spatial tasks can provide valuable insights into the differences observed between virtual and real environments. Other research, utilizing

PET imaging, revealed distinct activation patterns in the dorsal and ventral visual streams when pointing to objects within arm's reach (near) or beyond (far). We are investigating this specific phenomenon in order to identify similarities and differences in patterns of neural activity for participants viewing real or virtual environments. Furthermore, factors like field of view (FOV) size and peripheral visual stimulation are known to significantly impact spatial performance in VR, but their underlying mechanisms remain unclear. By using the difference between the dorsal and ventral streams as a measurement tool, which naturally distinguishes between near and far spaces, it may be possible to investigate whether visual factors like FOV are affecting perceived spatial boundaries. This involves determining the depths at which the transition between streams occurs. Understanding these relationships can shed light on the neural basis of spatial perception in virtual environments and the real world.

<u>Low-Persistence & Motion Quality in VR:</u> Low-persistence, a method to reduce motion blur in VR, is shockingly understudied. It involves displaying images on a headset's screen for a very brief period of time before it is replaced by a black screen, leading to more darkness than illumination. Although every modern headset uses this technique, its benefits to users have not been formally quantified in the literature. Existing research on low-persistence is scarce, and explanations for its benefits are conflicting. Some attribute the improvement to reduced pixel ghosting in hardware, while I find a perceptually motivated explanation more plausible. I believe it is likely that this explanation relates to persistence of vision, a phenomenon where light stimulation continues to be perceived in a location after its source has moved, causing perceived blurring. This is analogous to the photographic trend of "light painting" with long-exposure photography. Breaking up image presentation reduces this phenomenon, resulting in a series of distinct images being perceived as opposed to a continuous blur. Despite limited literature on the topic, low-persistence is widely used and often considered desirable in VR headsets. Since most headsets implement it, researchers should better understand why it works. Even researchers who do not specifically study motion blur should be mindful of low-persistence, as it is inherently used in their work by default.

<u>Evaluating Artificial Intelligence with Virtual Reality:</u> Recent advancements in Artificial Intelligence (AI) have shown significant promise in improving many facets of science and engineering. However, we do not fully understand its limitations and capabilities. In fact, it is very difficult to quantify how well AI can perform some tasks. We are investigating a novel way to rigorously evaluate a specific type of AI known as a Neural Radiance Field (NeRF). NeRFs are artificial neural networks that can reconstruct 3D scenes based on a small number of photographs. However, are these AIs providing accurate information and within what bounds can their results be trusted? These are open questions that must be addressed if we are to trust and effectively use these new technologies. One of the major challenges in validating NeRFs is that their results must be compared against known geometries in the real-world. Comparisons against the real-world are problematic as they are only as valid as your physical measurements. Consistency and reliability of physical measurements is difficult to ensure and relies largely on how your real-world measurements are taken. However, in virtual reality (VR) the world is synthetically generated and stored within a computer. As a result, we have exact, unambiguous measurements of all geometry in VR. We are putting NeRFs into VR to test their performance within a fully quantifiable virtual world. This is an approach which has not previously been used to evaluate NeRFs or similar AIs.

C. <u>Current service/administrative assignments</u>

1. Public service and off-campus professional service activities (non-assessment activities, such as guest lectures and presentations, external committee/board memberships, business/industry/stakeholder advisement, etc. with dates, organizations, & places):

- Associate Editor, IEEE Transactions on Visualization and Computer Graphics, 2022 - Present
- Associate Editor, Frontiers in Virtual Realty – Perception and Human Behaviour, 2022 – Present
- Advisory Board Member, NSF Building Capacity in STEM Education Research, Award #2125377, 2021 - Present

2. Professional association service, as offices held, etc.:
   - 2022 – Present, Faculty Sponsor for MSU Chapter of the Association of Computing Machinery (ACM), 146 members.

3. University and departmental committee and administrative accomplishments:

***For committee and service accomplishments prior to 2023, please see Section II.3.C of this form.

<u>COMMITTEES:</u>

2024.1	**[MSU]** Bagley College of Engineering Research Advisory Council, 2024
2024.2	**[MSU]** CSE Graduate Committee, 2024
2024.3	**[MSU]** Hackathon 2024 Planning Committee, 2024

2023.1 **[MSU]** Bagley College of Engineering Graduate Student Award Committee, 2023
2023.2 **[MSU]** Hackathon 2023 Planning Committee, 2023
2023.3 **[MSU]** CSE Undergraduate Committee, 2023
2023.4 **[MSU]** ABET Assessment Committee, 2023

D. Other

II. **Activities since last promotion (or initial appointment for tenure):**

A. Teaching

1. Evidence of quality of instruction, both credit and non-credit (check items submitted):
(The faculty member should provide material describing their teaching activities and documentation supporting effectiveness. This material must include a summary statement of student survey responses and may include any of the following or any other items deemed appropriate:

☒ peer evaluations (internal or external),

☒ course syllabi and exams,

☒ non-credit education program plans with assessment,

☐ non-credit education program outcomes and impacts,

☒ student input in the form of letters, emails, faculty nominations, etc.,

☐ recordings of teaching sessions, graduate student theses and dissertations, and other materials demonstrating teaching effectiveness.)

2.

Number of Students Supervised	**Major Professor**	**Minor Professor**
Undergraduate Students	Total: 60 0 – Mississippi State 60 – University of Mississippi	
Undergraduate Research	Total: 42 11 – Mississippi State 31 – University of Mississippi	
Clinical Interns & Residents		
Master's Students	Total: 12 5 – Mississippi State 7 – University of Mississippi	
Specialist Students		
Doctoral Students	Total: 3 2 – Mississippi State 1 – University of Mississippi	
Postdoctoral Students		
Visiting Scientist		

3. Courses initiated or innovations instituted:

*** Courses from Mississippi State University are prefaced with **[MSU]**. Courses from University of Mississippi are prefaced with **[UM]**.

NEW COURSES INITIATED:

- **[MSU]** CSE 4433: Virtual and Extended Reality Development
- **[UM]** Csci 447: Immersive Media
- **[UM]** Csci 343: Fundamentals of Data Science
- **[UM]** Csci 581: Special Topics in Human-Centered Computing
- **[UM]** Csci 581: Special Topics in Virtual Reality (precursor to Csci 447)
- **[UM]** Engr 691: Special Topics in Physiological Considerations for Virtual Environments
- **[UM]** Engr 691: Special Topics in Spatial Perception in Virtual Environments

REDESIGNED COURSES:

- **[MSU]** CSE 2383: Data Structures and Analysis of Algorithms
- **[UM]** Engr 694: Research Methods in Computer Science
- **[UM]** Csci 391: Introduction to Computer Graphics
- **[UM]** Csci 390: Special Topics in C/C++ Programming

4. Non-credit educational programs initiated or instituted (documented, non-credit instruction/teaching with student assessment, such as certification programs, short courses, workshops, in-service trainings, workshops, etc.):

5. Other (academic advisement may be described here or as service):

B. Research, creative endeavor, or performances

1. Publications, performances or creative activities:
(For books, indicate date of publication and publisher; for articles, indicate refereed journals; for art shows, indicate judged competition; for musical shows, attach copies of programs; for reports, indicate those done for in-house use.)

***All **[Journal], [Conference], [Workshop],** and **[Abstract]** entries are peer-reviewed. **[Book Chapters]** are **not** peer-reviewed.

PATENTS & APPLICATIONS:

2023.1 U.S. Provisional Patent Application #63/586,503. *DriVR: A Virtual Reality System*. Inventors: **J. Adam Jones**, Kasee Stratton-Gadke, Lalitha Dabbiru, Zacheus Ahonle, Kris Geroux. Filed: September 29, 2023.

2019.1 U.S. Patent #US10416453B2. *Control of Ambient and Stray Lighting in a Head Mounted Display.* Inventors: Mark Bolas, **J. Adam Jones**, David M. Krum. Applicant: University of Southern California. Granted: September 17, 2019.

2017.1 U.S. Patent #US009645395B2. *Dynamic Field of View Throttling as a Means of Improving User Experience in Head Mounted Virtual Environments.* Inventors: Mark Bolas, **J. Adam Jones**, Ian McDowall, Evan Suma. Applicant: University of Southern California. Granted: May, 9, 2017.

PUBLICATIONS:

2024.1 **[Journal]** Joao Paulo Oliveira Marum, H. Conrad Cunningham, **J. Adam Jones**, Yi Liu. 2024. *Following the Writer's Path to the Dynamically Coalescing Reactive Chains Design Pattern*. Algorithms – Special Issue on Algorithms for Virtual and Augmented Environments. 17(2), p.56.

2024.2 **[Conference]** Katharine E. Johanesen, **J. Adam Jones**, Territa Poole, Katherine Ryker, Christopher Green. 2024. *VR Geoscience Education: Building Spatial Reasoning Skills*. IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops, pp. 167-172.

2024.3 **[Journal]** Hunter Derby, Nathan Conner, Jacob Hull, Faith Hagan, Sally Barfield, Rosemary Boland, Timothy Stewart, **J. Adam Jones**, Adam Knight, Harish Chander. 2024. *Effects of Acute Virtual Reality Exposure on Dynamic Postural Stability.* International Journal of Exercise Science: Conference Proceedings. 16(3). p.75.

2024.4 **[Abstract] J. Adam Jones**, W. Woody Watson, Timothy Stewart, Jacob Brewington, Connor Chrismond, Lalitha Dabbiru, Zaccheus J Ahonle, Emily S Wall, Kris Geroux, Kasee Stratton-Gadke. 2024. *DriVR: Extending Driver Training for Persons with Disabilities.* IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops, pp. 1218-1219.

2024.5 **[Abstract]** Ander Talley, **J. Adam Jones**. 2024. *Evaluating NeRF Fidelity using Virtual Environments*. 2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops, pp. 1027-1028.

2023.1 **[Journal]** Hunter Derby, Nathan O. Connor, Jacob M. Hull, Faith Hagan, Sally Barfield, Timothy Stewart, **J. Adam Jones**, Adam C. Knight, Harish Chander. 2023. *Effects of Acute Exposure to Virtually Generated Slip Hazards during Overground Walking*. Applied Sciences, 13(23), p.12848.

2023.2 **[Journal]** Hannah R. Freeman, Harish Chander, Sachini N. K. Kodithuwakku Arachchige, Alana J. Turner, **J. Adam Jones**, Zhujun Pan, Christopher Hudson, Adam C. Knight. 2023. *Postural Control Behavior in a Virtual Moving Room Paradigm*. Biomechanics, 3(4), pp.539-551.

2023.3 **[Abstract] J. Adam Jones**. 2023. *The Pre-History (and Future) of Virtual Reality.* Proceedings of the Mississippi Academy of Science, 1 page.

2022.1 **[Journal]** Nathan O. Conner, Hannah R. Freeman, **J. Adam Jones**, Tony Luczak, Daniel Carruth, Adam C. Knight, Harish Chander. *Virtual Reality Induced Symptoms and Effects: Concerns, Causes, Assessment & Mitigation*. Virtual Worlds, 17 pages.

2022.2 **[Journal]** Harish Chander, Hannah R Freeman, Christopher M Hill, Christopher Hudson, Sachini N. K. Kodithuwakku Arachchige, Alana J Turner, **J. Adam Jones**, Adam C Knight. 2022. *The Walls are Closing in: Postural Responses to a Virtual Reality Claustrophobic Simulation*. Clinical and Translational Neuroscience, 15 pages.

2022.3 **[Conference]** Anthony Luczak. Charles Freeman, Reuben Burch, David Saucier, Harish Chander, John Barlow, John Ball, Patrick Nelsen, E. Parker, C. Middleton, Leslie Strawderman, J. Mohammadi-Aragh, Zachery Gillen, B. Smith, **J. Adam Jones**, Martin Duclos, Michael Taquino, Steven Grice. *Athlete Engineering BaseLine Ecosystem: innovative technologies to enhance human performance.* TechConnect World Innovation Conference, Washington DC, page 136-140.

2022.4 **[Workshop]** Anthony Luczak, **J. Adam Jones**, Reuben Burch, Jonathan Barlow, Patrick Nelsen, Steven M. Grice, Michael Taquino, Martin Duclos, Caleb Morgan. 2022. *Advancing tangible augmented game objects for use in a golf swing, self-service training environment: Report of Work-in-Progress with a Multidisciplinary Emphasis*. IEEE Virtual Reality Workshop on Augmented Reality Enabling Superhuman Sports + Serious Games (IEEE VR – ARES), Christchurch, New Zealand, pages: 136 - 140.

2022.5 **[Abstract]** Tony Luczak, Kait Jackson, Long Tian, **J. Adam Jones**, Reuben Burch, Jonathan Barlow, Patrick Nelsen, Caleb Morgan, Harish Chander, Martin Duclos, Michael Taquino, and Steven M. Grice. *The Challenges of Coaching Using 2D Golf Swing Video Data Compared to the Challenges of Building a 3D Technology Based Coach*. World Scientific Congress of Golf 2022. 1 page.

2022.6 **[Abstract]** Zachery Garris, Maggie Pettus, Daniel Molsbarger, Harish Chander, **J. Adam Jones**. 2022. *EEG Electrode Localization Using Off-the-shelf Virtual Reality Systems*. Proceedings of the Mississippi Academy of Sciences Annual Meeting.

2022.7 **[Abstract]** Dylan Devenny, Harish Chander, **J. Adam Jones**. 2022. *The Influence of Optical Flow on the Perception of Impossible Spaces in Virtual Reality*. Proceedings of the Mississippi Academy of Sciences Annual Meeting.

2022.8 **[Abstract]** Jonathan Hopper, David M. Krum, **J. Adam Jones**. 2022. *Perception of Target Eccentricity and Depth in Virtual Reality*. Proceedings of the Mississippi Academy of Sciences Annual Meeting.

2020.1 **[Conference]** Benjamin Creel, Caitlin Rinz-Jones, Colin Jackson, **J. Adam Jones.** 2020. *Bacterial Load of Virtual Reality Headsets*. ACM Symposium on Virtual Reality Software and Technology (VRST), Ottawa, Canada, pp.1-8.

2020.2 **[Conference]** Hunter Finney, **J. Adam Jones.** 2020. *Asymmetric Effects of the Ebbinghaus Illusion on Depth Judgments*. IEEE Conference on Virtual Reality and 3D User Interfaces (IEEE VR), Atlanta, GA, pages 573 - 578.

2020.3 **[Conference]** João Paulo Oliveira Marum, H. Conrad Cunningham, **J. Adam Jones**. 2020. *Unified Library for Dependency Graph Reactivity on Web and Desktop User Interfaces*. ACM Southeast Conference (ACMSE), Tampa, FL, USA, pages: 26 - 33.

2020.4 **[Workshop]** Collin Roth, Ethan Luckett, **J. Adam Jones**. 2020. *Latency Detection and Illusion in a Head-Worn Virtual Environment.* IEEE Virtual Reality Workshop on Perceptual and Cognitive Issues in Augmented Reality (PERCAR), Atlanta, GA, pages: 215 - 218.

2020.5 **[Workshop]** João Paulo Oliveira Marum, H. Conrad Cunningham, **J. Adam Jones**. 2020. *Dependency Graph-based Reactivity for Virtual Environments*. IEEE Virtual Reality Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS), Atlanta, GA, pages: 246 - 253.

2019.1 **[Journal] J. Adam Jones**, Jonathan E. Hopper, Mark T. Bolas, David M. Krum. 2019. *Orientation Perception in Real and Virtual Environment.* IEEE Transactions on Visualization and Computer Graphics (TVCG), 25(5), pages 2050-2060.

2019.2 **[Book Chapter, Invited] J. Adam Jones**, 2019. *A Tinkerer's Perspective on VR Displays*. "VR Developer Gems", AK Peters/CRC Press, ed. William Sherman, May 2019.

2019.3 **[Conference]** João Paulo Oliveira Marum, **J. Adam Jones**, H. Conrad Cunningham. 2019. *Towards a Reactive Game Engine*. IEEE Southeast Conference, Huntsville, AL.

2019.4 **[Conference]** William Panlener, David M. Krum, **J. Adam Jones.** 2019. *Effects of Horizontal Field of View Extension on Spatial Judgments in Virtual Reality*. IEEE Southeast Conference, Huntsville, AL.

2019.5 **[Workshop] J. Adam Jones**. 2019. *Optical and Neural Properties of Vision as Applied to Virtual Reality.* IEEE Virtual Reality Workshop on Neuroscience & Virtuality (NeuroVirt), Osaka, Japan, pages 1667-1670.

2019.6 **[Workshop]** Jonathan E. Hopper, Hunter Finney, **J. Adam Jones**. 2019. *Field of View and Forward Motion Discrimination in Virtual Reality*. IEEE Virtual Reality Workshop on Neuroscience & Virtuality (NeuroVirt), Osaka, Japan, pages 1663-1666.

2019.7 **[Workshop]** Ethan Luckett, Tykeyah Key, Nathan Newsome, **J. Adam Jones**. 2019. *Metrics for the Evaluation of Tracking Systems for Virtual Environments*. IEEE Virtual Reality Workshop on Novel Input Devices and Interaction Techniques (NIDIT), Osaka, Japan, pages 1711-1716.

2019.8 **[Abstract]** Benji Creel, **J. Adam Jones**, Collin R. Jackson. 2019. *Bacterial Load in Virtual Reality Headsets.* Proceedings of the American Society of Microbiology, South Central Branch Meeting. Oxford, MS, page 36.

2019.9 **[Abstract] J. Adam Jones**, Ethan Luckett, Tykeyah Key, Nathan Newsome. 2019. *Latency Measurement in Head-Mounted Virtual Environments*. Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces (IEEE VR), Osaka, Japan, pages 1000-1001.

2019.10 **[Abstract] J. Adam Jones**, Thai Phan. 2019. *A Comparison of Gait Measurement Methods.* Proceedings of the IEEE Southeast Conference, Huntsville, AL.

2019.11 **[Abstract]** Khoa Tran, Yaoxin Zhang, Yafei Jia, **J. Adam Jones**. 2019. *A Prototype Visualization Tool for Hurricane Flood Simulations in Virtual Reality.* Proceedings of the IEEE Southeast Conference, Huntsville, AL.

2018.1 **[Abstract]** McKennon McMillian, Hunter Finney, Jonathan Hopper, **J. Adam Jones**. 2018. *The Depth Light.* Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces (IEEE VR), Reutlingen, Germany, pages 834-835.

2017.1 **[Journal] J. Adam Jones**, David M. Krum, Mark T. Bolas. 2017. *Vertical Field of View Extension and Walking Characteristics in Head-worn Virtual Environments*. ACM Transactions on Applied Perception (ACM TAP). 14, 2, pages 9:1-9:17.

2016.1 **[Conference] J. Adam Jones**, Darlene Edewaard, Richard A. Tyrrell, Larry F. Hodges. 2016. *A Schematic Eye for Virtual Environments*. Proceedings of the IEEE Symposium of 3D User Interfaces, Greeneville, South Carolina, pages 221-230.

2016.2 **[Abstract]** Joao Paulo Marum, **J. Adam Jones**, Conrad H. Cunningham. 2016. *Functional Reactive Augmented Reality: Proof of Concept Using an Extended Augmented Desktop with Swipe Interaction.* Proceedings of the Eurographics Symposium on Virtual Environments, pages 13-14.

2016.3 **[Abstract]** Alexander Gunter, Andrew Robb, **J. Adam Jones**. 2016. *Real-Time Marker Tracking with Microsoft Kinect*. Proceedings of the Eurographics Symposium on Virtual Environments, pages 15-16.

2015.1 **[Conference] J. Adam Jones**, Lauren Cairco Dukes, David, M. Krum, Mark T. Bolas, Larry F. Hodes. 2015. *Correction of Geometric Distortions and the Impact of Eye Position in Virtual Reality Displays*. Proceedings of the International Conference on Collaboration Technologies and Systems, Atlanta, Georgia, pages 77-83.

2015.2 **[Abstract]** Elhman Ebrahimi, Bliss M. Altenhoff, Christopher C. Pagano, Sabarish V. Babu, **J. Adam Jones**. 2015. *Investigating the Impact of Perturbed Visual and Proprioceptive Information in Near-Field Immersive Virtual Environments*. Poster compendium, Proceedings of the IEEE Virtual Reality Conference, Arles Camargue, France, pages 171-172.

2. Professional papers read; indicate whether invited, refereed, or volunteered.
   Cite organization, date, and title:

3. Grants for research or study:

   Proposals submitted since last promotion and total dollar amount: **<u>23 proposals; $11,258,977</u>**

   ***Please note that all proposals submitted while at Mississippi State University are prefaced with **[MSU]** while those submitted at University of Mississippi are prefaced with **[UM]**.

2023.1 **[Funded - MSU]** *CAREER: A Neuro-Ophthalmic Approach to Virtual Reality Research*. National Science Foundation - Faculty Early Career Development Program (CAREER). PI: **J. Adam Jones**. $624,999. (May 2024 – April 2029).

2023.2 **[Funded - MSU]** *Movement Disorders and Cognitive Impairments from SARS-CoV-2 Infection in Older Adults*. Mississippi Center for Clinical and Translational Research. PI: Harish Chander, Co-PIs: Reuben

Burch, David Saucier, **J. Adam Jones**, John E. Ball, David Vandenheever, Jennifer Reneker. $53,928. (July 2023 – June 2025).

2023.3 **[Funded - MSU]** *AI2VR: Placing Artificial Intelligence into Virtual Reality to Evaluate Its Performance*. Mississippi State University – Office of Research and Economic Development. PI: **J. Adam Jones**. $1,500. (Fall 2023).

2023.4 **[Declined - MSU]** *NSF Convergence Accelerator Track H – Phase 2: Advancement of Driving Technology for Vocational Enablement*. National Science Foundation. PI: Kasee Stratton-Gadke, Co-PIs: **J. Adam Jones**, Lalitha Dabbiru, Zacheus Ahonle, Kris Geroux. $4,999,999. (January 2024 – December 2026).

2023.5 **[Decline - MSU]** *NIH-R15 - Multimodal Application for Head Injury Assessment.* National Institute of Health – R15 Research Enhancement Award Program. PI: David Vandenheever. Co-PIs: Harish Chander, **J. Adam Jones**, Jennifer Reneker, Pierre Viviers. $420,375. (July 2023 – June 2027).

2023.6 **[Declined - MSU]** *Movement Disorders and Cognitive Impairments from SARS-CoV-2 Infection in Older Adults*. National Institute of Health – R03 Small Grants Program. PI: Harish Chander, Co-PIs: Reuben Burch, David Vandenheever, David Saucier, **J. Adam Jones**. $142,083. (September 2023 – October 2025).

2023.7 **[Declined - MSU]** *Understanding the social, cognitive, and learning aspects of using a digital co-creation and collaborative education platform: Teaching and Learning in the Metaverse (TLM)*. National Science Foundation. PI: Zack Gillen, Co-PIs: **J. Adam Jones**, John Lamberth, O.P. McCubbins, Sudip Mittal. $399,988. (July 2023 – June 2026).

2022.1 **[Funded - MSU]** *NSF Convergence Accelerator Track H – Phase 1: Advancement of Driving Technology for Vocational Enablement*. National Science Foundation. PI: Kasee Stratton-Gadke, Co-PIs: **J. Adam Jones**, Lalitha Dabbiru, Zacheus Ahonle, Kris Geroux. $750,000. (December 2022 – November 2023).

2022.2 **[Funded - MSU]** *Creating and Testing a Virtual Reality (VR) Tool Kit for Fall Prevention Training*. Mississippi State University – Office of Research and Economic Development. PI: Harish Chander, Co-PIs: **J. Adam Jones**, David Vendenheever. $2,000. (Spring 2023 – Fall 2023).

2022.3 **[Funded - MSU]** *VXR - Virtual and Extended Reality Software Development: Providing Immersive Experiences in the Classroom using Virtual Reality*. MSU Ottilie Schillig Teaching Projects Grant. PI: **J. Adam Jones**. $3,000. (June 2022 – May 2023).

2022.4 **[Declined - MSU]** *CAREER: Fidelity Engineering for Virtual and Extended Reality*. National Science Foundation - Faculty Early Career Development Program (CAREER). PI: **J. Adam Jones**. $574,568. (January 2023 – January 2028).

2022.5 **[Declined - MSU]** *Movement Disorders and Cognitive Impairments from SARS-CoV-2 Infection in Older Adults.* Mississippi Center for Clinical and Translational Research (MCCTR). PI: Harish Chander, Co-PIs: David Vandenheever, Reuben Burch, **J. Adam Jones**, David Saucier, John Ball, Jennifer Reneker. (January 2023 – December 2023).

2022.6 **[Declined - MSU]** *CySP: Social Engineering in the Metaverse.* Department of Defense. PI: **J. Adam Jones**, Co-PIs: Cindy Bethel, Sudip Mittal. $150,000. (July 2022 – June 2023).

2022.7 **[Declined - MSU]** *Incorporation of VALD Health™ in Assessing Fall Risk in Elderly: A Biomechanical and Cognitive Approach*. VALD Health. PI: Harish Chander, Co-PIs: Reuben Burch, David Saucier, **J. Adam Jones**. $39,782. (July 2022 – June 2023).

2022.8 **[Declined - MSU]** *EEG-based Decoding of Conceptual Representations in the Human Brain Using Deep Learning*. Mississippi State University – Advancing Collaborative Research Program. PI: David Vandenheever, Co-PIs: **J. Adam Jones**, Harish Chander, Hossein Karimi. $15,000. (August 2022 – July 2024).

2022.9 **[Declined - MSU]** *NCS-FO: Determining Brain Activity Preceding Free Decisions Using Deep Neural Networks*. National Science Foundation. PI: David Vandenheever, Co-PIs: Hossein Karimi, **J. Adam Jones**. $998,692. (July 2022 – June 2026).

2022.10 **[Declined - MSU]** *Virtual Moving Room Paradigm: A Virtual Reality Tool Kit for Fall Prevention in Elderly*. Mississippi Center for Clinical and Translational Research (MCCTR). PI: Harish Chander. Co-PIs: Reuben Burch, Charles Freeman, David Saucier, Tony Luczak, **J. Adam Jones**, Jennifer Reneker. $39,988. (July 1, 2022 – June 30, 2023).

2022.11 **[Declined - MSU]** *Pre-proposal - Theme 6B: Machine Learning for Skill Acquisition in Individuals with Disabilities (MSAID)*. National Science Foundation. PI: Kasee Stratton-Gadke, Co-PIs: Ali Gurbuz, **J. Adam Jones**, Junfeng Ma, Bo Tang. (January 2023 - December 2027).

2021.1 **[Declined - MSU]** *Understanding the social, cognitive, and learning aspects of using a digital co-creation and collaborative education platform: Teaching and Learning in the Metaverse (TLM)*. National Science Foundation. PI: Zachary Gillen, Co-PIs: John Lamberth, OP McCubbins, Anthony Luczak, **J. Adam Jones**. $849,699. (2022 - 2024).

2021.2 **[Declined - MSU]** *SCH: EEG-based Decoding of Conceptual Representations in the Human Brain Using Deep Learning*. National Science Foundation. PI: David Vandenheever, Co-PIs: Hossein Karimi, **J. Adam Jones**. $1,192,376. (July 2022 - June 2026).

2019.1 **[Funded - UM]** *Virtual Reality Learning Center*, Mississippi Department of Human Services. PI: Albert Nylander, Technical Lead: **J. Adam Jones**. $550,000. FY: 2019-2020.

2018.1 **[Funded - UM]** *Undergraduate Research in Data Science*, University of Mississippi Summer Undergraduate Research Groups Grant (SURGG), Co-PIs: Naeemul Hassan, Dawn E. Wilkins, H. Conrad Cunningham, Yixin Chen, **J. Adam Jones**. $20,450. Summer 2018.

2017.1 **[Declined - UM]** *CHS: Small: Collaborative Research: The Effect of Peripheral Field of View on the Perception of Virtual and Augmented Environments seen in Head-Mounted Displays*, National Science Foundation, Principal Investigators: J. Edward Swan II, **J. Adam Jones**, $249,053. 2017.

2017.2 **[Declined - UM]** *360 and Immersive Journalism Best Practices*, John S. and James L. Knight Foundation, Ji Hoon Heo (Co-PI), James Adam Jones (Co-PI), $33,967. 2017.

2016.1 **[Declined - UM]** *A High Performance Computing Approach to Data Science and Cyber Security at Mississippi's Universities*, EPSCOR Preproposal, Principal Investigators: David Dampier, Sherif Abdelwahed, John A. Hamilton, Maxwell Young, Patrick Pape, Frances Dancer, April Tanner, Philip Rhodes, **J. Adam Jones**, 2016.

2015.1 **[Funded - UM]** *Intel Galileo University Donations.* Intel Higher Education. Awarded to **J. Adam Jones**. 5 Intel Galileo Generation 2 Development Boards and Seed Studio Grove Starter Kits. Estimated value: $1,500. Fall 2015.

Proposals funded (cite source, title of project, role [PI, etc.], $ amount, dates): **9 proposals; $2,007,377**

***Please note that all proposals submitted while at Mississippi State University are prefaced with **[MSU]** while those submitted at University of Mississippi are prefaced with **[UM]**.

2023.1 **[MSU]** *CAREER: A Neuro-Ophthalmic Approach to Virtual Reality Research*. National Science Foundation - Faculty Early Career Development Program (CAREER). PI: **J. Adam Jones**. $624,999. (May 2024 – April 2029).

2023.2 **[MSU]** *Movement Disorders and Cognitive Impairments from SARS-CoV-2 Infection in Older Adults*. Mississippi Center for Clinical and Translational Research. PI: Harish Chander, Co-PIs: Reuben Burch, David Saucier, **J. Adam Jones**, John E. Ball, David Vandenheever, Jennifer Reneker. $53,928. (July 2023 – June 2025).

2023.3 **[MSU]** *AI2VR: Placing Artificial Intelligence into Virtual Reality to Evaluate Its Performance*. Mississippi State University – Office of Research and Economic Development. PI: **J. Adam Jones**. $1,500. (Fall 2023).

2022.1 **[MSU]** *NSF Convergence Accelerator Track H – Phase 1: Advancement of Driving Technology for Vocational Enablement*. National Science Foundation. PI: Kasee Stratton-Gadke, Co-PIs: **J. Adam Jones**, Lalitha Dabbiru, Zacheus Ahonle, Kris Geroux. $750,000. (December 2022 – November 2023).

2022.2 **[MSU]** *Creating and Testing a Virtual Reality (VR) Tool Kit for Fall Prevention Training*. Mississippi State University – Office of Research and Economic Development. PI: Harish Chander, Co-PIs: **J. Adam Jones**, David Vandenheever. $2,000. (Spring 2023 – Fall 2023).

2022.3 **[MSU]** *VXR - Virtual and Extended Reality Software Development: Providing Immersive Experiences in the Classroom using Virtual Reality*. MSU Ottilie Schillig Teaching Projects Grant. PI: **J. Adam Jones**. $3,000. (June 2022 – May 2023).

2019.1 **[UM]** *Virtual Reality Learning Center*, Mississippi Department of Human Services. PI: Albert Nylander, Technical Lead: **J. Adam Jones**. $550,000. FY: 2019-2020.

2018.1 **[UM]** *Undergraduate Research in Data Science*, University of Mississippi Summer Undergraduate Research Groups Grant (SURGG), Co-PIs: Naeemul Hassan, Dawn E. Wilkins, H. Conrad Cunningham, Yixin Chen, **J. Adam Jones**. $20,450. Summer 2018.

2015.1 **[UM]** *Intel Galileo University Donations.* Intel Higher Education. Awarded to **J. Adam Jones**. 5 Intel Galileo Generation 2 Development Boards and Seed Studio Grove Starter Kits. Estimated value: $1,500. Fall 2015.

4. Other:

C. Service

1. Public service, non-assessment activities such as guest lectures and presentations, external committee/board memberships, business/industry/stakeholder advisement, etc. (with dates, organizations, places):

- 2024, Starkville-Oktibbeha School District – Summer Enrichment Program, Virtual Reality World Building (Grades 5-8), 20 students – *NSF CAREER Award sponsored outreach*

- 2022 – Present, Faculty Sponsor for MSU Chapter of the Association of Computing Machinery (ACM), 146 members.

2. Professional association service (offices held, journals edited, etc.):

EDITORIAL SERVICE:

- Associate Editor, IEEE Transactions on Visualization and Computer Graphics, 2022 - Present
- Associate Editor, Frontiers in Virtual Realty – Perception and Human Behaviour, 2022 - Present
- Guest Editor, Frontiers in Virtual Reality – Perception & Neuroscience in XR, 2020 - 2021
- Guest Editor, Multimodal Technologies and Interaction – Perception and Cognition in XR, 2020 - 2021
- Editorial Board, Frontiers in Virtual Reality, 2019 - Present
- Editorial Board, Frontiers in Robotics and AI – Virtual Environments, 2014 - 2019

CONFERENCE ORGANIZATION:

- IEEE Intl. Symposium on Mixed & Augmented Reality, Organizing Committee, Program Co-chair, 2022
- IEEE Virtual Reality Conference, Organizing Committee, Web Co-chair, 2021
- IEEE Virtual Reality Conference, Organizing Committee, Web Co-chair, 2020
- IEEE Virtual Reality Conference, Organizing Committee, Videos Co-chair, 2019
- IEEE Virtual Reality Conference, Organizing Committee, Videos Co-chair, 2018
- IEEE Virtual Reality Conference, Organizing Committee, Videos Co-chair, 2017
- IEEE Virtual Reality Conference, Organizing Committee, Videos Co-chair, 2016
- IEEE Virtual Reality Conference, Organizing Committee, Videos Co-chair, 2014
- IEEE Virtual Reality Conference, Organizing Committee, Videos Co-chair, 2013
- IEEE Virtual Reality Conference, Organizing Committee, Publications Co-chair, 2012
- IEEE Virtual Reality Conference, Organizing Committee, Publications Co-chair, 2011
- IEEE Virtual Reality Conference, Organizing Committee, Publications Co-chair, 2010

PROGRAM COMMITTEES:

- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2020
- IEEE VR Workshop on Perceptual & Cognitive Issues in Augmented Reality, 2020
- IEEE Conference on Virtual Reality and 3D User Interfaces (IEEE VR), 2019
- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2019
- ACM Symposium on Applied Perception (SAP), 2018
- ACM Symposium on Virtual Reality Software & Technology (VRST), 2018
- IEEE Artificial Intelligence and Virtual Reality (AIVR), 2018
- IEEE Conference on Virtual Reality and 3D User Interfaces (IEEE VR), 2018
- IEEE VR Workshop on Perceptual & Cognitive Issues in Augmented Reality (PERCAR), 2017
- IEEE VR Workshop on Collaboration and Virtual Environments (CoVE), 2016
- IEEE VR Workshop on Perceptual & Cognitive Issues in Augmented Reality (PERCAR), 2016
- ACM Symposium on Virtual Reality Software & Technology (VRST), 2016
- ACM Symposium on Virtual Reality Software & Technology (VRST), 2015
- IEEE ISMAR Workshop on Human Perception and Psychology in Augmented Reality (HPPAR), 2015
- IEEE Virtual Reality Conference (IEEE VR), 2015
- IEEE VR Workshop on Collaboration and Virtual Environments (CoVE), 2015
- IEEE VR Workshop on Perceptual & Cognitive Issues in Augmented Reality (PERCAR), 2015

- International Symposium on Visual Computing (ISVC), 2015
- IEEE Virtual Reality Conference (IEEE VR), 2014

JOURNAL REVIEWER:

- ACM Transactions on Applied Perception (ACM TAP), 2023
- IEEE Transactions on Visualization and Computer Graphics (TVCG), 2021
- Frontiers in Virtual Reality, 2022
- Frontiers in Virtual Reality, 2021
- Frontiers in Virtual Reality, 2020
- ACM Transactions on Applied Perception (ACM TAP), 2019
- Frontiers in Robotics and Artificial Intelligence, 2019
- IEEE Transactions on Visualization and Computer Graphics (TVCG), 2019
- ACM Transactions on Applied Perception (ACM TAP), 2018
- Frontiers in Robotics and Artificial Intelligence, 2018
- IEEE Transactions on Visualization and Computer Graphics (TVCG), 2018
- International Journal of Human-Computer Studies (IJHCS), 2018
- Presence: Teleoperators and Virtual Environments, 2018
- University of Mississippi - Undergraduate Research Journal, 2018
- ACM Transactions on Applied Perception (ACM TAP), 2017
- Frontiers in Robotics and Artificial Intelligence, 2017
- IEEE Transactions on Visualization and Computer Graphics (TVCG), 2017
- International Journal of Human-Computer Studies (IJHCS), 2017
- Frontiers in ICT, 2016
- IEEE Transactions on Visualization and Computer Graphics (TVCG), 2016
- Quarterly Journal of Experimental Psychology, 2015
- Presence: Teleoperators and Virtual Environments, 2015
- Frontiers in Virtual Environments, 2014
- ACM Transactions on Applied Perception (ACM TAP), 2014
- International Journal of Human-Computer Studies (IJHCS), 2014
- ACM Transactions on Applied Perception (ACM TAP), 2013
- IEEE Transactions on Visualization and Computer Graphics (TVCG), 2013
- International Journal of Human-Computer Studies (IJHCS), 2013
- International Journal of Human-Computer Studies (IJHCS), 2012
- International Journal of Human-Computer Studies (IJHCS), 2011
- International Journal of Human-Computer Studies (IJHCS), 2010
- International Journal of Human-Computer Studies (IJHCS), 2009
- International Journal of Human-Computer Studies (IJHCS), 2008

CONFERENCE REVIEWER:

- IEEE Conference on Virtual Reality and 3D User Interfaces (IEEE VR), 2020
- IEEE VR Workshop on Workshop on Immersive Sickness Prevention (WISP), 2020
- ACM Symposium on Applied Perception (SAP), 2019
- IEEE VR IEEE VR Workshop on Novel Input Devices and Interaction Techniques (NIDIT), 2019
- ACM Southeast Conference (ACMSE), 2018
- ACM Spatial User Interaction (SUI), 2018
- ACM Special Interest Group on Computer–Human Interaction (SIGCHI), 2018
- ACM Symposium on Applied Perception (SAP), 2018
- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2018
- ACM Special Interest Group on Computer–Human Interaction (SIGCHI), 2017
- IEEE Symposium on 3D User Interfaces (3DUI), 2017
- IEEE Virtual Reality Conference (IEEE VR), 2017
- IEEE Virtual Reality Conference (IEEE VR), 2016
- ACM Spatial User Interaction (SUI), 2015
- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2015
- IEEE Symposium on 3D User Interfaces (3DUI), 2015
- IEEE Virtual Reality Conference (IEEE VR), 2015
- ACM Special Interest Group on Computer-Human Interaction (SIGCHI), 2014
- IEEE Information Visualization, 2014
- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2014
- ACM Special Interest Group on Computer–Human Interaction (SIGCHI), 2013
- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2013
- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2012
- ACM Intl. Conference on Multimodal Interactions (ICMI), 2012
- IEEE Virtual Reality Conference (IEEE VR), 2012
- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2011
- IEEE Virtual Reality Conference (IEEE VR), 2011
- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2010
- IEEE Virtual Reality Conference (IEEE VR), 2010
- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2009
- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2007
- IEEE Virtual Reality Conference (IEEE VR), 2006

REVIEW PANELS:

- National Science Foundation (NSF) review panel, 2016

OTHER PROFESSIONAL SERVICE:

- Session Chair, IEEE Virtual Reality Conference (IEEE VR), 2022
- Session Chair, ACM Symposium on Virtual Reality Software and Technology (VRST), 2020
- Session Chair, IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2020
- Session Chair, IEEE Virtual Reality Conference (IEEE VR), 2014

- Judge, IEEE 3D User Interfaces Contest (IEEE 3DUI), 2013
- Judge, IEEE 3D User Interfaces Contest (IEEE 3DUI), 2012

3. University service (committees, administrative accomplishments, etc.):

******For committee and service accomplishments from 2023 to present, please see Section I.3.C of this form.***
***Please note that all service performed at Mississippi State University is prefaced with **[MSU]** while all service performed at University of Mississippi is prefaced with **[UM]**.

COMMITTEES:

2022.1	**[MSU]** CSE Undergraduate Committee, 2022
2022.2	**[MSU]** CSE Graduate Committee, 2022
2022.3	**[MSU]** ABET Assessment Committee, 2022
2021.1	**[MSU]** CSE Graduate Committee, 2021
2021.2	**[UM]** Computer & Information Science Undergraduate Committee, 2021
2021.3	**[UM]** Neuroscience Core Faculty (Curriculum Committee), 2021
2020.1	**[UM]** Computer & Information Science Undergraduate Committee, 2020
2020.2	**[UM]** Neuroscience Core Faculty (Curriculum Committee), 2020
2019.1	**[UM]** Computer & Information Science Undergraduate Committee, 2019
2019.2	**[UM]** Faculty Search Committee, 2019
2019.3	**[UM]** Neuroscience Core Faculty (Curriculum Committee), 2019
2018.1	**[UM]** Computer & Information Science Undergraduate Committee, 2018
2018.2	**[UM]** Neuroscience Core Faculty (Curriculum Committee), 2018
2017.1	**[UM]** Computer & Information Science Undergraduate Committee, 2017
2017.2	**[UM]** Neuroscience Core Faculty (Curriculum Committee), 2017
2016.1	**[UM]** Faculty Search Committee, 2016
2016.2	**[UM]** Neuroscience Core Faculty (Curriculum Committee), 2016
2009.1	**[MSU]** Imaging Center of Excellence Quality Control Group, 2009
2008.1	**[MSU]** Institute for Neurocognitive Science & Technology Steering Committee, 2008
2008.2	**[MSU]** MSU Imaging Center of Excellence Quality Control Group, 2008
2007.1	**[MSU]** Institute for Neurocognitive Science & Technology Steering Committee, 2007
2007.2	**[MSU]** Imaging Center of Excellence Quality Control Group, 2007

4. Other (academic advisement may be described here or as teaching):

III. Awards and distinctions (title, date, organization):

- Inductee, 2019, Sigma Xi: Scientific Research Honors Society
- Inductee, 2016, Upsilon Pi Epsilon Honors Society

IV. **Memberships in learned and professional societies. Society, dates of membership, and offices held**:

- Association of Computing Machinery (ACM), 2010 to present
- Sigma Xi: Scientific Research Honors Society, 2019 to present
- Upsilon Pi Epsilon Honors Society, 2016 to present

V. **Previous academic ranks, institutions, and dates**:

- Assistant Professor, University of Mississippi, August 2015 – August 2021
- Postdoctoral Research Associate, Clemson University, February 2014 – June 2015
- Postdoctoral Research Associate, University of Southern California, February 2012 – February 2014

VI. **Non-academic positions held prior to appointment at MSU**:

- Systems Administrator - Institute for Neurocognitive Science & Technology
  Mississippi State University, May 2007 – May 2008 & April 2010 – October 2010
- Assistant to the Systems Administrator - John C. Longest Health Center
  Mississippi State University, April 2003 – July 2005
- Programmer, Web Developer (ASP) – Burton Computer Resources, Inc.
  Laurel, Mississippi, October 2001 – March 2003
- Programmer, Web Developer (C/C++/PHP) – Cybergate, Inc.
  Laurel, Mississippi, August 1998 – October 2001

VII. **Summary listing of all required and supporting documentation (items 6 and 7 on the cover of the application form). This listing should be less than one page in length.**

- Table of Contents
- Application Letter
- Promotion and Tenure Form
- Initial Offer Letter
- Curriculum Vitae
- Teaching Accomplishments
- Research Accomplishments
- Service Accomplishments
- Selected Publications

Pages 21 - 29 omitted for external review.

Curriculum Vitae

James Adam Jones

Assistant Professor, Computer Science & Engineering
Mississippi State University

Office: 323 Butler Hall
Lab: 201 Butler Hall

Phone: 662-325-7510
E-mail: jadamj@acm.org
Web: www.hi5lab.org

EDUCATION:

Doctor of Philosophy in Computer Science
Department of Computer Science & Engineering
Mississippi State University (December 2011)

Cognitive Science Graduate Certificate
Department of Psychology
Mississippi State University (December 2011)

Master of Science in Computer Science
Department of Computer Science & Engineering
Mississippi State University (August 2007)

Bachelor of Science in Computer Science
Department of Computer Science & Engineering
Mississippi State University (December 2004)
Minor: Software Engineering & Mathematics

Associate of Art in Computer Science
Jones County Junior College (May 2002)

APPOINTMENTS & EXPERIENCE:

Assistant Professor - Mississippi State University
Computer Science and Engineering, August 2021 – Present

Assistant Professor - University of Mississippi
Computer & Information Science, Neuroscience Program, August 2015 – August 2021

Postdoctoral Research Associate - Clemson University
Human Centered Computing, School of Computing, February 2014 – June 2015

Postdoctoral Research Associate - University of Southern California
Institute for Creative Technologies – Mixed Reality Laboratory, February 2012 – February 2014

Instructor of Record - Mississippi State University

Department of Computer Science & Engineering, January 2011 – December 2011

Graduate Research Assistant - Mississippi State University
Augmented & Virtual Reality Perception Laboratory, August 2005 – December 2011

Systems Administrator - Institute for Neurocognitive Science & Technology
Mississippi State University, May 2007 – May 2008 & April 2010 – October 2010

Assistant to the Systems Administrator - John C. Longest Health Center
Mississippi State University, April 2003 – July 2005

Programmer, Web Developer (ASP) – Burton Computer Resources, Inc.
Laurel, Mississippi, October 2001 – March 2003

Programmer, Web Developer (C/C++/PHP) – Cybergate, Inc.
Laurel, Mississippi, August 1998 – October 2001

HONORS & MEMBERSHIPS:

1. NSF CAREER Award, 2024
2. Sigma Xi: Scientific Research Honors Society, 2019
3. Upsilon Pi Epsilon Honors Society, 2016
4. Spirit of State - Mississippi State University Service Award (highest student honor), 2010
5. Association of Computing Machinery (ACM), Member, 2010

COURSES TAUGHT:

Year, Term	Course #	Course Title	Credits	Enrolled
[2024, Fall]	CSE 4433	Virtual and Extended Reality Development	3	29
[2024, Spring]	CSE 2383	Data Structures and Analysis of Algorithms	3	55
[2023, Fall]	CSE 4433	Virtual and Extended Reality Development	3	23
[2023, Spring]	CSE 2383	Data Structures and Analysis of Algorithms	3	59
[2022, Fall]	CSE 2383	Data Structures and Analysis of Algorithms	3	60
[2022, Fall]	CSE 4990	Special Topics: Virtual & Extended Reality Development	3	30
[2022, Fall]	CSE 6990	Special Topics: Virtual & Extended Reality Development	3	3
[2022, Spring]	CSE 4990	Special Topics: Virtual & Extended Reality Development	3	27

[2022, Spring]	CSE 6990	Special Topics: Virtual & Extended Reality Development	3	3
[2021, Fall]	CSE 2383	Data Structures and Analysis of Algorithms	3	42
[2021, Spring]	CSci 343	Fundamentals of Data Science	3	35
[2021, Spring]	CSci 447	Immersive Media (Virtual Reality)	3	15
[2020, Fall]	CSci 343	Fundamentals of Data Science	3	35
[2020, Fall]	Engr 694	Research Methods in Computer Science	3	3
[2020, Spring]	CSci 343	Fundamentals of Data Science	3	38
[2020, Spring]	CSci 447	Immersive Media (Virtual Reality)	3	15
[2020, Spring]	Engr 598	Special Projects: Immersive Media	3	1
[2020, Spring]	Engr 697	Thesis	6 - 9	2
[2020, Spring]	Engr 797	Dissertation	9	1
[2019, Fall]	CSci 343	Fundamentals of Data Science	3	39
[2019, Fall]	Engr 691	Special Topics: Physiological Considerations for VR	3	5
[2019, Fall]	Neu 491	Directed Research in Neuroscience: Physiological Considerations for VR	3	1
[2019, Fall]	Engr 697	Thesis	3 - 9	2
[2019, Fall]	Engr 797	Dissertation	3	1
[2019, Spring]	CSci 447	Immersive Media (Virtual Reality)	3	25
[2019, Spring]	Engr 691	Special Topics: Spatial Perception in VR	3	5
[2019, Spring]	Neu 579	Special Topics in Neuroscience: Spatial Perception in VR	3	2
[2019, Spring]	Neu 491	Directed Research in Neuroscience: Orientation and Frames of Reference	3	1
[2019, Spring]	Engr 697	Thesis	6	2
[2018, Fall]	CSci 343	Fundamentals of Data Science	3	44
[2018, Fall]	CSci 581	Special Topics: Human-Centered Computing	3	16
[2018, Fall]	Engr 596	Special Projects in Engineering Science	3	1
[2018, Fall]	Engr 691	Special Topics in Engineering Science	3	1
[2018, Fall]	Engr 697	Thesis	3	1
[2018, Spring]	CSci 343	Fundamentals of Data Science	3	47
[2018, Spring]	CSci 447	Immersive Media (Virtual Reality)	3	27
[2017, Fall]	CSci 343	Fundamentals of Data Science	3	43
[2017, Fall]	CSci 581	Special Topics: Computer Graphics	3	11
[2017, Fall]	CSci 693	Research Topics in Engineering Science	3	1

[2017, Spring]	CSci 343	Fundamentals of Data Science	3	45
[2017, Spring]	CSci 447	Immersive Media (Virtual Reality)	3	30
[2017, Spring]	Engr 691	Special Topics: Immersive Media (Virtual Reality)	3	7
[2017, Spring]	Engr 597	Special Projects in Engineering Science	3	1
[2017, Spring]	Engr 693	Research Topics in Engineering Science	3	1
[2016, Fall]	CSci 343	Fundamentals of Data Science	3	42
[2016, Fall]	Engr 597	Special Projects in Engineering Science	3	1
[2016, Fall]	Engr 697	Thesis	3	1
[2016, Fall]	Engr 797	Dissertation	3	1
[2016, Spring]	CSci 581	Special Topics: Virtual Reality	3	16
[2016, Spring]	Engr 693	Research Topics in Engineering	3	1
[2016, Spring]	Hon 301	Honors Individual Study	3	1
[2015, Fall]	CSci 390	Special Topics: C/C++ Programming	3	29
[2015, Fall]	CSci 391	Computer Graphics	3	8
[2015, Fall]	CSci 581	Special Topics: Computer Graphics	3	4
[2011, Fall]	CSE 1233	Computer Programming with C	3	51
[2011, Spring]	CSE 1233	Computer Programming with C	3	52

RESEARCH PROPOSALS:

2023.1 **[Funded]** *CAREER: A Neuro-Ophthalmic Approach to Virtual Reality Research*. National Science Foundation - Faculty Early Career Development Program (CAREER). PI: **J. Adam Jones**. $624,999. (May 2024 – April 2029).

2023.2 **[Funded]** *Movement Disorders and Cognitive Impairments from SARS-CoV-2 Infection in Older Adults*. Mississippi Center for Clinical and Translational Research. PI: Harish Chander, Co-PIs: Reuben Burch, David Saucier, **J. Adam Jones**, John E. Ball, David Vandenheever, Jennifer Reneker. $53,928. (July 2023 – June 2025).

2023.3 **[Funded]** *AI2VR: Placing Artificial Intelligence into Virtual Reality to Evaluate Its Performance*. Mississippi State University – Office of Research and Economic Development. PI: **J. Adam Jones**. $1,500. (Fall 2023).

2023.4 **[Declined]** *NSF Convergence Accelerator Track H – Phase 2: Advancement of Driving Technology for Vocational Enablement*. National Science Foundation. PI: Kasee Stratton-Gadke, Co-PIs: **J. Adam Jones**, Lalitha Dabbiru, Zacheus Ahonle, Kris Geroux. $4,999,999. (January 2024 – December 2026).

2023.5 **[Decline]** *NIH-R15 - Multimodal Application for Head Injury Assessment.* National Institute of Health – R15 Research Enhancement Award Program. PI: David Vandenheever. Co-PIs: Harish Chander, **J. Adam Jones**, Jennifer Reneker, Pierre Viviers. $420,375. (July 2023 – June 2027).

2023.6 **[Declined]** *Movement Disorders and Cognitive Impairments from SARS-CoV-2 Infection in Older Adults*. National Institute of Health – R03 Small Grants Program. PI: Harish Chander, Co-PIs: Reuben Burch, David Vandenheever, David Saucier, **J. Adam Jones**. $142,083. (September 2023 – October 2025).

2023.7 **[Declined]** *Understanding the social, cognitive, and learning aspects of using a digital co-creation and collaborative education platform: Teaching and Learning in the Metaverse (TLM).* National Science Foundation. PI: Zack Gillen, Co-PIs: **J. Adam Jones**, John Lamberth, O.P. McCubbins, Sudip Mittal. $399,988. (July 2023 – June 2026).

2022.1 **[Funded]** *NSF Convergence Accelerator Track H – Phase 1: Advancement of Driving Technology for Vocational Enablement.* National Science Foundation. PI: Kasee Stratton-Gadke, Co-PIs: **J. Adam Jones**, Lalitha Dabbiru, Zacheus Ahonle, Kris Geroux. $750,000. (December 2022 – November 2023).

2022.2 **[Funded]** *Creating and Testing a Virtual Reality (VR) Tool Kit for Fall Prevention Training.* Mississippi State University – Office of Research and Economic Development. PI: Harish Chander, Co-PIs: **J. Adam Jones**, David Vendenheever. $2,000. (Spring 2023 – Fall 2023).

2022.3 **[Funded]** *VXR - Virtual and Extended Reality Software Development: Providing Immersive Experiences in the Classroom using Virtual Reality.* MSU Ottilie Schillig Teaching Projects Grant. PI: **J. Adam Jones**. $3,000. (June 2022 – May 2023).

2022.4 **[Declined]** *CAREER: Fidelity Engineering for Virtual and Extended Reality.* National Science Foundation - Faculty Early Career Development Program (CAREER). PI: **J. Adam Jones**. $574,568. (January 2023 – January 2028).

2022.5 **[Declined]** *Movement Disorders and Cognitive Impairments from SARS-CoV-2 Infection in Older Adults.* Mississippi Center for Clinical and Translational Research (MCCTR). PI: Harish Chander, Co-PIs: David Vandenheever, Reuben Burch, **J. Adam Jones**, David Saucier, John Ball, Jennifer Reneker. (January 2023 – December 2023).

2022.6 **[Declined]** *CySP: Social Engineering in the Metaverse.* Department of Defense. PI: **J. Adam Jones**, Co-PIs: Cindy Bethel, Sudip Mittal. $150,000. (July 2022 – June 2023).

2022.7 **[Declined]** *Incorporation of VALD Health™ in Assessing Fall Risk in Elderly: A Biomechanical and Cognitive Approach.* VALD Health. PI: Harish Chander, Co-PIs: Reuben Burch, David Saucier, **J. Adam Jones**. $39,782. (July 2022 – June 2023).

2022.8 **[Declined]** *EEG-based Decoding of Conceptual Representations in the Human Brain Using Deep Learning*. Mississippi State University – Advancing Collaborative Research Program. PI: David Vandenheever, Co-PIs: **J. Adam Jones**, Harish Chander, Hossein Karimi. $15,000. (August 2022 – July 2024).

2022.9 **[Declined]** *NCS-FO: Determining Brain Activity Preceding Free Decisions Using Deep Neural Networks*. National Science Foundation. PI: David Vandenheever, Co-PIs: Hossein Karimi, **J. Adam Jones**. $998,692. (July 2022 – June 2026).

2022.10 **[Declined]** *Virtual Moving Room Paradigm: A Virtual Reality Tool Kit for Fall Prevention in Elderly*. Mississippi Center for Clinical and Translational Research (MCCTR). PI: Harish Chander. Co-PIs: Reuben Burch, Charles Freeman, David Saucier, Tony Luczak, **J. Adam Jones**, Jennifer Reneker. $39,988. (July 1, 2022 – June 30, 2023).

2022.11 **[Declined]** *Pre-proposal - Theme 6B: Machine Learning for Skill Acquisition in Individuals with Disabilities (MSAID)*. National Science Foundation. PI: Kasee Stratton-Gadke, Co-PIs: Ali Gurbuz, **J. Adam Jones**, Junfeng Ma, Bo Tang. (January 2023 - December 2027).

2021.1 **[Declined]** *Understanding the social, cognitive, and learning aspects of using a digital co-creation and collaborative education platform: Teaching and Learning in the Metaverse (TLM)*. National Science Foundation. PI: Zachary Gillen, Co-PIs: John Lamberth, OP McCubbins, Anthony Luczak, **J. Adam Jones**. $849,699. (2022 - 2024).

2021.2 **[Declined]** *SCH: EEG-based Decoding of Conceptual Representations in the Human Brain Using Deep Learning*. National Science Foundation. PI: David Vandenheever, Co-PIs: Hossein Karimi, **J. Adam Jones**. $1,192,376. (July 2022 - June 2026).

2019.1 **[Funded]** *Virtual Reality Learning Center*, Mississippi Department of Human Services. PI: Albert Nylander, Senior Personnel/Technical Lead: **J. Adam Jones**. $550,000. FY: 2019-2020.

2018.1 **[Funded]** *Undergraduate Research in Data Science*, University of Mississippi Summer Undergraduate Research Groups Grant (SURGG), Co-PIs: Naeemul Hassan, Dawn E. Wilkins, H. Conrad Cunningham, Yixin Chen, **J. Adam Jones**. $20,450. Summer 2018.

2017.1 **[Declined]** *CHS: Small: Collaborative Research: The Effect of Peripheral Field of View on the Perception of Virtual and Augmented Environments seen in Head-Mounted Displays*, National Science Foundation, Principal Investigators: J. Edward Swan II, **J. Adam Jones**, $249,053. 2017.

2017.2 **[Declined]** *360 and Immersive Journalism Best Practices*, John S. and James L. Knight Foundation, Ji Hoon Heo (Co-PI), James Adam Jones (Co-PI), $33,967. 2017.

2016.1 **[Declined]** *A High Performance Computing Approach to Data Science and Cyber Security at Mississippi's Universities*, EPSCOR Preproposal, Principal Investigators: David Dampier, Sherif Abdelwahed, John A. Hamilton, Maxwell Young, Patrick Pape, Frances Dancer, April Tanner, Philip Rhodes, **J. Adam Jones**, 2016.

2015.1 **[Funded]** *Intel Galileo University Donations.* Intel Higher Education. Awarded to **J. Adam Jones**. 5 Intel Galileo Generation 2 Development Boards and Seed Studio Grove Starter Kits. Estimated value: $1,500. Fall 2015.

2013.1 **[Funded]** *Mapping the Field of View*, The Office of Naval Research 6.1 Long Range Navy and Marine Corps Science and Technology (N00014-13-1-0237). Principal Investigator: Mark Bolas (on behalf of postdoc **J. Adam Jones**). $506,141.00. FY:2013-2014. Other roles: Proposal Author, Research Personnel.

2013.2 **[Funded]** *Is this Real Life? Is This Just Fantasy? An Investigation of the Understanding of Complex Molecular Geometry Through Virtual Reality and Interaction*, Gordon Research Conference Visionary Grant. Co-Principal Investigators: Robert W. Kojima, Stephan Schwan, **J. Adam Jones**, Minyoung Song. $5,000.00. FY: 2013.

PATENTS:

2023.1 U.S. Provisional Patent Application #63/586,503. *DriVR: A Virtual Reality System.* Inventors: **J. Adam Jones**, Kasee Stratton-Gadke, Lalitha Dabbiru, Zacheus Ahonle, Kris Geroux. Filed: September 29, 2023.

2019.1 U.S. Patent #US10416453B2. *Control of Ambient and Stray Lighting in a Head Mounted Display.* Inventors: Mark Bolas, **J. Adam Jones**, David M. Krum. Applicant: University of Southern California. Granted: September 17, 2019.

2017.1 U.S. Patent #US009645395B2. *Dynamic Field of View Throttling as a Means of Improving User Experience in Head Mounted Virtual Environments.* Inventors: Mark Bolas, **J. Adam Jones**, Ian McDowall, Evan Suma. Applicant: University of Southern California. Granted: May, 9, 2017.

PUBLICATIONS:

Impact factors (IF) and Acceptance Rates (AR) are provided where available.
*All [Journal], [Conference], [Workshop], and [Abstract] entries are peer-reviewed. [Book Chapters] are **not** peer-reviewed.*

2024.1 **[Journal]** Joao Paulo Oliveira Marum, H. Conrad Cunningham, **J. Adam Jones**, Yi Liu. 2024. *Following the Writer's Path to the Dynamically Coalescing Reactive Chains Design Pattern*. Algorithms – Special Issue on Algorithms for Virtual and Augmented Environments. 17(2), p.56.

2024.2 **[Conference]** Katharine E. Johanesen, **J. Adam Jones**, Territa Poole, Katherine Ryker, Christopher Green. 2024. *VR Geoscience Education: Building Spatial Reasoning Skills*. IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops, pp. 167-172.

2024.3 **[Journal]** Hunter Derby, Nathan Conner, Jacob Hull, Faith Hagan, Sally Barfield, Rosemary Boland, Timothy Stewart, **J. Adam Jones**, Adam Knight, Harish Chander. 2024. *Effects of Acute Virtual Reality Exposure on Dynamic Postural Stability*. International Journal of Exercise Science: Conference Proceedings. 16(3). p.75.

2024.4 **[Conference] J. Adam Jones**, W. Woody Watson, Timothy Stewart, Jacob Brewington, Connor Chrismond, Lalitha Dabbiru, Zaccheus J Ahonle, Emily S Wall, Kris Geroux, Kasee Stratton-Gadke. 2024. *DriVR: Extending Driver Training for Persons with Disabilities*. IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops, pp. 1218-1219.

2024.5 **[Abstract]** Ander Talley, **J. Adam Jones**. 2024. *Evaluating NeRF Fidelity using Virtual Environments*. 2024 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops, pp. 1027-1028.

2023.1 **[Journal]** Hunter Derby, Nathan O. Connor, Jacob M. Hull, Faith Hagan, Sally Barfield, Timothy Stewart, **J. Adam Jones**, Adam C. Knight, Harish Chander. 2023. *Effects of Acute Exposure to Virtually Generated Slip Hazards during Overground Walking*. Applied Sciences, 13(23), p.12848.

2023.2 **[Journal]** Hannah R. Freeman, Harish Chander, Sachini N. K. Kodithuwakku Arachchige, Alana J. Turner, **J. Adam Jones**, Zhujun Pan, Christopher Hudson, Adam C. Knight. 2023. *Postural Control Behavior in a Virtual Moving Room Paradigm*. Biomechanics, 3(4), pp.539-551.

2023.3 **[Abstract] J. Adam Jones**. 2023. *The Pre-History (and Future) of Virtual Reality*. Proceedings of the Mississippi Academy of Science, 1 page.

2022.1 **[Journal]** Nathan O. Conner, Hannah R. Freeman, **J. Adam Jones**, Tony Luczak, Daniel Carruth, Adam C. Knight, Harish Chander. *Virtual Reality Induced Symptoms and Effects: Concerns, Causes, Assessment & Mitigation*. Virtual Worlds, 17 pages.

2022.2 **[Journal]** Harish Chander, Hannah R Freeman, Christopher M Hill, Christopher Hudson, Sachini N. K. Kodithuwakku Arachchige, Alana J Turner, **J. Adam Jones**, Adam C Knight. 2022. *The Walls are Closing in: Postural Responses to a Virtual Reality Claustrophobic Simulation*. Clinical and Translational Neuroscience, 15 pages.

2022.3 **[Conference]** Anthony Luczak. Charles Freeman, Reuben Burch, David Saucier, Harish Chander, John Barlow, John Ball, Patrick Nelsen, E. Parker, C. Middleton, Leslie Strawderman, J. Mohammadi-Aragh, Zachery Gillen, B. Smith, **J. Adam Jones**, Martin Duclos, Michael Taquino, Steven Grice. *Athlete Engineering BaseLine*

Ecosystem: innovative technologies to enhance human performance. TechConnect World Innovation Conference, Washington DC, page 136-140.

2022.4 **[Workshop]** Anthony Luczak, **J. Adam Jones**, Reuben Burch, Jonathan Barlow, Patrick Nelsen, Steven M. Grice, Michael Taquino, Martin Duclos, Caleb Morgan. 2022. *Advancing tangible augmented game objects for use in a golf swing, self-service training environment: Report of Work-in-Progress with a Multidisciplinary Emphasis*. IEEE Virtual Reality Workshop on Augmented Reality Enabling Superhuman Sports + Serious Games (IEEE VR – ARES), Christchurch, New Zealand, pages: 136 - 140.

2022.5 **[Abstract]** Tony Luczak, Kait Jackson, Long Tian, **J. Adam Jones**, Reuben Burch, Jonathan Barlow, Patrick Nelsen, Caleb Morgan, Harish Chander, Martin Duclos, Michael Taquino, and Steven M. Grice. *The Challenges of Coaching Using 2D Golf Swing Video Data Compared to the Challenges of Building a 3D Technology Based Coach*. World Scientific Congress of Golf 2022. 1 page.

2022.6 **[Abstract]** Zachery Garris, Maggie Pettus, Daniel Molsbarger, Harish Chander, **J. Adam Jones**. 2022. *EEG Electrode Localization Using Off-the-shelf Virtual Reality Systems*. Proceedings of the Mississippi Academy of Sciences Annual Meeting.

2022.7 **[Abstract]** Dylan Devenny, Harish Chander, **J. Adam Jones**. 2022. *The Influence of Optical Flow on the Perception of Impossible Spaces in Virtual Reality*. Proceedings of the Mississippi Academy of Sciences Annual Meeting.

2022.8 **[Abstract]** Jonathan Hopper, David M. Krum, **J. Adam Jones**. 2022. *Perception of Target Eccentricity and Depth in Virtual Reality*. Proceedings of the Mississippi Academy of Sciences Annual Meeting.

2020.1 **[Conference, AR: 26.5%]** Benjamin Creel, Caitlin Rinz-Jones, Colin Jackson, **J. Adam Jones.** 2020. *Bacterial Load of Virtual Reality Headsets*. ACM Symposium on Virtual Reality Software and Technology (VRST), Ottawa, Canada, *in press.*

2020.2 **[Conference, AR: 21.4%]** Hunter Finney, **J. Adam Jones.** 2020. *Asymmetric Effects of the Ebbinghaus Illusion on Depth Judgments*. IEEE Conference on Virtual Reality and 3D User Interfaces (IEEE VR), Atlanta, GA, pages 573 - 578.

2020.3 **[Conference, AR: 35%]** João Paulo Oliveira Marum, H. Conrad Cunningham, **J. Adam Jones**. 2020. *Unified Library for Dependency Graph Reactivity on Web and Desktop User Interfaces.* ACM Southeast Conference (ACMSE), Tampa, FL, USA, pages: 26 - 33.

2020.4 **[Workshop]** Collin Roth, Ethan Luckett, **J. Adam Jones**. 2020. *Latency Detection and Illusion in a Head-Worn Virtual Environment.* IEEE Virtual Reality Workshop on Perceptual and Cognitive Issues in Augmented Reality (PERCAR), Atlanta, GA, pages: 215 - 218.

2020.5 **[Workshop]** João Paulo Oliveira Marum, H. Conrad Cunningham, **J. Adam Jones**. 2020. *Dependency Graph-based Reactivity for Virtual Environments*. IEEE Virtual

Reality Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS), Atlanta, GA, pages: 246 - 253.

2019.1 **[Journal, AR: 23.3%, IF: 3.78] J. Adam Jones**, Jonathan E. Hopper, Mark T. Bolas, David M. Krum. 2019. *Orientation Perception in Real and Virtual Environment*. IEEE Transactions on Visualization and Computer Graphics (TVCG), pages 2050-2060.

2019.2 **[Book Chapter, Invited] J. Adam Jones**, 2019. *A Tinkerer's Perspective on VR Displays*. "VR Developer Gems", AK Peters/CRC Press, ed. William Sherman, May 2019.

2019.3 **[Conference]** João Paulo Oliveira Marum, **J. Adam Jones**, H. Conrad Cunningham. 2019. *Towards a Reactive Game Engine*. IEEE Southeast Conference, Huntsville, AL.

2019.4 **[Conference]** William Panlener, David M. Krum, **J. Adam Jones.** 2019. *Effects of Horizontal Field of View Extension on Spatial Judgments in Virtual Reality*. IEEE Southeast Conference, Huntsville, AL.

2019.5 **[Workshop] J. Adam Jones**. 2019. *Optical and Neural Properties of Vision as Applied to Virtual Reality*. IEEE Virtual Reality Workshop on Neuroscience & Virtuality (NeuroVirt), Osaka, Japan, pages 1667-1670.

2019.6 **[Workshop]** Jonathan E. Hopper, Hunter Finney, **J. Adam Jones**. 2019. *Field of View and Forward Motion Discrimination in Virtual Reality*. IEEE Virtual Reality Workshop on Neuroscience & Virtuality (NeuroVirt), Osaka, Japan, pages 1663-1666.

2019.7 **[Workshop]** Ethan Luckett, Tykeyah Key, Nathan Newsome, **J. Adam Jones**. 2019. *Metrics for the Evaluation of Tracking Systems for Virtual Environments*. IEEE Virtual Reality Workshop on Novel Input Devices and Interaction Techniques (NIDIT), Osaka, Japan, pages 1711-1716.

2019.8 **[Abstract]** Benji Creel, **J. Adam Jones**, Collin R. Jackson. 2019. *Bacterial Load in Virtual Reality Headsets*. Proceedings of the American Society of Microbiology, South Central Branch Meeting. Oxford, MS, page 36.

2019.9 **[Abstract] J. Adam Jones**, Ethan Luckett, Tykeyah Key, Nathan Newsome. 2019. *Latency Measurement in Head-Mounted Virtual Environments*. Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces (IEEE VR), Osaka, Japan, pages 1000-1001.

2019.10 **[Abstract] J. Adam Jones**, Thai Phan. 2019. *A Comparison of Gait Measurement Methods*. Proceedings of the IEEE Southeast Conference, Huntsville, AL.

2019.11 **[Abstract]** Khoa Tran, Yaoxin Zhang, Yafei Jia, **J. Adam Jones**. 2019. *A Prototype Visualization Tool for Hurricane Flood Simulations in Virtual Reality*. Proceedings of the IEEE Southeast Conference, Huntsville, AL.

2018.1 **[Abstract]** McKennon McMillian, Hunter Finney, Jonathan Hopper, **J. Adam Jones**. 2018. *The Depth Light*. Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces (IEEE VR), Reutlingen, Germany, pages 834-835.

2017.1 **[Journal, IF: 1.02] J. Adam Jones**, David M. Krum, Mark T. Bolas. 2017. *Vertical Field of View Extension and Walking Characteristics in Head-worn Virtual Environments*. ACM Transactions on Applied Perception (ACM TAP). 14, 2, pages 9:1-9:17.

2016.1 **[Conference, AR: 33%] J. Adam Jones**, Darlene Edewaard, Richard A. Tyrrell, Larry F. Hodges. 2016. *A Schematic Eye for Virtual Environments*. Proceedings of the IEEE Symposium of 3D User Interfaces, Greeneville, South Carolina, pages 221-230.

2016.2 **[Abstract]** Joao Paulo Marum, **J. Adam Jones**, Conrad H. Cunningham. 2016. *Functional Reactive Augmented Reality: Proof of Concept Using an Extended Augmented Desktop with Swipe Interaction.* Proceedings of the Eurographics Symposium on Virtual Environments, pages 13-14.

2016.3 **[Abstract]** Alexander Gunter, Andrew Robb, **J. Adam Jones**. 2016. *Real-Time Marker Tracking with Microsoft Kinect.* Proceedings of the Eurographics Symposium on Virtual Environments, pages 15-16.

2015.1 **[Conference, AR: 38.3%] J. Adam Jones**, Lauren Cairco Dukes, David, M. Krum, Mark T. Bolas, Larry F. Hodes. 2015. *Correction of Geometric Distortions and the Impact of Eye Position in Virtual Reality Displays*. Proceedings of the International Conference on Collaboration Technologies and Systems, Atlanta, Georgia, pages 77-83.

2015.2 **[Abstract]** Elhman Ebrahimi, Bliss M. Altenhoff, Christopher C. Pagano, Sabarish V. Babu, **J. Adam Jones**. 2015. *Investigating the Impact of Perturbed Visual and Proprioceptive Information in Near-Field Immersive Virtual Environments*. Poster compendium, Proceedings of the IEEE Virtual Reality Conference, Arles Camargue, France, pages 171-172.

2014.1 **[Conference, AR: 46%]** Elham Ebrahimi, Bliss Altenhoff, Leah Hartman, **J. Adam Jones**, Sabarish V. Babu, Christopher C. Pagano, Timothy A. Davis. 2014. *Effects of Visual and Proprioceptive Information in Visuo-motor Calibration During a Closed-loop Physical Reach Task in Immersive Virtual Environments*. Proceedings of the ACM SIGGRAPH Symposium on Applied Perception, Vancouver, Canada, pages 103-110.

2014.2 **[Abstract] J. Adam Jones**, Lauren Cairco Dukes, Mark Bolas. 2014. *Automated Calibration of Display Characteristics (ACDC) for Head-Mounted Displays and Arbitrary*

Surfaces. Poster compendium, Proceedings of the IEEE Virtual Reality Conference, Minneapolis, Minnesota, pages 85-86.

2014.3 **[Abstract] J. Adam Jones**, David M. Krum, Mark Bolas. 2014. *The Effect of Eye Position on the View of Virtual Geometry*, Poster compendium, Proceedings of the IEEE Virtual Reality Conference, Minneapolis, Minnesota, pages 87-88.

2013.1 **[Journal, IF: 3.78, AR: 21.6%] J. Adam Jones**, J. Edward Swan II, Mark Bolas. 2013. *Peripheral Stimulation and its Effect on Perceived Spatial Scale in Virtual Environments*. IEEE Transactions on Visualization & Computer Graphics (TVCG). 19, 4, pages 701-710.

2012.1 **[Conference, AR: 52.5%] J. Adam Jones**, J. Edward Swan II, Gurjot Singh, Sujan Reddy, Kenneth Moser, Chunya Hua, Stephen R. Ellis. 2012. *Improvements in Visually Directed Walking in Virtual Environments Cannot be Explained by Changes in Gait Alone*. Proceedings of the ACM SIGGRAPH Symposium on Applied Perception, Playa Vista, California, pages 11-16.

2012.2 **[Abstract, Best Poster] J. Adam Jones**, Evan A. Suma, David M. Krum, Mark Bolas. 2012. *Comparability of Narrow and Wide Field-Of-View Head-Mounted Displays for Medium-Field Distance Judgments*, Poster compendium, Proceedings of the ACM SIGGRAPH Symposium on Applied Perception, Playa Vista, California, page 119.

2012.3 **[Abstract]** Gurjot Singh, J. Edward Swan II, **J. Adam Jones**, and Stephen R. Ellis. 2012. *Depth Judgments by Reaching and Matching in Near-Field Augmented Reality*. Poster compendium, Proceedings of the IEEE Virtual Reality Conference, Irvine, California, pages 165-166.

2011.1 **[Conference, AR: 82.6%] J. Adam Jones**, J. Edward Swan II, Gurjot Singh, Stephen R. Ellis. 2011. *Peripheral Visual Information and Its Effect on Distance Judgments in Virtual and Augmented Environments*. Proceedings of the ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization. Toulouse, France, pages 29-35.

2011.2 **[Abstract] J. Adam Jones**, J. Edward Swan II, Gurjot Singh, Stephen R. Ellis. 2011. *Peripheral Visual Information and Its Effect on the Perception of Egocentric Depth in Virtual and Augmented Environments*. Poster compendium, Proceedings of the IEEE Virtual Reality Conference. Singapore, pages 215-216.

2011.3 **[Abstract]** Gurjot Singh, J. Edward Swan II, **J. Adam Jones**, Stephen R. Ellis. 2011. *Depth Judgment Tasks and Environments in Near-Field Augmented Reality*. Poster compendium, Proceedings of the IEEE Virtual Reality Conference. Singapore. pages 241-242.

2010.1 **[Conference, AR: 56.3%]** Gurjot Singh, J. Edward Swan II, **J. Adam Jones**, Stephen R. Ellis. 2010. *Depth Judgment Measures and Occluding Surfaces in Near-Field Augmented Reality*. Proceedings of the ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization. Los Angeles, California, pages 149-156.

2009.1 **[Abstract] J. Adam Jones**, J. Edward Swan II, Gurjot Singh, Joshua Franck, Stephen R. Ellis. 2009. *The Effects of Continued Exposure to Medium Field Augmented and Virtual Reality on the Perception of Egocentric Depth*. Poster compendium, Proceedings of the ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization. Crete, Greece, page 138.

2009.2 **[Abstract]** Gurjot Singh, J. Edward Swan II, **J. Adam Jones**, Joshua Franck, Stephen R. Ellis. 2009. *Depth Judgment Measures and Occluders in Near-Field Augmented Reality*. Poster compendium, Proceedings of the ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization. Crete, Greece, page 127.

2008.1 **[Conference, AR: 54.8%] J. Adam Jones**, J. Edward Swan II, Gurjot Singh, Stephen R. Ellis. 2008. *The Effects of Virtual Reality, Augmented Reality, and Motion Parallax on Egocentric Depth Perception*. Proceedings of the ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization. Los Angeles, California, pages 9-14.

2008.2 **[Abstract] J. Adam Jones**, J. Edward Swan II, Gurjot Singh, Eric W. Kolstad. 2008. *The Effects of Virtual Reality, Augmented Reality, and Motion Parallax on Egocentric Depth Perception*. Poster compendium, Proceedings of the IEEE Virtual Reality Conference. Reno, Nevada. pages 267-268.

2007.1 **[Journal, IF: 3.78]** J. Edward Swan II, **J. Adam Jones**, Eric W. Kolstad, Mark A. Livingston, Harvey S. Smallman. 2007. *Egocentric Depth Judgments in Optical See-Through Augmented Reality*. IEEE Transactions on Visualization & Computer Graphics (TVCG). 13, 3, pages 429-442.

PRESENTATIONS:

2023.1 **[Conference]** *The End of Algorithms (or How I Learned to Stop Worrying and Love the Machine),* Mississippi Academy of Science Annual Meeting, Gulf Port, Mississippi, February 23, 2023.

2023.2 **[Conference]** *The Pre-History (and Future) of Virtual Reality,* Mississippi Academy of Science Annual Meeting, Gulf Port, Mississippi, February 24, 2023.

2022.1 **[Conference]** *The Influence of Optical Flow on the Perception of Impossible Spaces in Virtual Reality,* Mississippi Academy of Science Annual Meeting, Gulf Port, Mississippi, March 31, 2022.

2022.2 **[Conference]** *Perception of Target Eccentricity and Depth in Virtual Reality,* Mississippi Academy of Science Annual Meeting, Gulf Port, Mississippi, March 31, 2022.

2021.1 **[Seminar]** *Extended Reality: Extending Human-Centered Research*, ACCESS: Applied Cognitive Science Seminar Series, Mississippi State University, Mississippi, September 3, 2021.

2021.2 **[Seminar]** *Extended Reality: Extending Human-Centered Research*, Computer Science & Engineering Seminar Series, Mississippi State University, Mississippi, October 22, 2021.

2021.3 **[Guest Lecture]** *Extended Reality: Extending Human-Centered Research,* CSE 1011 - Introduction to CSE, Mississippi State University, Mississippi, November 15, 2021.

2020.1 **[Keynote]** *Virtual and Augmented Reality in Engineering*, 2020 Southeast Symposium on Contemporary Engineering Topics (SSCET), Little Rock, Arkansas, September 2022 (delayed until 2022 due to COVID-19).

2020.2 **[Seminar]** *Augmented and Virtual Reality in STEM,* ARISE@UM - A Research Immersive STEM Experience at the University of Mississippi, July 14, 2020.

2020.3 **[Conference]** *Asymmetric Effects of the Ebbinghaus Illusion on Depth Judgments*. IEEE Conference on Virtual Reality and 3D User Interfaces (IEEE VR), Atlanta, GA, March 25, 2020.

2020.4 **[Conference]** *Latency Detection and Illusion in a Head-Worn Virtual Environment*. IEEE Virtual Reality Workshop on Perceptual and Cognitive Issues in Augmented Reality (PERCAR), Atlanta, GA, March 22, 2020.

2019.1 **[Invited]** *Media Arts Panel – Virtual Reality*, Oxford Film Festival, Oxford, Mississippi, February 7, 2019.

2019.2 **[Conference]** *A Comparison of Gait Measurement Techniques*, IEEE SoutheastCon 2019, Huntsville, Alabama.

2019.3 **[Conference]** *Field of View and Forward Motion Discrimination in Virtual Reality,* IEEE VR Workshop on Neuroscience and Virtuality (NeuroVirt), March 23, 2019.

2019.4 **[Conference]** *Optical and Neural Properties of Vision as Applied to Virtual Reality,* IEEE VR Workshop on Neuroscience and Virtuality (NeuroVirt), March 23, 2019.

2019.5 **[Conference]** *Metrics for the Evaluation of Tracking Systems for Virtual Environments,* IEEE VR Workshop on Novel Input Devices and Interaction Techniques (NIDIT), March 24, 2020.

2019.6 **[Conference]** *Orientation Perception in Real and Virtual Environments*, IEEE Virtual Reality and 3D User Interfaces Conference (IEEE VR), March 25, 2019.

2018.1 **[Invited]** *Seagulls & Solar Flares: Science, Data, & Data Science.* Pi Mu Epsilon Honors Society Meeting, Oxford, Mississippi, April 26, 2018.

2018.2 **[Invited]** *Virtual Reality, Augmented Reality, and the Classroom*. ADMI Symposium on Computing at Minority Institutions, New Orleans, Louisiana, April 7, 2018.

2017.1 **[Invited]** *The State of Augmented Reality*. FNC/Core Logic – Lunch and Learn, Oxford, MS, August 18, 2017. Host: Bethany Cooper.

2016.1 **[Conference]** *A Schematic Eye for Virtual Environments,* IEEE Symposium on 3D User Interfaces, Greenville, South Carolina, March 19, 2016.

2015.1 **[Invited]** *The State of the Art and Science of Virtual Reality*, University of Memphis, November 6, 2015.

2015.2 **[Conference]** *Correction of Geometric Distortions and the Impact of Eye Position in Virtual Reality Displays*, International Conference on Collaboration Technologies and Systems (CTS), June 2, 2015.

2015.3 **[Invited]** *The State of Virtual Reality,* Invited speaker in Business Telecommunications Technologies, Georgia State University, Atlanta, Georgia, February 23, 2015, Host: Jung P. Shim, PhD.

2015.4 **[Invited]** *Spatial Perception in Virtual Reality: It's Not Just What You See But How You See It*, Invited speaker to Engineering Seminar Series at University of Georgia, Athens, Georgia, February 20, 2015, Host: Kyle Johnson, PhD.

2014.1 **[Seminar]** *The State of Virtual Reality: A Talk of Many Things,* School of Computing Seminar Series, Clemson University, South Carolina, October 31, 2014.

2013.1 **[Keynote]** *Learning and Perceiving in Augmented (and Virtual) Environments,* Invited speaker at the Gordon Research Conference on Visualization in Science and Education, Bryant University, Smithfield, Rhode Island, July 22, 2013.

2013.2 **[Conference]** *Peripheral Stimulation and its Effect on Perceived Spatial Scale in Virtual Environments*, IEEE Virtual Reality and 3D User Interfaces Conference (IEEE VR), Orlando, Florida, March 20, 2013.

2013.3 **[Report]** *Mapping the Field of View,* Office of Naval Research Year One Project Overview, USC Institute for Creative Technologies, Playa Vista, California, February 10, 2013.

2012.1 **[Conference]** *Improvements in Visually Directed Walking in Virtual Environments Cannot be Explained by Changes in Gait Alone*, ACM SIGGRAPH Symposium on Applied Perception, Playa Vista, California, August 3, 2012.

2011.1 **[Invited]** *Going With The Flow: Motion-based Visual Cues in the Extreme Periphery and Their Effect on Distance Judgments in Virtual & Augmented Reality*, Invited speaker at the School of Interactive Arts & Technology - Simon Frasier University, Vancouver, BC, Canada, November 24, 2011, Host: Bernhard Riecke.

2011.2 **[Seminar]** *Going With The Flow: Distance Judgments in Virtual Reality, Augmented Reality, and the Real-World*, Applied Cognitive Science Seminar Series, Mississippi State University, Mississippi, October 28, 2011.

2011.3 **[Invited Guest Lecture]** *Graphics: Making Games Look Good*, Invited presentation to the Starkville High School Game Design Club, Starkville, Mississippi, October 25, 2011, Host: Andrew Stamps.

2011.4 **[Invited]** *External Visual Cues and Their Effect on Walked Distance Judgments in Virtual and Augmented Environments*, Invited speaker at the NASA Ames Research Center, Moffett Field, California, November 22, 2011, Host: Stephen R. Ellis.

2010.1 **[Invited]** *Experiences Abroad - Japanese Technology & Culture*, Invited presentation at the Colloquium on Understanding More About Japanese Culture, Business, and Information Technology, Sponsored by the Japan Foundation and Mississippi State University, Mississippi State University, Mississippi, August 19, 2010, Host: J.P. Shim.

2010.2 **[Seminar]** *A Mostly Visual Introduction to Augmented and Virtual Reality*, Association of Computer Machinery (ACM) Student Chapter, Mississippi State University, Mississippi, September 20, 2010.

2010.3 **[Invited]** *Virtual & Augmented Environments: A Matter of Perception*, Invited speaker at the Imaging Media Research Center - Korea Institute for Science & Technology, Seoul, South Korea, May 4, 2010, Hosts: Heedong Ko & Yong-Moo Kwon.

2010.4 **[Guest Lecture]** *A Review of the AR Lab's Perceptual Experiments in Virtual & Augmented Environments*, Invited guest lecture to CSE-8283: Empirical Software Engineering, Mississippi State University, Mississippi, April 1, 2010.

2009.1 **[Seminar]** *Depth Judgments and Adaptation in Augmented and Virtual Reality*, Empirical Software Engineering Research Group, Mississippi State University, Mississippi, November 2, 2009.

2009.2 **[Seminar]** *Visual Perception in Augmented and Virtual Environments*, Graduate Seminars in Computer Science, Mississippi State University, Mississippi, October 7, 2009.

2009.3 **[Guest Lecture]** *Perceptual Cues in Visual Presentations*, Invited guest lecture to CSE-6990: Principles of Digital Visual Communication and Analysis, Mississippi State University, Mississippi, September 4, 2009.

2009.4 **[Guest Lecture]** *Studying Human Perception in Virtual and Augmented Reality*, Invited guest lecture to ECE-8990: Level of Detail and Virtual Environments, Mississippi State University, Mississippi, March 26, 2009.

2007.1 **[Seminar]** *Augmented and Virtual Reality: A Matter of Perception*, Graduate Seminars in Computer Science, Mississippi State University, Mississippi, November 10, 2007.

2007.2 **[Seminar]** *Depth Perception in Augmented and Virtual Reality*, ACCESS: Applied Cognitive Science Seminar Series, Mississippi State University, Mississippi, September 7, 2007.

2006.1 **[Seminar]** *Egocentric Depth Judgments in Optical See-Through Augmented Reality*, ACCESS: Applied Cognitive Science Seminar Series, Mississippi State University, Mississippi, November 17, 2006.

2006.2 **[Colloquium]** *Egocentric Depth Judgments in Optical See-Through Augmented Reality*. Institute for Neurocognitive Science & Technology Colloquium. Mississippi State University, Mississippi, October 26, 2006.

ARTISTIC & TECHNICAL EXHIBITIONS:

2024.1 **[Technical]** *DriVR: Extending Driver Training for Persons with Disabilities*. Research Demonstration, IEEE Virtual Reality Conference 2024, Orlando, FL.

2024.2 **[Artistic]** *Trinummus or Three Dollar Day - A Roman Comedy by Plautus adapted with Robots and Extended Reality*, Starkville, MS.

2019.1 **[Technical]** *Imagine the Possibilities – Virtual Reality Development at UM Exhibit,* Tupelo, MS.

2018.1 **[Artistic]** *Oxford Fringe Film Festival – 10 Steps to Plotting Data*, Oxford, MS.

2018.2 **[Technical]** *CSpire Tech MVMT Expo – Student-Built Virtual Reality Exhibit*, Jackson, MS.

2018.3 **[Technical]** *eSports Egg Bowl – Virtual Reality at UM Exhibit*, Oxford, MS.

2017.1 **[Technical]** *CSpire CTX Technology Expo – Student-Built Virtual Reality Exhibit*, Oxford, MS.

UNIVERSITY SERVICE:

ADVISING – CURRENT GRADUATE STUDENTS:

1. **[MS, MSU, Advisor]** Christiam Johnson
2. **[MS, MSU, Advisor]** Jacob Adams
3. **[MS MSU, Advisor]** Ander Talley
4. **[MS, MSU, Advisor]** Jacob Brewington
5. **[PhD, MSU, Advisor]** Zeenat Islam
6. **[PhD, MSU, Advisor]** David Huang
7. **[PhD, MSU, Committee]** Alexander Sommers

ADVISING - DOCTORAL DISSERTATIONS - COMPLETED: [1]

1. **[PhD, UM, Co-Advisor, Summer]** João Paulo Oliveira Marum
2. **[PhD, UM, Co-Advisor, Incomplete]** William Panlener
3. **[PhD, UM, Co-Advisor, Incomplete]** Daniel Coto

ADVISING – DISSERTATIONS/THESES/PROJECTS - COMPLETED:

2023.1 **[MS, MSU, Advisor, Summer]** Zachery Garris, *Localization of Spatial Audio in Matched and Mismatched Virtual Environments*

2023.2 **[MS, MSU, Advisor, Fall]** Daniel Molsbarger, *Using Sketched-based Methods to Evaluate Spatial Scale in Real and Virtual Environments*

2021.1 **[MS, UM, Co-Advisor, Summer]** João Paulo Oliveira Marum, *Dependency-based Reactive Change Propagation Design Pattern Applied in Environments with Unpredictable Interactions*

2020.1 **[MS, UM, Advisor, Spring]** Ethan Luckett, *Assessing Distance Perception in Virtual and Augmented Realities with Electroencephalography*

2020.2 **[MS, UM, Advisor, Spring]** Jonathan Hopper, *Simulation of Virtual Reality Display Characteristics: A Method for the Evaluation of Motion Perception*

---

[1] *When working with PhD students at the University of Mississippi, Assistant Professors are required to co-advise with a tenured professor.*

2019.1 **[MS, UM, Advisor, Fall]** McKennon McMillian, *The Effects of Differing Optical Stimuli on Depth Perception in Virtual Reality*

2019.2 **[MS, UM, Advisor, Spring]** Collin Roth, *Latency Detection Thresholds in HMD-based Virtual Environments*

2017.1 **[MS, UM, Committee, Spring]** Everett Price, *Naïve Bayes With Sentiment Analysis Features as a Means of Class Identification of the Property Conditions Field Text of the 1004 Appraisal Form*

2017.2 **[MS, UM, Advisor, Fall]** William Panlener, *Effects of Field of View Restriction on Spatial Judgments*

2017.3 **[MS, UM, Advisor, Incomplete]** Aussie Warren, *Haptic Feedback Devices for Enhanced Spatial Interactions*

2016.1 **[MS, UM, Advisor, Spring]** Daniel Coto, *Camera-based Correction of Optical Distortions in Head-Mounted Displays*

2016.2 **[MS, UM, Committee, Fall]** Justin Trotter, *A Web Based Graphical User Interface for a Decision Support System*

ADVISING - UNDERGRADUATE HONORS THESES:

2024.1 **[Directed, MSU]** Christian Johnson (Computer Science, In Progress), *Exploring Spatial Judgments in Virtual Reality*

2024.2 **[Directed, MSU]** Isaac Martinolich (Computer Science, In Progress), *Utilizing Machine Learning in the Teaching of Foreign Languages through Popular Media*

2022.1 **[Directed, UM]** Logan Parker (Computer Science, Spring 2022), *Diminished and Mixed Reality in the Oculus Quest 2*

2020.1 **[Committee, UM]** Benjamin Creel (Biology, Spring 2020), *Bacterial Load on Virtual Reality Headsets*

2020.2 **[Committee, UM]** Jennifer Lauriello (Computer Science, Spring 2020), *Methods to Detect Forgeries in Static Signatures*

2020.3 **[Committee, UM]** Olivia Eustice (Biomedical Engineering, Spring 2020), *A Comparison of Neural Activity for Peripersonal and Extrapersonal Viewing in Real and Virtual Environments*

2019.1 **[Directed, UM]** Dylan Devenny (Computer Science, Spring 2019), *The Effect of Optical Flow on Redirection Techniques in Virtual Reality*

2018.1 **[Directed, UM]** Ethan Luckett (Computer Science, Spring 2018), *Quantifying the HTC Vive's Lighthouse Tracking Accuracy*

2017.1 **[Committee, UM]** Reid Barber (Computer Science, Spring 2017), *Autonomous Vehicle Communication Using Blockchain*

2016.1 **[Directed, UM]** Sean O'Hara (Computer Science, Spring 2016), *Multi-body, Infrared Motion Tracking with Off-The-Shelf Components*

ADVISING - UNDERGRADUATE CAPSTONE PROJECTS:

2023.1 **[Spring]** ECE Senior Design Team 1, *Safer Virtual Reality for Electric Wheelchair Users*

2023.2 **[Spring]** ECE Senior Design Team 2, Hand-*Control Driving Interface for VR-based Training for Persons with Disabilities*

2021.1 **[Fall]** Logan Parker, *Augmented Home: An Augmented Reality Interior Design Tool*
2021.2 **[Spring]** Zachery Garris, *Hologram-like Projections using a Persistence-of-Vision Display*
2021.3 **[Spring]** Paul Budz, *Arduino-Based Flight Telemetry Computer for Amateur Rocketry*
2021.4 **[Spring]** Thong Tra, *Video-based Augmented Reality with Hand Interactions for Video Conferencing and Education*
2021.5 **[Spring]** Parker Mathis, *Framework for Simulating Novel Augmented Reality Hardware Configurations in Virtual Reality*
2021.6 **[Spring]** Ariana Howell, *Inverse Kinematics in Python using Off-The-Shelf VR Tracking*

2020.1 **[Fall]** Nayan Chawla, *Video-Overlay Augmented Reality Interface for Off-The-Shelf Video Conferencing Tools*
2020.2 **[Fall]** Andrew Jelson, *Neural Network Polygonization of Point Cloud Data*
2020.3 **[Fall]** James Shaver, *Neural Network Polygonization of Point Cloud Data*
2020.4 **[Spring]** Hunter Finney, *Redirected Walking: Exploiting Change Blindness in Virtual Reality*
2020.5 **[Spring]** William Brozovic, *Computational Archaeology: Using Data Science to Understand the Past*

2019.1 **[Fall]** Robert Lampton, *Textual Optical Flow Variations: A Study of Altered Peripheral Vision on Spatial Perception*
2019.2 **[Spring]** Marcus Higgins, *StatsViz: A Tool for Visual Analysis of Data*
2019.3 **[Spring]** Cole Smith, *Random Dot Stereograms and Dynamic Maze Generation*

2018.1 **[Spring]** Ethan Luckett, *Quantifying the HTC Vive's Lighthouse Tracking Accuracy*
2018.2 **[Spring]** Jonathan Hopper, *Optical Flow Presentation in Virtual Reality*

2017.1 **[Fall]** Rashad Collier, *Walking Recalibration in VR, Proof of Concept*
2017.2 **[Spring]** McKennon McMillian, *Depth Light: A Mixed Reality Interaction Device*
2017.3 **[Spring]** Collin Roth, *Tabletop Optical See-Through Augmented Reality System*
2017.4 **[Spring]** Ricky Bojorquez, *Gait Tracking and Analysis in Immersive Virtual Reality*
2017.5 **[Spring]** Kathrine Hewlett, *Real-Time Correction of Off-Axis Projections*

2016.1 **[Fall]** Alex Gunter, *General Purpose Motion Tracking using Retroreflective Markers and the MS Kinect*
2016.2 **[Fall]** Joseph Woestendiek, *Stereoscopic 3D Table-top Augmented Reality Display*
2016.3 **[Spring]** Matt Brown, *Mechanical, 3D Position Tracking for the Oculus Rift DK1*
2016.4 **[Spring]** Shaylen Patel, *DIY 3D Shutter Glasses Using the Intel Galileo*

ADVISING - UNDERGRADUATE SUMMER SCHOLARS & INTERN PROJECTS:

2021.1 Ronnie Davis (McNair Scholar), *Distance Perception in Virtual Reality*
2021.2 Robert Ingraham (Biomedical Intern), *Distance Perception in Virtual Reality*

2020.1 Alessandriel Harper (McNair Scholar), *Remote Video Based Marker Tracking for Augmented Reality*
2018.1 Tykeyah Key (McNair Scholar), *Determining Vive Motion Sensitivity Using a Robotic Device*

2018.2 Hunter Finney (CREX Intern), *Effects of the Ebbinghaus Illusion on Depth Perception*
2018.3 Khoa Tran (SURGG Summer Intern), *Flood Flow Visualization in Virtual Reality*

2017.1 Jonathan Hopper (CREX Intern), *Effects of Field of View on Optical Flow Discrimination*

COMMITTEES:

2024.1 **[MSU]** Bagley College of Engineering Research Advisory Council, 2024
2024.2 **[MSU]** CSE Graduate Committee, 2024
2024.3 **[MSU]** Hackathon 2024 Planning Committee, 2024

2023.1 **[MSU]** Bagley College of Engineering Graduate Student Award Committee, 2023
2023.2 **[MSU]** Hackathon 2023 Planning Committee, 2023
2023.3 **[MSU]** CSE Undergraduate Committee, 2023
2023.4 **[MSU]** ABET Assessment Committee, 2023

2022.1 **[MSU]** CSE Undergraduate Committee, 2022
2022.2 **[MSU]** CSE Graduate Committee, 2022
2022.3 **[MSU]** ABET Assessment Committee, 2022

2021.1 **[MSU]** CSE Graduate Committee, 2021
2021.2 **[UM]** Computer & Information Science Undergraduate Committee, 2021
2021.3 **[UM]** Neuroscience Core Faculty (Curriculum Committee), 2021

2020.1 **[UM]** Computer & Information Science Undergraduate Committee, 2020
2020.2 **[UM]** Neuroscience Core Faculty (Curriculum Committee), 2020

2019.1 **[UM]** Computer & Information Science Undergraduate Committee, 2019
2019.2 **[UM]** Faculty Search Committee, 2019
2019.3 **[UM]** Neuroscience Core Faculty (Curriculum Committee), 2019

2018.1 **[UM]** Computer & Information Science Undergraduate Committee, 2018
2018.2 **[UM]** Neuroscience Core Faculty (Curriculum Committee), 2018

2017.1 **[UM]** Computer & Information Science Undergraduate Committee, 2017
2017.2 **[UM]** Neuroscience Core Faculty (Curriculum Committee), 2017

2016.1 **[UM]** Faculty Search Committee, 2016
2016.2 **[UM]** Neuroscience Core Faculty (Curriculum Committee), 2016

2009.1 **[MSU]** Imaging Center of Excellence Quality Control Group, 2009

2008.1 **[MSU]** Institute for Neurocognitive Science & Technology Steering Committee, 2008
2008.2 **[MSU]** MSU Imaging Center of Excellence Quality Control Group, 2008

2007.1 **[MSU]** Institute for Neurocognitive Science & Technology Steering Committee, 2007
2007.2 **[MSU]** Imaging Center of Excellence Quality Control Group, 2007

SPONSORING & MENTORING:

2023.1 ECE Senior Program Sponsor (9 students), 2023

2021.1 McNair Postbaccalaureate Achievement Program Mentor, 2021

2020.1 McNair Postbaccalaureate Achievement Program Mentor, 2020

2019.1 University of Mississippi - Association of Computing Machinery, Sponsor, 2019

2018.1	University of Mississippi - Computer Science Research Experiences Sponsor, 2018
2018.2	McNair Postbaccalaureate Achievement Program Mentor, 2018
2018.3	University of Mississippi - Association of Computing Machinery, Sponsor, 2018
2017.1	University of Mississippi - Computer Science Research Experiences Sponsor, 2017
2017.2	University of Mississippi - Association of Computing Machinery, Sponsor, 2017
2016.1	University of Mississippi - Association of Computing Machinery, Sponsor, 2016
2015.1	University of Mississippi - Association of Computing Machinery, Sponsor, 2015

OTHER UNIVERSITY SERVICE:

2011.1	MSU Globe Trotters - International Education, *Mississippi State University*, 2011
2010.1	Liaison - Future Exchange Student Meeting, *Sookmyung Women's University (Seoul, South Korea)/ Mississippi State University*, May 2010
2010.2	MSU Globe Trotters - International Education, *Mississippi State University*, 2010
2009.1	Korean Exchange Student Orientation, *Mississippi State University*, August 2009.
2009.2	MSU Globe Trotters - International Education, *Mississippi State University*, 2009
2009.3	Forum Moderator: Leadership Styles and Successful Leaders in Korea and the U.S., *Kwangwoon University (Seoul, South Korea)*, 2009

PROFESSIONAL SERVICE:

CONFERENCE ORGANIZATION:

- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), Program Co-chair, 2022
- IEEE Virtual Reality Conference (IEEE VR), Web Co-chair, 2021
- IEEE Virtual Reality Conference (IEEE VR), Web Co-chair, 2020
- IEEE Virtual Reality Conference (IEEE VR), Videos Co-chair, 2019
- IEEE Virtual Reality Conference (IEEE VR), Videos Co-chair, 2018
- IEEE Virtual Reality Conference (IEEE VR), Videos Co-chair, 2017
- IEEE Virtual Reality Conference (IEEE VR), Videos Co-chair, 2016
- IEEE Virtual Reality Conference (IEEE VR), Videos Co-chair, 2014
- IEEE Virtual Reality Conference (IEEE VR), Videos Co-chair, 2013
- IEEE Virtual Reality Conference (IEEE VR), Publications Co-chair, 2012
- IEEE Virtual Reality Conference (IEEE VR), Publications Co-chair, 2011
- IEEE Virtual Reality Conference (IEEE VR), Publications Co-chair, 2010

CONFERENCE PROGRAM COMMITTEES:

- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2020
- IEEE VR Workshop on Perceptual & Cognitive Issues in Augmented Reality, 2020
- IEEE Conference on Virtual Reality and 3D User Interfaces (IEEE VR), 2019
- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2019
- ACM Symposium on Applied Perception (SAP), 2018
- ACM Symposium on Virtual Reality Software & Technology (VRST), 2018
- IEEE Artificial Intelligence and Virtual Reality (AIVR), 2018
- IEEE Conference on Virtual Reality and 3D User Interfaces (IEEE VR), 2018

- IEEE VR Workshop on Perceptual & Cognitive Issues in Augmented Reality (PERCAR), 2017
- IEEE VR Workshop on Collaboration and Virtual Environments (CoVE), 2016
- IEEE VR Workshop on Perceptual & Cognitive Issues in Augmented Reality (PERCAR), 2016
- ACM Symposium on Virtual Reality Software & Technology (VRST), 2016
- ACM Symposium on Virtual Reality Software & Technology (VRST), 2015
- IEEE ISMAR Workshop on Human Perception and Psychology in Augmented Reality (HPPAR), 2015
- IEEE Virtual Reality Conference (IEEE VR), 2015
- IEEE VR Workshop on Collaboration and Virtual Environments (CoVE), 2015
- IEEE VR Workshop on Perceptual & Cognitive Issues in Augmented Reality (PERCAR), 2015
- International Symposium on Visual Computing (ISVC), 2015
- IEEE Virtual Reality Conference (IEEE VR), 2014

EDITORSHIPS:

- Associate Editor, IEEE Transactions on Visualization and Computer Graphics, 2022 - Present
- Associate Editor, Frontiers in Virtual Realty – Perception and Human Behaviour, 2022 - Present
- Lead Guest Editor, Frontiers in Virtual Reality – Perception & Neuroscience in XR, 2021
- Guest Editor, Multimodal Technologies and Interaction – Perception and Cognition in XR, 2021
- Lead Guest Editor, Frontiers in Virtual Reality – Perception & Neuroscience in XR, 2020
- Guest Editor, Multimodal Technologies and Interaction – Perception and Cognition in XR, 2020
- Editorial Board, Frontiers in Virtual Reality, 2023
- Editorial Board, Frontiers in Virtual Reality, 2022
- Editorial Board, Frontiers in Virtual Reality, 2021
- Editorial Board, Frontiers in Virtual Reality, 2020
- Editorial Board, Frontiers in Virtual Reality, 2019
- Editorial Board, Frontiers in Robotics and AI – Virtual Environments, 2019
- Editorial Board, Frontiers in Robotics and AI – Virtual Environments, 2018
- Editorial Board, Frontiers in Robotics and AI – Virtual Environments, 2017
- Editorial Board, Frontiers in Robotics and AI – Virtual Environments, 2016
- Editorial Board, Frontiers in Robotics and AI – Virtual Environments, 2015
- Editorial Board, Frontiers in Robotics and AI – Virtual Environments, 2014

PROFESSIONAL REVIEWING:

JOURNAL REFEREE:

- ACM Transactions on Applied Perception (ACM TAP), 2023
- IEEE Transactions on Visualization and Computer Graphics (TVCG), 2021
- Frontiers in Virtual Reality, 2022

- Frontiers in Virtual Reality, 2021
- Frontiers in Virtual Reality, 2020
- ACM Transactions on Applied Perception (ACM TAP), 2019
- Frontiers in Robotics and Artificial Intelligence, 2019
- IEEE Transactions on Visualization and Computer Graphics (TVCG), 2019
- ACM Transactions on Applied Perception (ACM TAP), 2018
- Frontiers in Robotics and Artificial Intelligence, 2018
- IEEE Transactions on Visualization and Computer Graphics (TVCG), 2018
- International Journal of Human-Computer Studies (IJHCS), 2018
- Presence: Teleoperators and Virtual Environments, 2018
- University of Mississippi - Undergraduate Research Journal, 2018
- ACM Transactions on Applied Perception (ACM TAP), 2017
- Frontiers in Robotics and Artificial Intelligence, 2017
- IEEE Transactions on Visualization and Computer Graphics (TVCG), 2017
- International Journal of Human-Computer Studies (IJHCS), 2017
- Frontiers in ICT, 2016
- IEEE Transactions on Visualization and Computer Graphics (TVCG), 2016
- Quarterly Journal of Experimental Psychology, 2015
- Presence: Teleoperators and Virtual Environments, 2015
- Frontiers in Virtual Environments, 2014
- ACM Transactions on Applied Perception (ACM TAP), 2014
- International Journal of Human-Computer Studies (IJHCS), 2014
- ACM Transactions on Applied Perception (ACM TAP), 2013
- IEEE Transactions on Visualization and Computer Graphics (TVCG), 2013
- International Journal of Human-Computer Studies (IJHCS), 2013
- International Journal of Human-Computer Studies (IJHCS), 2012
- International Journal of Human-Computer Studies (IJHCS), 2011
- International Journal of Human-Computer Studies (IJHCS), 2010
- International Journal of Human-Computer Studies (IJHCS), 2009
- International Journal of Human-Computer Studies (IJHCS), 2008

CONFERENCE REFEREE:

- IEEE Conference on Virtual Reality and 3D User Interfaces (IEEE VR), 2020
- IEEE VR Workshop on Workshop on Immersive Sickness Prevention (WISP), 2020
- ACM Symposium on Applied Perception (SAP), 2019
- IEEE VR IEEE VR Workshop on Novel Input Devices and Interaction Techniques (NIDIT), 2019
- ACM Southeast Conference (ACMSE), 2018
- ACM Spatial User Interaction (SUI), 2018
- ACM Special Interest Group on Computer–Human Interaction (SIGCHI), 2018
- ACM Symposium on Applied Perception (SAP), 2018
- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2018
- ACM Special Interest Group on Computer–Human Interaction (SIGCHI), 2017
- IEEE Symposium on 3D User Interfaces (3DUI), 2017
- IEEE Virtual Reality Conference (IEEE VR), 2017
- IEEE Virtual Reality Conference (IEEE VR), 2016
- ACM Spatial User Interaction (SUI), 2015
- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2015
- IEEE Symposium on 3D User Interfaces (3DUI), 2015

- IEEE Virtual Reality Conference (IEEE VR), 2015
- ACM Special Interest Group on Computer-Human Interaction (SIGCHI), 2014
- IEEE Information Visualization, 2014
- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2014
- ACM Special Interest Group on Computer–Human Interaction (SIGCHI), 2013
- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2013
- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2012
- ACM Intl. Conference on Multimodal Interactions (ICMI), 2012
- IEEE Virtual Reality Conference (IEEE VR), 2012
- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2011
- IEEE Virtual Reality Conference (IEEE VR), 2011
- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2010
- IEEE Virtual Reality Conference (IEEE VR), 2010
- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2009
- IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2007
- IEEE Virtual Reality Conference (IEEE VR), 2006

REVIEW PANELS:

- National Science Foundation (NSF) review panel, 2016

OTHER PROFESSIONAL SERVICE:

- Session Chair, IEEE Virtual Reality Conference (IEEE VR), 2022
- Session Chair, ACM Symposium on Virtual Reality Software and Technology (VRST), 2020
- Session Chair, IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR), 2020
- Session Chair, IEEE Virtual Reality Conference (IEEE VR), 2014
- Judge, IEEE 3D User Interfaces Contest (IEEE 3DUI), 2013
- Judge, IEEE 3D User Interfaces Contest (IEEE 3DUI), 2012
- Judge, Mississippi School for Math & Science – Game Design Contest, 2011
- Judge, Mississippi School for Math & Science – Game Design Contest, 2010
- Student Volunteer, IEEE Virtual Reality Conference (IEEE VR), 2010

Teaching Accomplishments

Teaching Accomplishments

Contents

Highlights

- Teaching evaluation scores have consistently exceeded departmental averages by roughly 10% points each semester across both MSU and UM.
- Developed new courses in Data Science, Immersive Media, and Virtual Reality Development at both the undergraduate and graduate levels.
- Awarded an Ottilie Schillig Special Teaching Projects Grant ($3000) for building a Virtual and Extended Reality Development course.
- Mentored 35 students through research-based experiential cap-stone projects with 45.7% of those students entering graduate programs.
- Taught classes "virtually face-to-face" using virtual reality during COVID-19 pandemic.
- Developing new interdisciplinary minor in Neuroscience.

Having served as an assistant professor at Mississippi's two leading research universities, I have had the opportunity to develop and refine my approach to teaching. In this chapter, I will share my teaching accomplishments and philosophy, as well as metrics indicating the effectiveness of my approach. When joining Mississippi State University (MSU), I was generously granted 2 years credit towards tenure for work conducted at my previous institution, University of Mississippi (UM). As will become apparent in the *Motivation* section, I consider this collective work to be *a continuous endeavor to serve the students of Mississippi*. My continued work at MSU has allowed me to broaden my network of students and collaborations within our state in ways that would not have otherwise been possible having served a single institution.

In the following sections, I demonstrate my enthusiasm for teaching, how I approach teaching both philosophically and practically, and metrics indicating the effectiveness of my approach. This includes a discussion of the philosophies and motivations that I bring to the classroom, my passion for traditional and experiential learning, student feedback from teaching evaluations, and metrics describing pursuit of graduate education by students that I mentored through experiential learning programs.

Teaching Philosophy

Motivation: A single motivation has run through nearly every choice that I have made in my professional life – the desire to influence cultural improvement in education, society, and science. More specifically, I am interested in influencing cultural growth within Mississippi. As an EPSCoR jurisdiction, Mississippi has been identified as a location of particular need for improved STEM capacities. I have a somewhat unique perspective on this topic. I was born, raised, and received much of my education in Mississippi. As a result, I have been fortunate to see the many facets of our educational system from kindergarten through graduate school. To gain perspective, I have traveled extensively, worked with top labs and researchers in my field, and pursued postdoctoral education from one coast of the US to the next. I did these things with the goal of coming back to Mississippi and sharing my experiences and perspectives. Using these insights, I am investing my career in building a scholarly community in Mississippi that is based on the philosophical principle that we are at our best when education cuts across classrooms, labs, disciplines, and ways of thought. In short, excellence in Mississippi is my mission. Below, I provide specific details and metrics regarding my teaching goals and outcomes in two specific categories: Traditional Learning and Experiential Learning. I firmly believe that in order for our students to excel and push our discipline forward, they must incorporate both traditional, classroom-based education and experiential, hands-on education.

Be Honestly Wrong: A thread that runs deeply through my teaching and personal philosophy is that we should not be afraid of being honestly wrong. I tend to make my lectures very interactive and attempt to provoke conversation around our specific topics. To support these conversations, I try to provide an environment where students are not afraid to be wrong, misunderstand, or simply not know something. I like to tell them that "we learn by making mistakes and there are too many mistakes to make them all yourself" thus we must learn from each other's honest mistakes. If someone answers a question incorrectly, that isn't a point of failure. It is an opportunity to learn something while also helping your classmates. Usually, if one student misunderstands something, there are several others too. I also apply this philosophy to myself. Occasionally, students might ask me something that I honestly do not have an answer to. In this case, I will tell them "I don't know, but let's find out!". Sometimes this requires us to write some experimental code on the fly or do a quick internet search on the spot. Regardless, someone is going to learn something from the exercise.

Question, Question, Question: I usually finish my lectures around 5 to 10 minutes early by design. This is not the end of class, however. Instead, this is the beginning of what I call the "Question Game". In the Question Game, I tell the students that they can leave as soon as they ask X number of questions (usually between 7 and 12 depending on how much time it left). To make it insightful but also fun, the questions must follow this pattern: initial questions are on the topic of today's lecture, followed by questions on the course topic, followed by general computer science questions, and ending with one (almost) anything-goes question. This exercise gets the students more comfortable asking questions throughout the class and helps them to identify areas where they may need more help. Student comments included on our anonymous teaching evaluations also support the effectiveness of this exercise, see *Teaching Evaluations*.

Bridge the Last Mile: If you give students an easy way out, even if it is clearly not in their favor, they will often take it when frustrated and tired. I have particularly observed this in classes where programming is a significant requirement. Students will get frustratingly close to a workable solution, become discouraged, and simply turn in their code for whatever partial credit they can get. We see this pattern in many areas of education, science, and engineering. It is sometimes referred to as the "last mile problem" because the last portion of a task feels toughest or even insurmountable. In the classroom, this is burdensome for both the student and the teacher. For the student, this prevents them from bridging the often small, remaining gaps in their knowledge before gaining proficiency. The teacher is then burdened with the task of accurately and consistently gauging what level of completion and correctness was achieved by each individual student.

I design my assignments to circumvent the last mile problem as much as possible by distilling the core knowledge that each lecture and assignment is trying to convey into a form that is assuredly completable given sufficient time and study. Students are then required to demonstrate their working code in real-time to the instructor or grader, often actually composing the code during the demonstration. They are not graded on whether or not the code functions as full function is required for the demonstration. Instead, the students are graded on how long it takes them to achieve a successful demonstration over the course of a week. It has shown itself to be effective in improving

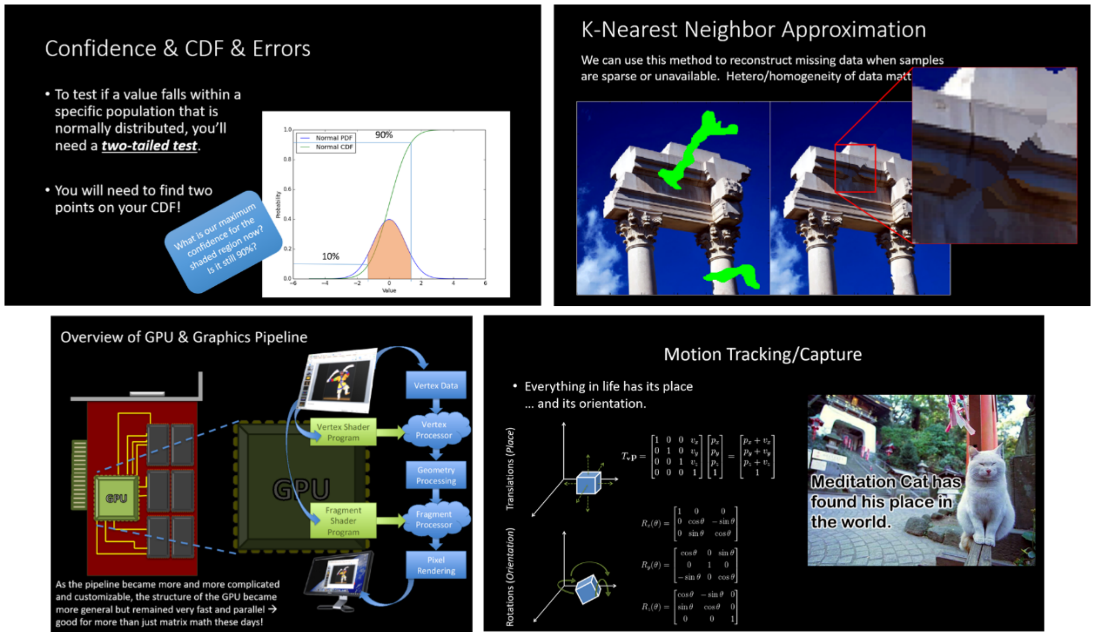


Figure T 1 - A sample of representative slides illustrating the style I typically use when lecturing. The top row slides are from Fundamentals of Data Science. The bottom row slides are from Computer Graphics and Virtual and Extended Reality Development respectively.

overall performance and completion of the assignments. For instance, in *CSE 2383 – Data Structures and Analysis of Algorithms* by the time students reach our assignment on Binary Search Trees (an assignment that we assess for ABET accreditation) historically roughly 30% of the students simply do not submit the assignment under a traditional assignment paradigm. Under the demonstration-based paradigm, 10% of the students in my section did not submit the assignment with 83.33% of the class earning a grade of 70% or higher. This exceeds our ABET accreditation criteria. We have since adopted this approach for all sections of the course.

Teaching Approach

I enjoy studying how successful scientists, engineers, and educators approach teaching and elicit excitement among students. For instance, Richard Feynman and Carl Sagan are known for their talent in capturing a topic's beauty and elegance while preserving its complexity and nuance. Santiago Ramón y Cajal and Richard Hamming are synonymous with continuous improvement and the relentless pursuit of excellence. Edward O. Wilson and Nancy Kanwisher embody the height of scientific enthusiasm and curiosity. These are some of the scholars that I most admire and try to emulate in the classroom. I have studied their writings, lectures, and interviews in order to better understand their ability to inspire, educate, and motivate excellence among students.

We cannot afford to neglect, however, that we are firmly in the Internet age. The preferred styles and methods of content delivery are fundamentally different for students now versus 10 or more years ago. For this reason, I also try to incorporate styles of presentation seen with quality online educators, such as Destin Sandlin (*Smarter Every Day*), Hank Green (*Crash Course*), and Tony (*This Old Tony*). To further pursue this idea, I have attended a YouTube sponsored educator event in 2018 in order to meet and learn firsthand from professional educational content creators (*YouTube Thinkercon 2018*).

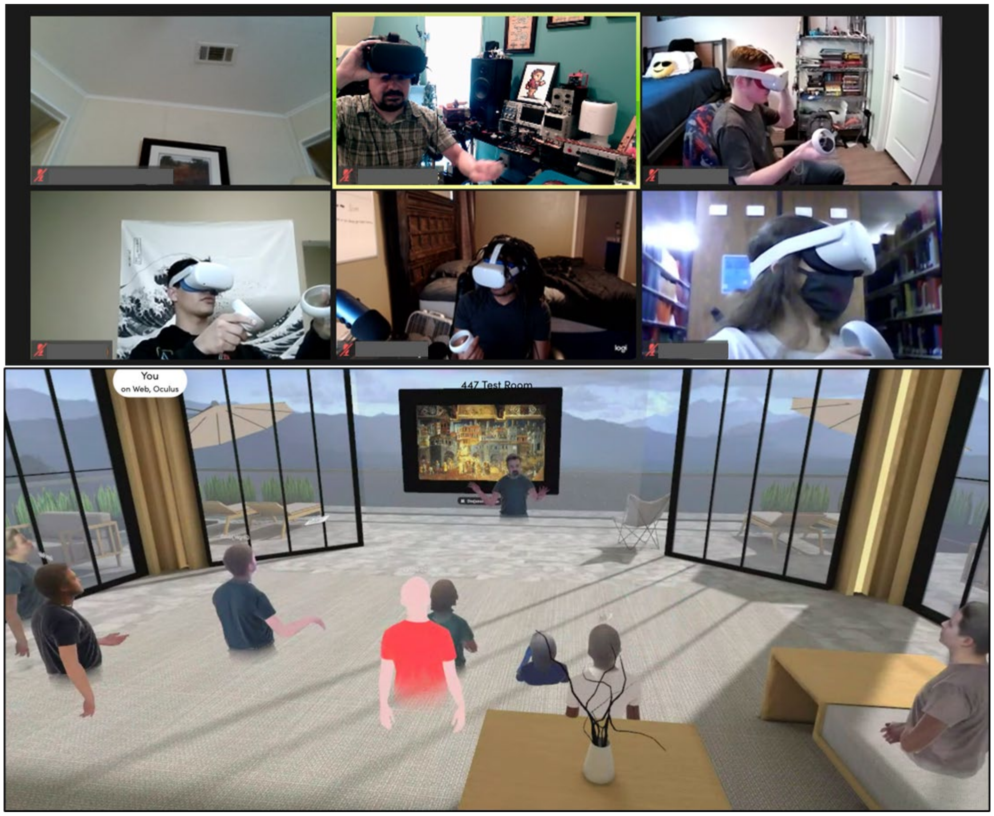


Figure T 2 - (top) Students wearing VR headsets to attend class during COVID-19, (bottom) A view from the virtual classroom. Students had the option of having realistic self-avatars.

Traditional Teaching

When I say *Traditional Learning* I am specifically referring to lecture-style teaching. As you will see in this section, this usually refers to teaching in a classroom but can, as necessitated by campus closures due to COVID-19, also refer to online or video-based teaching. I firmly believe that there is no substitute for learning by listening to and conversing with an expert on a topic. This is why I place strong emphasis on maintaining an open and honest dialogue within my classes. I have also found it important to present course content in a modern and engaging way. Though it is difficult to convey the atmosphere and style of a presentation through figures, I have included several representative slides from a variety of courses including *Virtual and Extended Reality Development*, *Data Science*, and *Computer Graphics*, see Figure T1.

University of Mississippi: While at the University of Mississippi, I taught a variety of regular and special topics courses. The departmental tradition at the time was to teach a new course at least once per year, preferably per semester. As a result, I taught a total of 9 unique courses over the span of 6 years. I was fortunate, however, that *Fundamentals of Data Science* was a requirement for our Data Science Minor and that *Immersive Media (Virtual Reality)* was in high demand. These were both courses that I had created. This enabled me to teach these courses on a per-semester and per-year basis respectively.

Course Title	New	Redesign	Cross-listed
CSE 4433: Virtual and Extended Reality Development	X		
CSE 2383: Data Structures and Analysis of Algorithms		X	
Engr 691: Topics in Physiological Considerations for Virtual Environments	X		Neu 579
Engr 691: Topics in Spatial Perception in Virtual Environments	X		Neu 579
CSci 581: Topics in Human-Centered Computing	X		
Csci 447: Immersive Media	X		
Csci 343: Fundamentals of Data Science	X		
Csci 581: Special Topics in Virtual Reality (precursor to Csci 447)	X		
Engr 694: Research Methods in Computer Science		XX	
Csci 391: Introduction to Computer Graphics		XX	
Csci 390: Special Topics in C/C++ Programming		X	

Table T 1 - Newly Developed and Redesigned Classes at MSU (orange) and UM (blue)
*For redesigned classes, **XX** denotes a major redesign while **X** denotes a minor but critical redesign.*

I thoroughly enjoy teaching in a classroom setting. As you probably know, teaching is a "multiplayer experience" with the students picking up on the instructor's enthusiasm and vice versa. This creates a natural connection between the instructor and the class. However, when COVID-19 necessitated the closure of in-person classes during 2020 and 2021, many courses that would have otherwise been taught in the classroom were now being taught by video. This was disheartening for me as teacher, and I believe many of the students felt the same way. However, I decided to put my research into practice and teach my classes using virtual reality using a one-to-many paradigm. I setup a VR-based classroom complete with lectern, markerboards, projection screens, and a motion-captured avatar of myself. I delivered my remote lectures from this virtual classroom. The students still saw me over video just as they would any other instructor at the time, but from my perspective, I was in a real classroom[1]. The students reported that my presentations felt more natural than typical Zoom classes, which was not a surprise as I felt more natural delivering my lectures in this way. After demonstrating VR's viability for delivering lectures, I was able to acquire funds as part of the CARES Act to purchase VR headsets to distribute to students in my classes. This enabled us to hold in-person class meetings completely in VR for a group of 12 students, see Figure T2. The most striking thing that I saw when teaching in this way was that students interacted socially in a way very similar to how they would in the real world. Students would arrive early to the classroom sit with their friends and have conversations about TV shows, movies, and life. Once class was over, they would often hang out and continue conversing or discuss topics from the lecture that they may not have understood.

Developing new courses that accompany and support my long-term research goals is a fundamental part of my strategy to build a strong academic and research culture among my students. While at the UM, I developed five new classes in Computer Science, Neuroscience, and Engineering at both the graduate and undergraduate levels and performed a major redesigned of three existing courses, see Table T1. These classes were designed such that their computer science content overlapped and was contextualized with respect to a wide cross-section of other disciplines. Of these, I am particularly proud of CSci 447: Immersive Media, which was a first-of-its-kind class in Mississippi. Immersive Media is a hands-on VR software development course. Additionally, this class took a strong

[1] Example Video: https://youtu.be/zBSG0K4NqAI

interdisciplinary approach to developing virtual environments by incorporating software engineering, psychology, human factors, art, and neuroscience. I am also proud of CSci 343: Fundamentals of Data Science. This class takes a "data science from scratch" approach where students implemented numerous analysis and visual analytics methods over the course of the semester using Python. This course was the core course in UM's interdisciplinary Data Science minor. A core philosophy that I taught in this class is that data has little meaning without context and that all data are indirect representations of what we are attempting to measure. As such, a mantra of the class was "data approximates reality".

Mississippi State University: Since joining Mississippi State University, I have been under a reduced teaching load of 1:1 for Fall 2021 and Spring 2022. Additionally, I was fortunate to have been able to fund a course buyout through a recent NSF funded project for both Spring and Fall 2023. So far, I have taught *CSE 2383: Data Structures and Analysis of Algorithms* three times and *CSE 4990: Special Topics in Virtual and Extended Reality Development* twice. These are the first in-person classes that I have taught since mid-Spring 2020, and I am very glad to be back in the physical classroom again. Though teaching in VR was a reasonable approximation, it is not yet a replacement for being in the physical classroom.

I have twice piloted updated versions of my previous virtual reality development course as a special topics course. The first time was very similar to my original UM class. This course is designed to be a hybrid lection/studio-style experience. Lectures are typically detailed and technical while the projects are design focused providing significant room for creativity and innovation. Though this class has historically worked well, I believe it needed small changes for this student population. For instance, students struggled with some of the design and studio-oriented aspects of the class, often requesting *less room for interpretation* and *more constrained parameters* for their projects. I think this speaks to the quality of the engineering mindset that is taught here: MSU students expect rigorous problem definitions and deliverable specifications. During the second offering of this course, I provided more detailed specifications for assignments and offered more personalized technical feedback by holding "team report" days in class. The only major issue during this offering was the lack of lab access due to Butler Hall's renovation. This necessitated that students be issued VR headsets to take home for development. This, however, is an expensive route since each headset costs roughly $300. To facilitate this, I applied for the *2022 Ottilie Schillig Special Teaching Grants Program* and was awarded the maximum amount of $3000. These funds were used to purchase 10 additional headsets to support this and future iterations of the course. In future offerings, I plan on using these headsets in conjunction with the equipment available in the Hi5 Lab so that students can get experience developing for a breadth of off-the-shelf and industrial technologies. We have applied to add this as a new course entitled *CSE 4433: Virtual and Extended Reality Development* in time for inclusion on the Fall 2023 schedule.

I have also implemented a partial redesign of *CSE 2383: Data Structures and Analysis of Algorithms* after having first taught the class in Fall 2021. The basic content has remained the same, but two components have been improved based on feedback from students and my own observations. I first taught this course in Fall 2021 where I duplicated the lectures and assignments that have been traditionally offered in this course. My first observation was that it is not safe to assume an elementary level C++ proficiency. This is a foundational assumption of the class, and students may not succeed

Institution	CCIHE	Degrees
Mississippi State University	R1	MS
University of Mississippi	R1	MS
Jackson State University	R2	MBA
William Carey University	*NA*	MBA
Louisiana State University	R1	MD
University of Central Florida	R1	PhD
University of Utah	R1	PhD
Virginia Polytechnic	R1	PhD
USMC Officer Candidate School	*NA*	2ndLt
US Army Officer Candidate School	*NA*	2LT

Table T 2 - Postgraduate schools attended by students who partook in one of my experiential learning projects. This table includes their CCIHE classification and the degree/rank program into which the student was accepted.

if they are not able to code in C++ with at least partial proficiency. When I observed the wide variation of skill levels, I made several supplemental videos covering a "hit the ground running" introduction to C++. The feedback from students was that this was very helpful, so I have now dedicated the first three lectures of the class to providing a C++ crash course. The second observation was that students were spending excessive time working on programming assignments. When asking the students about where they were spending the most time, I found that it was on parts of the assignments that were not directly related to the data structures (e.g. creating a command line interface, implementing matrix mathematics, etc.). Too often students would become discouraged and either submit broken assignments or not submit anything at all. In subsequent offerings of this course, I changed the style of assignment to be more focused on understanding the anatomy and function of data structures. I also now require that their code must work, and they are graded on how quickly they develop a solution, see the above section *Bridge the Last Mile* for more details.

An addition to my curriculum development focuses on the creation of new, interdisciplinary Minor in Neuroscience. This was one of my primary goals when joining MSU and led me to seek out collaborators to assist in developing this program. Development of this minor is also a key component to the *Education Plan* in my current NSF CAREER grant. I have gathered a team of faculty who are working in neuroscience-oriented areas to sit on a committee to develop the curriculum and submit for approval to UCCC. These faculty are from Computer Science & Engineering, Psychology, Kinesiology, Agricultural & Biological Engineering, and Religion & Philosophy. In addition to these faculty, we have solicited the cooperation of 7 departments who host courses that fit well in the neuroscience minor curriculum. Our goal in developing this minor is to provide organized neuroscience education to students while not increasing course burdens for participating departments. As such, the curriculum is composed entirely of existing class but will likely necessitate the creation of a new prefix (NEU) to facilitate cross-listing of courses. Informed by the curricula from similar programs at peer institutions, we have developed a draft that consists of 18 hours (9 core and 9 elective courses). A draft of this curriculum is shown in Table T3.

Core Courses (9 Hours)
HON 3163/NEU XXXX - Intro to Neuroscience
PSY 4424 - Sensation & Perception
PSY 3713 - Cognitive Pyschology ***(Choose One)***
PSY 3723 - Cognitive Neuroscience

Elective Courses (9 Hours) - Organized by Topic Area	
Computational Neuroscience	**Cognitive Neuroscience**
CSE 4633 - Artificial Intelligence	PSY 3723 - Cognitive Neuroscience
CSE 4683 - Machine Learning and Soft Computing	PSY 4713 - Language and Thought
CSE 4643 - AI Robotics	PSY 4653 - Cognitive Science
CSE 4433 - Virtual and Extended Reality	PSY 4743 - Psychology of HCI
CSE 4663 - Human-Computer Interaction	PSY 4753 - Applied Cognitive Psychology
Neural Engineering	**Neuromechanics & Neurobiology**
ABE 3413 - Bioinstrumentation	EP 4703 - Neural Control & Human Movement
ABE 4613 - Biomechanics	PE 3223 - Motor Development & Movement
ABE 4803 - Biosystems Simulation	PE 4853 - Motor Learning & Skill Analysis
ABE XXXX - Neural Engineering	BCH 4113 - Essentials of Molecular Genetics
ABE XXXX - Biomedical Signals and Sensors	BCH 4713 – Molecular Biology
ABE 4323 - Physiological Systems in Biomedical Eng.	BIO 4133 - Human Genetics
	BIO 3004 - Human Anatomy
Philosophy of Neuroscience	
HON 3163/NEU XXXX - Discoveries in Neuroscience	
PHI 4223 - Philosophy of Cognitive Science	

Table T 3 -Draft curriculum for interdisciplinary Neuroscience Minor. Course numbers whose addition to the catalog are pending are shown as XXXX.

Experiential Teaching

Since beginning a tenure-track appointment, I have focused on establishing a culture within my classes and my lab that integrates traditional teaching with experiential learning that focuses on teaching through research. I believe that research without teaching risks becoming myopic, neglecting to address the *longitudinal development and growth of our discipline*. Similarly, teaching without incorporating aspects of research overlooks the innovation needed to *advance the practice of our discipline*. I leverage my unique position in academia to bridge the gap between teaching and research by facilitating both traditional and experiential learning in my classes. I strive to engage students at all educational levels in the development of fundamentally new scientific directions. In the following sections, I present the story and associated data describing the success of this particular approach.

University of Mississippi: At my first tenure-track appointment in the Department of Computer and Information Science (CIS) at the University of Mississippi, I was a strong proponent of actively involving undergraduates in research as part of their experiential education requirement. In order to graduate, students in this department were required to individually complete a semester long capstone project. Starting in Fall 2016, I began offering students the opportunity to perform research in my lab to fulfill their capstone project requirement for graduation. This was atypical as these projects were historically software development oriented. My approach to capstone projects was met with resistance, but I fought to give students the opportunity to experience the excitement of research with the hope of getting talented students interested in pursuing a graduate education. The metrics that I have collected indicate that this program has been successful. <u>When surveyed three months post-graduation[2], 18% of UM CIS graduates reported an intent to pursue a graduate degree. However, 50.0% (12 of 24) of the students who participated in a research-based capstone project were accepted into graduate programs. Out of all undergraduate students who have participated in projects, not solely capstone projects, in my UM lab, 45.7% (16 of 35) pursued graduate education.</u> Table T2 shows the graduate programs to which these students were accepted. In all honesty, sampling bias is almost certainly a factor since students who are interested in going to graduate school are certainly more likely to want to do a research project. However, some of these students reported that they had not previously considered graduate school until they had a chance to experience research firsthand.

<u>Mississippi State University:</u> My current appointment is in Computer Science & Engineering (CSE) at Mississippi State University. Though the undergraduate curriculum is structured differently than my previous institution, I am utilizing a similar strategy to engage undergraduate students in experiential learning. Since Spring 2022, I have mentored 20 undergraduates and 1 high school student (MSMS). These students are from a wide range of majors including Computer Science, Psychology, Computer Engineering, Software Engineering, and Electrical Engineering. They are participating in a variety of capacities including Senior Design courses, Honors Theses, CSE 4800, and volunteering. Although only two of these CSE undergraduate students have reached graduation, both have applied to graduate school, and several continuing students have also expressed interest in pursuing graduate education. Of the students that I have mentored in this capacity, several have received internships in both industry and government labs (e.g. Camgian and Air Force Research Lab).

Teaching Evaluations

My teaching evaluations since joining MSU have been largely positive and are comparable to those from my previous institution. I have included a plot of the per semester average evaluation scores in Figure T3. Departmental averages from MSU are shown in maroon while departmental averages from UM are shown in yellow. Evaluations from UM are reported on a 5-point while MSU reports on a 4-point/percentage scale. I have normalized the UM and MSU scores such that they are on a percentage scale.

Numerically, my evaluations have been quite positive (UM: 90.18%, MSU: 96.96%) and exceed the departmental averages for both UM (79.39%) and MSU (86.28%). My overall evaluation scores from UM averaged 10.79% points higher than the departmental average. Similarly, my overall evaluation scores from MSU averaged 10.68% points higher than the departmental average. The student comments from both institutions have also been very positive and are included below. I take my teaching obligations very seriously, and I believe this is reflected in both the numerical data and the students' comments.

[2] These are the most recent statistics available from my previous institution and are current as of December 2020.

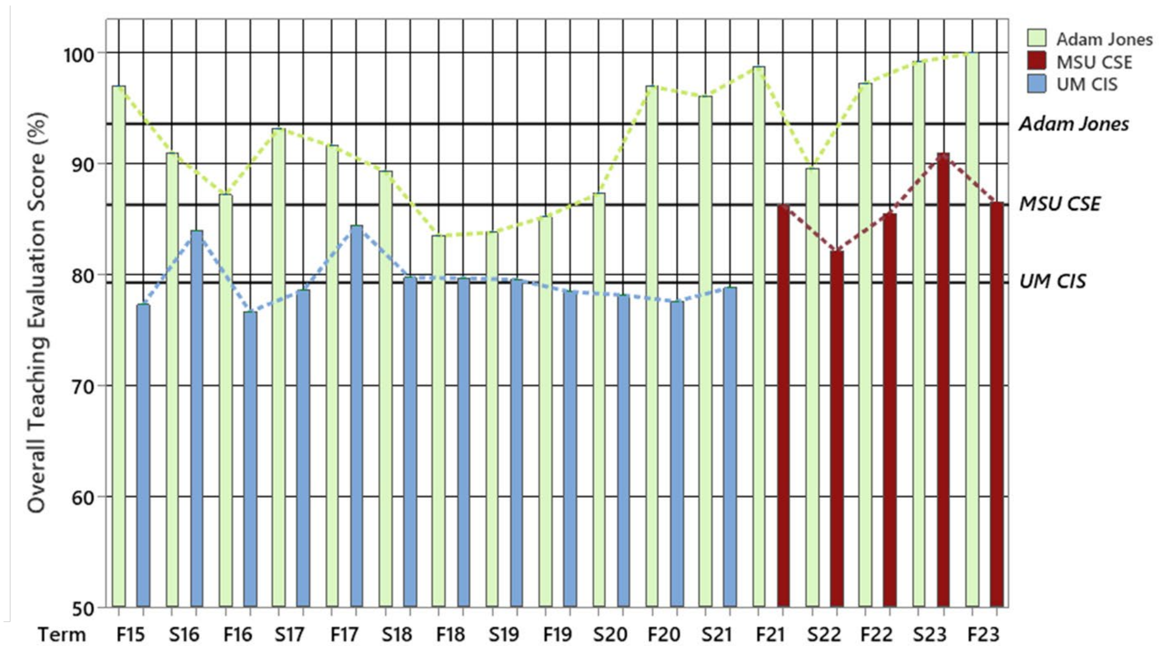


Figure T 3 - Average teaching evaluation scores. Both UM and MSU departmental averages are provided for reference. Horizontal lines representing the average across all semesters for each institution. Note that these are presented on a percentage scale to facilitate direct comparisons as UM and MSU use a 5-point and 4-point/percent scales respectively.

Though the student comments are generally quite positive, I receive the most personal benefit from constructive feedback. I have included samples of these below as well. For instance, constructive feedback from UM led me to make changes in the way I taught students to code and collaborate in Unity 3D, as well as bringing better contextualization to the lecture content with respect to the programming assignments. The constructive feedback from MSU has also been very helpful, especially given the differences in the student population. For instance, I substantially restructured the programming assignments in *Data Structures* because the reviews were right – the assignments were too large with respect to the topics being evaluated. I also applied for and was awarded a teaching grant to get more headsets for the *VXR Development* class, addressing an important criticism from the evaluations. The third comment is harder to address but important none-the-less. The nature of an emerging technology is that it undergoes fast evolution, and its associated SDK evolves at a similar speed. I now request that the students use specific versions of the Oculus SDK to avoid accidentally bumping into "experimental feature" integration problems. Perhaps the most important lesson learned from comments like this is that I needed to better manage the students' expectations when working with immerging technologies and emphasize engineering around the limitations that they may face.

UM Positive Feedback
"I wouldn't make any changes to this class. <u>This has easily been the most organized CSCI class I have taken at Ole Miss to date</u>." – Data Science, Spring 2017 *"Dr. Jones is by far my favorite professor that I've had thus far at Ole Miss. He is one of the most enthusiastic and engaging professors I've had and he is a fantastic instructor. He makes all of his lectures engaging and fun and make every topic apply to your life and makes everything that much more fun to learn and do."* – Data Science, Fall 2017

"He is very approachable and easygoing. He loves teaching this course and loves to see what comes out of it." – VR Dev, Spring 2018

"Dr. Jones is very enthusiastic about data science, and the projects we completed were very appropriate and fun in understanding the material. I enjoyed it! Especially the structure of lectures on a topic, getting your hands on some data and practicing, and then getting evaluated." – Data Science, Spring 2020

UM Constructive Feedback
"I think the course could be better if the Dr. Jones went more straight to the topic. That we went more over the code for the challenges in class." – VR Dev, Spring 2017 *"The Unity program that is used in the course does not allow for more than three people to collaborate on a single project. For groups bigger than that it becomes increasingly difficult for the whole group to participate on the projects."* – VR Dev, Spring 2018 *"A little better consistency between the lectures and the assignments."* – VR Dev, Spring 2019

MSU Positive Feedback
"Dr. Jones is very charismatic and kept the class interesting. He does a really great job of explaining data structures in simple terms and relating them to relevant topics and/or metaphors. When working on assignments, he was really great at explaining how you needed to approach a programming assignment and/or change your approach to the current assignment." – Data Struct, Fall 2021 *"The structures are broken down in a way that makes them genuinely interesting and easier to understand. The "X questions and then we'll leave" when lecture content gets finished up early is a good way of filling out the time while still being productive."* – Data Struct, Fall 2022 "*This was one of the best teachers I have had. Dr. Jones made class interesting making me want to go to class everyday instead of dreading it. He also was very informative we really got to understand the material, while also answering any questions that we might have at the end of class. I can say that I never left class feeling confused about an assignment we had gone over that day."* – Data Struct, Fall 2022

MSU Constructive Feedback
"This class was overly difficult. Programming assignments were virtually impossible to complete. They were designed for someone with a much higher skill level than the students in this class possessed. It was one of the hardest classes I have taken in college so far." – Data Struct, Fall 2021 *"Maybe more headsets and probably guidance on how to do more stuff within Unity. However, I do love searching for answers. Sometimes it can be helpful though."* – VXR Dev, Fall 2022 *"I think that the third project could be changed to not involve an experimental Oculus feature, like the hand-tracking. It was needlessly difficult to set that up."* – VXR Dev, Spring 2022

Attachments: Sample Syllabi

I am including three examples of my syllabi from both MSU and UM. These include *CSE 4433: Virtual and Extended Reality Development*, *CSci 343: Fundamentals of Data Science, and CSE 2383: Data Structures and Analysis of Algorithms*. Each of these classes are among my favorites to teach and span a breadth of topics.

- *CSE 4433: Virtual and Extended Reality Development* is a course that I have created to provide students with the unique software development skills needed for virtual, augmented, and extended reality. Software development in these contexts requires knowledge, skills, and experience that is not typical for on-screen software. I have loosely modeled this course around Carnegie Mellon University's Building Virtual Worlds, which serves as a cornerstone for their Entertainment Technology degree program.

- *CSci 343: Fundamentals of Data Science* is a course that I created at the University of Mississippi to teach students the fundamentals of data science and visual analytics. However, I believe the two most important skills from this course were: 1) data-oriented thinking required to be a capable scientist, engineer, and modern citizen, 2) fundamentals of data handling, organization, and processing. Both of these skills were aspects of the UM curriculum that I felt needed further support and thus were incorporated into the course's structure from the ground up.

- *CSE 2383: Data Structures and Analysis of Algorithms* is a required part of our undergraduate curriculum that I regularly teach. After my first semester teaching this course, I redesigned the assignment and grading structure of the class in a way that more closely reflects the students' development of their data structure skills and also scales better with the growing enrollment of CSE students.

CSE 4433/6433: Virtual and Extended Reality Development - Fall 2023

General Information

This is an evolving document and will almost certainly be updated as the semester progresses. Please keep an updated view of the online version of this document.

Meeting

When: Tuesday, Thursday - 3:30PM to 4:45PM
Where: Butler Hall - Room 100, Hi5 Lab in Butler Hall - Room 201

Description

This course covers applications, methods, research, and technologies used in the development of interactive virtual and extended reality environments. Students will develop immersive environments and interactive experiences using industry standard technologies. Topics include virtual reality, mixed/augmented reality, human factors, motion capture, 3D content authoring, scientific applications, and current trends.

With the growing popularity and wide-spread adoption of inexpensive virtual reality (VR) and extended reality (XR) devices (especially those geared toward gaming), there is growing demand for software developers who have experience working with these technologies. Demand for VR/XR developers is out pacing the supply. This course aims to fill that gap by providing students with hands-on VR/XR development experience while gaining an in-depth understanding of the principals underlying these technologies.

Instructional Staff

Instructor

J. Adam Jones, PhD
E-mail: jaj33@msstate.edu [mailto:jaj33@msstate.edu]
Office: Butler Hall - Room 323
Office Hours:
T, Th from 12:30PM to 3:30PM
Others by appointment
(If you can't make it to my regular office hours, don't hesitate to ask for an appointment at a convenient time!)

OFFICE HOURS POLICY:
Office Hours are first-come-first-serve and visits are limited to 20 minutes. Please keep in mind that office hours are for all my students. This includes students from classes, research projects, honors projects, teaching/research assistants, and laboratories.

Topics

- History of VR/XR
- Anatomy of VR/XR Systems
- Introduction to Computer Graphics
- Immersive & Pictorial Communication

- Ethics in VR/XR
- Unity3D
- 3D Modeling
- Immersive Displays
- Optical Systems
- Motion Tracking
- Graphics Hardware
- Human Factors
- Spatial Sensation & Perception
- Simulator Sickness
- Presence & Immersion
- Development & Interaction Methods
- Navigation/Locomotion in VR
- Hand-based Interaction
- Virtual Agents
- Telepresence

Grading Method

Students in the class will be graded using three evaluations: Programming Assignments, Presentations, and Participation. Programming assignments are graded in studio-style demonstrations of the VR/XR environments built by the students. Late assignments will be accepted with a penalty of 10 points per day for up to 9 days. Presentations will consist of students discussing specified topics in a professional, proficient, and effective manner. Participation is evaluated as participation in group and in-class activities as indicated by team evaluations. Students taking this course for graduate credit will be expected to write a research paper on a specified VR/XR topic. This paper will contribute to their grade as if it were an additional programming assignment. Students taking this course for graduate credit (CSE 6433), will also be required to submit a research assignment prior to the last day of class. For grading purposes, this research assignment will count as an additional programming assignment. Evaluations will contribute to the students' final grade based on the following percentages:

```
Programming Assignments 70%
Presentations   20%
Participation   10%
------------------
Total   100%
```

Grading Scale

All evaluations are graded on a 100-point scale. Their cumulative value (adjusted according to the above percentages) determine the final grade in the course. The "letter grade" to point mapping is as follows:

```
A       90] and above
B       (90 – 80]
C       (80 – 70]
D       (70 – 60]
F       (60 and below
```

Communication

The primary means of communication for this class is through your official MSU email address. This preferred method of directly contacting the instructor. Students may expect a response within 24-48 hours.

We will also have a class Discord where students can discuss class topics and share the virtual environments that they build during this class. The link to our Discord server will be posted on Canvas. You are encouraged to post questions, problems, bugs (and fixes), and such on Discord. You are free to help each other learn and overcome challenges in this course!

Assignments will be posted on Canvas. Assignments are graded in class during "demo week" (or in some cases "demo day"). Assignments are due on the first day of demo week.

University Policies

The Mississippi State University Syllabus contains all policies and procedures that are applicable to every course on campus and online. The policies in the University Syllabus describe the official policies of the University and will take precedence over those found elsewhere. It is the student's responsibility to read and be familiar with every policy. The University Syllabus may be accessed at any time on the Provost website under Faculty and Student Resources and at https://www.provost.msstate.edu/faculty-student-resources/university-syllabus [https://www.provost.msstate.edu/faculty-student-resources/university-syllabus]

CSci 343: Fundamentals of Data Science
Class Syllabus – Spring 2020

Instructor	J. Adam Jones, PhD Email: jajone13@olemiss.edu Office: Kinard 275, Insight Park 215 Office Hours: Tuesday & Thursday 10:00AM – 2:00PM *(Appointments are strongly encouraged)*
Teaching Assistant	Deep Phuyal Email: dphuyal@go.olemiss.edu Office: Weir 232 Office Hours: Monday, Wednesday, & Friday 8:00AM – 12:00PM Tuesday & Thursday 4:00PM – 5:30PM (Others by appointment)
Class Meeting	Tuesday & Thursday 2:30PM – 3:45PM Weir Hall, Room 106
Prerequisite	CSci 112, CSci 203, CSci 256, or CSci 251

Description

Data science is the study of discovering knowledge from data. This course explores the field using a broad perspective. Topics include data collection and integration, exploratory data analysis, descriptive statistics, prediction and regression, evaluating and communicating results. Significant programming is required.

Textbook

Joel Grus, *Data Science from Scratch: First Principles with Python*. 2015. O'Reilly Press.

Topics Covered

For this class, we plan to cover the follow topics: Basic Statistical Analysis, Hypothesis Testing, Data Distributions, Sentiment Analysis, Image Analysis, Data Modeling, Genetic Algorithms, Linear Regression, Polynomial Regression, Logistic Regression, Nearest Neighbor Approximation, Decision Trees, Neural Networks. This list is contingent on how far and how quickly we progress in lecture. The objective of this course is to enable you to learn and gain proficiency in the application of basic data analysis, visualization, and modeling.

Grading Policy

Programming Challenges, Tests, Quizzes, Readings, etc. (collectively referred to as Challenges) will be assessed on a points basis. As you successfully face Challenges, you will be awarded Experience Points (referred to as XP). Challenges will be issued approximately weekly and cover the readings, chapters, assignments, independent student research, and lecture material. **There will be no make-up Challenges for any reason**. Programming Challenges have a point value of 250XP and are graded based on two criteria: time taken to correctly complete the assignment (200XP) and the understandability & readability of the submitted code (50XP). There is a submission window of 6 weekdays for each Programming Challenge (and weekends don't count as being late!). Here is the breakdown of points for each day of the submission window:

Day 1: 200XP
Day 2: 180XP
Day 3: 160XP
Day 4: 140XP
Day 5: 120XP
Day 6: 100XP

If you submit your Programming Challenge solution after the submission window has closed, you may only receive points for the second grading criteria (understandability & readability) and receive a maximum of 50XP. NEVER SKIP SUBMITTING AN ASSIGNMENT! As you will learn in this class, most things exist on a gradient. Your Challenge grades are no different. The only way to get 0XP is to *not submit anything*. Partial credit will always be awarded as long as you adhere to the above guidelines and make a good faith effort. See the section on *Submitting Programming Challenges* for the specific guidelines on submitting your challenges.

All other challenges are worth 100XP each unless otherwise specified. Bonus XP will be awarded throughout the semester. Sources of bonus XP will be assigned by the instructor and may include optional assignments, in-class participation, engagement in out of class activities, or research activities.

Your final grade will be calculated as a percentage of the total XP (excluding bonus points) available during the semester. A letter grade of A, B, C, D, or F will be assigned based on this percentage (no plus/minus).

Submitting Programming Challenges

You may begin making your solution to a programming challenge as soon as it is posted. Once the submission window for the challenge opens, you have 6 days to demonstrate your working solution to the TA. You can either demonstrate your solution during the TA's office hours or schedule a time to meet with the TA. The opening of the submission window is not a "due date". You are expected to have completed the assignment (and asked for any needed help) prior to the submission window. Once you have successfully demonstrated your solution, you must upload it to Blackboard within 24 hours. Otherwise, your grade will be assessed based on the next the day within the submission window that it was uploaded. Be sure to have completely read the "Deliverables" section of each Challenge document ***before*** submitting to Blackboard. Not following these specific instructions will result in loss of points. Questions and emails to the instructor about a programming challenge must be submitted before 5PM the day before the submission window opens.

Class Attendance

Attendance is mandatory for this course. Attendance is taken at the beginning of class. If you come in late, you must notify the instructor at the end of class. Otherwise, you will be counted as absent. There are no make-up challenges. You may make arrangements to submit programming challenge solutions early. If you miss 3 days or less, you will earn a *drop-quiz*. This *drop-quiz* will allow you to replace your lowest quiz grade with the average of your remaining quizzes. It is important to note, that the university has a policy of verifying that all students have attended class during the first two weeks of the semester. If you are not counted as present during the first two weeks, this may affect enrollment and financial aid. Since this class requires you to be present for all meetings, this should not be a problem for you. However, understand that all classes at the university have to perform this verification procedure. As such, it is vitally important that you notify the instructor (after class has ended) if you were late to class.

Academic Misconduct

This is a data science class – a class that combines computers, programming, and statistical analysis. We use these tools to check for cheating and plagiarism. Your challenge solutions will be statistically analyzed and compared with the solutions of your classmates and submissions from previous semesters. This is an automated process. Let me be candid about this… The analysis software is REALLY good. It's almost creepy how good it is at catching people. You don't want to try cheating. It will catch you. Plus, it complicates both of our lives. Keep life simple. Earn your points (even if they are small). Seriously, you will be better off not demoing your code than turning in something that is not your own work.

The academic discipline policy of the University of Mississippi will be strictly followed in this course. Violation of the policy may result in anything from failure on an assignment to expulsion from the course, depending on the severity of the violation. The student should refer to the "M" book for general definitions of academic misconduct.

Course Specific Academic Honesty Rules

You are encouraged to work together on homework problems. All challenges, tests, quizzes, and graded programming assignments (including bonus assignments) are to be individual work unless noted otherwise in the assignment. Students can discuss general strategies for programming assignments using examples from the book or lecture. Students may not share code (verbally, in writing, or electronically) for a programming assignment. This includes all direct or indirect communications. Never show your code to anyone except the TA or the instructor. Never look at another student's code. If a student uses code from any source other than the textbook, he or she must mention the source in the program as comments (and also in any assigned report). Both sharing code with another student and failing to mention the origin of code taken from another source will constitute plagiarism. **Any** student involved in plagiarism will be reported.

Email

Every student will be required to use his/her official email address that is *student_webid*@go.olemiss.edu. All email communications will be made using this address. It is not uncommon for additional instructions or guidance to be sent by email, so check your email often. Students will be responsible for any instructions sent by email more than 24 hours old. The instructor checks email at least every 24 hours (and often more frequently than that) so email is the best way to contact the instructor.

Student Disability Services

It is University policy to provide, on a flexible and individual basis, reasonable accommodations to students who have disabilities that may affect their ability to participate in course activities or meet course requirements. Students with disabilities are encouraged to contact their instructors to discuss their individual needs for accommodations.

Students should contact the Office of Student Disability Services (http://sds.olemiss.edu/) in 234 Martindale to inquire about the services available and how to request them. Students must submit an Instructor Notification of Classroom Accommodation form to each instructor before direct classroom accommodations will be provided. This must be done on a timely basis, at the beginning of the semester or at least one week before needed, so that appropriate accommodations can be arranged.

Emergency Information

http://emergency.olemiss.edu provides information about campus-related emergencies due to weather or other circumstances. Know what you will do in the event of an emergency. Read RebAlert texts and emails, and respond accordingly. RebAlerts allow the university to communicate essential information to the campus community when a disaster occurs.

Subject to Change

All the dates, descriptions, locations, and other information on this syllabus are subject to change. Updates will be posted on the course website

CSE 2383: Data Structures and Algorithm Analysis - Spring 2023

General Information

This is an evolving document and will almost certainly be updated as the semester progresses. Please keep an updated view of the online version of this document.

Meeting

When: Tuesday, Thursday - 9:30AM to 10:45AM
Where: Butler Hall, Room 100

Description

Three hours lecture. Non-linear data structures and their associated algorithms. Trees, graphs, hash tables, relational data model, file organization. Advanced software design and development.
Significant programming is required

Prerequisites

A grade of C or higher is required in the following classes:

- CSE 1384 - Introduction to Computer Programming
- MA 1713 - Calculus I

Textbook

No text is required for this course.

Instructional Staff

Instructor

J. Adam Jones, PhD
E-mail: jaj33@msstate.edu
Office: Butler Hall - Room 323
Office Hours:
Thursdays, 11AM to 12PM
Others by appointment
(If you can't make it to my regular office hours, don't hesitate to ask for an appointment at a convenient time!)

SPECIAL ANNOUCEMENT: I'll be traveling for a conference from February 21 to 26. Please work closely with the class TAs/helpers during this period.

OFFICE HOURS POLICY:
Office Hours are first-come-first-serve and visits are limited to 15 minutes. Please keep in mind that office hours are for all my students. This includes students from classes, research projects, honors projects, teaching/research assistants, and laboratories.

Class Helpers

Ander Tally, ajt432@msstate.edu, Office Hours
Derek Willis, dpw9@msstate.edu (← ***Expert! Ask him your tough questions!***), Office Hours

Topics

- Introduction to the course (1 hr)
- C++ Refresher (4 hr)
- Memory & Computation (2 hr)
- Algorithm Analysis Lists (2 hr)
- Stacks (2 hr)
- Queues (2 hr)
- Tree Structures Binary Trees (1 hr)
- Tree Traversal Algorithms (2 hr)
- Search Trees (3 hr)
- Balanced Trees (3 hr)
- Heaps (3 hr)
- Graphs Definitions (1 hr)
- Representations (2 hr)
- Minimum Spanning Tree Algorithms (3 hr)
- Shortest Path Algorithms (3 hr)
- Hash Tables (3 hr)
- Tests (3 hr) - is this really a *topic* though? You decide.

Information for Students

- You are required to have access to a computer that connects to the internet.
- The course materials will be accessed through Canvas.
- You are required to check your MSU email account regularly. This is considered an official means of communication by MSU for distance education students.
- Assignment submissions will utilize Canvas unless otherwise specified by the instructor.
- Students should direct correspondence to the instructor directly related to the class via his official MSU email address.
- Students can correspond with each other via the general course discussion board. Please note that collaboration on individual projects is not acceptable unless otherwise indicated by the instructor.
- Students should expect to log in to Canvas no less than 3 times per week to access course information, lectures, and updates.

Grading

Numerics

Assessments (percentage of final grade):

```
Exams (60%)
Quizzes/Class Activities (10%)
Programming Challenges (25%)
C++ Refresher Challenge (5%)
```

Grading Scale:

```
(FG = final grade)
A - FG >=90%
B - 80% <= FG < 90%
C - 70% <= FG < 80%
D - 60% <= FG < 70%
F - FG < 60%
```

Assessments

Various assessments will be given throughout this course to measure each student's progress on learning course material. The type of assessments given in this course will include the following:

- Exams
- Quizzes/Class Activities
- Programming Challenges

All assessments in this course (including exams, quizzes, programming challenges, etc.) must be completed individually unless otherwise stated by the instructor. The use of notes, lecture slides, textbooks, and online materials on quizzes and exams is prohibited unless given permission by the instructor. If the Instructor or Teaching Assistant suspects any form of academic misconduct on any assessment given in this course, the student will be referred to the Honor Code Office.

Exams

3 tests will be given in this course. A comprehensive final exam will also be offered. If you choose to take the final exam, your final exam score will replace your lowest test grade, provided that any one of your test scores is lower than your final. No student is required to take the final exam. Henceforth, the final exam is optional.

Students will have a set duration of time to take the exam. No makeup exams will be offered after the exam has been given UNLESS the absence is excused. Any students that need an extension must provide documentation of the absence, which must meet University Guidelines to be accepted. Students must contact the instructor before the due date of the exam to schedule a makeup.

Quizzes

Quizzes will be given throughout the semester. Quizzes can cover material from any previous lecture. Quizzes will typically cover material from the most recent lecture but may have questions from earlier lectures. Quizzes will be proctored like exams but will be given a shorter time limit. No make-up quizzes will be given, but if you miss 3 days or less of class, you will be given a replacement quiz grade. The replacement grade will be the average of all quiz grades inclusive of your lowest grade.

If you have disability accommodations for quizzes you would like to use, contact the instructor either at the beginning of the semester or as soon as you are given accommodations to discuss arrangements. *The instructor must receive official notification of accommodations from the Disability Resource Center (DRC).*

Weekly Activities

Weekly activities will be given throughout the semester to support your learning and understanding of the material. These could include discussion on the current course topic, code analysis exercises, code readability improvement, and more. These will be graded at the instructor's discretion. Students taking this course for honors credit are required to participate in discussions via Canvas forums. For honors students, these forum discussions will constitute 25% of the Class Activity portion of their final grade.

Programming Challenges

Programming challenges will be given as part of this course. In programming challenges, students will be given a problem and produce a program solution that applies techniques learned in this course to solve the problem. The purpose of these challenges is to engage and develop the student's critical thinking and analyzing skills, which is important for success in an Engineering career. Students will use the C++ programming language to complete each challenges.

Course Policies

Attendance

This section is a face-to-face instructional class. Per Academic Operating Policy 12.09, students are expected to attend all class meetings in person. Should a student expect a university-excused absence from a class, the student should contact the course instructor of record to inform them of the absence and the reason for it. Attendance is required, but you are allowed to miss up to three days without penalty. If you miss three or less days, you will be allowed a replacement quiz grade. See the Quizzes section for more details.

Computer & Software

All students in the Bagley College of Engineering are expected to have a laptop that meets the requirements described at the following link: https://www.bagley.msstate.edu/computer/ NOTE: Students that use a Mac book should NOT expect support/troubleshooting from any instructor/TA and Bagley IT. Most software used in the Computer Science/Software Engineering curriculum is only compatible with Windows. If students choose to use a Mac, there is Virtual Machine software available online such as Oracle Virtual Box (free, but buggy) and Parallels(recommended). Students are expected to write programs using the C++ language. The recommended compiler/environment for C++ is Microsoft Visual Studio. The link for Visual Studio can be found at https://visualstudio.microsoft.com/downloads/. The Community edition is free for students. If you do not have access to a Windows machine or your system does not support Visual studio, you can use alternative compilers, but all code will be graded in Visual studio. Code that does not compile with the C++ standard will NOT be acceptable.

Alternatives to Visual Studio:

- Online GDB: https://www.onlinegdb.com/online_c++_compiler
  Works on all systems, as it is a webapp.
- Code::Blocks: http://www.codeblocks.org/
  Works on Windows and Linux/Mac OS.
- X Code: https://developer.apple.com/xcode/
  Mac OS only. Available through the Apple Store.
- g++ via WSL or Cygwin with a reasonable text editor (emacs, vi, notepad++, etc.)

Note: The alternative compilers above have native support for code accepted by the C language. Using these compilers will compile valid C code that is not acceptable by C++ standard. Be cautious of using these alternatives and ensure that your code is fully accepted by C++ standards.

Program Grading

Unless otherwise specified by the instructor, any Programming Challenge that requires you to write code MUST be completed in the C++ language. Any Challenge not completed in C++ will automatically be given a grade of zero. All Programming Challenges are completable and must be fully functional at the time of submission. Programming Challenge have a point value of 100 points and are graded based on two criteria: “completion points” - time taken to correctly complete the challenge (90 points) and “submission points” - proper submission of code (10 points). There is a submission window of 6 weekdays for each Programming Challenge (and weekends don’t count as being late!). Here is the breakdown of points for each day of the submission window:

- Day 1: 90 points
- Day 2: 80 points
- Day 3: 70 points
- Day 4: 60 points
- Day 5: 50 points
- Day 6: 40 points

Also, it is expected that students submit code that is syntactically correct. Code that contains syntactical issues will be penalized at the instructor's/Teaching Assistant's discretion, up to a grade of 0 for any points associated with completing code. **It is your responsibility to test/run your code BEFORE submitting!**

Submissions

All Submissions must be composed of the correct file formats. These formats are listed as followed:

- Code: .cpp, .h, .hpp
- Everything else: .pdf or .txt

Any different file formats will generally NOT be accepted.

Late Policy

Every programming challenge and activity MUST be turned in within the submission window. Programming challenges must be completed within the assigned submission window to receive “completion points”. “Submission points” (up to 10 points) can be awarded for programming challenges submitted at any point prior to the last day of class.

Any extensions on assignments due to illness or planned travel/events MUST be discussed with the instructor in a reasonable time frame, and any such extension will be granted at the discretion of the instructor.

Homework, Quizzes, and Tests are strictly due on the day they are due. There are no late opportunities for submissions after the due date of these items.

Instructional Staff Availability

Throughout the semester, the instructor and teaching assistants for this course will be available by appointment. Students are expected to schedule an appointment using the Instructors/TAs by email. Online appointments will be handled using WebEx software, where access is provided free of charge to Faculty/Staff and students. Students are expected to have this software set up and ready no later than the first week of class.

University Policies

The Mississippi State University Syllabus contains all policies and procedures that are applicable to every course on campus and online. The policies in the University Syllabus describe the official policies of the University and will take precedence over those found elsewhere. It is the student's responsibility to read and be familiar with every policy. The University Syllabus may be accessed at any time on the Provost website under Faculty and Student Resources and at https://www.provost.msstate.edu/faculty-student-resources/university-syllabus

Research Accomplishments

Research Accomplishments

Contents

Highlights

- Total of $2,007,720 in funding across both MSU and UM. $1,435,427 has come to MSU.
- NSF CAREER proposal successfully funded ($624,999).
- NSF Convergence Accelerator proposal successfully funded ($750,000).
- Since Fall 2021, submitted 21 proposals with 6 funded (3 external, 3 internal).
- 28 peer reviewed papers (20 since tenure-track), 2 US patents, 1 US provisional patent application, 1 invited book chapter, and 12 abstracts. According to Google Scholar, these have resulted in 1042 citations and an h-index of 13.

Research Narrative

My core research focuses on applying neuroscience and ophthalmological approaches to virtual reality. At face value, this may appear to be an unusual fit for a traditional computer scientist. However, in this section, I present the narrative of how I planned my education, training, and research to uniquely support the pursuit of these research directions. I do not exaggerate when I say that this work has been decades in the planning.

While pursuing my graduate degrees in computer science, I explored virtual reality (VR) and augmented reality (AR). Specifically, I investigated whether the distance compression issue observed in VR also existed in AR. It became clear that addressing VR/AR usability challenges, particularly in spatial performance, required a multidisciplinary approach beyond computer science. This led me to delve into the psychology and physiology of vision, pursuing coursework in sensation, perception, cognitive modeling, neuroimaging, and visual attention. It was during this time that I familiarized myself with the basic scaffolding of perceptual psychology research: sensation drives perception, perception drives action. This framework has shaped my research philosophy and is a key inspiration for my ongoing research.

Before my tenure-track appointments, I completed two postdoctoral fellowships. The first, at the University of Southern California (USC), focused on studying the impact of VR headset designs on spatial performance. I had the privilege of working alongside VR display experts Mark Bolas (founder of Fakespace Labs) and Palmer Luckey (founder of Oculus, creator of the Oculus Rift) on creating low-cost VR prototypes. This experience equipped me with skills in rapid prototyping and still influences my approach to developing experimental apparatuses. I was also fortunate enough to fund my own research at USC via my first grant proposal, sponsored through the Office of Naval Research, entitled “Mapping the Field of View” ($500,000, 2 years). This was the point at which my work started branching from traditional computer science into ophthalmological and neuro-perceptual research. I saw this as an unexplored region where no one was presently going, thus providing an opportunity to conduct high-impact, novel research while establishing myself as an authority in this new field. During my second postdoctoral fellowship in Human-Centered Computing at Clemson University, I further developed my expertise in psychology and human factors by attending classes and working closely with faculty in the Department of Psychology. It was also while there that I developed, tested, and published an ophthalmic schematic eye model for use with VR/AR that could provide submillimeter registration between real and virtual environments. The state of the art at the time was 5mm, thus resulting in an order of magnitude improvement over existing methods. During this fellowship, I was extremely fortunate to work under Larry Hodges. Hodges, one of the founding fathers of VR research, is well known for his talents in running highly productive and innovative labs. This is a skillset that I believe is essential to a successful research career. Indeed, I turned down an opportunity to join the faculty of one of our peer institutions to work with and learn from Hodges. The lessons learned from working with him have equipped me with the confidence and skills to manage a highly productive, cross-disciplinary lab of vibrant, enthusiastic students.

My first tenure-track appointment was at the University of Mississippi (UM) in the Department of Computer and Information Science, including an appointment as Neuroscience Program Faculty. This is where I founded the High Fidelity Virtual Environments Lab (Hi5 Lab). My lab's goal is to improve virtual environments by expanding our understanding of how humans process, perceive, think, and act when performing spatial tasks. While at UM, I began to explore combining VR with EEG-based neuroimaging and collected very exciting preliminary data just prior to the university being moved to remote operation during the COVID-19 pandemic. This data was a key component of my recently funded NSF CAREER proposal. I also worked with the UM McLean Institute for Public Service and Community Engagement in acquiring $550,000 to establish the Virtual Reality Learning Center. This was a center dedicated to applying VR to a wide variety of educational and training endeavors. Despite these achievements, one of the most challenging aspects of my time at UM was growing a graduate student population. Unfortunately, we did not have a PhD program in Computer Science. This proved to be a rather significant obstacle when trying to recruit students external to the university. As a result, the majority of my research at the time was aided by talented undergraduate and master students who were already in the UM programs.

When the possibility of joining Mississippi State University's Department of Computer Science and Engineering (CSE) presented itself, my family and I were very excited for the opportunity. Upon receiving my formal offer to join the faculty in Summer 2021, I began to formulate a strategy, including a transition plan, to rapidly build the personnel and infrastructure needed to continue my research. I immediately began to identify and remotely meet with potential collaborators on campus. These included Edward Swan (CSE), Deborah Eakin (Psychology), Tonya Hayes (Communications/Theatre), Dan Gadke (College of Education), and Harish Chander (Kinesiology). Each of these people had knowledge, expertise, and resources within the university community that I saw as essential to my long-term research goals. I am now actively collaborating with 4 of these 5 departments. Furthermore, 3 of these departments are presently working with me on externally funded research. I believe these would not have been possible had I not begun building this network before physically coming to campus.

I joined the faculty of CSE in Fall 2021 and was generously granted two years credit toward tenure for work at my previous institution. Having credit toward tenure can potentially be a double-edged sword, providing both advantages and challenges. This, of course, saves you time on going up for review, but also reduces the amount of time you have to bring your research up to speed. I knew I needed to optimize my efforts. As such, I developed a transition plan to migrate my research to MSU in three phases: 1) Lab Building, 2) Research Support, and 3) Independent Research. Regarding "Lab Building", working in my lab involves a significant amount of institutional knowledge associated with our procedures, protocols, and methods. Training and managing student researchers in order to get them to the point necessary to be productive is incredibly time consuming. As such, I negotiated in my start-up package to bring one of my most productive students from UM with me to help get my lab off the ground. Having another person who was already familiar with how my lab works and my expected level of quality sped up my lab's development by at least a year. This was perhaps one of the wisest decisions that I have made. Additionally, since I had established connections with people doing related research prior to coming to campus, I was able to almost immediately get students working on research projects that resulted in co-authorship on publications.

The "Research Support" phase of my transition plan focused on serving in software development and research support roles on projects led by collaborators. This would serve to solidify partnerships, establish value as a colleague, and secure co-PI-ship on external funding that I may not have otherwise been competitive for this early in my career at MSU. It is important, however, to note that my long-term research goals cannot be met by operating in this phase alone. As such, part of my transition plan was to serve in a support capacity until I had secured at least one large, externally funded grant as co-PI, specifically through NSF. Anecdotal accounts from numerous mentors indicated that getting an NSF proposal funded as lead PI would be significantly easier if I had first served as a co-PI. Fortunately, a proposal led by Kasee Stratton-Gadke at the T.K. Martin Center was funded through the highly exclusive NSF Convergence Accelerator program ($750,000, 1 year). This project aimed to assist persons with disabilities in learning to drive using off-the-shelf VR equipment. I served as co-PI and technical lead on this project. Convergence Accelerator projects are extremely fast-paced and provide funding for 1 year with the option for an additional 1 year no-cost extension. These projects are intended to rapidly develop emerging research into product prototypes with the potential for large societal impact, thus providing a large amount of funding for a somewhat short window of time. Over the course of the project, I have supervised 15 total students from Computer Science & Engineering, Computer Engineering, and Electrical & Computer Engineering. We have developed several software and hardware prototypes that we have now presented at the most impactful VR conference (IEEE Virtual Reality) and the most prestigious driver rehabilitation conference (ADED Association for Driver Rehabilitation Specialists). This research also resulted in a US provisional patent application for the technologies we have developed. We are presently investigating ways to make this technology available to the public either through government-sponsored programs or commercialization.

We are now solidly in the "Independent Research" phase of my transition plan. After having served as co-PI on the aforementioned NSF project, I submitted an NSF CAREER proposal entitled "A Neuro-Ophthalmic Approach to Virtual Reality Research" which was successfully funded ($624,999, 5 years). This proposal laid out a very detailed and focused plan for establishing myself and my lab as authorities in applying neuroscience and ophthalmology methods to VR research. Prior to submitting this proposal, my students and I conducted several pilot studies producing preliminary results for the proposed research, giving us both evidence of viability and a head-start on the work. I have also acquired seed funding from the Office of Research and Economic Development for undergraduate research projects to investigate the viability of combining VR and AI to study spatial behaviors from both the computational and biological perspectives. This brings elements of computational neuroscience into my current arsenal of psychophysical and biometric methods. The preliminary results are promising, and I plan on developing this further for a potential proposal submission to a venue such as NSF CISE.

I have now effectively completed my transition plan and am focusing on direct contributions to my long-term research goals. The research areas for the next five years are centered around the following themes. Though these topics are delineated by years, I anticipate significant cross-over between years as I prepare for new phases of my research. It is also important to note that I do plan on continuing to support colleagues in their research efforts as well, but not as a primary goal of my lab. Further details about these areas as well as current research supporting them are provided in the *Research Program* section of this chapter.

Year 2025 - Ophthalmic Schematic Eye Models for VR: I plan on incorporating VR rendering techniques and ophthalmic schematic eye models, used in clinical practice to closely approximate the anatomy and optics of the human eye, to improve the accuracy of virtual environment registration. This addresses a critical issue that hinders the real-world applicability of VR and AR. Current research suggests that achieving registration below 5mm may be challenging with existing pupillary distance (PD) based eye models. I suggest that this limitation is due to inadequate ocular modeling. While using pupils as a convenient approximation for rendering realistic VR is appealing, it becomes problematic due to the eye's complex nature. The eye has separate optical, rotational, and physical centers that are not coincident. Furthermore, the eye's nodal point, which serves as the optical center of projection, moves along multiple axes during eye rotation. It is important to note that the nodal point is an internal feature of the eye and cannot be directly observed. This makes using an ophthalmic modeling approach a promising way of approximating its position.

Year 2026 - Misalignment of Display Optics and the Eye: My preliminary studies suggest that achieving a high degree of geometric fidelity is possible in VR when accurately modeling the eye, but this scenario is not realistic for most immersive environments. In practice, misalignment between the observer's eyes and headset optics is common, especially in headsets with limited or no PD adjustment options. This misalignment can lead to image distortions due to disparities between the display optics and ocular optics. As such, we must ask how much error is introduced to the eye's retinal projection when there is a misalignment between these two optical systems? Preliminary data shows image distortions of up to 3° for a reasonable range of eye separations (64±5mm) when the observer's PD did not match that of the headset. This misalignment can potentially cause disorientation, misperceived motion, or simulator sickness as observers shift their gaze or move within a virtual scene.

Year 2027 & 2028 - Localizing Neural Activity While in VR: At the heart of VR perception research lies a pivotal question: Does the human brain process VR in the same way as it does the real, physical world? If not, what factors contribute to this disparity? These are questions that can only be answered from the neural perspective. Studying neural activity patterns during spatial tasks can provide valuable insights into the differences observed between virtual and real environments. Other research, utilizing PET imaging, revealed distinct activation patterns in the dorsal and ventral visual streams when pointing to objects within arm's reach (near) or beyond (far). I propose investigating this specific phenomenon in order to identify similarities and differences in patterns of neural activity for participants viewing real or virtual environments. Furthermore, factors like field of view (FOV) size and peripheral visual stimulation are known to significantly impact spatial performance in VR, but their underlying mechanisms remain unclear. By using the difference between the dorsal and ventral streams as a measurement tool, which naturally distinguishes between near and far spaces, it may be possible to investigate whether visual factors like FOV are affecting perceived spatial boundaries. This involves determining the depths at which the transition between streams occurs. Understanding these relationships can shed light on the neural basis of spatial perception in virtual environments and the real world.

Year 2029 - Low-Persistence & Motion Quality: Low-persistence, a method to reduce motion blur in VR, is shockingly understudied. It involves displaying images on a headset's screen for a very brief period of time before it is replaced by a black screen, leading to more darkness than illumination. Although every modern headset uses this technique, its benefits to users have not been formally quantified in the literature. Existing research on low-persistence is scarce, and explanations for its benefits are conflicting. Some attribute the improvement to reduced pixel ghosting in hardware, while I find a perceptually motivated explanation more plausible. I believe it is likely that this explanation relates to persistence of vision, a phenomenon where light stimulation continues to be perceived in a location after its source has moved, causing perceived blurring. This is analogous to the photographic trend of "light painting" with long-exposure photography. Breaking up image presentation reduces this phenomenon, resulting in a series of distinct images being perceived as opposed to a continuous blur. Despite limited literature on the topic, low-persistence is widely used and often considered desirable in VR headsets. Since most headsets implement it, researchers should better understand why it works. Even researchers who do not specifically study motion blur should be mindful of low-persistence, as it is inherently used in their work by default.

Research Funding

Establishing funding for research activities has been a priority for me both at MSU and UM. Collectively, I have contributed to bringing $2,007,720 in funding across both institutions. Since joining MSU, I have been actively searching for groups and individuals on campus who have research directions that are complementary to my own. I have found colleagues in Psychology, Kinesiology, CAVS, T.K. Martin Center, Agriculture and Biological Engineering, Counseling and Educational Psychology, and Electrical Engineering with whom I have been very active in pursuing projects, papers, and proposals. These collaborations have been very successful, resulting in the submission of 16 proposals (13 external, 3 internal) in under 18 months. I was very fortunate to have a total of 3 proposals funded (1 external, 2 internal) for a total amount of funding of $1,435,427. The proposals selected for funding are:

1. **[External]** *NSF CAREER: A Neuro-Ophthalmic Approach to Virtual Reality Research*. National Science Foundation. $624,999. (May 2024 – April 2029).
2. **[External]** *NSF Convergence Accelerator Track H: Advancement of Driving Technology for Vocational Enablement*. National Science Foundation. $750,000. (December 2022 – November 2024).
3. **[External]** *MCCTR: Movement Disorders and Cognitive Impairments from SARS-CoV-2 Infection in Older Adults*. Mississippi Center for Clinical and Translational Research. $53,928. (December 2023 – November 2024).
4. **[Internal]** *ORED Undergraduate Research - Creating and Testing a Virtual Reality (VR) Tool Kit for Fall Prevention Training*. $2000. (Spring 2023 – Fall 2023).
5. **[Internal]** *ORED Undergraduate Research - AI2VR: Placing Artificial Intelligence into Virtual Reality to Evaluate Its Performance*. $1500. (Fall 2023 – Spring 2024).
6. **[Internal]** *Ottilie Schillig Teaching Projects: VXR - Virtual and Extended Reality Software Development: Providing Immersive Experiences in the Classroom using Virtual Reality*. $3000. (June 2022 – May 2023).

Prior to my time at MSU, I also received a combination of external and internal funding through both grants and donations. These totaled $572,202 in funds to the University of Mississippi. These include:

1. **[External]** *Virtual Reality Learning Center*, Mississippi Department of Human Services. $550,000. FY: 2019-2020. $34,000 of which came to CIS in the form of graduate support, undergraduate support, and faculty salary.
2. **[External]** *Intel Galileo University Donations*. Intel Higher Education. 5 Intel Galileo Generation 2 Development Boards and Seed Studio Grove Starter Kits. Estimated value: $1,500. Fall 2015.
3. **[Internal]** *Undergraduate Research in Data Science,* University of Mississippi Summer Undergraduate Research Groups Grant (SURGG), $20,450. Summer 2018.

Publications

Publication of scholarly work in a variety of venues and forms is an important part of my mission to disseminate the results of my research. Since first accepting a tenure-track appointment, I have published 20 peer-reviewed manuscripts, 2 US patents, 1 provisional patent application, 1 book chapter, and 12 abstracts.

Facilitated by successful collaborations and the recruitment of talented students, I have had 2 journal articles, 2 conference proceedings, and 5 abstracts published since joining MSU. These are given below:

1. **[Journal]** Joao Paulo Oliveira Marum, H. Conrad Cunningham, **J. Adam Jones**, Yi Liu. 2024. *Following the Writer's Path to the Dynamically Coalescing Reactive Chains Design Pattern.* Algorithms – Special Issue on Algorithms for Virtual and Augmented Environments. 17(2), p.56.
2. **[Journal]** Hunter Derby, Nathan O. Connor, Jacob M. Hull, Faith Hagan, Sally Barfield, Timothy Stewart, **J. Adam Jones**, Adam C. Knight, Harish Chander. 2023. *Effects of Acute Exposure to Virtually Generated Slip Hazards during Overground Walking.* Applied Sciences, 13(23), p.12848
3. **[Journal]** Hannah R. Freeman, Harish Chander, Sachini N. K. Kodithuwakku Arachchige, Alana J. Turner, **J. Adam Jones**, Zhujun Pan, Christopher Hudson, Adam C. Knight. 2023. *Postural Control Behavior in a Virtual Moving Room Paradigm*. Biomechanics, 3(4), pp.539-551.
4. **[Journal]** Nathan O. Conner, Hannah R. Freeman, **J. Adam Jones**, Tony Luczak, Daniel Carruth, Adam C. Knight, Harish Chander. *Virtual Reality Induced Symptoms and Effects: Concerns, Causes, Assessment & Mitigation*. Virtual Worlds, 17 pages.
5. **[Journal]** Harish Chander, Hannah R Freeman, Christopher M Hill, Christopher Hudson, Sachini N. K. Kodithuwakku Arachchige, Alana J Turner, **J. Adam Jones**, Adam C Knight. 2022. *The Walls are Closing in: Postural Responses to a Virtual Reality Claustrophobic Simulation*. Clinical and Translational Neuroscience, 15 pages.
6. **[Conference]** Anthony Luczak. Charles Freeman, Reuben Burch, David Saucier, Harish Chander, John Barlow, John Ball, Patrick Nelsen, E. Parker, C. Middleton, Leslie Strawderman, J. Mohammadi-Aragh, Zachery Gillen, B. Smith, **J. Adam Jones**, Martin Duclos, Michael Taquino, Steven Grice. *Athlete Engineering BaseLine Ecosystem: innovative technologies to enhance human performance.* TechConnect World Innovation Conference, Washington DC, page 136-140.
7. **[Conference]** Anthony Luczak, **J. Adam Jones**, Reuben Burch, Jonathan Barlow, Patrick Nelsen, Steven M. Grice, Michael Taquino, Martin Duclos, Caleb Morgan. 2022. *Advancing tangible augmented game objects for use in a golf swing, self-service training environment: Report of Work-in-Progress with a Multidisciplinary Emphasis.* IEEE Virtual Reality Workshop on Augmented Reality Enabling Superhuman Sports + Serious Games (IEEE VR – ARES), Christchurch, New Zealand, pages: 136 - 140.
8. **[Abstract]** Tony Luczak, Kait Jackson, Long Tian, **J. Adam Jones**, Reuben Burch, Jonathan Barlow, Patrick Nelsen, Caleb Morgan, Harish Chander, Martin Duclos, Michael Taquino, and

Steven M. Grice. *The Challenges of Coaching Using 2D Golf Swing Video Data Compared to the Challenges of Building a 3D Technology Based Coach*. World Scientific Congress of Golf 2022. 1 page.

9. **[Abstract]** Zachery Garris, Maggie Pettus, Daniel Molsbarger, Harish Chander, **J. Adam Jones**. 2022. *EEG Electrode Localization Using Off-the-shelf Virtual Reality Systems*. Proceedings of the Mississippi Academy of Sciences Annual Meeting.
10. **[Abstract]** Dylan Devenny, Harish Chander, **J. Adam Jones**. 2022. *The Influence of Optical Flow on the Perception of Impossible Spaces in Virtual Reality*. Proceedings of the Mississippi Academy of Sciences Annual Meeting.
11. **[Abstract]** Jonathan Hopper, David M. Krum, **J. Adam Jones**. 2022. *Perception of Target Eccentricity and Depth in Virtual Reality*. Proceedings of the Mississippi Academy of Sciences Annual Meeting.
12. **[Abstract] J. Adam Jones**. 2023. *The Pre-History (and Future) of Virtual Reality*. Proceedings of the Mississippi Academy of Science, 1 page.

At my previous institution, I also had a successful record of publication in high-quality venues. My publications while at UM consisted of 13 peer-reviewed manuscripts, 2 US patents, 1 invited book chapter, and 7 abstracts. These are given below:

1. **[Journal] J. Adam Jones**, Jonathan E. Hopper, Mark T. Bolas, David M. Krum. 2019. *Orientation Perception in Real and Virtual Environment*. IEEE Transactions on Visualization and Computer Graphics (TVCG), pages 2050-2060.
2. **[Journal] J. Adam Jones**, David M. Krum, Mark T. Bolas. 2017. *Vertical Field of View Extension and Walking Characteristics in Head-worn Virtual Environments*. ACM Transactions on Applied Perception (ACM TAP). 14, 2, pages 9:1-9:17.
3. **[Conference] J. Adam Jones**, Darlene Edewaard, Richard A. Tyrrell, Larry F. Hodges. 2016. *A Schematic Eye for Virtual Environments*. Proceedings of the IEEE Symposium of 3D User Interfaces, Greeneville, South Carolina, pages 221-230.
4. **[Conference]** Ethan Luckett, Tykeyah Key, Nathan Newsome, **J. Adam Jones**. 2019. *Metrics for the Evaluation of Tracking Systems for Virtual Environments*. IEEE Virtual Reality Workshop on Novel Input Devices and Interaction Techniques (NIDIT), Osaka, Japan, pages 1711-1716.
5. **[Conference]** João Paulo Oliveira Marum, **J. Adam Jones**, H. Conrad Cunningham. 2019. *Towards a Reactive Game Engine*. IEEE Southeast Conference, Huntsville, AL.
6. **[Conference]** William Panlener, David M. Krum, **J. Adam Jones.** 2019. *Effects of Horizontal Field of View Extension on Spatial Judgments in Virtual Reality*. IEEE Southeast Conference, Huntsville, AL.
7. **[Conference] J. Adam Jones**. 2019. *Optical and Neural Properties of Vision as Applied to Virtual Reality*. IEEE Virtual Reality Workshop on Neuroscience & Virtuality (NeuroVirt), Osaka, Japan, pages 1667-1670.
8. **[Conference]** Jonathan E. Hopper, Hunter Finney, **J. Adam Jones**. 2019. *Field of View and Forward Motion Discrimination in Virtual Reality*. IEEE Virtual Reality Workshop on Neuroscience & Virtuality (NeuroVirt), Osaka, Japan, pages 1663-1666.
9. **[Conference]** Benjamin Creel, Caitlin Rinz-Jones, Colin Jackson, **J. Adam Jones.** 2020. *Bacterial Load of Virtual Reality Headsets*. ACM Symposium on Virtual Reality Software and Technology (VRST), Ottawa, Canada.

10. **[Conference]** Hunter Finney, **J. Adam Jones.** 2020. *Asymmetric Effects of the Ebbinghaus Illusion on Depth Judgments*. IEEE Conference on Virtual Reality and 3D User Interfaces (IEEE VR), Atlanta, GA, pages 573 - 578.
11. **[Conference]** João Paulo Oliveira Marum, H. Conrad Cunningham, **J. Adam Jones**. 2020. *Unified Library for Dependency Graph Reactivity on Web and Desktop User Interfaces*. ACM Southeast Conference (ACMSE), Tampa, FL, USA, pages: 26 - 33.
12. **[Conference]** Collin Roth, Ethan Luckett, **J. Adam Jones**. 2020. *Latency Detection and Illusion in a Head-Worn Virtual Environment*. IEEE Virtual Reality Workshop on Perceptual and Cognitive Issues in Augmented Reality (PERCAR), Atlanta, GA, pages: 215 - 218.
13. **[Conference]** João Paulo Oliveira Marum, H. Conrad Cunningham, **J. Adam Jones**. 2020. *Dependency Graph-based Reactivity for Virtual Environments*. IEEE Virtual Reality Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS), Atlanta, GA, pages: 246 - 253.
14. **[Book Chapter, Invited] J. Adam Jones**, 2019. *A Tinkerer's Perspective on VR Displays*. "VR Developer Gems", AK Peters/CRC Press, ed. William Sherman, May 2019.
15. **[Patent]** US10416453B2. *Control of Ambient and Stray Lighting in a Head Mounted Display*. Inventors: Mark Bolas, **J. Adam Jones**, David M. Krum. Applicant: University of Southern California. Granted: September 17, 2019.
16. **[Patent]** US009645395B2. *Dynamic Field of View Throttling as a Means of Improving User Experience in Head Mounted Virtual Environments*. Inventors: Mark Bolas, **J. Adam Jones**, Ian McDowall, Evan Suma. Applicant: University of Southern California. Granted: May, 9, 2017.
17. **[Abstract]** Benjamin Creel, **J. Adam Jones**, Collin R. Jackson. 2019. *Bacterial Load in Virtual Reality Headsets*. Proceedings of the American Society of Microbiology, South Central Branch Meeting. Oxford, MS, page 36.
18. **[Abstract] J. Adam Jones**, Ethan Luckett, Tykeyah Key, Nathan Newsome. 2019. *Latency Measurement in Head-Mounted Virtual Environments*. Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces (IEEE VR), Osaka, Japan, pages 1000-1001.
19. **[Abstract] J. Adam Jones**, Thai Phan. 2019. *A Comparison of Gait Measurement Methods*. Proceedings of the IEEE Southeast Conference, Huntsville, AL.
20. **[Abstract]** Khoa Tran, Yaoxin Zhang, Yafei Jia, **J. Adam Jones**. 2019. *A Prototype Visualization Tool for Hurricane Flood Simulations in Virtual Reality*. Proceedings of the IEEE Southeast Conference, Huntsville, AL.
21. **[Abstract]** McKennon McMillian, Hunter Finney, Jonathan Hopper, **J. Adam Jones**. 2018. *The Depth Light*. Proceedings of the IEEE Conference on Virtual Reality and 3D User Interfaces (IEEE VR), Reutlingen, Germany, pages 834-835.
22. **[Abstract]** Joao Paulo Marum, **J. Adam Jones**, Conrad H. Cunningham. 2016. *Functional Reactive Augmented Reality: Proof of Concept Using an Extended Augmented Desktop with Swipe Interaction*. Proceedings of the Eurographics Symposium on Virtual Environments, pages 13-14.
23. **[Abstract]** Alexander Gunter, Andrew Robb, **J. Adam Jones**. 2016. *Real-Time Marker Tracking with Microsoft Kinect*. Proceedings of the Eurographics Symposium on Virtual Environments, pages 15-16.

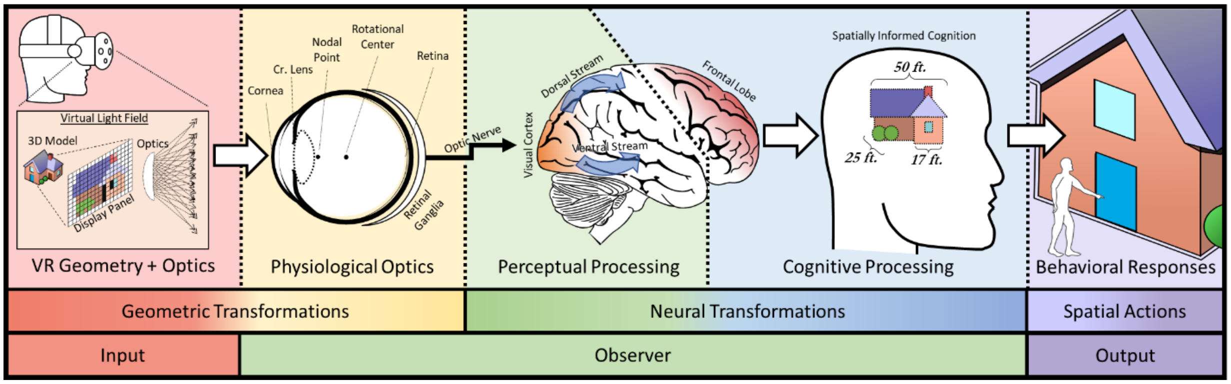


Figure R1 - The progression of visuospatial information in the pipeline approach used to guide my on-going research. This pipeline can have multiple levels of granularity, expanding with our understanding. The top and middle rows are the perspectives taken in the current proposal. The bottom row is the lowest granularity and common perspective for much of existing VR perception research.

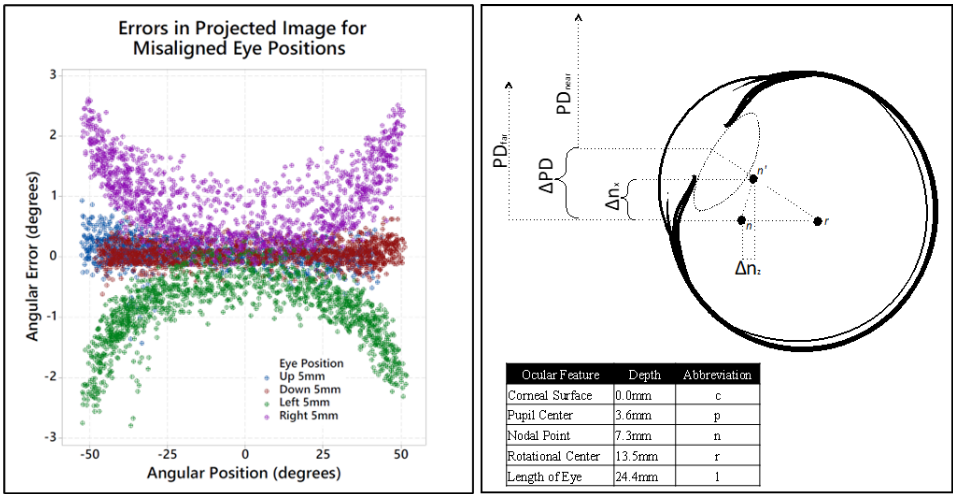
Research Program

The general focus of my research is summarized well in my lab's name: *High Fidelity Virtual Environments Laboratory*. My goal is to understand and improve the fidelity and usability of virtual environments. However, the definition of high fidelity depends on the context under which a virtual environment will be used. What constitutes high fidelity for a user playing an immersive video game may not constitute high fidelity for a surgeon using VR to assist in a delicate operation. This brings about an interesting problem: *how do we determine what features and designs will constitute a high fidelity virtual environment for a given scenario*? This is clearly an engineering question. However, VR is an intimate technology that, in most cases, physically interfaces with the user's body. Thus, the answer to this question goes beyond traditional engineering, incorporating elements from computer science, neuroscience, ophthalmology, and psychology. In an attempt to better understand the user while also engineering the needed level of fidelity, I suggest that we need to systematically decompose and analyze the human-technology relationship in terms of a processing pipeline where visuospatial sensory information undergoes three distinct stages of transformations before producing resultant actions in the user. These stages encompass the *geometric transformations, neuro-perceptual transformations, neuro-cognitive transformation, and behavioral actions* for a given virtual environment. Each of these stages offers an opportunity to engineer an environment's spatial fidelity in unique ways, see Figure R1.

Though the engineering benefits of such an approach are clear, perhaps an even more important aim of this approach is to elucidate the visuospatial mechanisms internal to the human observer (e.g. neural processes and representations) that are relying upon these features. This approach can also be used to systematically study virtual environments and spatial behaviors in a more general sense. Substantial previous work, including my own, has attempted to address visuospatial behavior in virtual environments. Despite this work, relatively incremental progress has been made in understanding the breadth of factors affecting virtual environments. One of my goals is to replace the historically ad-hoc approach to studying virtual environments with a coherent scaffolding upon which we can build future hypotheses, experiments, and theories.

Research Areas

If we approach studying visuospatial processing and behavior as an information processing problem, we can generalize it as consisting of three distinct stages of transformation: geometric, perceptual, and cognitive. This provides a convenient way to discuss the specific areas of research that I am currently pursuing. Below, I present an explanation of each of these areas, as well as a sample of my prior work, followed by on-going research efforts.



Ocular Feature	Depth	Abbreviation
Corneal Surface	0.0mm	c
Pupil Center	3.6mm	p
Nodal Point	7.3mm	n
Rotational Center	13.5mm	r
Length of Eye	24.4mm	l

Figure R2 - (left) Errors introduced by misalignment of an observer's eye with the lens of a VR headset. (right) A schematic eye model designed to predict the position of the eye's optical center.

<u>Geometric Transformations</u>

The vast majority of VR research has focused on accurately generating an *environmental signal* in terms of rendering, resolution, magnification, aspect, focal depth, lighting, and the like. Though most of this research acknowledges that people have differing visual acuity, essentially no research has tried to examine whether the individual differences in the eye's *physiology* play an important role in the spatial fidelity of VR. This has led to a gap in the current research that necessitates an in-depth evaluation of the interplay between environmental and physiological geometry.

We can conceptualize the factors affecting geometric fidelity as the features that are encoded in the <u>physical properties of both the environment and the system detecting the environment</u>. For instance, the geometric features of the space where you are currently sitting are being generated by the physical interaction of light and surfaces producing a light field that contains an echo-like representation of your surroundings. In the context of an eye seeing a scene, these features include the environment's light field, classical optic transformations applied to the light field, physiological optic transformations performed by the pre-retinal portions of the eye, and the detection of light by the retina itself.

The system detecting the environment, the retina in our specific case, has intrinsic geometric properties that bound the attributes of the environment that can be detected. We can think of this detection system as having both *physical geometry* (size, location, receptor density, etc.) and *signal geometry* (dynamic detection ranges, response curves, sampling rates, etc.). The signal geometry pertains to the shape, range, and temporal properties of the signal being processed by the detecting system. In real-world settings, the signal geometry is generally a continuous gradient of visible properties. However, in most immersive headsets the signal geometry is somewhat more discrete. The most obvious example of this is image pixelation that leads to "screen door"-type effects. However, signal geometry also encompasses the focal geometry and temporal characteristics of the displayed images. For example, many headsets utilize a technique referred to as *low persistence* to decrease motion blur by manipulating the temporal geometry of the visual signal, see Figure R3.

Schematic Eye Model: One of my favorite experiments that I have conducted was to determine if modeling the physiological geometry of an observer's eye would produce a more accurate retinal projection and, consequently, better registration between real and virtual scenes [Jones et al. 2016]. This required us to move beyond common approaches to rendering virtual environments and investigate ophthalmic schematic eye models, see Figure R2. What I found was contrary to the casual, pupil-based approaches usually employed when modeling an observer's eyes in VR. The eye is vastly complicated, having distinct optical, rotational, and physical centers that are noncoincident. This becomes particularly problematic since the eye's nodal point, the optical center of projection, moves along multiple axes as the eye rotates. My hypothesis was that better estimation of the eye's optical center would improve the registration between real and virtual scenes. We refer to this as the "nodal model" since it is estimating the location of the eye's optical nodal point. There are many other possible factors that could adversely affect registration, such as head tracking errors, focal mismatches, changing gaze direction, and position of the headset on the observer's head. I aimed to remove these factors by constructing a prototype headset that rigidly attached the observer's head to a stationary stereoscopic display. The display was constructed from a repurposed robotic positioning system, a 3D-printed head restraint, ophthalmic trial frame, optical filters, and a 3DTV. We were able to reliably measure distances from the observer's eye at just under 0.09mm resolution. The observer's gaze direction was also fixed using pinhole filters, which allowed the stimulus to only be visible when the eyes were in a known position. I compared the nodal model against common pupillary distance-based methods of modeling the observer's eyes. Consistent with related work, we found that several of the commonly used methods failed to produce registration below the 5mm level. Modeling the eye's optical center produced the best registration with mean ***accuracy below 0.5mm, an order of magnitude improvement*** over the state of the art.

Optical Alignment: As we can see, a high degree of geometric fidelity is possible when accurately modeling the eye in the absence of confounding factors such as display optics. Unfortunately, this is not a situation that is realistic for most immersive environments. In practice, the observer's eyes are seldom exactly aligned with the optics of a headset, especially for headsets that offer limited or no ability to adjust factors such as interpupillary distance and eye relief. This can be especially problematic for modern consumer-grade headsets that rely on graphical corrections to compensate for optical distortions. In these cases, misalignment of the display optics and ocular optics results in distortions of the imagery seen through the headset. As such, we asked the question, *how much error is introduced to the retinal projection when there is a misalignment between these two optical systems*? To acquire these measurements, we needed a way to position and re-position a camera relative to a headset in a very precise and repeatable manner. Our solution was to use methods from precision manufacturing where it is common for measurement tolerances to be below 1/1000" [Jones et al. 2014]. We modified a manual milling machine to move an Oculus Rift DK1 headset very precisely within the horizontal and depth axes while moving a camera along the vertical axis. We illustrated that for a reasonable range of eye separations (64±5mm) there can be image distortions as high as 3° when the observer's PD did not match that of the headset, see Figure R2. This is concerning in the static case where the eye is looking forward, but, as our previously discussed study indicates, the eye's optical center moves as the eye rotates. This leaves the potential for the retinal projection to change in unexpected ways as observers shift their gaze around a virtual scene, which has the potential to lead to disorientation and motion sickness.

On-going Research: We are resuming *Optical Alignment* work during the Summer of 2023. A colleague from the University of Utah will be spending June visiting Mississippi State in order to assist in performing an updated version of this experiment. We will be examining a wider variety of VR headsets and optical configurations, as well as using an improved CNC-controlled positioning apparatus. This is a very synergistic project as the group from Utah is interested in how optical distortions in headsets relate to motion sickness but have not been able to develop a way of systematically measuring it in a way that would be comparable across headsets. Fortunately, in our prior work, we have developed highly precise methods for measuring optical distortions using a combination of computer vision and precision manufacturing equipment.

I have a PhD student, Zeenat Islam, who is currently investigating the differences in temporal signal geometry produced by VR headsets and whether this affects their usability. This is typically referred to as low-persistence. Our preliminary data has already shown that the temporal presentation of imagery in common headsets varies drastically, see Figure R3. This figure shows that there exist large differences in how headsets apply low-persistence when presenting graphics. This work is in its very early stages but has already proved to be a rich source of novel insights. It is clear that these patterns are engineered by the headset manufacturers. However, no work has been published on how low-persistence improves the viewing experience. Our aim is to isolate the specific physical, sensory, or perceptual mechanisms involved in low-persistence.

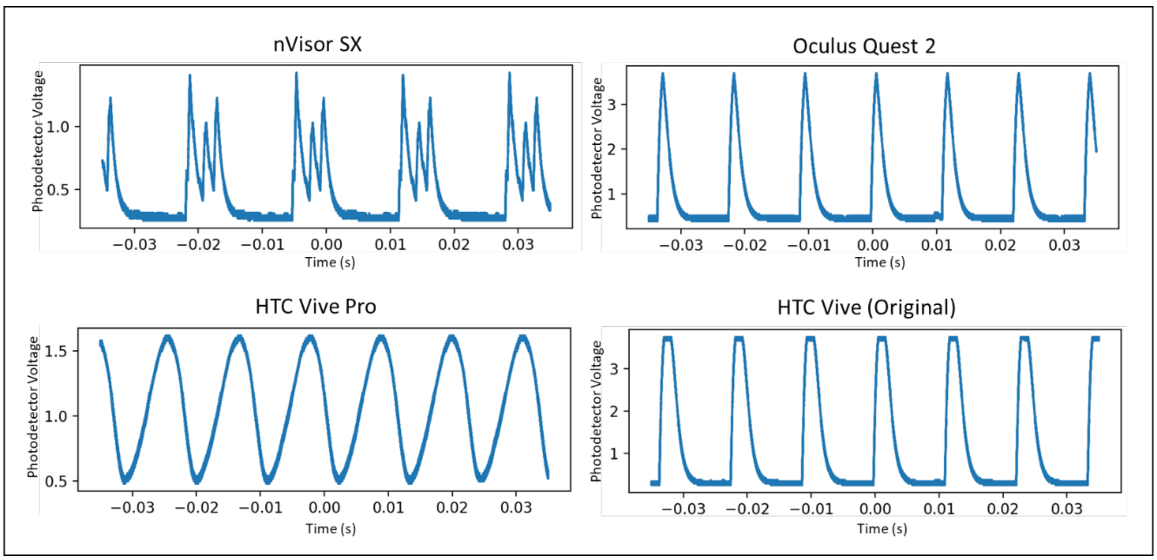


Figure R 3 - Low-persistence profile of a sample of commonly used VR headsets. This can be conceptualized as how long and at what brightness is a single frame of video displayed in the headset. These are preliminary data as measured by a simplistic photodetector circuit connected to an oscilloscope. Consequently, the vertical axis is not well calibrated between datasets and loosely correlates to the brightness of the displays screen.

<u>Neuro-Perceptual Transformations</u>

Much of the work involving measuring the fidelity of virtual environments focuses on so-called *perceptual issues*, most notably in terms of psychophysical measurement of distance perception. Psychophysics operates on the assumption that patterns in behavior infer patterns in perception. Accordingly, all psychophysical measures must *couple a percept with a physical, measurable action*. Though psychophysics is an extremely powerful tool for studying perception-action couplings, it is limited in what it can tell us about perception itself. Spatial perception research in VR has been exclusively psychophysical in nature. As such, it is not known if the distance *judgments* in these studies are, in fact,

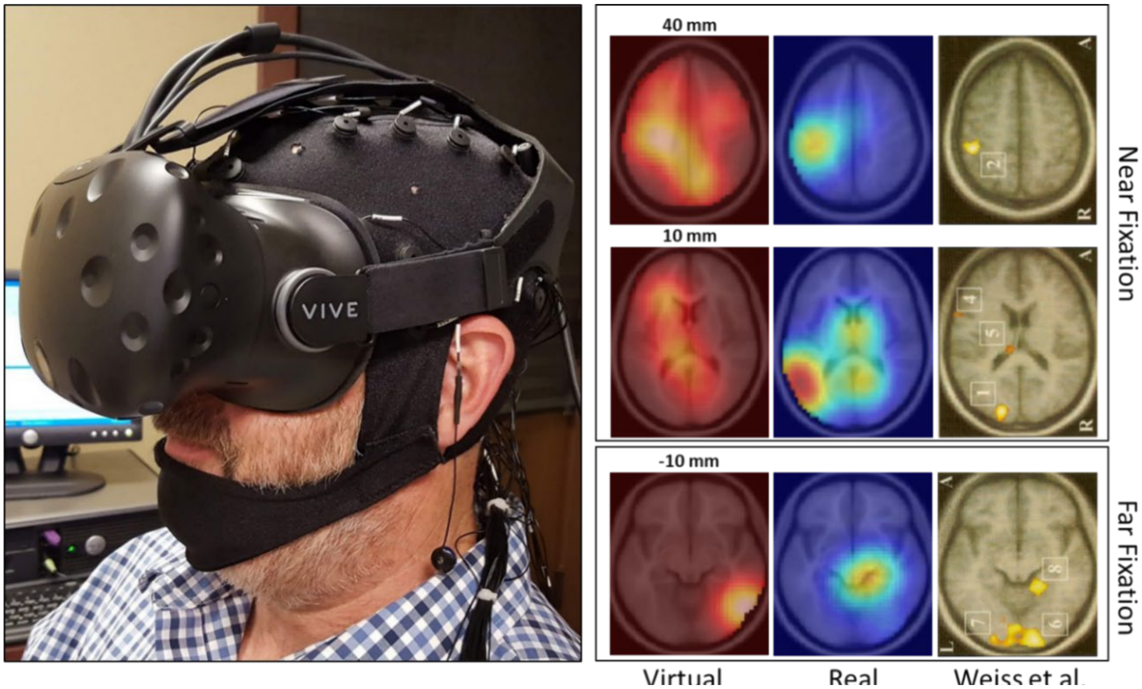


Figure R 4 - Left: EEG-enabled Virtual Reality Headset. Right: Single subject source localization using EEG during viewing of virtual targets (red shaded) and real targets (blue shaded) compared to expected results from real targets (yellow shaded) as detected using PET by Weiss et al. [2000].

reflective of distance *perception*. Bridging this gap is one of the biggest challenges faced by VR perception research. Ultimately, perception is a matter of neural processing and representation; thus, a percept is a strictly neural construct. For this reason, I have been incorporating techniques from neuroscience to measure and compare the neural correlates of spatial perception in real and virtual environments.

Neuroimaging in VR: In March 2020, I conducted a small study using electroencephalography (EEG) to test the viability of this approach and found that we could reproduce established neuroimaging results using a method compatible with off-the-shelf VR headsets. Unfortunately, only a tantalizingly small amount of data was collected before the university closed due to the COVID-19 pandemic. This experiment adapted the procedures from work conducted by Weiss et al. [2000] who demonstrated that there are detectable differences in neural activation along the dorsal and ventral visual streams when observers pointed to objects within arm's reach (near) or beyond (far). They found that dorsal activity was greater when acting in near space while ventral activity was greater when acting in far space. We examined this phenomenon in both VR and the real world. Observing differences in activation imply that the disparities between real and virtual scenes are being introduced earlier in the neural processing of vision than the dorsal and ventral streams, significantly narrowing the possible regions where perceptual differences are originating. Unfortunately, we were only able to collect three subjects' worth of data before having to halt activities. The data for real stimuli closely match the results seen by Weiss et al. [2000], see Figure R4. While the results for the virtual stimuli appear promising, the data were significantly noisier and appear to have a counterclockwise rotation relative to expected patterns. We suspect that this was due to electrode alignment errors introduced when wearing headset over the EEG cap.

By the time the pandemic was sufficiently under control to allow this form of research to resume, I had already changed institutions. I am currently seeking funding to continue this work at MSU.

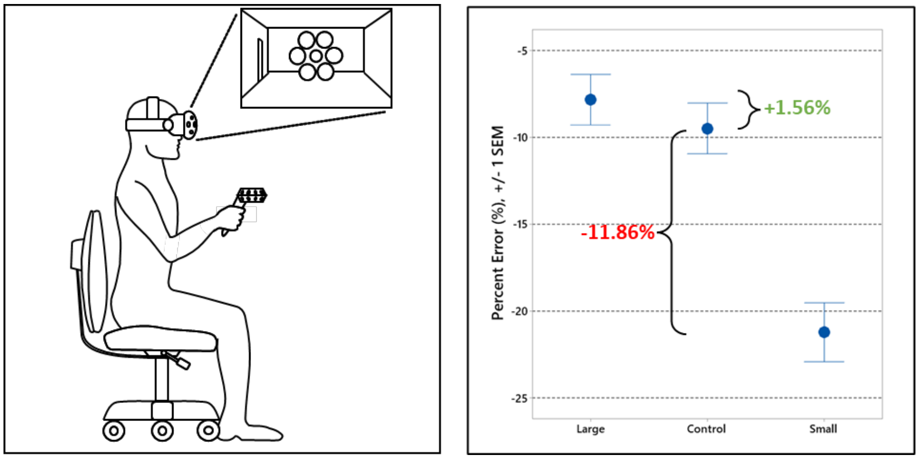


Figure R 4 - Left: Illustration of a participant in the Ebbinghaus Illusion study. Right: The results from this study illustrating an asymmetric effect of anulus size on the strength of the illusion.

However, this has led to the creation of novel EEG and VR integration techniques at MSU, which have led to undergraduate research as well as an abstract and presentation at the Mississippi Academy of Science in 2022.

Behavioral Measurements: Another facet of my work is using virtual environments to take a new look at classic behavioral neuroscience experiments. For instance, the Ebbinghaus Illusion has long been held as an example of how action-based tasks are not affected by visual illusions while judgment-based tasks are affected [Aglioti et al. 1995]. However, in recent years, this view has been hotly debated [Bruno 2001, Carey 2001, Franz 2001]. The Ebbinghaus Illusion consists of a central disc surrounded by an anulus of either large or small discs. When surrounded by small discs, the central disc is perceived as larger than its actual size while it appears smaller when surrounded by large discs, see Figure R4. Since the Ebbinghaus Illusion alters the perceived frontal 2D size of an object, it is difficult to study in the real-world without providing unintentional 3D cues. I designed a study using a virtual environment and the psychophysical task of blind reaching to measure if the illusion, in fact, affected action-based tasks [Finney and Jones 2020]. I found that the influence of the illusion was asymmetric depending on the anulus configuration. The small anulus produced a large illusory response while the large anulus did not significantly differ from the control (no anulus). However, I also found that the response time associated with reporting judgments for both anulus types were significantly higher (by roughly 200ms) than when no annulus was present. Response time is often used to indicate the relative difficulty of cognitively processing a given stimulus. These results imply that both anulus types were eliciting a higher processing load than the control despite having asymmetric effects on reaching. Since publication of these results, others have found a similar pattern of asymmetry [Todd et al. 2021].

On-going Research: An area that I am currently pursuing is perceptual integration of spatial information from multiple sensory channels. In particular, very little work has addressed possible connections between visual and auditory spatial perception. Just as we see in 3D, we also hear in 3D. However, it is not known if conflicting auditory and visual spatial cues have an appreciable effect on a person's spatial behaviors. Our preliminary data indicate that the location of audio sources may be

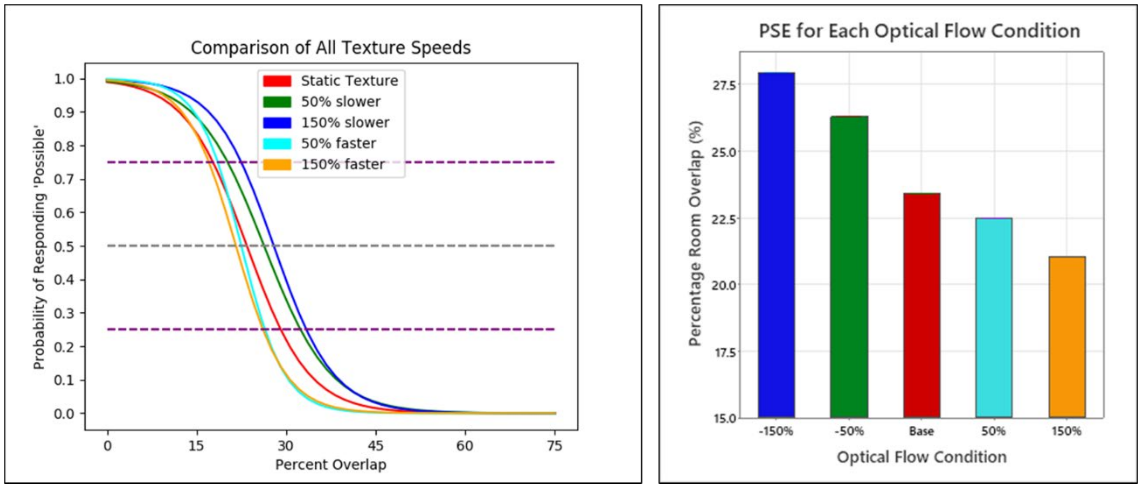


Figure R 5 – Preliminary results from our investigation into the effects of textural flow of spatial judgments. Left: a series of psychometric curve representing observer responses when presented with an Impossible Spaces paradigm. Right: A plot of the point-of-subject-equality (PSE) for each of the psychometric curves illustrating changes in PSE as a function of textural flow speed.

misperceived if the reverberation properties of the sound does not match that of the visual environment. This is particularly interesting since a sound's reverberation is a direct function of the geometry of the space within which it was produced. This is a question that is especially difficult to study in the real world due to a number of logistical issues moving observers from one physical location to the next, as well as mitigating uncontrolled auditory contamination. Thus, this study lends itself well to being evaluated in a virtual environment where both visual and auditory environmental factors can be programmatically controlled. The results of this study will allow us to understand how the brain integrates spatial information from these two sources and what effect it may have on spatial behavior.

The thesis work of MS student Jacob Brewington is aimed at systematically decomposing motion-based properties of vision to isolate the specific parameters that affect our internal metric of self-motion. In other words, what visual cues are we using to perceptually calibrate our sense of motion? It has been well-known for decades that optic flow (the continuous change in the retinal projection caused by movement) drives perceptual motor recalibration. However, no work has addressed the fact that optic flow is actually an amalgam of several visual cues (textural flow and geometric flow) which, by necessity, covary in the real world. It is unlikely that our brains are processing all the components of optic flow in order to calibrate self-motion. Unfortunately, in the real world, we cannot reasonably decompose optic flow into its constituent components. However, this is easily achievable in VR. We have conducted a small pilot study that strongly implicates the textual component of optic flow as the driver of perceptual recalibration, see Figure R5. In this pilot study, we examined constant geometric flow while systematically varying textural flow and then measured how observers walked through a virtual environment. We are currently collecting data on a more robust version of this study in order to see if this hypothesis is further supported.

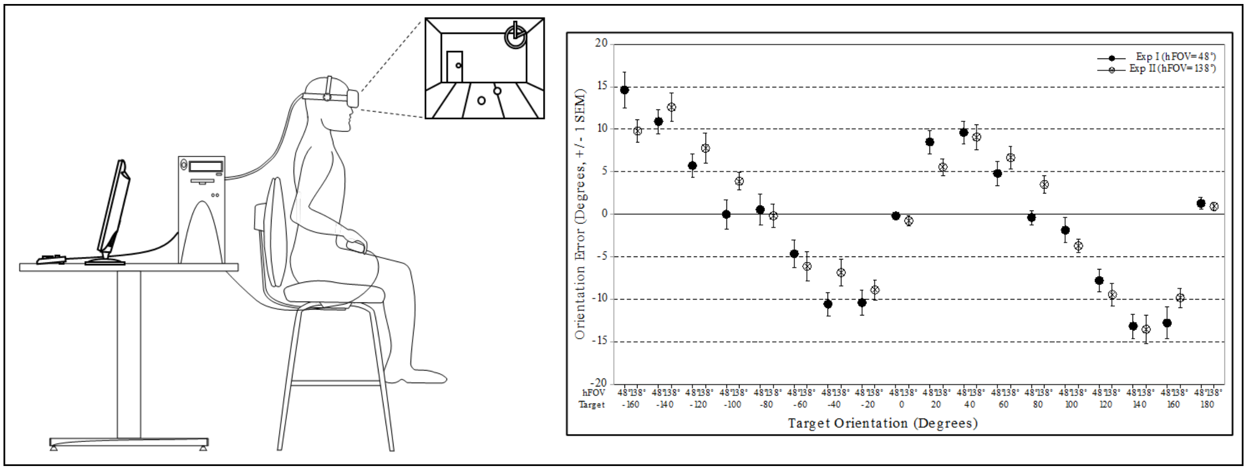


Figure R 6 – Right: Apparatus used by Jones et al.[2019] for measuring orientation judgments. Left: Sinusoidal pattern of error in orientation judgments seen in both small and large field of view virtual environments.

Neuro-Cognitive Transformations

Though cognition is informed by perception, it has flexibility that enables adaptation to occur through feedback, practice, and reasoning. According to the processing pipeline framework, the perceptual representations derived from environmental geometry serve as input for spatial cognition. It is important to note that, in the current context, we are using the term spatial cognition to refer to specifically post-perceptual, consciously accessible spatial information from which actionable decisions are made. This is an important distinction since spatial cognition, depending on the context, can refer to anything from neural representations to decision making (yet another reason to adopt a unified framework for doing this kind of research). Enabling accurate decisions based on spatial information is important for a wide variety of fields. Therefore, VR has the potential to have a substantial impact by facilitating easier and more accurate spatial cognition. It is important to understand the practical implications of using VR to make decisions based on the presentation of spatial information. My research addresses this specific problem as well.

Spatial Biases using VR: The concept of making spatially informed decisions utilizing computer-generated imagery is not new and has been studied in the context of a wide variety of tasks ranging from architectural simulation to air-traffic control. Many of these cases have shown the presence of systematic biases in spatial judgments that can alter the decisions that an observer makes when acting upon the visual information. A strong example of this was made by Tharp and Ellis [1990], who found that observers judging the orientation of objects on a 3D display systematically misreported their azimuth relative to a defined reference. There has also been substantial work examining people's innate abilities to make spatially oriented decisions based on relative or representative information. A classic example that has been applied broadly is *perspective taking* [Tarampi et al. 2016]. In this task, observers are asked to assume the role of an observer in a hypothetical arrangement of landmarks and are then asked to indicate the direction of a given target as if they were standing at a specific position and orientation within the scene. This has been used as a means of evaluating spatial ability [Tarampi et al. 2016] and is a predictor of success in fields related to STEM [Chang et al. 2017, Shea et al. 2001].

I replicated the procedures used in the study reported by Tharp and Ellis [1990] which was originally part of an effort to examine air-traffic control scenarios using computer graphics. I found that patterns of sinusoidal, allocentric orientation errors seen in 3D perspective displays were also present in both real and virtual environments [Jones et al. 2019]. I then examined this effect under several varying field of view extents and found effectively identical results. To ensure that these errors were not the result of the specific task adapted from Tharp and Ellis [1990], I introduced several other measurement methods that relied on judging the orientation of targets. These tasks produced nearly identical results in both the real and virtual worlds. This is a very intriguing finding as it indicates that there is an apparent radial anisotropy, a systematically non-isotropic scene representation, when judging spatial orientations, see Figure R6. Perhaps the most remarkable part of this, however, is the high degree of consistency in errors seen between observers. This implies that the bias affecting orientation-related spatial cognition is highly predictable for most people. This can enable us to design interfaces that actively predict and correct biases in spatial cognition in real-time.

On-going Research: I am currently working with faculty at Juniata College in Geology on using VR to study and improve spatial reasoning skills. Interestingly, geologists are trained in making accurate spatial judgments about geological formations and topography. These are difficult skills to train and often necessitate substantial hands-on practice. We are investigating whether VR can be used to improve spatial reasoning skills through simulated hands-on training exercises. This work is supported by the National Science Foundation.

In a perfect world, we could peer directly at the image seen in the mind of an observer and measure how it changes under varying viewing conditions. Unfortunately, this is not yet possible. However, students in the visual arts are often trained to make hand-drawn images based on their visual impressions. In a way, drawing gives us an indirect image of the world as seen by the mind's eye. Daniel Molsbarger, for his MS project, developed perspective drawing tasks to produce an image of a scene recalled after viewing either a real or virtual environment. Similar sketch-based tasks have been employed by others to study changes in the recall of scenes, but it has not been applied to the study of spatial cognition in virtual environments. Since our prior work indicates that field of view restriction affects perceived scale, we will have observers view both real and virtual scenes under varying field of view restrictions and have them recreate the recalled scene using perspective drawing techniques. We plan on analyzing the perspective lines used to construct the drawing in order to measure the perceived scale of the recalled scene.

Student Research Mentorship

As a final addition to this chapter, I want to say that science does not happen in a vacuum. Fellow scientists and other collaborators are essential not only to supporting my research agenda but also for advancing the future of scientific discovery. As such, we must acknowledge that scientists are not born but made through academic, technical, and philosophical education. I have made it my mission in academia to train as many potential scientists as possible. All of the projects, experiments, and discoveries detailed here have been enabled by the work of students. I make it a practice that if a student demonstrates clear competence and talent, I will assist them in finding a research or development project that fits their skills. I am grateful to these students for their enthusiasm and effort. Below, I have included a table listing the students I have mentored on these projects. Undergraduate students are highlighted in blue while graduate students are highlighted in green.

Name	Project Type	Year	Project Title/Topic
Alex Gunter	Senior Project	2016	General Purpose Motion Tracking using Retroreflective Markers and the MS Kinect
Joseph Woestendiek	Senior Project	2016	Stereoscopic 3D Table-top Augmented Reality Display with Hand Tracking
Matt Brown	Senior Project	2016	Mechanical, 3D Position Tracking for the Oculus Rift DK1
Shaylen Patel	Senior Project	2016	DIY 3D Shutter Glasses Using the Intel Galileo 2
Sean O'Hara	Honors Thesis	2016	Multi-body, Infrared Motion Tracking with Off-The-Shelf Components
Rashad Collier	Senior Project	2017	Walking Recalibration in VR, Proof of Concept
McKennon McMillian	Senior Project	2017	Depth Light: A Mixed Reality Interaction Device
Collin Roth	Senior Project	2017	Tabletop Optical See-Through Augmented Reality System
Ricky Bojorquez	Senior Project	2017	Gait Tracking and Analysis in Immersive Virtual Reality
Katherine Hewlett	Senior Project	2017	Real-Time Correction of Off-Axis Projections
Dylan Devenny	Honors Thesis	2018	The Effect of Optical Flow on Redirection Techniques
Ethan Luckett	Honors Thesis	2018	Quantifying the HTC Vive's Lighthouse Tracking Accuracy
Jonathan Hopper	Senior Project	2018	Optical Flow Presentation in Virtual Reality
Tykeyah Key	McNair Scholar	2018	Determining Vive Motion Sensitivity Using a Robotic Device
Khoa Tran	SURGG Intern	2018	Flood Flow Visualization in Virtual Reality
Robert Lampton	Senior Project	2019	Perceptual Effects of Textual Optical Flow Variations and Peripheral Vision
Marcus Higgins	Senior Project	2019	StatsViz: A Tool for Visual Analysis of Data
Cole Smith	Senior Project	2019	Random Dot Stereograms and Dynamic Maze Generation
Alana Reemsnyder	Neuroscience Interm	2019	Frontal Orientation Judgments as a Function of Eccentricity, an Eye-Tracking Study
Nayan Chawla	Senior Project	2020	Video-Overlay AR Interface for Off-The-Shelf Video Conferencing Tools
James Shaver	Senior Project	2020	Neural Network Polygonization of Point Cloud Data
Andrew Jelson	Senior Project	2020	Neural Network Polygonization of Point Cloud Data
Hunter Finney	Senior Project	2020	Redirected Walking: Exploiting Change Blindness in Virtual Reality
William Brozovic	Senior Project	2020	Computational Archaeology: Using Data Science to Understand the Past
Benjamin Creel	Honors Thesis	2020	Measuring the Bacterial Load of Virtual Reality Headsets
Olivia Eustice	Honors Thesis	2020	EEG Source Localization While Performing Depth Judgments in Virtual Reality
Alessandriel Harper	McNair Scholar	2020	Remote Video-Based Marker Tracking for Augmented Reality
Paul Budz	Senior Project	2021	Arduino-Based Flight Telemetry Computer for Amateur Rocketry
Zachary Garris	Senior Project	2021	Hologram-like Projections using a Persistence-of-Vision Display
Ariana Howell	Senior Project	2021	Inverse Kinematics in Python using Off-The-Shelf VR Tracking
Parker Mathis	Senior Project	2021	Framework for Simulating Novel AR Hardware Configurations within VR
Thong Tra	Senior Project	2021	Video-based Augmented Reality with Hand Interactions for Video Conferencing
Ronnie Davis	McNair Scholar	2021	Effects of Target Eccentricity on Perceived Depth in Virtual Reality
Robert Ingraham	Biomedical Intern	2021	Effects of Target Eccentricity on Perceived Depth in Virtual Reality
Logan Parker	Honors Thesis	2022	Increasing the Perceived Realism of Mixed Reality using "Diminished Virtual Reality"
Maggie Pettus	CSE 4800	2022	EEG Electrode Localization using Off-The-Shelf Components
Timothy Stewart	NSF Sponsored	2023	DriVR: Extending Driver Training for Persons with Disabilities
Ander Talley	AFRL Sponsored	2023	AI2VR: Placing Artificial Intelligence into Virtual Reality to Evaluate Its Performance
Kyler Smith	ECE Senior Team 1	2024	Enhancing Safety and Usability of VR for Wheelchair Users
Garrett Bradshaw	ECE Senior Team 1	2024	Enhancing Safety and Usability of VR for Wheelchair Users
Slade Hicks	ECE Senior Team 1	2024	Enhancing Safety and Usability of VR for Wheelchair Users
Brandon Waldrup	ECE Senior Team 1	2024	Enhancing Safety and Usability of VR for Wheelchair Users
Peyton Jackson	ECE Senior Team 2	2024	Enhanced Training for Disabled Drivers using Modified Hand-Controls
Kylar Fielder	ECE Senior Team 2	2024	Enhanced Training for Disabled Drivers using Modified Hand-Controls
Hunter Reinike	ECE Senior Team 2	2024	Enhanced Training for Disabled Drivers using Modified Hand-Controls
Robert Carver	ECE Senior Team 2	2024	Enhanced Training for Disabled Drivers using Modified Hand-Controls
Blayne Brister	ECE Senior Team 2	2024	Enhanced Training for Disabled Drivers using Modified Hand-Controls
Christian Johnson	Honors Thesis	2024	The Influence of Spatial Anxiety on Egocentric Distance Judgments in VR
Isaac Martinolich	Honors Thesis	2024	Keitaiman: A Readability Analyzer for Anime
Eddie Caveness	CSE 4800	Incomplete	DriVR: Extending Driver Training for Persons with Disabilities
Woody Watson	NSF Sponsored	In Progress	DriVR: Extending Driver Training for Persons with Disabilities
Jewell Norris	CSE 4800	In Progress	EEG Electrode Localization using Off-The-Shelf Components
Caden Thompson	CSE 4800	In Progress	Effects of the Ebbinghaus Illusion on Stereoscopic Depth
Connor Chrismond	NSF Sponsored	In Progress	DriVR: Extending Driver Training for Persons with Disabilities
Michael Tvarkunas	Hi5 Intern	In Progress	AI2VR: Placing Artificial Intelligence into Virtual Reality to Evaluate Its Performance
Daniel Coto	MS Project	2016	Camera-based Correction of Optical Distortions in Head-Mounted Displays
William Panlener	MS Thesis	2018	Effects of Field of View Restriction on Spatial Judgments
McKennon McMillian	MS Thesis	2019	The Effects of Return Protocol on Depth Judgments as Measured with Blind Walking
Collin Roth	MS Project	2019	Latency Detection Thresholds in HMD-based Virtual Environments
Ethan Luckett	MS Thesis	2020	Assessing Distance Perception in Virtual and Augmented Realities with EEG
Jonathan Hopper	MS Thesis	2020	Simulation of VR Display Characteristics for Motion Perception Evaluation
Joao Paulo Marum	PhD Dissertation	2021	Dependency-based Reactive Change Propagation in Virtual Environments
Daniel Molsbarger	MS Project	2023	Using Sketched-based Methods to Evaluate Spatial Scale in Reality and VR
Zachary Garris	MS Thesis	2023	Localization of Spatial Audio in Matched and Mismatched Virtual Environments
Aussie Warren	MS Thesis	Incomplete	Haptic Feedback Devices for Enhanced Spatial Interactions
Jacob Brewington	MS Thesis	In Progress	The Effects of Decoupling Textural and Geometric Components of Optical Flow
Zeenat Islam	PhD Dissertation	In Progress	Effects of the Temporal Profile of Low-Persistence Displays in Virtual Environments
David Huang	PhD Dissertation	In Progress	TBD
Ander Talley	MS Thesis	In Progress	TBD
Christian Johnson	MS Thesis	In Progress	TBD

References

Aglioti, S., DeSouza, J.F. and Goodale, M.A., 1995. Size-contrast illusions deceive the eye but not the hand. *Current biology*, 5(6), pp.679-685.

Bruno, N., 2001. When does action resist visual illusions? *Trends in cognitive sciences*, 5(9), pp.379-382.

Carey, D.P., 2001. Do action systems resist visual illusions? *Trends in cognitive sciences*, 5(3), pp.109-113.

Chang, J.S.K., Yeboah, G., Doucette, A., Clifton, P., Nitsche, M., Welsh, T. and Mazalek, A., 2017, October. Evaluating the effect of tangible virtual reality on spatial perspective taking ability. *In Proceedings of the 5th symposium on spatial user interaction* (pp. 68-77).

Finney, H. and Jones, J.A., 2020, March. Asymmetric effects of the ebbinghaus illusion on depth judgments. *In 2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)* (pp. 573-578). IEEE.

Franz, V.H., 2001. Action does not resist visual illusions. *Trends in cognitive sciences*, 5(11), pp.457-459.

Jones, J.A., Edewaard, D., Tyrrell, R.A. and Hodges, L.F., 2016, March. A schematic eye for virtual environments. *In 2016 IEEE Symposium on 3D User Interfaces (3DUI)* (pp. 221-230). IEEE.

Jones, J.A., Krum, D.M. and Bolas, M., 2014, March. The effect of eye position on the view of virtual geometry. *In 2014 IEEE Virtual Reality (VR)* (pp. 87-88). IEEE.

Jones, J.A., Swan II, J.E. and Bolas, M., 2013. Peripheral stimulation and its effect on perceived spatial scale in virtual environments. *IEEE transactions on visualization and computer graphics*, 19(4), pp.701-710.

Shea, D.L., Lubinski, D. and Benbow, C.P., 2001. Importance of assessing spatial ability in intellectually talented young adolescents: A 20-year longitudinal study. *Journal of Educational Psychology*, 93(3), p.604.

Tarampi, M.R., Atit, K., Petcovic, H.L., Shipley, T.F. and Hegarty, M., 2016. Spatial skills in expert structural geologists.

Tharp, G.K. and Ellis, S.R., 1990. The effects of training on errors of perceived direction in perspective displays, *NASA Technical Report.* (No. NASA-TM-102792).

Todd, R., Zhu, Q. and Banić, A., 2021, March. Temporal Availability of Ebbinghaus Illusions on Perceiving and Interacting with 3D Objects in a Contextual Virtual Environment. *In 2021 IEEE Virtual Reality and 3D User Interfaces (VR)* (pp. 817-825). IEEE.

Weiss, P.H., Marshall, J.C., Wunderlich, G., Tellmann, L., Halligan, P.W., Freund, H.J., Zilles, K. and Fink, G.R., 2000. Neural consequences of acting in near versus far space: a physiological basis for clinical dissociations. *Brain*, 123(12), pp.2531-2541.

Service Accomplishments

Service Accomplishments

Contents

Highlights

- Served as Major Professor for 15 graduate students across two institutions (12 MS, 3 PhD).
- A long history of mentoring students on capstone, honors, and sponsored research projects.
- Actively engaged in broadening participation in computing and other STEM disciplines.
- Serves as an Associate Editor for the two leading journals in my field (IEEE TVCG, Frontiers in Virtual Reality).
- Served on Editorial Boards of numerous journals.
- Has served on Program and Organizing committees for top-tier conferences for more than 10 years (IEEE VR, IEEE ISMAR).

University Service

Student Advising

At MSU, I have advised seven graduate students as their Major Professor. These include five MS students, two of whom have graduated (*Current:* Jacob Brewington, Jacob Adams, Ander Talley; *Graduated:* Zackery Garris, Daniel Molsbarger) and two current PhD students (Zeenat Islam, David Huang). I serve as a committee member for two PhD students, one at MSU and another at University of Utah (Alexander Sommers, MSU; Hunter Finney, Utah). I also serve as honors thesis advisor for two students (Christian Johnson, Isaac Martinolich). As part of an NSF-funded project, I served as advisor for two ECE senior design teams (10 total students). From my perspective guiding students to graduation is a necessary but insufficient approach to advising. I believe it is paramount to aid these students in developing a scientific thought process and becoming critical thinkers. As such, I spend a significant amount of time working with these students in my lab, teaching them about topics ranging from experimental design and analysis to the nuances of human subjects research.

As part of our departmental service, faculty are asked to perform *professional mentoring* for our students. Each semester I hold two to three professional mentoring group sessions where I answer questions and discuss courses, concentrations, minors, internships, employment possibilities, and graduate education. I usually have roughly 30 students per session. On average, I interact with between 50 and 70 students per semester for professional mentoring.

While at UM, I served as academic advisor for 60 undergraduate and 8 graduate students. I advised 6 MS students to completion, and 1 PhD student to completion. One student did not complete his MS due to receiving a lucrative job offer. I also directed 16 undergraduate capstone projects, 4 honors theses, and 6 sponsored undergraduate interns. I served on numerous other graduate and undergraduate thesis committees across multiple departments.

In order to better promote the involvement of undergraduate and graduate students in the areas of Human and Visual Computing, Perceptual Psychology, and Neuroscience, I have developed supplemental documentation to help guide their course selections. These include guides of suggested courses, recommended minors, milestones, and degree progress checklists. I typically distribute these to students that I am mentoring in my lab but also make them publicly available when asked. I am including representative samples of these documents from both UM and MSU for reference.

Broadening Participation in Computing

I have actively sought to broaden participation in our discipline by supporting and mentoring a diverse population of students. These students have been from a variety of demographics that are commonly underrepresented in STEM fields including students from various ethnic and racial backgrounds, genders, LGBTQ+ identities, and physical impairments.

As part of this goal, I have served since 2018 as a mentor in the *Ronald E. McNair Post-Baccalaureate Achievement Program (McNair Scholars)*, a nationally focused program dedicated to increasing the achievement of doctoral degrees by students from typically underrepresented groups through experiential research education and one-on-one mentorship. After completing their sponsored research experience, I maintain yearly contact with all of these former students in order to provide help, references, and other assistance. I am presently working with MSU's *TRIO and GEM Lab programs*, sister programs to the McNair Scholars, in order to facilitate recruitment and retention of underrepresented students.

University Committees & Service

During my time at UM, I served on several committees. From 2016 to 2021, I served on the *Neuroscience Curriculum Committee* which worked to continuously improve the quality and variety of neuroscience classes available to students in the Neuroscience Minor program. From 2017 to 2021, I served on the *CIS Undergraduate Committee*. I also served on two *faculty search committees* resulting in addition of three faculty to our department.

Since joining MSU, I have served on the CSE *Graduate Committee*, *Undergraduate Committee*, and *Assessment Committee*. When I previously worked professionally for MSU during 2007 and 2008, I also served on the *Steering Committee for the Institute for Neurocognitive Science and Technology* and the *Imaging Center for Excellence Quality Control Committee*. I include these in this list in order to communicate that I have a historical understanding of the progress of neuroscience related research at MSU.

Presently, I serve on the *BCoE Research Advisory Committee*, an appointment of two years. The purpose of the committee is to broadly advise on policies and directions associated with research support for the Bagley College of Engineering. The goal for the 2024 year is to advise on faculty incentives and other research support related programs to promote the pursuit of external funding and also retain faculty.

I have served on the *Organizing Committee for the 2024 Innovate State Hackathon*. The hackathon focused on building Lego-based battle bots. Teams of students from across campus competed to see whose battle bot would prevail. This was intended to be an event to attract young spectators and promote STEM education.

I presently serve as advisor for our local Association of Computing Machinery (ACM) student chapter. With the aid of a new group of student officers, we have performed a major reboot of our ACM chapter. We now have over 80 registered members and over 100 members in our online community. As part of the reboot, we have instigated several local "Extended Interest Groups" on a range of professional and social topics ranging from High Performance Computing to Animation.

Online Content

All video content from my classes, study-sessions, and tutorials that I taught remotely during COVID-necessitated remote instruction are shared publicly through my lab's website and corresponding YouTube channel[1]. I also began formally recording and disseminating VR-specific education and research content during the summer of 2020 when I organized the *Hi5 Seminar Series for Early-Career Researchers*. This seminar series was intended to keep members of our research community connected and engaged during the COVID-19 pandemic. The seminar was live-streamed via YouTube and is still ongoing to present. Though it began with a focus on early-career researchers, we now host speakers at all stages of their careers. The YouTube channel has had over 8,500 individual views and 553 hours of watch time since it began in June 2020.

Neuroscience Minor Program at MSU

I am presently working with colleagues in Kinesiology, Agriculture and Biological Engineering, Psychology, and Philosophy to create a Neuroscience Minor program at MSU. Though this work is in its early stages, we currently have support from our respective department heads and dean's offices. I have drafted a curriculum which has distributed to participating departments in order to gather additional feedback and suggestions before submitting the minor for review by the University Committee on Courses & Curricula (UCCC). I have specifically developed the draft curriculum such that it will not necessitate the addition of new courses. Fortunately, the classes typically required for a neuroscience minor at comparable R1 institutions are currently taught at MSU but are spread across multiple departments.

Professional Service

Conference Organizing

I have served on the organizing committees for my field's two largest conferences. In particular, I have served on the IEEE Virtual Reality (VR) organizing committee in a variety of chair capacities form 2010 through 2021. In 2022, I began serving on the organizing of IEEE Intl. Symposium on Mixed and Augmented Reality (ISMAR) in the role of Program Chair. I was invited to also serve in this capacity for 2023 but deferred due to the recent birth of my child.

Program Committees

Since 2014, I have served on 19 program committees across 10 conferences and workshops. A representative sample of these includes: ACM Symposium on Applied Perception (SAP), IEEE Virtual Reality (VR), International Symposium on Visual Computing (ISVC), IEEE Artificial Intelligence and Virtual Reality (AIVR), and IEEE Intl. Symposium on Mixed & Augmented Reality (ISMAR).

Professional Reviewing

I have a long history of professional reviewing service dating back to 2006. During this span I have reviewed for at least 22 venues, many of which for multiple years. Of these, 10 were journals from a variety of fields ranging from computer graphics to psychology. A representative sample of these

[1] https://www.youtube.com/channel/UCe_yTsQ0Tap4NMLJQcbNezw/

include: IEEE Transactions on Visualization and Computer Graphics, ACM Transactions on Applied Perception, Quarterly Journal of Psychology and Frontiers in AI and Robotics.

I have also served as a reviewer for at least 12 conferences, many of which for multiple years. A representative sample of these include: IEEE Virtual Reality (VR), IEEE Intl. Symposium on Mixed and Augmented Reality (ISMAR), ACM SIGCHI, ACM Spatial User Interfaces (SUI), and IEEE Information Visualization (InfoVis).

Editorial Service

I presently serve as an Associate Editor for two of my field's leading journals: IEEE Transactions on Visualization and Computer Graphics (TVCG) and Frontiers in Virtual Reality. Previously, I was guest editor for two special edition journals: Frontiers in Virtual Reality – Perception and Neuroscience in XR (2021) and Multimodal Technologies and Interaction – Perception and Cognition in XR (2020-2021).

I began editorial service in 2014 when I was invited to the Editorial Board of Frontiers in Robotics and AI to work within the Virtual Reality topic area. I served on this board through 2019 when this topic area had become large enough to become a separate journal, Frontiers in Virtual Reality. I have served on the Frontiers in Virtual Reality editorial board from 2019 until present. In 2022, I was invited to accept the role of Associate Editor for this journal. As such, I have served on the editorial board of this journal in its various forms for the past 10 years.

Attachment: Example Advising Checklists

Attached advising checklists:

University of Mississippi – Neuroscience Minor for CIS students recommend curriculum

Mississippi State University – MS in Computer Science Milestones & Checklist

Mississippi State University – PhD in Computer Science Milestones & Checklist

Neuroscience Minor for Ole Miss Computer Science Students

Recommended Course of Study

If you are an undergraduate majoring in Computer Science and interested in pursuing VR research, I have put together a list of recommendations that may help you achieve this goal. In addition to your CS major, I recommend pursuing a minor in Neuroscience since it relates directly to much of our research. The following list of courses will satisfy the requirements for the Neuroscience minor as well as needed electives for the BS in Computer Science. Be aware that not all courses are offered every semester, and their availability may change over time.

The full requirements for the Bachelor of Science in Computer Science (BSCS) degree can be found here.

Computer Science Electives (15 hours)

- CSCI 343 - Data Science
- CSCI 447 - Immersive Media (Virtual Reality)
- CSCI 391 - Computer Graphics
- CSCI 443 or 444 - Advanced Data Science or Information Visualization
- Any other CSCI 300+ elective

Neuroscience Minor Electives (18 hours)

- PSY 319 - Brain & Behavior
- BISC 327 - Introductory Neuroscience
- PSY 326 - Sensation & Perception
- Neu 491 - Directed Research (in Hi5 Lab)
- Any other Neuroscience ***500***+ elective
- Any other Neuroscience ***300***+ elective

Social Science Electives (6 hours)

- Psy 201 - General Psychology (required)
- Psy 202 - Statistics for Behavioral Sciences
- Psy 320 - Cognitive Psychology

Optional

- Bisc 533 - Advanced Neuroscience (counts as a Science elective)
- Phil 101 - Introduction to Philosophy (counts as a Humanities elective)
- Phil 103 - Logic: Critical Thinking (counts as a Humanities elective)

MS Thesis Requirement Completion Checklist

(Edited January 25, 2023)

Student Name:____________________________

Date this form was issued: ____________________________

Desired date of graduation: ____________________________

Action	Expectation	Date Completed
Select a Major Professor *(for Hi5 students this is Dr. Jones)*	Complete within 1st semester.	________________
Complete CITI human subjects research training and send certificate to Major Professor *(required for all Hi5 Lab students)*	Complete within 1st semester.	________________
Select a thesis topic area	As early as possible, no later than 2nd semester.	________________
Write a literature review of the topic area consisting of at least 10 sources. *(Provide to your Major Professor)*	Complete at end of 2nd semester.	________________
Select two faculty members to serve on your Thesis Committee *(Rec. for most Hi5 projects: 1 CSE and 1 Psych/CogSci/Neuro faculty)*	Within 1st month of 3rd semester.	________________
Draft your Thesis Proposal *(reuse previous literature review)*	Complete before mid-term of 3rd semester.	________________
Schedule Proposal Defense with Committee	Complete before mid-term of 3rd semester.	________________
Send Proposal Draft to Major Professor	At least three weeks prior to Proposal Defense.	________________
Send final Proposal Draft to Committee	At least one week prior to Proposal Defense.	________________

This is not an official Computer Science & Engineering form. *This is only intended to be used as an aid in gauging progress toward completing your degree.*

Dry-run of Proposal presentation with Major Professor	At least one day prior to Proposal Defense.	________________
Proposal Defense	Complete before end of 3rd semester.	________________
Design & Conduct Thesis Experiments	***Begin as early as possible (immediately)!*** ***Preliminary experiments may be conducted prior to Proposal with Major Professor's guidance.***	
Draft your Thesis Document	Complete before mid-term of 4th semester.	________________
Schedule Thesis Defense with Committee	Complete within 1st month of 4th semester.	________________
Apply for Graduation *(without late fees)*	Deadline is early-October in Spring, early-March in Fall. ***Check Graduate Academic Calendar for exact dates.***	________________
Send Thesis Draft to Major Professor	At least three weeks prior to Thesis Defense.	________________
Send final Thesis Draft to Committee	At least one week prior to Thesis Defense.	________________
Dry-run of Thesis presentation with Major Professor	At least one day prior to Thesis Defense.	________________
Thesis Defense	Defense deadline is usually mid-March in Spring, mid-October in Fall. ***Check Graduate Academic Calendar for exact dates.***	________________
Submit Thesis to Library for review *(must include all edits requested by your committee)*	Deadline is usually early-April in Spring, early-November in Fall. ***Check Graduate Academic Calendar for exact dates***.	________________
Pay University account balance to ensure graduation	Deadline is usually early-May in Spring, early-December in Fall. ***Check Graduate Academic Calendar for exact dates***.	________________

This is not an official Computer Science & Engineering form. *This is only intended to be used as an aid in gauging progress toward completing your degree.*

Coursework Checklist – Human-Centered Computing Research Concentration

This document provides loose coursework guidance based on CSE requirements. Recommended courses are suggested for students conducting research in the Hi5 Lab who are interested in pursuing careers in human-centered and/or visual computing.

Requirement	Course	Completed/Grade
Graduate Seminar: ***Take within your first two semesters.*** *Recommended seminars series to also attend:* *CSE Seminars* *Hi5 Seminars* *ACCESS CogSci Seminars*	CSE 8011	_________ /____
Choose one "CSE Theory": *CSE 8813*, CSE 8833, CSE 8843 *Recommended*	________________	_________ /____
Choose three "Research Depth Courses": *CSE 6663*, CSE 8283, CSE 8613, *PSY 6423**, *IE 8153* *Recommended* *Recommended* *Recommended*	________________	_________ /____
or others approved courses **technically requires approval*	________________	_________ /____
	________________	_________ /____
Choose two "Research Breadth Courses": Any approved non-HCC 6XXX or 8XXX courses	________________	_________ /____
CSE 8413, *PSY 8214* *Recommended* *Recommended*	________________	_________ /____
CSE Thesis Research Hours:		
Thesis Research (3 hours)	CSE 8000	_________ /____
Thesis Research (3 hours)	CSE 8000	_________ /____
Choose two "Free Electives (6XXX/8XXX)":		
	________________	_________ /____
	________________	_________ /____

This is not an official Computer Science & Engineering form. *This is only intended to be used as an aid in gauging progress toward completing your degree.*

Pre-Approved Non-CSE Research Depth Courses for Human-Centered Computer and Visualization:

- OTHER COURSES MAY BE APPROVED BY THE STUDENT'S GRADUATE COMMITTEE
- PSY 6653 Cognitive Science (cross-listed as CSE 6653) (Prerequisite: CS 4633/6633 or PSY 4713 or PHI 4143/6143 or AN 4623/6623 or EN 4403/6403). Three hours lecture. The nature of human cognition from an interdisciplinary perspective, primarily utilizing a computational model, including insights from philosophy, psychology, linguistics, artificial intelligence, anthropology, and neuroscience. (Same as CS 4653/6653).
- PSY 6743 Psychology of HCI (Prerequisites: PSY 3713 or CS 4663/6663 or IE 4113/6113 or consent of the instructor). Two hours lecture. Two hours laboratory. Exploration of psychological factors that interact with computer interface usability. Interface design techniques and usability evaluation methods are emphasized. (Same as CS 4673/6673 and IE 4123/6123).
- PSY 8723 Cognitive Skills Models (Prerequisite: Graduate Standing). Three hours lecture. Introduction to cognitive modeling, with a focus on computational models of skill acquisition and expert skill (Same as CS 8613).
- IE 6113 Human Factors Eng (Prerequisite: Junior standing in engineering). Two hours lecture. Three hours laboratory. Human capabilities and limitations affecting communications and responses in man-machine systems. Emphasis on physiological and psychological fundamentals.
- IE 8143 Applied Ergo Methods Three hours lecture. Provide practical usage and theoretical background of select tools for ergonomic evaluation of workers and workplaces, tasks, and environments using real world scenarios.
- IE 8153 Cognitive Engr Three hours lecture. Implications of human perceptual, cognitive, and psycho-motor capabilities on the design of systems for effective, efficient and safe human-machine performance.
- ECE special topics in visualization and graphics

PhD Dissertation Requirement Completion Checklist (MS Admit)

(Edited April 12, 2024)

Student Name:___________________________

Date this form was issued: ___________________________

Desired date of graduation: ___________________________

Action	Expectation	Date Completed
Select a Major Professor *(for Hi5 students this is Dr. Jones)*	Complete within 1st semester.	_______________
Complete CITI human subjects research training and send certificate to Major Professor *(required for all Hi5 Lab students)*	Complete within 1st semester.	_______________
Select a dissertation topic area	As early as possible, no later than 2nd semester.	_______________
Write a literature review of the topic area consisting of at least 10 sources. *(Provide to your Major Professor)*	Complete at end of 2nd semester.	_______________
Qualifying Exam	Complete within first two semesters.	_______________
Select three faculty members to serve on your Dissertation Committee *(Rec. for most Hi5 projects: 2 CSE and 1 Psych/CogSci/Neuro faculty)*	Within 1st month of 3rd semester.	_______________
Notify your Major Professor that you are ready for your Preliminary Exam and Dissertation Proposal. *(in the Hi5 Lab this is combined with the Proposal Defense)*	At least one month prior to your Proposal Defense date.	_______________
Draft your Dissertation Proposal *(reuse previous literature review)*	Complete before mid-term of 3rd semester.	_______________
Schedule Proposal Defense with Committee	Complete before mid-term of 3rd semester.	_______________

This is not an official Computer Science & Engineering form. *This is only intended to be used as an aid in gauging progress toward completing your degree.*

Send Proposal Draft to Major Professor	At least three weeks prior to Proposal Defense.	________________
Send written responses to Preliminary Exam to your Major Professor	At least one week prior to Proposal Defense.	
Send final Proposal Draft to Committee	At least one week prior to Proposal Defense.	________________
Dry-run of Proposal presentation with Major Professor	At least one day prior to Proposal Defense.	________________
Proposal Defense	Complete before end of 3rd semester.	________________
Design & Conduct Dissertation Experiments	***Begin as early as possible (immediately)! Preliminary experiments may be conducted prior to Proposal with Major Professor's guidance.***	
Draft your Dissertation Document	Complete before mid-term of 4th semester.	________________
Schedule Dissertation Defense with Committee	Complete within 1st month of 4th semester.	________________
Apply for Graduation *(without late fees)*	Deadline is early-October in Spring, early-March in Fall. ***Check Graduate Academic Calendar for exact dates.***	________________
Send Dissertation Draft to Major Professor	At least three weeks prior to Dissertation Defense.	________________
Send final Dissertation Draft to Committee	At least one week prior to Dissertation Defense.	________________
Dry-run of Dissertation presentation with Major Professor	At least one day prior to Dissertation Defense.	________________

This is not an official Computer Science & Engineering form. This is only intended to be used as an aid in gauging progress toward completing your degree.

Dissertation Defense	Defense deadline is usually mid-March in Spring, mid-October in Fall. ***Check Graduate Academic Calendar for exact dates.***	________________
Submit Dissertation to Library for review *(must include all edits requested by your committee)*	Deadline is usually early-April in Spring, early-November in Fall. ***Check Graduate Academic Calendar for exact dates***.	________________
Pay University account balance to ensure graduation	Deadline is usually early-May in Spring, early-December in Fall. ***Check Graduate Academic Calendar for exact dates***.	________________

This is not an official Computer Science & Engineering form. *This is only intended to be used as an aid in gauging progress toward completing your degree.*

PhD Coursework Checklist – MS Admit

Human-Centered Computing Concentration

This document provides loose coursework guidance based on CSE requirements. Recommended courses are suggested for students conducting research in the Hi5 Lab who are interested in pursuing careers in human-centered and/or visual computing. ***You must have more 8000-level courses than 6000-level once your course work has been completed.***

Requirement	Course	Completed/Grade
Graduate Seminar: ***Take within your first two semesters.*** *Recommended seminars series to also attend:* *CSE Seminars* *Hi5 Seminars* *ACCESS CogSci Seminars*	_CSE 8011________	_________ /____
ETD Format and Submission Course: ***Must be taken in the semester you plan to defend your dissertation***	_LIB 9010_________	_________ /____
Choose two "CSE Core": *CSE 8813*, CSE 8833, CSE 8843 *Recommended*	________________ ________________	_________ /____ _________ /____
Choose four "Depth Courses": *CSE 6663*, CSE 8283, CSE 8613, *PSY 6423**, *IE 8153* *Recommended* (CSE 6663), *Recommended* (PSY 6423*), *Recommended* (IE 8153) *Other courses available with approval.*	________________ ________________ ________________ ________________	_________ /____ _________ /____ _________ /____ _________ /____
Choose two "Breadth Courses": Any approved non-HCC 6XXX or 8XXX courses *CSE 8413*, *PSY 8214* *Recommended* *Recommended*	________________ ________________	_________ /____ _________ /____
Choose three "Free Electives": Could also be additional CSE 9000 courses	________________ ________________ ________________	_________ /____ _________ /____ _________ /____

This is not an official Computer Science & Engineering form. *This is only intended to be used as an aid in gauging progress toward completing your degree.*

CSE Dissertation Research Hours:	CSE 9000	Hours:	________/____
Dissertation Research (21 hours)	CSE 9000	Hours:	________/____
	CSE 9000	Hours:	________/____
	CSE 9000	Hours:	________/____
	CSE 9000	Hours:	________/____
	CSE 9000	Hours:	________/____
	CSE 9000	Hours:	________/____

Pre-Approved Non-CSE Research Depth Courses for Human-Centered Computer and Visualization:

- OTHER COURSES MAY BE APPROVED BY THE STUDENT'S GRADUATE COMMITTEE
- PSY 6653 Cognitive Science (cross-listed as CSE 6653) (Prerequisite: CS 4633/6633 or PSY 4713 or PHI 4143/6143 or AN 4623/6623 or EN 4403/6403). Three hours lecture. The nature of human cognition from an interdisciplinary perspective, primarily utilizing a computational model, including insights from philosophy, psychology, linguistics, artificial intelligence, anthropology, and neuroscience. (Same as CS 4653/6653).
- PSY 6743 Psychology of HCI (Prerequisites: PSY 3713 or CS 4663/6663 or IE 4113/6113 or consent of the instructor). Two hours lecture. Two hours laboratory. Exploration of psychological factors that interact with computer interface usability. Interface design techniques and usability evaluation methods are emphasized. (Same as CS 4673/6673 and IE 4123/6123).
- PSY 8723 Cognitive Skills Models (Prerequisite: Graduate Standing). Three hours lecture. Introduction to cognitive modeling, with a focus on computational models of skill acquisition and expert skill (Same as CS 8613).
- IE 6113 Human Factors Eng (Prerequisite: Junior standing in engineering). Two hours lecture. Three hours laboratory. Human capabilities and limitations affecting communications and responses in man-machine systems. Emphasis on physiological and psychological fundamentals.
- IE 8143 Applied Ergo Methods Three hours lecture. Provide practical usage and theoretical background of select tools for ergonomic evaluation of workers and workplaces, tasks, and environments using real world scenarios.
- IE 8153 Cognitive Engr Three hours lecture. Implications of human perceptual, cognitive, and psycho-motor capabilities on the design of systems for effective, efficient and safe human-machine performance.
- ECE special topics in visualization and graphics

This is not an official Computer Science & Engineering form. This is only intended to be used as an aid in gauging progress toward completing your degree.

Attachment: IEEE ISMAR Program & Message from Chairs

Attached materials include:

IEEE ISMAR 2022 Program Chairs Message

IEEE ISMAR 2022 Complete Program Overview

Message from the ISMAR 2022 Science and Technology Conference Program Chairs

Henry Duh
La Trobe University, Australia

Ian Williams
Birmingham City University, UK

Jens Grubert
Coburg University, Germany

J. Adam Jones
Mississippi State University, USA

Jianmin Zheng
Nanyang Technological University, Singapore

2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR) | 978-1-6654-5325-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISMAR55827.2022.00006

We are delighted to welcome you to the 21st IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2022) conference program. Following the success of 2021, this year's conference again featured a separate conference paper track, therefore continuing the clear distinction between the contributions of the journal and conference papers. The Science and Technology program of ISMAR 2022 collates both the journal and conference papers and presents them equally in terms of presentation time and sessions. This supports a clear collegial vision of the contributions within ISMAR 2022 showcasing cutting edge research in our field. We are confident this program will inspire further research while provoking discussions and creating a platform for co-creation of future work.

This year we received 441 submissions to the conference paper track, representing a near twofold increase in submissions from the previous year. This high number of submissions is especially outstanding given the complex situation of the COVID-19 pandemic over the last few years. During this period, researchers have all faced unforeseeable difficulties in coordinating research activities, collecting data, and conducting laboratory experiments. This growth in submissions illustrates the resilience of our community to not only continue but to grow in strength and size.

Regarding the specifics of the conference paper submission and review process, there was again a single paper submission category, with papers from 4 to 8 pages in length (plus two pages of references) being permitted. All papers were reviewed by an International Program Committee (IPC) which comprised 60 internationally renowned experts from the Asia-Pacific region, the Americas, and Europe. All submissions underwent a rigorous review process that encompassed a single review cycle overseen by a coordinator from the IPC. After IPC members had declared their conflicts and provided their preferences, the program chairs assigned coordinators. Where there was a conflict of interest, both the reviewer assignments and reviewer names were hidden from the IPC and program chairs. Additionally, as with prior ISMAR conferences, we again applied a double-blind process and the external reviewers were, therefore, not aware of the identity of the authors. Before the reviewing began, a desk rejection and a quick rejection process was applied, which focused on applying the publication guidelines available on the ISMAR 2022 website. During the review cycle, each submission received reviews from two external experts and one review from a secondary IPC member. New for ISMAR 2022, after the reviewing period, we implemented the opportunity for the authors to submit a response letter to the reviewers. The authors' response letters provided an opportunity to highlight the strengths and merits of their paper and address concerns raised in the reviews. Following this response phase, the coordinator checked the review quality and asked for an improvement of insufficient reviews or for additional reviews where required. Then, reviewers of each submission, under the guidance of the coordinator, anonymously discussed the submission and the author's response letter to reach a consensus decision. The coordinator was then in charge of providing a meta-review and an initial recommendation following the outcome of the discussion. Four recommendations were possible at this stage: (1) accept as conference paper, (2) conditionally accept as conference paper, (3) reject or (4) no consensus reached. In the case of a lack of consensus after the discussion phase, the Conference Paper Chairs recruited an additional reviewer from the IPC. Finally, the program chairs met to discuss each paper with an emphasis on maintaining consistency in the recommendations from the coordinators, to validate the decisions, and to discuss the papers still missing a final recommendation.

This process resulted in a selection of 93 papers, corresponding to an overall acceptance of 21.1%. The papers cover a wide range of topics relevant to and advancing AR, VR, and MR, including: visual tracking methods, learning-based rendering techniques, studies on perception and collaboration, explorations into the metaverse, and many more. All the conference papers will be presented at ISMAR 2022, alongside journal papers in thematically organized sessions. In addition to these papers, 98 were recommended by the coordinators for consideration as poster papers with 49 of these submissions finally accepted.

We would personally like to thank all who made the ISMAR 2022 Conference Papers track possible. Special thanks go to the IPC members who donated their time to ensure a fair selection process during a very short review period. Additional thanks go to the 391 external reviewers for their insightful and thorough reviews. We would also like to thank the General Chairs, Frank Guan, Yiyu Cai, and Michele Fiorentino, for their support throughout the entire process, and the members of the ISMAR Steering Committee for their continuing active support. We also wish to acknowledge James Stewart for his outstanding and timely support with the PCS review system, the Publications Chairs Mohammed Safayet Arefin, Xinxing Xia, and Weitao Song for collecting materials and assisting in the production of the proceedings, and Lisa O'Conner for excellent support with the IEEE Computer Society Conference Publishing Services. Finally, thank you to all of our ISMAR community for your support of our conference and for continuing your excellent research.

2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR) | 978-1-6654-5325-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISMAR55827.2022.00004

2022 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)

ISMAR 2022

Table of Contents

Pages 132 - 134 omitted for external review.

Selected Publications

Selected Publications

I have included a small number of publications that I feel are most representative of my work. These include three journal articles and a U.S. patent that was developed as a direct result of my research.

- **Jones, J.A.**, Swan II, J.E. and Bolas, M., 2013. Peripheral Stimulation and Its Effect on Perceived Spatial Scale in Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics, 19*(4), pp.701-710.

  This paper is a good example of the incorporation of vision science and computer science. Though this paper is from my time as a postdoc, it served as an inflection point in my career. It challenged the findings of prior research using a series of straightforward, simply designed experiments. It was met with substantial resistance at the time of its publication but has now become one of my most highly cited (and most highly replicated) papers. Though initially controversial, the findings of this paper are now considered common knowledge in my field.

- **Jones, J.A.**, Hopper, J.E., Bolas, M.T. and Krum, D.M., 2019. Orientation Perception in Real and Virtual Environments. *IEEE Transactions on Visualization and Computer Graphics*, *25*(5), pp.2050-2060.

  This paper is among my favorites as it addresses a topic frequently overlooked by VR perception research. In addition to exploring an understudied area in VR, it revealed aspects of basic human spatial vision that had not previously been observed.

- Oliveira Marum, J.P., Cunningham, H.C., **Jones, J.A.** and Liu, Y., 2024. Following the Writer's Path to the Dynamically Coalescing Reactive Chains Design Pattern. *Algorithms, 17*(2), p.56.

  This paper is the culmination of my first PhD student's dissertation research. This research began by studying ways to produce highly reliable simulations in virtual environments where multiple elements may interact in unpredictable ways. However, it became apparent that we had developed a new form of software design pattern that could be broadly applied to any dynamic, self-updating system.

- Bolas, M., **Jones, J.A.**, McDowall, I. and Suma, E., 2017. Dynamic Field of View Throttling as a Means of Improving User Experience in Head Mounted Virtual Environments. *U.S. Patent* 9,645,395.

  This patent addresses ways to mitigate visual discomfort, including motion sickness, for users of VR systems. This method was found as result of my research on field of view manipulation. This approach is now used in numerous video games and virtual environment applications; examples include Google Earth VR and Assassin's Creed Nexus VR.

Peripheral Stimulation and its Effect on Perceived Spatial Scale in Virtual Environments

J. Adam Jones, J. Edward Swan II, *Member, IEEE*, and Mark Bolas, *Member, IEEE*

Abstract—The following series of experiments explore the effect of static peripheral stimulation on the perception of distance and spatial scale in a typical head-mounted virtual environment. It was found that applying constant white light in an observer's far periphery enabled the observer to more accurately judge distances using blind walking. An effect of similar magnitude was also found when observers estimated the size of a virtual space using a visual scale task. The presence of the effect across multiple psychophysical tasks provided confidence that a perceptual change was, in fact, being invoked by the addition of the peripheral stimulation. These results were also compared to observer performance in a very large field of view virtual environment and in the real world. The subsequent findings raise the possibility that distance judgments in virtual environments might be considerably more similar to those in the real world than previous work has suggested.

Index Terms—Virtual environments, spatial perception, distance judgments, field of view, periphery

1 Introduction

Users of virtual environments, when judging positions along the ground plane, have consistently indicated that the apparent distance of objects appear closer than their actual geometric position [5, 13, 14, 15, 19, 25, 30, 32, 33, 37, 38, 40, 44, 45, 46, 47]. This effect is confusing as great lengths are often taken to ensure that the retinal projection of the virtual environment closely matches that of an equivalent real world environment [2, 28, 31, 42]. This is even more interesting when comparing these results to the significantly larger body of distance judgment literature that exists for real world environments. These studies quite uniformly demonstrate veridical performance across a large range of distances when full visual cues are available [19,25,29,41,43]. However, it is frequently the case that not all visual signals available in the real world can be reasonably reproduced in the virtual world. It is an open question as to which of these signals are necessary to enable accurate spatial perception in virtual environments. Jones et al. [14, 16] provide evidence indicating that visual stimulation in the far periphery may be an important source of information when performing spatial judgments. This was an intriguing result as many virtual environments seldom present visual stimulation outside of the near periphery, see Figure 1. However, their work was not conclusive as to the exact manner by which this stimulation provided benefit. This document details a series of five experiments intended to expand upon the work described by Jones et al. [14, 16] and to further inform the nature of the benefit provided by peripheral stimulation.

2 Related Work

It has long been established that vision is a strong guide to action, especially with regard to our movements within an environment [9, 10, 11, 41]. As such, visually guided movements, including walking, jumping, pointing, or reaching, are generally well calibrated with regard to our spatial understanding of the world around us [8, 10, 18, 29, 32, 41]. This connection between movement and vision has led to a considerable body of research devoted to how we judge distances under varying viewing restrictions. Cutting and Vishton [6, 7] offer an informative introduction to the basic visual cues indicative of distance and their effectiveness across three ranges of distances: personal, action, and vista space. An often used alternative nomenclature for these spaces is respectively near, medium, and far field. These labels are effectively equivalent, with near field distances being those within arm's length, medium field being distances extending to roughly 30m, and the far field representing all distances beyond 30m.

There is no lack of prior work in the field of distance judgments as visually guided movements are an important part of our everyday lives. In real world viewing conditions, people are quite accurate in judging positions across a large range of distances using a variety of methods [19,24,29,32,33,37,41]. Observers have been shown to accurately judge distances within arm's length using blind reaching tasks. In such tasks, an observer is shown a target displaced in space and then asked to blindly reach to its location [8, 32, 33]. In the medium field, blind walking (otherwise referred to as visually directed walking) is commonly used as a means of reliably measuring distance judgments out to roughly 20 meters with remarkable accuracy [19, 20, 29, 41]. A modest sampling of real world blind walking results can be seen in Figure 2. In blind walking, observers are typically shown a target located along the ground plane at a given distance. They then either close their eyes or have their vision blocked and walk until they feel as though they are standing at the target distance. The walked distance is used as an indication of the perceived distance to the target. Variants of this technique, such as imagined walking and triangulated walking, have been shown to provide similar results [18, 44].

In head-mounted virtual environments, however, a curious trend of underestimated distances has been documented [5, 13, 14, 15, 19, 25, 30, 32, 33, 37, 38, 40, 44, 45, 46, 47]. Many of these studies employ one or more of the psychophysical measurement techniques described above. The magnitude of the underestimation varies appreciably from one set of results to the next. Many researchers have attempted to offer speculative explanations. Unfortunately, no consensus has been reached as to why these underestimations occur in virtual environments. Some have investigated the possibility that improper modeling of the display device's optical and geometric characteristics is to blame, but found only somewhat negligible effects on distance estimation errors [17, 22]. This is a reasonable concern as proper modeling of the optical characteristics of the image producing system (the display device) and the image receiving system (the human eye) is necessary to produce an image of the virtual environment that would result in a retinal projection comparable to that of an equivalent real world environment. Considerable effort, however, has been invested in calibration techniques that ensure that these parameters are controlled [2, 28, 31, 42]. Other work has indicated that humans are ca-

- *J. Adam Jones is with the University of Southern California, Institute for Creative Technologies. E-mail: jadamj@acm.org.*
- *J. Edward Swan II is with the Mississippi State University, Department of Computer Science & Engineering. E-mail: swan@acm.org.*
- *Mark Bolas is with the University of Southern California, Institute for Creative Technologies. E-mail: bolas@ict.usc.edu.*

Manuscript received 13 September 2012; accepted 10 January 2013; posted online 16 March 2013; mailed on 16 May 2013.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

1077-2626/13/$31.00 © 2013 IEEE Published by the IEEE Computer Society

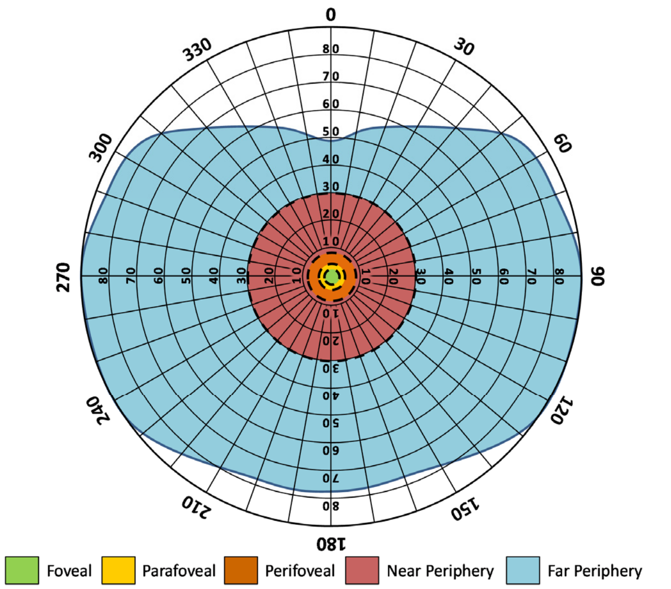


Fig. 1. Regions within the human binocular field of view. Adapted from Taylor [39], Boring et al. [3], and Strasburger et al. [36].

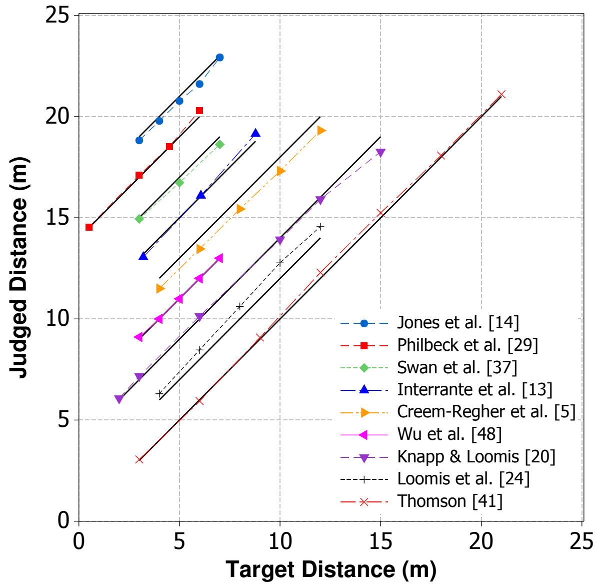


Fig. 2. Mean distance judgments from several real world blind walking experiments. Individual series of judgments have been vertically offset for ease of viewing. Each is presented with a bold, black line indicative of ideal performance.

pable of tolerating a certain amount of miscalibration in the modeling of their physical and display characteristics [4]. Another often speculated candidate is the variable depth of focus in the real world which triggers the ocular accommodative response [6]. Variable focus does not usually exist in virtual environments due to technical limitations of the fixed optical systems typically used in head-mounted displays. This results in a decoupling of the ocular accommodative response from other depth cues available in a virtual environment. To somewhat mitigate this factor, many head-mounted displays have fixed focal distances at or near optical infinity as the accommodative response of most humans is negligible beyond roughly 2 meters [6]. This, of course, reduces bias introduced by accommodative mismatch for distances at 2 meters and beyond, but not for closer distances. For most medium field virtual environments, this is an acceptable compromise and leads one to discount accommodation as a source of the oft cited underestimations. Others have speculated that the limited field of view of many head-mounted displays is a possible culprit. However, there exists conflicting results regarding field of view as the contributing factor. Work by Wu et al. [48] indicated that distance judgments in the real world suffered as the vertical, but not horizontal, field of view was restricted. However, Wu et al. [48] also found that this effect could be countered by allowing observers to pan their view vertically along the ground plane. However, both Knapp and Loomis [20] and Creem-Regher et al. [5] found no effect of field of view on distance judgments. Other work by Jones et al. [14] found evidence that indicated that a small field of view virtual environment could produce improved distance judgments when real world optical flow was added to the lower, far periphery. However, they were unable to explain why this was the case. A follow-up study by Jones et al. [16] attempted to determine if this far peripheral optical flow was causing observers to recalibrate their gait, enabling them to move more accurately in the virtual environment. They found that gait did seem to be a partial contributing factor but that it was insufficient to explain all the observed improvements. They speculated that the peripheral stimulation may have served as an additional reference to the location of the ground plane relative to the observers' eye position.

The findings described by Jones et al. [14, 16] are intriguing as the area within the visual field where the stimulation was provided is outside of that typically considered in most virtual environment experiments. The binocular field of view for most humans spans a range of $180^\circ \times 120^\circ$. Figure 1 shows a polar representation of the angular expanse of the typical, forward looking, binocular field of view. This figure is adapted from that described by Taylor [39] which is itself based on that described by Boring et al. [3]. Most current head-mounted displays seldom provide fields of view larger than 60°. As can be seen in Figure 1, this is a substantially small portion of that which would be available under real world viewing conditions. The remainder of our field of view is not without purpose. A large body of research in the visual sciences has been dedicated to studying differences in the detection of visual stimuli in the periphery. For instance, motion detection thresholds in the periphery are much higher than for central vision [23,27]. The ability to resolve minute variations in luminance in low light conditions is significantly better in the periphery [36]. Though detail resolution is quite low, peripheral vision provides finer temporal resolution than central vision [12]. Reports, dating back as far as the late 1800s, claim that our peripheral vision is used to aid in orientation and obstacle avoidance [1, 9]. Other studies have shown that when vision in the periphery is restricted, in either the real or virtual world, our ability to understand our surroundings and judge spatial relationships suffers greatly [1, 14, 16, 48]. These are fundamental features of how the periphery influences visually driven actions. However, the vast majority of the periphery is unavailable in most virtual environments.

Before continuing further, it is important to clarify the terminology to be used when discussing the periphery. For these purposes, we borrow and somewhat adapt the terminology described by Strasburger et al. [36]. We refer to foveal vision as the area within the radius of 2.5°, parafoveal extending from 2.5° to 4.5°, and perifoveal extending from 4.5° to 8.5°. We will collectively refer to these areas as central vision. The area beyond central vision extending to a radius of 30° we refer to as near periphery. All areas beyond a radius of 30° will be referred to as far periphery. A visual representation of these areas is shown in Figure 1.

3 Experiment I: Light Bar Stimulation

The work described by Jones et al. [14] demonstrated substantial improvements in distance judgments as participants were exposed to varying levels of peripheral visual information near the lower edge

Fig. 3. The virtual environment used in Experiments I through IV.

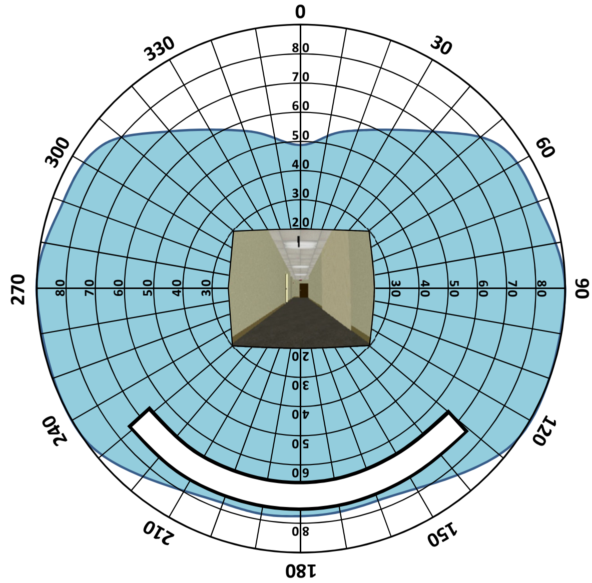


Fig. 5. Placement of virtual environment and light bar within the field of view. The shaded region denotes the extent of the typical human's field of view.

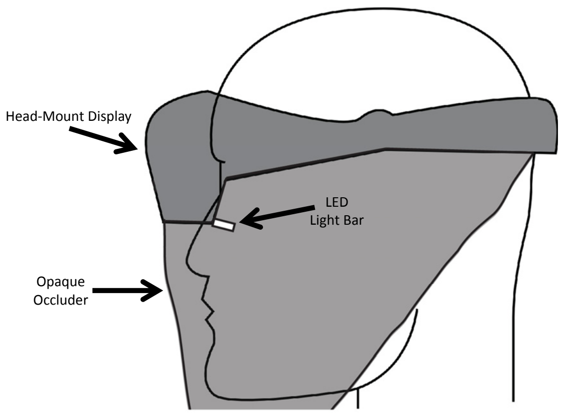


Fig. 4. Display, occluder, and light bar used in Experiment I.

of the observers' natural field of view. The speculative explanation offered for these results was that participants were using peripheral optical flow as a means of either correcting their movements or internalized scale of the environment. However, one of the conditions described in these experiments brings this explanation into question. Specifically, the *VR Partially Occluded with Attention* condition described by Jones et al. [14] provided remarkably little visual information, yet it yielded significant improvements in distance judgments. In the *VR Partially Occluded with Attention* condition, participants were provided with low fidelity optical flow in near darkness at the extreme lower edge of their periphery while viewing a small field of view virtual environment. Could it be possible that this improvement came from a source other than optical flow? Experiment I aimed to answer this question by providing visual stimulation but no optical flow in the observers' lower, far periphery.

3.1 Method

Eight participants (*4 males, 4 females, mean age: 25*) with normal or corrected to normal vision were recruited for this experiment. All participants were naive to the purposes of the experiment. Participants viewed a virtual environment through an NVIS nVisor ST60 head-mounted display. This display offers horizontal, vertical, and diagonal fields of view of $48^\circ \times 40^\circ \times 60^\circ$. The virtual environment, shown in Figure 3, consisted of a high fidelity model of a real world hallway which spanned 1.82m in width and 16.53m in length from the participants' viewing position. The virtual environment was realistically lit and textured to match its real world counterpart. Participants wore earphones that played continuous white noise in order to prevent auditory influences from affecting distance judgments. Instructions were communicated to the participants via a wireless microphone system. Participants performed blind walking as a means of judging distances to previously viewed objects placed along the ground plane. After each blind walking trial, the virtual environment was visually muted and participants were guided back to their starting position. Though no visual information was presented during the return walk, to maintain methodological consistency between this study and those described by Jones et al. [14, 16], participants were instructed to keep their eyes open until they were returned to the starting position. The object used to indicate target distances was a white, wireframe pyramid with a square base of 23.5cm and an apex of 23.5cm. Target distances ranged from 3 to 7 meters in 1 meter increments. Each distance was presented three times resulting in a total of 15 trials. Presentation order of target distances was determined by a restricted random shuffle, with the restriction criterion that no target distance could be presented twice in immediate succession. In Jones et al. [14], participants were exposed to varying levels of real world peripheral stimulation through a small gap between the display and the participants' faces. For this experiment, two stationary white LED light bars were placed in this gap, one corresponding to each of the display's eyepieces, as shown in Figure 4. Each light bar was constructed from a length of white plastic conduit with a single white LED placed in one end. The LED was powered by direct current from a small battery. The light bars were powered on after each participant donned the head-mounted display, and they remained illuminated for the duration of the experiment. The remainder of the periphery was fully occluded using an opaque, black cloth. Figure 5 shows the arrangement of the virtual environment and light bar in terms of their position within the field of view. All areas within the field of view, excepting the display area and light bars, were completely darkened. This *Light Bar* condition was intended to provide light stimulation in the same area as described by Jones et al. [14] but without the influence of optical flow. This condition should be no more spatially informative than having no stimulation in

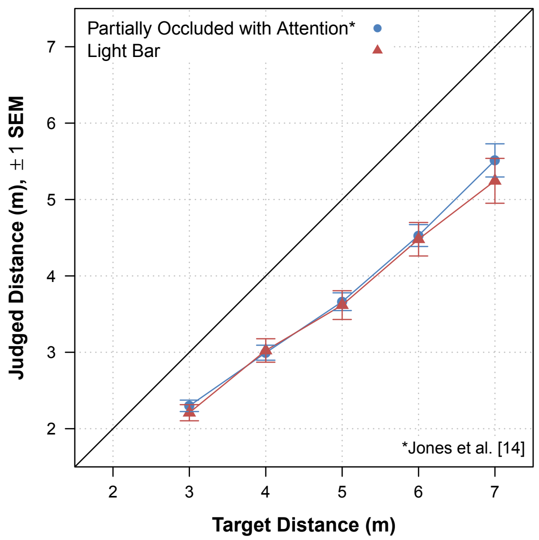


Fig. 6. Mean distance judgments from Experiment I and the *VR Partially Occluded with Attention* condition from Jones et al. [14].

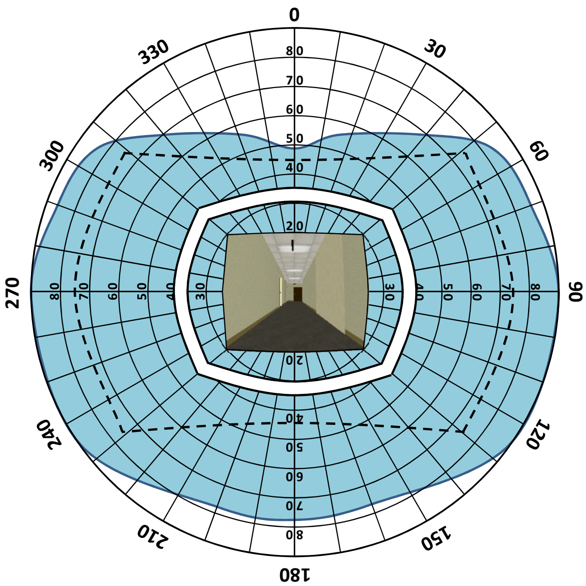


Fig. 7. Placement of the virtual environment and peripheral stimulation within the field of view for the *Frame* condition.

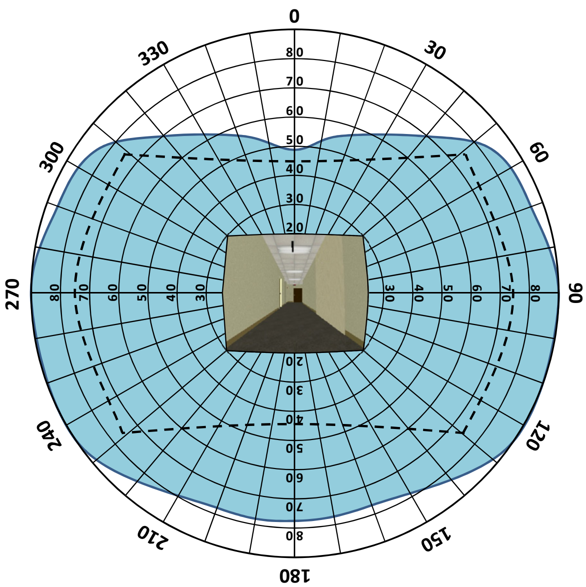


Fig. 8. Placement of the virtual environment within the field of view for the *No Frame* condition.

the periphery. Jones et al. [14] demonstrated that when participants viewed a virtual environment with the far peripheral fully occluded, they judged distances to be only 63.8% of their actual length. Given the similarity of these two conditions, one would expect no substantial difference. It is important to note that for this experiment the virtual environment, physical location, head-mounted display, and procedures exactly matched those used in Jones et al. [14].

3.2 Results

The accuracy of distance judgments in this and the following experiments are analyzed in terms of percent distance defined as the distance walked during the blind walking procedure divided by the target distance. An accurate distance judgment, therefore, would result in a percent distance of 100% while underestimations will result in a value less than 100% and overestimations higher than 100%. All judgments were analyzed using a mixed model analysis of variance (ANOVA). Figure 6 shows the main findings from this experiment. Participants in the *Light Bar* condition averaged 74.24% accuracy. When comparing this to the performance of participants in the *VR Partially Occluded with Attention* condition (75.79%) from Jones et al. [14] who were supposedly benefitting from optical flow, we find that they do not significantly differ ($F(1,14)$=0.048, p=0.831). Also, the improvements seen in the *VR Partially Occluded with Attention* condition were found to significantly increase as a function of time. However, this trend did not significantly express itself in the *Light Bar* condition ($F(2,14)$=2.210, p=0.147). Instead, the improved performance was more generally distributed over the course of the experiment. These are confusing results as the *Light Bar* condition essentially added no information to the scene, yet participants judged distances as accurately as those who were supposedly benefitting from optical flow indicative of their movements.

4 Experiment II: Peripheral Frame Stimulation

The results from Experiment I show that participants gained at least some benefit from the presence of a stationary light source in the lower, far periphery. However, is this benefit limited to the lower periphery? Would participants glean even more benefit from also receiving stimulation across the top and sides of their periphery? Experiment II aimed to answer these questions by introducing a *Frame* condition where white light stimulation was provided more broadly across the participants' far periphery.

4.1 Method

For this experiment, a Fakespace Wide5 head-mounted display was used. This display has horizontal, vertical, and diagonal fields of view of 150°×88°×150°. This enabled stimulation to be graphically rendered in the far periphery. To keep the view of the virtual environment consistent with Experiment I and those described by Jones et al. [14], the field of view of the NVIS nVisor ST60 was simulated by only using the central 48°×40°×60° portion of the screen area to display the virtual environment. A white frame was also displayed in the periphery at 80°×73°×84° and extended inward 5°. The vertical edges and corners of the frame fell outside the area of stereo overlap within the HMD's field of view, preventing convergence cues indicative of the frame's depth. The real world that would otherwise be visible in the far periphery was fully occluded using an opaque, black cloth that covered the sides and bottom of the display. The placement of the virtual environment and frame is illustrated with the full extent of the head-mounted display's field of view denoted by a dashed line in Figure 7.

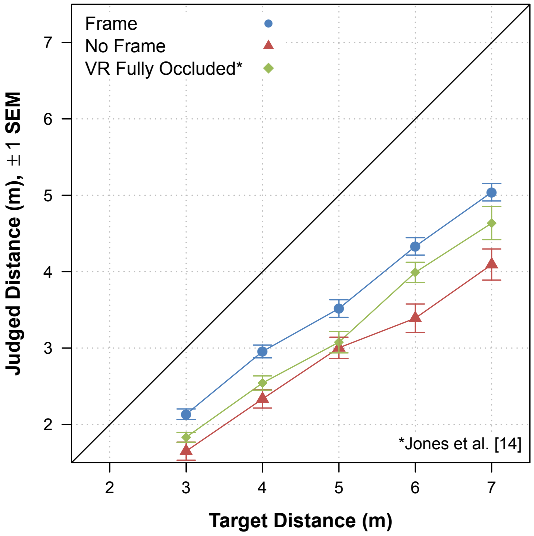


Fig. 9. Mean distance judgments from Experiment II and the *VR Fully Occluded* condition from Jones et al. [14].

Otherwise, the methods used in this experiment exactly matched those described in Experiment I. Since the display device used in this experiment differed from that of Experiment I and Jones et al. [14], an additional condition was performed without the peripheral frame. This *No Frame* condition, illustrated in Figure 8, was intended to serve as a control condition and establish baseline performance for this head-mounted display. Aside from the change in display device, this condition exactly matched the *VR Fully Occluded* condition described by Jones et al. [14]. A group of 15 participants with normal or corrected to normal vision were recruited for this experiment. All participants were naive to the purposes of the experiment. Eight participants experienced the *Frame* condition (*5 males, 3 females, mean age: 27*) while seven experienced the *No Frame* condition (*7 males, mean age: 27*).

4.2 Results

Figure 9 depicts the main findings from this experiment. Participants in the *No Frame* condition judged distances with an accuracy of 58.37% and did not significantly differ from those in the *VR Fully Occluded* condition, which averaged 63.8% ($F(1,13)$=0.823, p=0.381). This indicates that simulating the smaller field of view of the nVisor ST60 inside the Wide5's larger display area provided mutually comparable distance judgments. Participants that experienced the *Frame* condition judged distances with an accuracy of 71.90%, which was significantly more accurate than their *No Frame* counterparts, ($F(1,13)$=6.123, p=0.028). Additionally, the *Frame* condition did not significantly differ from the *Light Bar* condition from Experiment I ($F(1,14)$=0.119, p=0.736). These results seem to indicate that adding visual stimulation to the periphery is sufficient to provide a substantial improvement in distance judgments as expressed by blind walking.

5 Experiment III: Effect of Periphery on Scale

Blind walking is typically considered to be a reliable method for gauging an observer's perceived egocentric distance to a given point in the surrounding environment. However, it is important to understand that blind walking consists of essentially two components: visuospatial and locomotive. Participants use their vision to encode the spatial location of a target and then express this encoded location by walking to the target's position. Blind walking, as a distance measure, relies on the assumption that an observer's vision and motor responses are well calibrated spatially. When a factor, such as peripheral stimulation, alters blind walking behavior, it can be unclear as to whether observers have changed their spatial understanding or simply altered their walking behavior. Jones et al. [16] attempted to disambiguate this relationship but found mixed results indicating that high fidelity optical flow in the far periphery did seem to have an impact on participants' gait while lower fidelity optical flow had none. They report, however, that both high and low fidelity optical flow significantly improved blind walking performance in virtual environments. This leaves open the possibility that peripheral stimulation may be modifying the participants' spatial representation of the environment. However, this could not be clearly determined with the experimental design utilized by Jones et al. [16]. In order to resolve this point of uncertainty, the current experiment aimed to determine whether or not participants were experiencing a change in the perceived scale of the environment as a result of the added peripheral stimulation.

5.1 Method

Ten participants (*8 males, 2 females, mean age: 27*) with normal or corrected to normal vision were recruited for this experiment. All participants were naive to the purposes of the experiment. Participants were shown the same head-mounted virtual environment through the simulated $48^\circ \times 40^\circ \times 60^\circ$ field of view used in Experiment II. The participants were asked to freely look about the environment until they felt they had a good sense of the size of the virtual space. When participants indicated their readiness, they were removed from the head-mounted display and walked a short distance to a screen depicting a projected, static image of the virtual environment as seen from their prior observation position. The projected image spanned an area of 0.67m $\times$ 0.56m on the projection screen and was positioned at the participants' approximate eye height. Participants stood 4 meters from the projection screen and were instructed to walk towards the projected image until they felt as though the size of the environment in the projected image matched that seen in the head-mounted display. Once the participants indicated they were standing at the distance where the projected image matched the scale of the head-mounted environment, their distance from the projection screen was recorded. This procedure was repeated six times, three with the presence of the peripheral frame (*Frame*) and three without (*No Frame*). Presentation order was alternated between subjects with half of the participants first experiencing the presence of the peripheral frame and the remainder without.

Gogel's Theory of Phenomenal Geometry [10] theorizes that any point within visually perceived space can be described in terms of three fundamental variables: 1) perceived direction relative to the observer, 2) perceived egocentric distance from the observer, and 3) perceived position of the observer. One can easily see these variables expressed in many commonly used spatial judgment tasks. For instance, perceived direction is a factor in triangulation based tasks, such as pointing. Other tasks, such as blind walking, provide an estimation of direct egocentric distance. We suggest that the visual scale task described here is driven by the third variable, perceived position of the observer. This judgment task operates by having observers reproduce their approximate viewing position within the virtual environment, as driven by the visual angle subtended by the projected image. This method is somewhat similar to natural perspective estimation procedures, but retains an explicit coupling between observation position and visual scale [34, 35].

5.2 Results

A repeated measures analysis of variance was conducted to determine if the alternating presentation order of the peripheral frame had an effect on responses. No significant effect of presentation order was detected ($F(1,8)$=2.495, p=0.153). Participants stood significantly further from the projection screen (1.293m) when having previously viewed the virtual environment with the peripheral frame than when having viewed the same environment without the frame (1.078m) ($F(1,8)$=7.556, p=0.025). Using the visual angle occupied by the back face of the hallway for each of these viewing distances, the change in visual scale of the virtual environment can be calculated. The width of the virtual hallway was 1.82m and extended 16.53m from the participants' viewing position. The width of the hallway's end occupied 3.86° of visual angle in the *Frame* condition and 4.61° in the

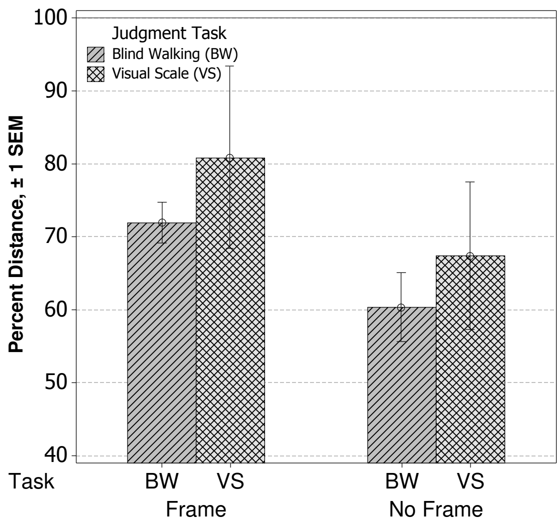


Fig. 10. Comparison of blind walking and visual scale tasks.

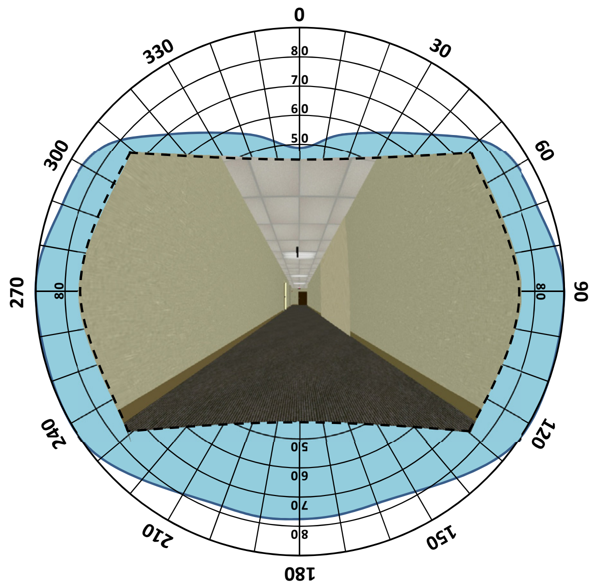


Fig. 11. Placement of the virtual environment within the field of view for the *Full Field of View* condition.

No Frame condition. Using the hall width as a scaling factor, we can now calculate the distance from the observer to the end of the hallway. It was found that the hallway appeared to be 13.49m long in the *Frame* condition and 11.28m long in the *No Frame* condition. These correspond to 80.8% and 67.4% of the hallway's actual length, respectively. Recall that the blind walking results indicated that participants in the *Frame* condition judged distance as 71.9% of their actual distance while participants in the *No Frame* condition judged distances as 58.4% of their actual distance. As can be seen in Figure 10, the results of the scaling task very closely mimic the trends seen in the previously discussed blind walking judgments. Interestingly, the percentage difference between the *Frame* and *No Frame* conditions in both distance judgments (13.5%) and scale judgments (13.4%) are quite identical. When comparing the percent distance judged in the blind walking task with the calculated percent distance from the scaling task, no significant difference could be found (*Frame*: $F(1,16)$=0.400, p=0.536; *No Frame*: $F(1,16)$=0.339, p=0.568). These results are consistent with the hypothesis that the addition of visual stimulation in the far periphery is not simply causing participants to walk farther but is introducing a change in the perceived scale of the virtual environment.

6 Experiment IV: Large Field of View Environment

Experiments I and II demonstrated appreciable improvements in the performance of distance judgments when participants viewing a virtual environment through a relatively narrow field of view were provided with a small amount of static peripheral stimulation outside of the display area. Previous work has indicated that larger fields of view allow individuals to perform distance judgments more accurately [48]. How does the improvement in performance seen with added peripheral stimulation compare to the performance of participants viewing a virtual environment through a very large field of view? Experiment IV sought to answer this question by introducing a *Full Field of View* condition that presented the same virtual environment seen in Experiments I and II but in a very large field of view.

6.1 Method

Eight participants (*5 females, 3 males, mean age: 33*) with normal or corrected to normal vision were recruited for this experiment. These participants viewed the virtual environment using the same Fakespace Wide5 display used in Experiment II. However, for this experiment, the display's full $150^\circ \times 88^\circ \times 150^\circ$ field of view, illustrated in Figure 11, was used to display the virtual environment. Otherwise, the apparatus and procedures for this experiment exactly matched those described in Experiments I and II.

6.2 Results

Participants in the *Full Field of View* condition judged distances (75.96%) significantly more accurately than participants who utilized the smaller field of view in the *No Frame* condition (58.37%) ($F(1,13)$=8.852, p=0.011). As can be seen in Figure 12, these participants did not significantly differ in average performance from their *Light Bar* (74.24%) and *Frame* (71.90%) counterparts (*Light Bar*: $F(1,14)$=0.059, p=0.811; *Frame*: $F(1,14)$=0.848, p=0.373). However, if we look at the *Full Field of View* distance judgments as a function of target distance, we see a pattern of significantly improved judgments as target distances increase ($F(4,28)$=5.971, p=0.001). This pattern does not manifest itself in either the *Light Bar* or *Frame* conditions (*Light Bar*: $F(4,28)$=0.645, p=0.635; *Frame*: $F(4,28)$=0.623, p=0.650). A possible explanation is that the larger field of view provides substantially more texture information along the ground plane than is available in the smaller field of view, giving more information with increasing distance.

7 Experiment V: Real World Occlusion

Most normally sighted individuals have a binocular field of view that spans roughly $180^\circ \times 120^\circ$. The field of view of the environment seen by participants in Experiment IV, $150^\circ \times 88^\circ$, is approaching that of the real world. It has been well documented that individuals judge distances with near 100% accuracy when performing blind walking in the real world [19,20,24,29,41]. This includes the *Real World* condition described by Jones et al. [14], where participants averaged 93.5% accuracy. Many blind walking experiments in virtual environments compare performance against a real world control condition. In this paradigm, participants experiencing the virtual environment are usually disallowed from seeing anything outside the display's narrow field of view and are typically presented with no visual stimulus between trials. However, participants in the real world condition may experience insufficient occlusion or occlusion that is not directly comparable to the experimental condition. Since minute, nonspecific visual stimulation between trials, particularly with regard to the periphery, has until recently not been considered as a possible influence on the outcome of blind walking tasks, such asymmetries between the experimental and control conditions may have gone unnoticed and undocumented. However, the results from Jones et al. [14] and Jones et al. [16] show

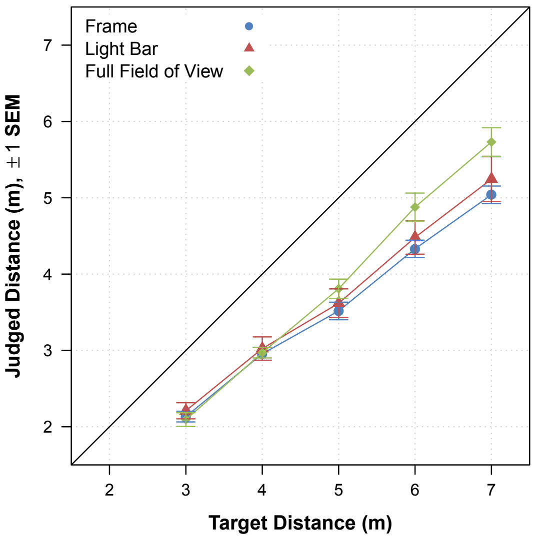


Fig. 12. Mean distance judgments from Experiment IV.

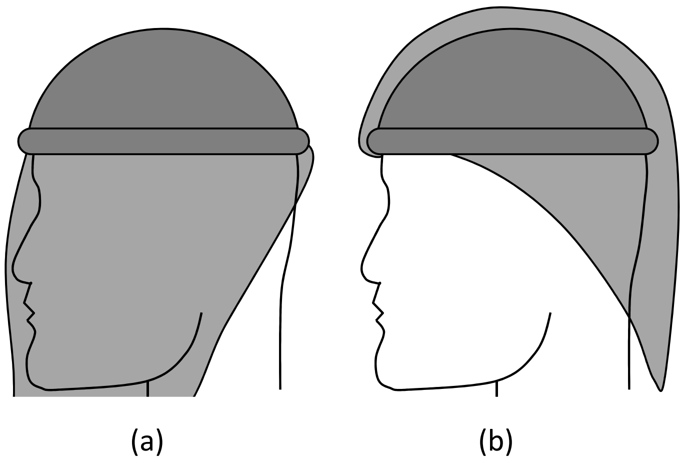


Fig. 13. Occluder used for Experiment V shown in (a) occluding and (b) viewing configurations.

that the presence of visual stimulation, even in near total darkness, during the return portion of the blind walking procedure quite strongly influences distance judgments. This brings up an interesting question regarding comparability between experimental and control conditions where occlusion procedures are incongruent or insufficient. Experiment V aimed to determine if rigorously disallowing visual stimulation during the return walk using occlusion procedures that very closely mimicked those used in Experiments I through IV would alter performance in an otherwise unencumbered real world condition. To test this possibility, we introduce a *Real World Occluded* condition.

7.1 Method

Eight participants (*5 males, 3 females, mean age: 22*) with normal or corrected to normal vision were recruited for this experiment. As depicted in Figure 13, participants wore a veil constructed of opaque, black cloth attached to a hard plastic cap. This veil, when pulled over the top of the head, allowed participants to have a full, unimpeded view of the real world environment. Participants performed the blind walking task with the veil in this position. When participants completed their judgment walk, the veil was lowered before they returned to their starting position. The experiment took place in the real world location that was modeled for the previous virtual environments. Otherwise, the experimental procedures matched those used in the previous experiments.

7.2 Results

Interestingly, the *Real World Occluded* participants judged distances with only 81.74% accuracy. This is considerably lower than has been reported by many studies examining real world blind walking [13, 14, 15, 19, 20, 24, 29, 41]. Work by Kuhl et al. [21] has shown that significant individual variations exist between participants in blind walking tasks, with roughly one third of participants being predisposed to underestimating real world distances relative to a population mean of 96% accuracy. This value is consistent with the 93.5% accuracy seen in the *Real World* control condition reported by Jones et al. [14]. In the current experiment, 95% confidence intervals were calculated per participant revealing that 7 of the 8 participants significantly underestimated distances relative to the mean performance reported by Kuhl et al. [21] and 6 of the 8 participants relative to the mean reported by Jones et al. [14]. Though it is possible that these results could be the product of randomly recruiting an above average number of participants predisposed to underestimation, this seems unlikely given the substantial number of participants exhibiting underestimation. This suggests that depriving participants of visual information during the return walk appreciably hinders the participant's ability to judge distances with blind walking. As can be seen in Figure 14, participants in the *Real World Occluded* significantly underestimated distances relative to those in Jones et al. [14]'s *Real World* condition ($F(1,16)$=10.905 p=0.004). Perhaps more interesting is the lack of a significant difference between the performance of the *Real World Occluded* participants and their *Full Field of View* counterparts ($F(1,14)$=1.254, p=0.282). This suggests that participants in a large field of view virtual environment may, in fact, judge distances as accurately as participants in an equivalently occluded real world viewing condition. A comparison of performance in the *Real World Occluded* and the *Light Bar* and *Frame* conditions reveals that participants also did not significantly differ in their distance judgments (*Light Bar*: $F(1,14)$=1.063, p=0.321; *Frame*: $F(1,14)$=4.364, p=0.055). The improvement in distance judgments with increasing target distance observed in Experiment IV did not manifest itself in this experiment ($F(4,28)$=1.106, p=0.373).

8 Discussion

Somewhat surprisingly, the results from Experiment I indicate that at least some of the performance improvements seen by Jones et al. [14] may not have been a result of optical flow, but from the presence of light in the far periphery. This is a curious result as the simple presence of a light source does not, in itself, convey any information about the surrounding environment or the observer's position within the environment. However, Experiments II and III continue to support the hypothesis that adding peripheral stimulation to a virtual environment improves one's ability to judge distance and scale. It is especially interesting that the effect of the peripheral stimulation expresses itself equally in terms of both egocentric distance and visual scale judgments. As with any psychophysical phenomenon, observers' responses are colored both by their internalized perceptual representations and the physical manifestation of those representations [26, 43]. For instance, changes in blind walking responses could be caused not only by changes in perceived target distance, but also by changes in the way one walks. Using a single technique, such as blind walking, cannot reveal which influence has caused a change in an observer's response. We suggest that by observing changes in the blind walking and visual scale tasks, both of which are driven by one's spatial understanding, that the peripheral stimulation provided in these experiments

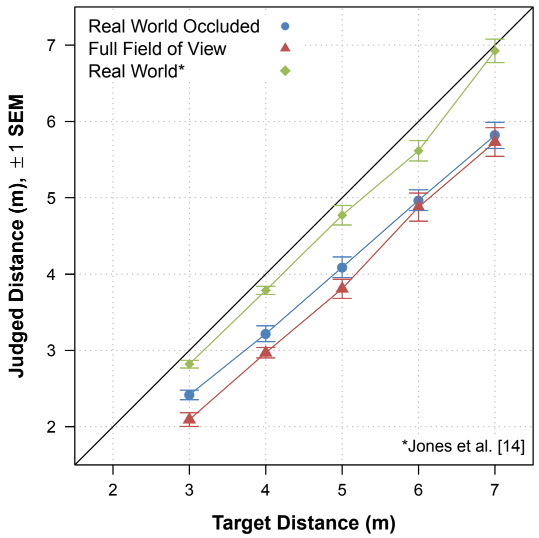


Fig. 14. Mean distance judgments from Experiments IV, V, and the *Real World* condition from Jones et al. [14].

is influencing behavior on a perceptual level.

Prior work by Jones et al. [16] speculated that peripheral stimulation may affect the internal representation of an environment's spatial scale but lacked sufficient evidence to substantiate this hypothesis. The present work, however, indicates that peripheral stimulation can cause an observer to perceive a change in scale when, in fact, there was none. A counterargument could be made that perhaps participants were using the peripheral stimulation as a reference to the screen edge and thereby consciously judging the virtual environment as only being a small portion of a much larger screen area. However, the comments of the participants after the experiment seem to counter indicate this hypothesis. All participants in the *Light Bar* condition were aware that the bar was present in the periphery, but this is expected as the device was manually powered on at the beginning of the experiment in a manner that was difficult to ignore. However, most of the participants in this condition commented that they stopped noticing the bar's presence. Some of the participants claimed the light bar disappeared as the experiment progressed, assuming that it had been remotely disabled by the experimenters. This was not the case, and the light bar was confirmed to be illuminated at the end of each experimental session. Perhaps even more interesting are the comments of the participants in the blind walking *Frame* condition. Since the peripheral frame was graphically rendered, there were no obvious physical cues to explicitly direct participants' attention to the periphery. Only two out of the eight participants remarked that the frame was present when queried. The remaining participants reacted skeptically when informed that the frame was, in fact, present in their periphery. This implies that the majority of participants in the *Frame* condition were not consciously aware of the frame's presence, making cognitive efforts an unlikely explanation for the observed improvement in distance judgments.

A rather large amount of research has focused on distance judgments in virtual environment with results varying substantially from one experiment to the next. One possible explanation for the variation seen across the literature is uncontrolled peripheral stimulation. A remarkably small amount of stimulation seems to be necessary to trigger an effect [14, 16]. From the perspective of a researcher wanting to perform tightly controlled visual experiments in virtual environments, this is a troublesome thought as minor variations in peripheral stimulation may inadvertently contaminate experiments. Achieving sufficient occlusion can be a nontrivial task. In the process of investigating methods of occlusion, several common techniques were implemented [13, 17, 29, 35, 37, 45]. These included opaque goggles, foam rubber baffles, blindfolds, night-masks, cloth shrouds, and darkening the physical environment by extinguishing light sources. It was found that most reasonable implementations of these methods still allowed at least a small amount of light to be detected by observers. However, if instead we look through the eyes of a practitioner wanting to enhance the performance of a virtual environment, these findings are quite encouraging. If we consider the *Light Bar* condition from Experiment I, simply adding two large, low resolution pixels to the lower periphery of a small field of view display enabled its users to perform as if they were utilizing a much larger field of view. Additionally, if the surrounding real world environment is sufficiently similar to the virtual world, Jones et al. [14]'s findings indicate that mixing a virtual environment with real world periphery may even allow performance comparable to that observed in the real world. This is quite exciting when considering that large field of view displays can be difficult to engineer and are often prohibitively expensive.

An intriguing proposition raised by Experiment V is that some virtual environment blind walking studies may be using an inappropriate control condition. A typical scenario for a blind walking study using a head-mounted virtual environment involves observers viewing a target located at some distance. The observers then close their eyes and have the virtual environment visually muted. During this period without vision, the observers then walk to the apparent location of the target. The observers then return to the starting position, still without vision, and the procedure repeats. In the real world, however, it is substantially more difficult to visually mute the scene. Even when observers close their eyes, they quite often have a visual awareness of their position in the world from light passing their closed eye lids. This is made obvious by simply closing your eyes while walking through a well-lit environment. It quickly becomes apparent that you can resolve subtle shifts in light and shadow even with eyes shut. Even very small luminance changes viewed through the black cloth occluder used by Jones et al. [14] was sufficient to alter blind walking behavior. As demonstrated by Jones et al. [14, 16] and reiterated here, near darkness is not equivalent to darkness. A very small amount of light seepage is capable of altering distance judgments. Finding an occluding configuration that both prevents light seepage from the surrounding environment and is not overly cumbersome for the observer is, in fact, quite challenging. For our purposes, we found the most effective occlusion to be achieved by using a tightly woven, black cloth veil that drapes over the shoulders and lays flat against the chest.

The results of Experiment V seem to indicate that, when the return portion of the blind walking task is made comparable, accuracy of blind walking in the real world and the large field of view virtual environment cease to differ. However, it is worth noting that despite the lack of comparability under these specific circumstances, the typical real world blind walking procedure is likely a more valid comparison for other real world based judgments. Though real world performance drops when visual stimulation is removed between trials, it has yet to be definitively seen if performance will improve when comparable stimulation is added in the virtual environment. The results shown by Jones et al. [14, 16] imply that this may be the case. However, the stimulation they provided during the return walk was strictly peripheral, leaving this possibility only partially addressed.

9 Conclusion & Future Work

The results of the preceding experiments can be summarized in terms of three main findings:

- Adding static, white light stimulation to the far periphery invokes a positive change in perceived scale of a small field of view virtual environment as expressed by blind walking and visual scale estimation,
- The effect of peripheral stimulation on blind walking performance was not significantly different than that observed in a very large field of view virtual environment,
- Rigorous occlusion during the return portion of the blind walking procedure impairs blind walking performance in an otherwise unencumbered real world environment.

Substantial future work is required in order to fully understand the mechanisms that underlie the perceived change in scale caused by stimulating the far periphery. Since the peripheral stimulation provided no additional information about the virtual environment or the observer's position within the environment, we speculate that perhaps providing visual signal, even if nonspecific, more broadly across the field of view may serve sufficient to activate perceptual mechanisms associated with spatial understanding. This, however, is a strictly speculative explanation that requires further investigation. Additionally, it is as yet unclear whether there are draw backs associated with the peripheral stimulation as presented here. For instance, does having a light source in the far periphery impact an observer's sense of presence or immersion in a virtual environment? Based on the previously discussed feedback from participants in Experiment II, we speculate that impact on presence or immersion would be minimal. However, this is beyond the scope of the current work and necessitates further investigation.

In the authors' prior work, which served as the basis for this series of experiments, typical implementations of the real world blind walking procedure resulted in performance very much comparable to those reported in many other studies, ranging from roughly 94% to 97% accuracy [14, 15, 37]. The real world occlusion procedures used in Experiment V were the first in this series of studies to both rigorously occlude and closely mimic the occlusion procedures used in the experimental conditions. The surprising amount of underestimation seen in this condition emphasizes the importance of comparability between experimental and control conditions. However, it has yet to be determined how little light is necessary to facilitate a change in distance judgments in the real world. It is our suspicion that very minor light seepage around the edges or through occluding apparatuses is quite possibly responsible for at least some of the variation in distance judgments reported through the contemporary literature. However, further investigation is needed to determine if this is, in fact, the case.

Acknowledgments

The projects or efforts depicted were or are in part sponsored by the U.S. Army. The content or information presented does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred. This material is in part based upon work supported by the National Science Foundation under grants IIS-0713609 and IIS-1018413. The authors would also like to express their gratitude to the Mississippi State University Institute for Imaging and Analytical Technologies and the University of Southern California Institute for Creative Technologies for use of their equipment and facilities. The authors would additionally like to thank Stephen R. Ellis and Carrick C. Williams for their thoughtful insights. Special thanks to Thai Phan, Gurjot Singh, Kennth Moser, Chunya Hua, and Sujan Anreddy for assistance in data collection.

References

[1] P. L. Alfano and G. F. Michel. Restricting the field of view: Perceptual and performance effects. *Perceptual and Motor Skills*, 70(1):35–45, 1990.

[2] M. Axholt, M. Skoglund, S. O'Connell, M. Cooper, and S. Ellis. Parameter estimation variance of the single point active alignment method in optical see-through head mounted display calibration. In *IEEE Virtual Reality Conference*, pages 27–34, march 2011.

[3] E. G. Boring, H. S. Langfield, and H. P. Weld. *Foundations of psychology*. John Wiley & Sons, New York, 1948.

[4] G. Bruder, A. Pusch, and F. Steinicke. Analyzing effects of geometric rendering parameters on size and distance estimation in on-axis stereographics. In *Proceedings of the ACM Symposium on Applied Perception*, pages 111–118, Los Angeles, CA, USA, 2012.

[5] S. H. Creem-Regehr, P. Willemsen, A. A. Gooch, and W. B. Thompson. The influence of restricted viewing conditions on egocentric distance perception: Implications for real and virtual indoor environments. *Perception*, 34:191–204, 2005.

[6] J. E. Cutting. How the eye measures reality and virtual reality. *Behavior Research Methods*, 29:27–36, 1997. 10.3758/BF03200563.

[7] J. E. Cutting and V. P. M. *Perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth in Handbook of Perception and Cognition*, pages 69–117. Academic Press, San Diego, CA, USA, 1995.

[8] S. R. Ellis and B. M. Menges. Localization of virtual objects in the near visual field. *Human Factors*, 40(3):415–431, 1998.

[9] E. Fuchs. *Text-Book On Ophthalmology*. D. Appleton & Company, New York, 1899.

[10] W. C. Gogel. *The Analysis of Perceived Space in Foundations of Perceptual Theory*, pages 113–182. Elsevier Science Publishers, 1993.

[11] W. C. Gogel and J. D. Tietz. Oculomotor adjustments in darkness and the specific distance tendency. *Perception & Psychophysics*, 13:284–292, 1973.

[12] E. Hartmann, B. Lachenmayr, and H. Brettel. The peripheral critical flicker frequency. *Vision Research*, 19:1019–1023, 1979.

[13] V. Interrante, B. Ries, and L. Anderson. Distance perception in immersive virtual environments, revisited. *Virtual Reality Conference, IEEE*, 0:3–10, 2006.

[14] J. A. Jones, J. E. Swan II, G. Singh, and S. R. Ellis. Peripheral visual information and its effect on distance judgments in virtual and augmented environments. In *Symposium on Applied Perception in Graphics and Visualization*, pages 29–35, Toulouse, France, 2011.

[15] J. A. Jones, J. E. Swan II, G. Singh, E. Kolstad, and S. R. Ellis. The effects of virtual reality, augmented reality, and motion parallax on egocentric depth perception. In *Symposium on Applied Perception in Graphics and Visualization*, pages 9–14, Los Angeles, USA, 2008.

[16] J. A. Jones, J. E. Swan II, G. Singh, S. Reddy, K. Moser, C. Hua, and S. R. Ellis. Improvements in visually directed walking in virtual environments cannot be explained by changes in gait alone. In *Proceedings of the Symposium on Applied Perception*, pages 11–16, Los Angeles, USA, 2012.

[17] F. Kellner, B. Bolte, G. Bruder, U. Rautenberg, F. Steinicke, L. M., and R. Koch. Geometric calibration of head-mounted displays and its effects on distance estimation. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):589–596, 2012.

[18] E. Klein, J. Swan, G. Schmidt, M. Livingston, and O. Staadt. Measurement protocols for medium-field distance perception in large-screen immersive displays. In *Virtual Reality Conference, 2009. VR 2009. IEEE*, pages 107 –113, march 2009.

[19] J. M. Knapp. *The Visual Perception of Egocentric Distance in Virtual Environments*. PhD thesis, University of California, Santa Barbara, Santa Barbara, California, 1999.

[20] J. M. Knapp and J. M. Loomis. Limited field of view of head-mounted displays is not the cause of distance underestimation in virtual environments. *Presence*, 13(5):572–577, 2004.

[21] S. A. Kuhl, S. H. Creem-Regehr, and W. B. Thompson. Individual differences in accuracy of blind walking to targets on the floor. *Journal of Vision*, 6(6):726, 2006.

[22] S. A. Kuhl, W. B. Thompson, and S. H. Creem-Regehr. Hmd calibration and its effects on distance judgments. *ACM Transactions on Applied Perception*, 35(9):19, 2009.

[23] D. Levi, S. Klein, and P. Aitsebaomo. Detection and discrimination of the direction of motion in central and peripheral vision of normal and amblyopic observers. *Vision Research*, 24(1):789–800, 1983.

[24] J. M. Loomis, J. A. Da Silva, N. Fujita, and S. S. Fukusima. Visual space perception and visually directed action. *Journal of Experimental Psychology: Human Perception and Performance*, 18(4):906–921, 1992.

[25] J. M. Loomis and J. M. Knapp. *Virtual and Adaptive Environments: Applications, Implications, and Human Performance*, chapter 2, pages 21–46. CRC Press, Mahwah, NJ, 2003.

[26] J. M. Loomis and J. W. Philbeck. *Measuring Spatial Perception with Spatial Updating and Action*, pages 1–43. Psychology Press, New York, 2008.

[27] S. McKee and K. Nakayama. The detection of motion in the peripheral visual field. *Vision Research*, 24(1):24–32, 1984.

[28] C. B. Owen, J. Zhou, A. Tang, and F. Xiao. Display-relative calibration for optical see-through head-mounted displays. *Mixed and Augmented Reality, IEEE / ACM International Symposium on*, pages 70–78, 2004.

[29] J. Philbeck, A. Woods, J. Arthur, and J. Todd. Progressive locomotor recalibration during blind walking. *Attention, Perception, & Psychophysics*, 70:1459–1470, 2008. 10.3758/PP.70.8.1459.

[30] A. R. Richardson and D. Waller. Interaction with an immersive virtual environment corrects users' distance estimates. *Human Factors*, 49(3):507–517, 2007.

[31] J. P. Rolland, W. Gibson, and D. Ariely. Towards quantifying depth and

size perception in virtual environments. *Presence: Teleoperators and Virtual Environments*, 4(3):24–49, 1995.

[32] G. Singh, J. E. Swan II, J. A. Jones, and S. R. Ellis. Depth judgment measures and occluding surfaces in near-field augmented reality. In *Symposium on Applied Perception in Graphics and Visualization*, pages 149–156, Los Angeles, California, USA, 2010.

[33] G. Singh, J. E. Swan II, J. A. Jones, and S. R. Ellis. Depth judgments by reaching and matching in near-field augmented reality. In *Poster Compendium of the IEEE Virtual Reality Conference*, pages 165–166, Irvine, CA, USA, 2012.

[34] F. Steinicke, G. Bruder, and S. Kuhl. Realistic perspective projections for virtual objects and environments. *ACM Transactions on Graphics*, 30(5):112, 2011.

[35] F. Steinicke, G. Bruder, S. Kuhl, P. Willemsen, M. Lappe, and K. H. Hinrichs. Natural perspective projections for head-mounted displays. *Visualization and Computer Graphics, IEEE Transactions on*, 17(7):888–899, 2011.

[36] H. Strasburger, I. Rentschler, and M. Juttner. Peripheral vision and pattern recognition: A review. *Journal of Vision*, 11(5):1–82, 2011.

[37] J. Swan, A. Jones, E. Kolstad, M. Livingston, and H. Smallman. Egocentric depth judgments in optical, see-through augmented reality. *Visualization and Computer Graphics, IEEE Transactions on*, 13(3):429–442, may-june 2007.

[38] J. Swan, M. Livingston, H. Smallman, D. Brown, Y. Baillot, J. Gabbard, and D. Hix. A perceptual matching technique for depth judgments in optical, see-through augmented reality. In *Virtual Reality Conference, 2006*, pages 19–26, march 2006.

[39] J. H. Taylor. *Vision in Bioastronautics Data Book*, pages 611–665. Scientific and Technical Information Office – NASA, 1973.

[40] W. B. Thompson, P. Willemsen, A. A. Gooch, S. H. Creem-Regehr, J. M. Loomis, and A. C. Beall. Does the quality of the computer graphics matter when judging distances in visually immersive environments. *Presence: Teleoper. Virtual Environ.*, 13:560–571, October 2004.

[41] J. A. Thomson. Is continuous visual monitoring necessary in visually guided locomotion? *Journal of Experimental Psychology: Human Perception and Performance*, 9(3):427–443, 1983.

[42] M. Tuceryan and N. Navab. Single point active alignment method (spaam) for optical see-through hmd calibration for ar. In *Augmented Reality, 2000. (ISAR 2000). Proceedings. IEEE and ACM International Symposium on*, pages 149 –158, 2000.

[43] M. Wagner. The metric of visual space. *Perception & Psychophysics*, 38(6):483–495, 1985.

[44] P. Willemsen, M. B. Colton, S. H. Creem-Regehr, and W. B. Thompson. The effects of head-mounted display mechanics on distance judgments in virtual environments. In *Proceedings of the 1st Symposium on Applied perception in graphics and visualization*, APGV '04, pages 35–38, New York, NY, USA, 2004. ACM.

[45] P. Willemsen, M. B. Colton, S. H. Creem-Regehr, and W. B. Thompson. The effects of head-mounted display mechanical properties and field of view on distance judgments in virtual environments. *ACM Trans. Appl. Percept.*, 6:8:1–8:14, March 2009.

[46] P. Willemsen and A. Gooch. Perceived egocentric distances in real, image-based, and traditional virtual environments. In *Virtual Reality, 2002. Proceedings. IEEE*, pages 275–276, 2002.

[47] B. G. Witmer and J. Sadowski, Wallace J. Nonvisually guided locomotion to a previously viewed target in real and virtual environments. *Human Factors*, 40:478–488, 1998.

[48] B. Wu, T. Ooi, and Z. He. Perceiving distance accurately by a directional process of integrating ground information. *Nature*, 428:73–77, Mar 2004.

Orientation Perception in Real and Virtual Environments

J. Adam Jones, Jonathan E. Hopper, Mark T. Bolas, and David M. Krum

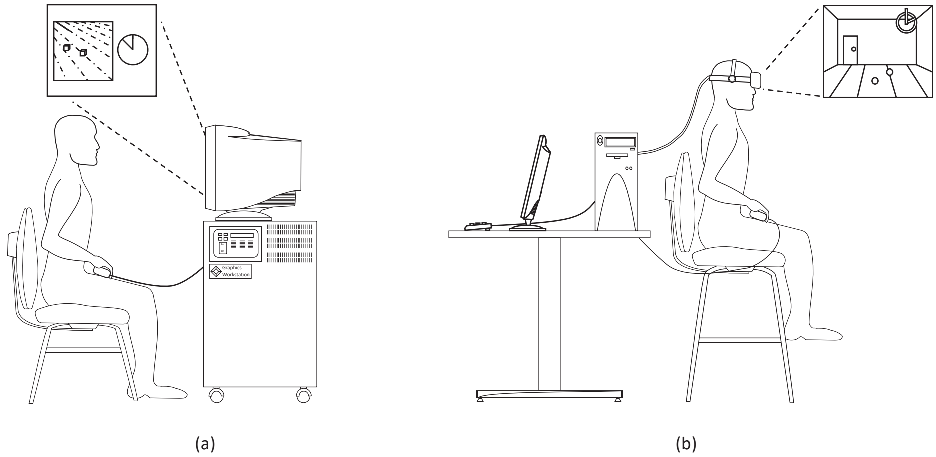


Fig. 1. (a), The setup used by Tharp and Ellis [50]. This image is derived from their setup descriptions and figures. The stimuli are presented as cubes. The "ahead" direction was indicated using a bold line. Judgments are reported by using a mouse to adjust a clock-like widget. (b), The setup used in the current work. The stimuli are presented as spheres and judgments are made on the widget. This setup is based on, but not a replication of, that used by Tharp and Ellis.

Abstract— Spatial perception in virtual environments has been a topic of intense research. Arguably, the majority of this work has focused on distance perception. However, orientation perception is also an important factor. In this paper, we systematically investigate allocentric orientation judgments in both real and virtual contexts over the course of four experiments. A pattern of sinusoidal judgment errors known to exist in 2D perspective displays is found to persist in immersive virtual environments. This pattern also manifests itself in a real world setting using two differing judgment methods. The findings suggest the presence of a radial anisotropy that persists across viewing contexts. Additionally, there is some evidence to suggest that observers have multiple strategies for processing orientations but further investigation is needed to fully describe this phenomenon. We also offer design suggestions for 3D user interfaces where users may perform orientation judgments.

Index Terms—Virtual Environments, Perception, Spatial Orientation, Visual Orientation

1 Introduction

Spatial perception in virtual environments has been well-studied over the past several decades with much of this work focusing on distance perception [5,17,21,22,25,28,29,31–33,35,36,39,49]. However, there have been significant efforts to understand *egocentric orientation* perception as well. The majority of this work has focused specifically on the how users judge their position as they either virtually or physically navigate an environment [38,40–42]. These works have a wide variety of applications, including virtual locomotion and redirected walking, and are undeniably important. However, less work has focused on *allocentric orientation*.

• *J. Adam Jones & Jonathan E. Hopper are with High Fidelity Virtual Environments Lab (Hi5 Lab) and the Department of Computer & Information Science at the University of Mississippi. E-mail: jadamj@acm.org.*
• *Mark T. Bolas is with Microsoft.*
• *David M. Krum is with the Institute for Creative Technologies - Mixed Reality Lab at the University of Southern California.*

Manuscript received 10 Sept. 2018; accepted 7 Feb. 2019.
Date of publication 17 Feb. 2019; date of current version 27 Mar. 2019.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TVCG.2019.2898798

When discussing navigation in virtual environments, position and rotation are sometimes collectively referred to as spatial orientation. However, in the context of the current work, we make a distinction between these and use the term *orientation* to refer specifically to *rotational* relationships. Furthermore, we emphasize that the current work focuses on *allocentric orientation*. Allocentric orientation refers to the orientation of objects within a frame of reference external to the observer.

There is substantial research examining *egocentric orientation*, the orientation of objects judged relative to the observer. Specifically, there has been much progress in the areas of navigating and way-finding. These typically involve measuring judgments of egocentric orientation in the form of observers indicating the direction to their starting position after having traversed a path in a virtual environment [38,42]. There are tasks, however, that require an observer to make judgments of objects in an allocentric frame of reference. Such tasks include air-traffic control, coordination of ground maneuvers relative to terrain features, and plotting the course of travel for a vehicle. In order to better understand how observers perform these kinds of tasks, we conducted the following set of experiments. An early work specifically examining such a task was conducted by Tharp and Ellis [50]. Their work aimed to study orientation judgments in scenarios where objects are positioned relative to each other while suspended above the ground plane (e.g., aircrafts). They found large, systematic errors in orientation judgments

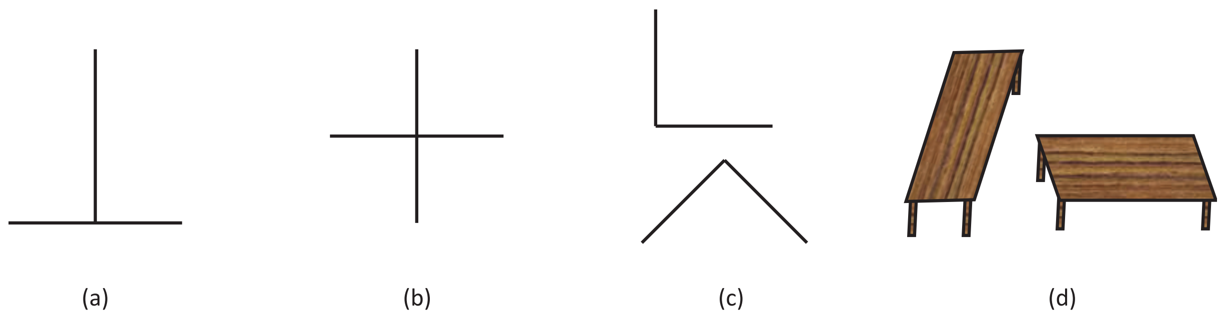


Fig. 2. Various depictions of the vertical-horizontal illusion. In (a,b), both the vertical and horizontal lines are the same length. In (c), we see two right angles. The bottom angle is often perceived as not as "right" as the top angle. In (d), we see Shepard's Tables [44]. Both table tops are the same size and shape.

when using a perspective display. It is unclear, however, whether or not similar errors occur in an immersive virtual environment.

In an attempt to answer this question, we adapt the general procedure presented by Tharp and Ellis [50] for use in an immersive virtual environment. Orientations were indicated using a clock-like widget, shown in Figure 1-a, where one arm was locked to represent the "ahead" direction while the other arm represented the ahead-relative angle formed between two objects as viewed on a monitor. Though similar, the current study examines the orientation judgment of objects *on* the ground plane using an immersive virtual environment. Judgments were recorded using a mouse and input widget similar to that of Tharp and Ellis, see Figure 1-a,b.

The goals of the work described here evolved over the course of the experiments. As aforementioned, our initial aim was to determine whether or not users of an immersive virtual environment were able to accurately judge allocentric orientation. Upon finding a series of systematic errors, we investigated whether or not field of view, a well-known factor affecting spatial judgments, altered this pattern of errors. This was not the case. We then tested several real-world conditions to determine if the observed errors existed outside the context of a virtual environment and, if so, under what viewing conditions. The results of these experiments provide insight into a radial anisotropy and further informs us on allocentric orientation perception, especially with regard to perspective views. Additionally, we discuss some design considerations for both real and virtual interfaces where users may need to perform orientation-dependent tasks.

2 Related Work

2.1 Anisotropy and Visual Orientation

The term *visual anisotropy* refers to cases where an isometric transformation is not preserved between the physical world and its corresponding visual percept. In other words, distortions are perceived where none exist. This has long been a topic of particular interest in the study of perceptual illusions. Much of the work in this area has focused on *linear anisotropies*, such as the famous *Vertical-Horizontal Illusion* [43]. In this illusion, vertical and horizontal lines of equal length intersect, forming a 90° angle. The vertical line is often judged to have a larger apparent length, Figures 2-a,b. Another common example is *Shepard's Tables* [44], Figure 2-d. In this illusion, the vertically oriented table appears to be significantly longer than its horizontally oriented counterpart when, in fact, they are of the same length.

Kunnapas [27] investigated a horizontal linear anisotropy where extents were over estimated toward the nasal edge of the field of vision. It was hypothesized that this may be an artifact of the asymmetric shape of the human field of vision. This was tested by having subjects view stimuli through "synthetic apertures" or, in contemporary terms, restricted fields of view (FOV). This was tested using a bisection task where observers were provided two reference posts and asked to position a third post such that it evenly bisected the extent between the references. It was found that small, symmetric FOVs resulted in increased accuracy while asymmetric FOVs resulted in biased judgments. Ritter [43] performed a similar task and found no effect of FOV shape on perceived asymmetries.

However, modern studies in virtual environments have indicated that the effects of FOV restriction can be difficult to detect [22, 23, 30]. In particular, Jones et al. [22] found that the size of an observer's FOV affected the perception of visual scale. Interestingly, this was found be to of the same magnitude as blind walking compression seen in the same study. Similar effects were seen by Li et al. [30] when examining distance compression with a consumer-grade VR display. Nilsson et al. [34] found that FOV size impacted the perceived naturalness of an observer's walking speed. In all of these cases, larger FOVs led to more accurate judgments. If similar FOV-related scaling errors affect orientation judgments, this could be problematic for the accurate judgment of object orientations within a scene.

There has been a long history of work examining similar illusions affecting angular judgments. An early account of angular anisotropies was made, perhaps not surprisingly, by Herman von Helmholtz [51]. Helmholtz's study was rather informal and little was documented of his procedures. He reported that vertically oriented, rightward-facing, right triangles were not judged as such. Instead, to achieve a judgment of a right triangle the vertical leg of the triangle had to be rotated roughly 18° counterclockwise. He further noted that angles appeared least like right angles when oriented at 45°. This is illustrated in Figure 2-c.

A detailed investigation of the relationship between the vertical-horizontal illusion and orientation was conducted by Shipley et al. [46]. They compared the subjective length of line pairs where each line was oriented at some angle between 0° and 90°. It was found that the apparent length of a line did continuously vary as its orientation changed. While vertically oriented lines were found to be consistently judged as longer than horizontal lines, diagonal lines of 60° were judged to be the longest.

Pollock and Chapanis [37] performed a similar study, but tested more orientations as well as lines of differing initial lengths. They found the same general pattern of errors as Shipley et al. [46] with the exception that diagonal lines of roughly 30° were judged the longest. They also noted that the effect was not universal across their subjects. Though most subjects showed some level of systematic error, a few appeared to exhibit no discernible anisotropy.

Goldmeier [15] wrote extensively on factors related to the subjective similarity between shapes. It was noted that right angles exhibited "singularity" in that they were easily detectable but only when in a vertical-horizontal configuration, top of Figure 2-c. Diagonal orientation, bottom of Figure 2-c, was observed to hinder an observer's sensitivity in detecting right angles.

Ferrante et al. [14] presented a study aiming to further inform the findings of Goldmeier et al. [15]. Their experiment aimed to determine if the perception of right angles was relative to a retinal or environmental reference frame. They examined the verbal descriptions of observers viewing a right angle and several distractor images as seen with the head either upright or tilted 45°. Subjects described the images they saw verbally. They found that right angles were only reported as such when they were oriented with their sides parallel to the vertical and horizontal axes of the retina regardless of head tilt. It is important to note, however, that the psychophysical measurement used in this study was highly subjective, providing only a loose threshold of discrimination.

Work more closely examining the ability to discriminate between two angles was conducted by Snippe and Koenderink [48]. In this study, observers were presented with several reference points in their visual frontal plane and asked to determine whether or not the angles defined by these points subtended the same angle. It was found that for most stimulus configurations, the threshold sensitivity fell between 5° and 10°. However, stimuli oriented at 90° and 180° exhibited significantly lower thresholds.

Selective preference for stimuli oriented along the cardinal directions is not uncommon. For instance, Appelle [1] provides an in-depth review of work focusing on the *oblique effect*. The oblique effect describes a general pattern of decreased performance in visually directed tasks for stimuli oriented at oblique angles. When stimuli are oriented vertically or horizontally, on the other hand, performance in such tasks substantially increases. A broad range of phenomena have been attributed to the oblique effect, including grating acuity and line tilt. This effect is often also associated with increased variability in responses to oblique stimuli.

One example that could be attributed to the oblique effect is work by Dick and Hochstein [7]. They studied the reported tilt of a luminous line against a dark background. The tested stimuli ranged from 0° to 90°. Interestingly, their results indicated that errors were lowest for stimuli oriented at 0°, 45°, and 90°. However, they observed a distinct change in the pattern of responses when subjects were asked not to estimate the tilt, but instead to estimate the time in minutes as though the line represented a clock hand.

Other work examining orientation judgments focuses on the interplay between multiple frames of reference, such as vision and gravity. Judgment of the "up" direction, both relative to gravity and to the head, can be measured in several ways and has been shown to be influenced by the tilt of both the observer and the environment about the roll axis. Barnett-Cowan and Harris [3] showed overestimations of physical tilt up to 11° when participants were in the dark and presented a vector model for determining the magnitude of the influence of both observer and environmental tilt relative to the influence of gravity. While they, unsurprisingly, found gravity to be the defining factor in allocentric tasks and body orientation in egocentric tasks, a significant leftward bias (3.5° for allocentric, 6° for egocentric) was found.

Dyde et al. [8] also studied this relationship. Their work examined perceptual upright (PU), and constructed the oriented character recognition test (OCHART) to measure this subjective direction. The test required subjects to identify a symbol as either a "p" or a "d" when the subject and the background environment were rotated. This data was compared to standard tests used to determine the subjective visual vertical (SVV), and a vector model was fit to both. This produced results similar to those of Barnett-Cowan and Harris, such that gravity dominated the SVV task and body position dominated the PU task. The vector weights for the PU were also predicted by the inverse of the variance of each cue, which was not true of the SVV weights. They speculate variance of the cues might be how the brain determines the weights for its own internal model.

Another such study is that of Barnett-Cowan et al. [2]. They found that participants, when rotated relative to their environment, perceive gravity to tilt in the direction they are rotated. This was found by asking rotated participants to determine whether an object which was rotated about the participant's roll axis and placed near the edge of a table would, if released, fall off the table or right itself. Participants tended to perceive the object as less stable (more likely to fall) when the object was rotated in the same direction as the participant, and vice versa.

Observer rotation has also been shown to affect the perception of distance. Harris and Mander [16] showed that by rotating participants or the environment such that the participant was or seemed to be supine, a line projected at 12ft directly in front of the viewer was perceived to be 9.8% closer. Similarly, a line projected 4ft from the subject was perceived as almost 10.8% closer, though only when the subject viewed the line monocularly, suggesting that, at close range, where binocular depth cues are more salient, such cues are able to supersede the effects of body position.

2.2 Orientation in Virtual Environments

Modern virtual environments are often associated with head-mounted displays (HMDs). However, early virtual environments, as well as some of their modern counterparts, were desktop-based perspective displays. In such setups an observer sees a two-dimensional (2D) projection of a three-dimensional (3D) scene on a monitor. This is essentially a monocular, 3D view where the viewpoint is calculated based on a modeled camera position. Perspective displays were of particular interest for applications such as air-traffic control where the 3D relationship between objects is important [11, 50]. Tharp and Ellis [50] discuss a series of experiments investigating the effectiveness of perspective displays in communicating orbital orientation. A recreation of their display is shown in Figure 1-a. They asked observers to recreate an object's orientation as seen from a perspective view using a 2D input widget. It was found that observers committed consistent, systematic pattern of sinusoidal errors with error approaching zero in the cardinal directions and reaching its peak near the diagonals.

Ellis and Hacisalihzade [10] conducted a followup study exploring a method of correcting biases in exocentric direction judgments seen in Tharp and Ellis [50]. Participants were shown one reference and one target cube on a 3D map, and, given a reference direction, were asked to give the orientation of the target cube relative to the reference direction. In then-previous experiments, participants had exhibited a systematic bias which was attributed to misinterpretations of the perspective display. Ellis and Hacisalihzade introduced a compass rose overlay to the scene with divisions of 15°, 30°, 45°, or 60°, and in half of the test cases, also included an on-screen dial which included the same or different divisions. Results indicated that, as more divisions were added to the rose and dial, smaller errors were made in judgments, until the 15° case, where errors began to increase. Additionally, increasing rose and dial densities resulted in subjects taking longer to make judgments; thus, divisions of 30° were recommended to optimize judgment accuracy and decision time. This result could suggest that the improved performance with the addition of directional references may be the result of cognitive as opposed to perceptual mechanisms.

Ellis et al. [13] performed two additional experiments to determine if the judgment errors that had been previously encountered were the result of misperceptions of the display surface, i.e. the computer screen. A new model had been proposed to explain said errors as the result of misjudgments in the direction from which the reference cube is viewed, and the experiments conducted attempted to alleviate this source of error by utilizing different display methods. The first experiment presented subjects with a stereoscopic version of the previously described orientation judgment test, achieved via anaglyphic presentation, and found that the judgment errors persisted. In the second experiment, the same procedures were recreated in a real environment. Cubes were placed on poles in a parking lot, and participants viewed the scene from a nearby building through windows which subtended 30° or 60°. Neither window size nor flight experience were found to have a significant effect on error; again, the target orientation did significantly affect error. However, both experiments resulted in errors which were, on average, smaller in magnitude than in the previous experiments using perspective displays.

Much spatial orientation work specific to immersive virtual environments has focused on how well an observer can establish and maintain their egocentric position relative to their surroundings while navigating an environment. A canonical example is the work of Riecke et al. [42]. Their work examined the ability of users of an immersive virtual environment to maintain spatial understanding under a variety of viewing conditions. It was found that users of a virtual environment, after having navigated a path, were able to accurately indicate the direction of their starting position by pointing. Their task can be thought of as an implicit measurement of egocentric orientation.

A related study by Ragan et al. [38] examined the effect of amplified head rotation on an observer's ability to navigate, locate objects, and self-orient in both head-mounted and projection-based virtual environments. Their results indicated that observers were able to accurately indicate the egocentric orientation of targets within the virtual environment. Though errors were generally low, they were significantly higher in the HMD. Instead of the pointing method used in Riecke et al. [42], this study used a method where the direction to targets was indicated on a circular input widget. This widget is quite similar to that used by Tharp and Ellis [50] and the current study.

Keeny-Kennicutt and Merchant [24] studied the effectiveness of chemistry education when mediated through the third-person virtual environment Second Life[1]. This environment is visually similar to many third-person-view video games and is effectively a perspective display not unlike those described by Ellis and others [10, 13, 50]. On the surface, it may be difficult to see how this is an application of spatial orientation, but a common task in chemistry is the estimation of bond angles. They found that participants in their study generally performed poorly on bond angle estimation except in the case of 90° bonds. This is somewhat consistent with the effects noted by Appelle [1], Goldmeier [15], and Ferrante et al. [14].

3 Experiment

3.1 Experiment I: Orientation in a Common Virtual Environment

The project originally motivating this work was aimed at studying the performance of spatial judgment tasks in virtual environments under varying FOV configurations. As such, we began our investigation examining allocentric orientation judgments in an immersive virtual environment utilizing a small, but commonly used FOV. We aimed to adapt the procedures and measurements described by Tharp and Ellis [50] but for use in a HMD. We were curious if similar orientation judgment errors as seen in their work would also occur in an immersive virtual environment.

3.1.1 Apparatus and Procedures

For the following experiments, we used a Fakespace Wide5 HMD. This display provides a FOV spanning $150° \times 90°$. Since most HMDs exhibit some form of geometric distortions or other visual errors, it was important to detect and correct these prior to experimentation. Such errors could be caused by a number of factors ranging from optical distortions to prolonged wear-and-tear. We corrected these errors using a manual variant of the calibration method described by Jones et al. [20]. Using a wide angle camera, we were only able to reasonably calibrate the central most $136° \times 88°$ of the FOV. For this reason, no more than this area was illuminated during any of the following experiments. Our investigation began by examining a virtual environment as seen through a $48° \times 40°$ ($60°$ diagonal) aperture, which has been, until quite recently, the most common FOV configuration used in virtual environments research. Pixels outside this area remained black during the experiment. This arrangement is shown in Figure 3. The participants' gaze direction and head position were tracked in real time using a PhaseSpace Fusion motion capture system operating at 240Hz. This enabled the participants to have a correct view that corresponded to their head position at any given point in time. There was no physical restraint of the participants' heads, but it is important to note that they were instructed to keep their view directed toward the stimuli over the course of the experiment.

Participants in this experiment viewed a realistically textured, stereoscopic virtual environment. Participants were presented with two spheres, measuring 20cm in diameter. A red sphere, referred to as the *Reference Sphere*, was located on the ground plane and centered with the participants' forward view at a distance of 4.5m. This sphere was stationary throughout the duration of the experiment. A second,

[1]https://www.secondlife.com

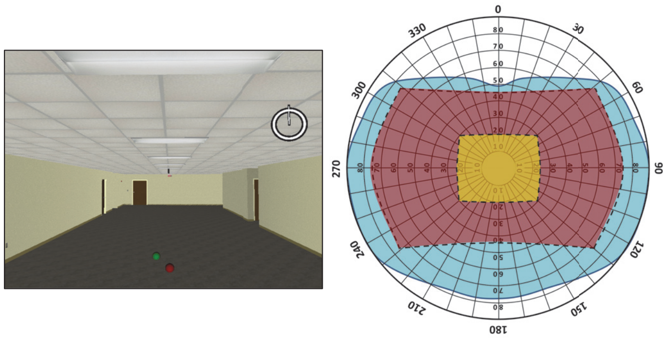


Fig. 3. (a), The virtual environment as seen in Experiment I & II. (b), An illustration of the fields of view used in Experiment I (yellow) and Experiment II (red) plotted on a polar coordinate representation of the typical human's field of vision (blue).

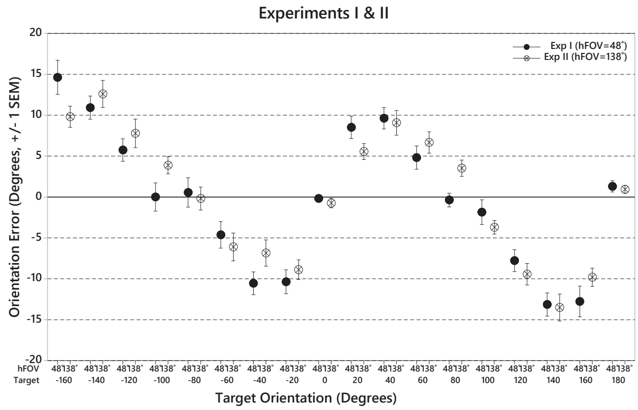


Fig. 4. The combined results from Experiments I and II. A pattern of sinusoidal error is apparent in both experiments. *hFOV* refers to the horizontal field of view for each condition. Error bars represent the Standard Error of the Mean (SEM) for each target orientation.

green sphere, referred to as the *Target Sphere* was positioned along the ground plane at various orbital orientations relative to the Reference Sphere at a radius of 1m. The participants' task was to judge the angle, relative to their forward view, formed by the position of the Target sphere around the Reference sphere. For example, if the Target Sphere was located along the observer's forward view and directly behind the Reference Sphere, its orientation would be 0°. If the Target Sphere was located at the same frontal depth as the Reference Sphere and directly to the right, its orientation would be $+90°$. Instead, if the Target Sphere was located directly to the left of the Reference Sphere, its orientation would be $-90°$.

Participants reported the judged orientation of the Target Sphere using a clock-like widget, shown in Figure 3. This was displayed within the participants' upper, right FOV and was offset from the visual center. The position of the widget was screen stabilized and, as such, was always visible regardless of head movements. This arrangement was visually similar to a head-up display. The widget consisted of two arms, the *ahead* arm, which represented the direction of the participants' forward view, and the *target* arm, which represented the line described by the position of the Reference and Target spheres. The target arm was adjusted using the scroll wheel of a computer mouse, which was held by the participants throughout the course of the experiment. Participants were instructed to imagine that the Reference sphere was located at the center of the widget and that the Target sphere was at the tip of the target arm. They were then instructed to use the scroll wheel of the mouse to adjust the widget until the angle formed by the ahead and target arms was equal to the angle formed by the Reference and Target spheres relative to the forward view. Once the observers were satisfied with their judgment, they were instructed to press the left mouse button to submit their response. At this point a new trial would begin.

We chose to present stimulus orientations ranging from -160° to 180° in 20° increments. This resulted in 18 unique orientations. Each orientation was repeated three times through the course of the experiment, resulting in 54 trials per participant. Presentation order of the trials was determined using a restricted random shuffle with the restriction criterion that no orientation was presented twice in immediate succession.

For this experiment, 10 participants (4 females, 6 males, mean age: 27.2 years) were recruited from the student, intern, and staff population of the University of Southern California. Participants either participated to fulfill a research participation requirement or were compensated $20 for participation. All participants were presented with documentation detailing the experiment before their informed consent was acquired. Participants were also screened by verbal report for epilepsy, visually induced seizures, pregnancy, and severe motion sickness prior to being allowed to participate. Additionally, all participants were screened for normal or corrected to normal vision prior to participation. All procedures were reviewed and approved by the Institutional Review Board of the University of Southern California.

3.1.2 Results

As can be seen in Figure 4, a strong pattern of sinusoidal errors presented itself in the data. The pattern was sufficiently strong as to be easily discernible in the judgments of each individual subject. Generally speaking, the pattern was quite cyclic with peaks very near $\pm 40^\circ$, $\pm 160^\circ$ with errors reaching approximately 15°.

Most standard analysis techniques used in VR spatial perception studies rely on the assumption that the data should adhere, either by default or by correction, to a normal distribution. This is generally a true assumption when working with Euclidian spaces. However, orientations are radial in nature and require directional analysis. Consequently, such techniques are based on the *von Mises distribution*, a circular normal distribution. We employed several directional statistical methods to analyze these data. First, we performed a Watson's U^2 test as described by Davis [6]. This method relies on a rank ordering of directional values in order to determine whether or not a significant portion of the values overlap with each other. We performed this test by pairing each orientation judgment with its target orientation. Thus, this test would reveal whether or not judgments were evenly distributed around the target orientation or if judgments tend to fall to one side. We found that, except for 0°, 80°, and 180°, all judged orientations significantly differed from their respective target orientations ($p < 0.001$). However, since the Watson's U^2 test is based solely on rank ordering of judgments, this may not be a sufficiently sensitive test as it does not strongly account for variability in the data. Consequently, we chose to also compute 99% confidence angles for each set of judgments. Confidence angles are the directional equivalent of confidence intervals seen in traditional linear statistics [6]. This test revealed that all target directions, excepting 0°, $\pm 80^\circ$, $\pm 100^\circ$, and 180°, fell outside the 99% confidence angle for their respective judgments.

The pattern of errors in orientation judgments seen in this experiment are very similar to those seen by Ellis et al. [13] and Tharp and Ellis [50]. Prior speculation was that these errors were likely due to an absence of robust depth cues in their perspective display. The current experiment indicates that similar errors also present in a textured, head-tracked, stereoscopic virtual environment. However, a possible explanation is that the errors could be due to anisotropies introduced by the presence of a frame enclosing the stimuli. In the studies reported by Tharp and Ellis [50], participants viewed virtual stimuli presented on a CRT monitor. Most monitors, especially those contemporary with their study, typically have a housing that encloses the display area. A number of studies have shown that the presence of an enclosing frame tends to alter an observer's judgment of various visual and spatial properties in both 2D and 3D scenes [9, 18, 19, 21, 22, 30]. Though there was no explicit frame present in the current experiment, recall that the FOV in the HMD was restricted to the central $48^\circ \times 40^\circ$. It is possible that the edges of the FOV could serve as an implicit frame and introduce perceptual errors similar to those seen with explicit visual frames [21, 22, 30].

3.2 Experiment II: Orientation in a Large Field of View

In Experiment I, we found that a pattern of sinusoidal errors similar to those seen in a 2D perspective display by Tharp and Ellis [50] exists in a typical virtual environment. However, it is well known that small FOVs and enclosing frames tend to alter spatial judgments under a number of viewing conditions [9, 18, 19, 22, 30, 34]. In particular, work by Jones et al. [21, 22] indicated that increased FOV improves spatial judgments in immersive virtual environments. In the current experiment, we investigate whether or not these errors were, in fact, due to viewing the scene through a small FOV.

3.2.1 Procedures

9 participants (4 females, 5 males, mean age: 27.0 years) were recruited from the student, intern, and staff population of the University of Southern California. Informed consent was acquired from all participants. All procedures were reviewed and approved by the Institutional Review Board of the University of Southern California. Participants either participated to fulfill a research participation requirement or were compensated $20 for participation. The procedures, apparatus, and environments exactly matched those used in Experiment I. The current experiment differed only in that the scene was viewed through a $136^\circ \times 88^\circ$ FOV (red region in Figure 3).

3.2.2 Results

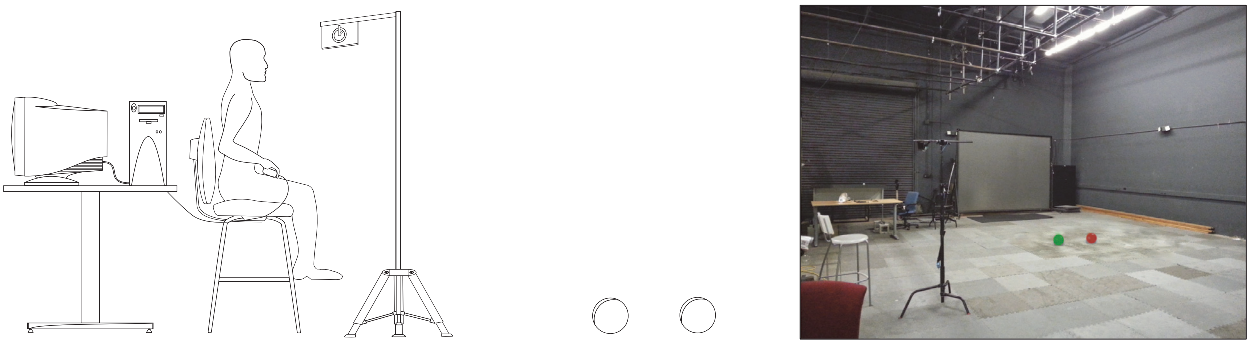
As can be seen in Figure 4, a pattern of sinusoidal errors similar to those seen in Experiment I were found. Errors are very close to zero near 0°, $\pm 80^\circ$, and 180° with peak errors ranging from 8.9° to 13.5° near targets of $\pm 40^\circ$ and $\pm 140^\circ$. A Watson's U^2 test revealed that judgments, excepting those for target orientations of 0°, -80°, and 180°, significantly differed from their respective targets ($p < 0.001$). When using a 99% confidence angle comparison, we find similar results with only 0°, $\pm 80^\circ$, and 180° falling within the confidence angle of their respective judgments. Additionally, when comparing the results of Experiments I and II, we found that their judgments fall within each other's 99% confidence angles at all positions except -160°. Essentially, no other difference was discernible between the results. This implies that FOV or framing effects are unlikely explanations for the observed pattern of errors.

3.3 Experiment III: Real World

Since FOV does not appear to be a contributing factor to the orientation judgment errors seen in the prior experiments, we now ask if the computer generated imagery itself could be causing these errors. It has been found that spatial judgment errors exist in both 3D immersive virtual environments and 2D displays, presumably due to insufficient spatial cues. Could these errors be an artifact of viewing a computer generated scene? We test this hypothesis by closely replicating the procedures of Experiments I and II in the real world.

3.3.1 Procedures

For the current experiment, participants were seated in the same chair with the same equipment, excepting the HMD, as used in Experiments I and II. The floor of the room was covered in foam matting that formed a somewhat regular grid. A representation of the room can be see in Figure 5. In order to prevent this grid from acting as a forward oriented reference, participants were seated such that their forward view was offset from the grid by 20°. Otherwise, the procedures closely mimic those of the previous experiments. A red, spherical balloon, inflated to a diameter of 20cm, was placed at a distance of 4.5m from the participant. This served as the reference sphere and was not moved over the course of the experiment. A green, spherical balloon, also inflated to a diameter of 20cm, acted as the Target sphere.



(a) (b)

Fig. 5. (a), A depiction of the experimental setup for Experiment III. (b), A recreated view of the environment in the room where Experiment III took place.

Inflating the balloons to the specific diameter of 20cm was achieved by constructing a circular, cloth belt with a circumference of 62.8cm and inflating the balloons within the belt until they fit snuggly. Untethered balloons tend to be difficult to control due to their extremely light weight. As such, both balloons also contained roughly 6 to 8 glass marbles which added weight and served to inhibit their movement after being placed on the floor. As before, the Target sphere was placed at one meter radii from the Reference sphere at 20° intervals. To ensure consistent placement, a series of small reference marks were made on the floor's surface that indicated each target position. These marks were not visible from the participants' viewing position as they were only slightly darker than the color of the floor and quite small (roughly 4mm in length). The stimuli could not be programmatically placed in the scene, as is possible in a virtual environment, so an experimenter did this manually. Since the experimenters were not visible in the prior two experiments, participants were asked to briefly close their eyes between each trial while an experimenter placed the stimuli within the room. The experimenter was hidden behind the participants during all other phases of the experiment.

For this experiment, 12 participants (5 males, 7 females, mean age: 36) were recruited from the student, staff, and surrounding community of the University of Southern California. All participants were paid $20 for their participation in this experiment. Participants viewed the same stimuli configurations as in the previous experiments with 18 total target orientations repeated 3 times each. Presentation order was shuffled in the same manner as in the previous experiments. Since the input widget used in the previous experiments was part of the virtual imagery, we could not display it in the same manner in the real world. Instead, a 7 inch liquid crystal display (LCD) panel was mounted on an arm attached to a tripod that extended into the FOV of the participants. The LCD was positioned so that it occupied the same visual angle as the widget seen in the virtual environment. The angular position of the display within the participants' FOV also very closely approximated that of the widget in the virtual environment when gaze was fixed on the Reference sphere. Its position was calculated for an average seated eye height. To maintain minimal visual intrusiveness of the display apparatus, all cables, mountings, and lightly colored surfaces were covered with black gaffer's tape. The pixel resolution of the widget shown on the display was also adjusted to match that of the widget as seen in the previous experiments.

3.3.2 Results

As can be seen in Figure 6, the pattern of sinusoidal errors from the previous experiments no longer holds. However, upon closer examination it was found that the subjects fell into two distinct groups, Figures 7 and 8. Interestingly, both groups exhibited a sinusoidal profile, but they differed in the distribution of error. These groups were unambiguously visible in the data and were subsequently confirmed by performing subject-per-subject Discrete Fourier Transformations (DFT) on mean error. The DFT was used to quantify the primary frequency of oscillation in error that occurred over the range of stimulus orientations. All participants exhibited a primary frequency in error of either 1.13 oscillations per cycle (a long period) or 2.25 oscillations per cycle (a short period). Based on the appearance of the data, we can speculate that the actual frequencies would likely be very close to 1.0 and 2.0 oscillations per cycle, but the frequency resolution of a DFT is limited by the number of samples taken per cycle. In our case, the number of samples per cycle was 18, the number of stimulus orientations presented to a participant. As a result, our frequency resolution was low but sufficient to robustly classify participants.

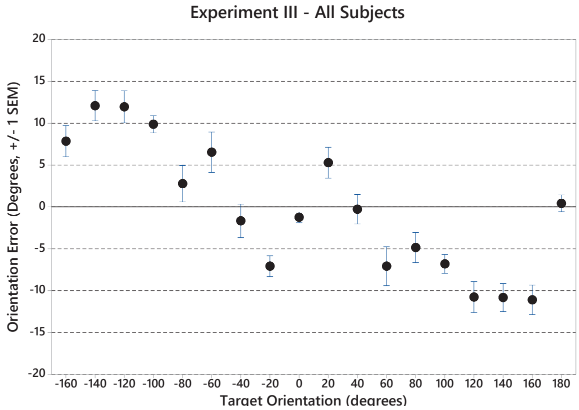


Fig. 6. Judgment error in degrees for all participants. Error bars represent the Standard Error of the Mean (SEM) for each target orientation.

Exactly half of the participants exhibited orientation judgments similar to those seen in Experiments I and II, with errors very close to zero near the cardinal directions (0°, ±90°, 180°) giving the appearance of a short period sine wave. The remaining half of our participants, however, demonstrated a pattern with errors very close to zero only near 0° and 180°. Errors were positively biased for targets with a negative orientation and negatively biased for targets with a position orientation giving the appearance of a long period sine wave. We will refer to

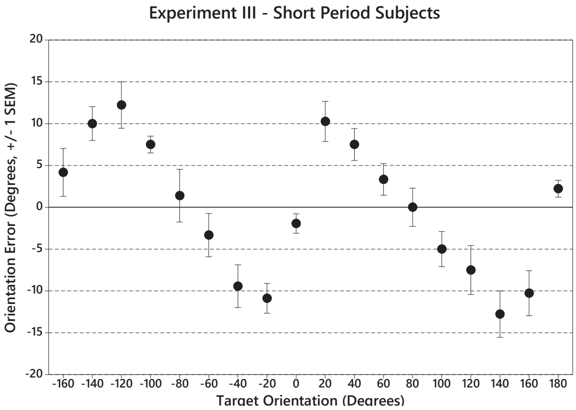


Fig. 7. Judgment error in degrees for participants exhibiting the *Short* pattern. Error bars represent the Standard Error of the Mean (SEM) for each target orientation.

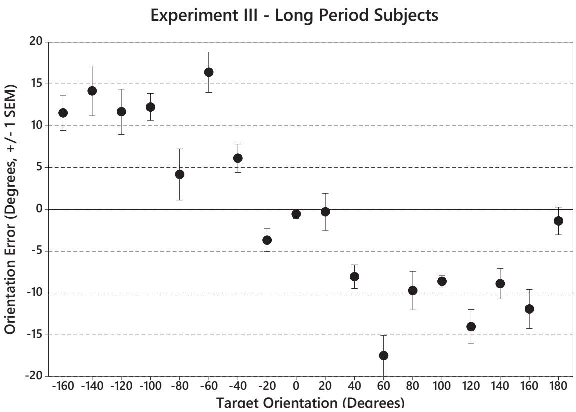


Fig. 8. Judgment error in degrees for participants exhibiting the *Long* pattern. Error bars represent the Standard Error of the Mean (SEM) for each target orientation.

these groups as *Short* and *Long* respectively. We statistically compared these two groups using the Watson's U^2 test to ensure that the observed differences were, in fact, statistically significant. The groups were found to significantly differ at all orientations, excepting 0° and 180° ($p < 0.001$).

When examining the *Short* group with the Watson's U^2 test, we found that all judged orientations significantly differed from their targets at the $p < 0.001$ level excepting 0°, 60°, ±80°, 180°. An examination of 99% confidence angles for the *Short* group revealed that the target angles only fell within the confidence angle for their respective judgments at 0°, −60°, ±80°, −160°, and 180°. The same analysis for the *Long* group revealed that only 0°, ±20°, −80°, 180° fell within their respective confidence angles. When examining the *Long* group with the Watson's U^2 test, we find that all judgments significantly differed from their targets at the $p < 0.001$ level except 0°, −80°, and 180°.

This experiment aimed to determine whether or not errors seen in the prior experiments could be explained by viewing computer generated scenes. However, these results do not fully support that hypothesis. Though the manner of the errors observed in this experiment do begin to diverge from the pattern seen in the previous experiments, the participants' judgments are still quite far from veridical.

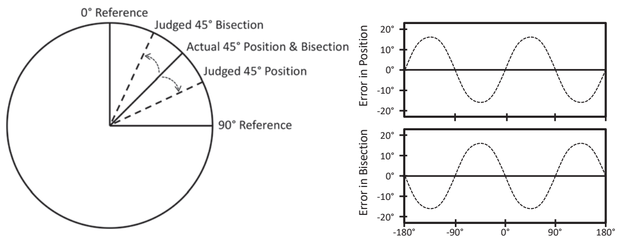


Fig. 9. Assuming that the previously seen errors manifest themselves in Experiment IV, a change in sign should be seen when using a bisection task. The overall pattern would be the same but flipped vertically as illustrated above.

3.4 Experiment IV: Angular Bisection

The previous experiments have shown that the presence of sinusoidal errors in orientation judgments cannot be explained by the absence of stereoscopic vision, limited FOV, or virtual viewing context. Thus far, our experiments have used the same general design and procedures described in Tharp and Ellis [50]. The motivations of their study was to measure, model, and potentially correct errors, specifically with regard to understanding and directing the movements of aircraft. As such, the tasks and measurement methods were similar to those that users may encounter when performing related real world tasks. A result of this was the design of the input widget. From a participant's point of view, the input widget is always located in the visual frontal plane while the stimuli are always located along a perspective plane, or in our case, the ground plane. Thus, observations are being made in one plane while judgments are being given in another. This requires observers to make an internal transformation between the perspective, ground plane and the frontal plane. It would not be unreasonable to question whether such a transformation may introduce biases in reported judgments. In this experiment, we aim to remove this disconnect between stimulus and response by performing both observations and judgments in the same plane. Additionally, it could be argued that the method used in the previous experiments was not fully allocentric since the orientation of the objects were reported relative to the observer's gaze direction. This is not the case for the bisection task as all information necessary to perform the task is encapsulated in the stimulus itself, thus removing the need for an egocentric reference direction.

This experiment utilized a fully allocentric judgment method to test the hypothesis that judgment errors were due to performing observations and judgments in two different planes. We tested this hypothesis using two conditions: one where the clock widget appeared to be laying on the surface of the floor and another where all arms of the clock widget subtended the same visual angle, giving it the appearance of being oriented in the observer's visual frontal plane. We refer to these two conditions as *Perspective* and *Frontal* respectively. If this hypothesis holds, then judgments in both conditions should be close to veridical.

3.4.1 Procedures

We utilized a perceptual bisection task somewhat similar to that used in Kunnapas et al. [27], but adapted for angular bisection. Often bisection tasks require an observer to place a mark between two references. Instead of presenting two reference points to be judged against one's forward viewing direction, we presented participants with a 90° L-shaped widget with its corner positioned on the floor at 4.5m from the participant, see Figure 5. This is referred to as the *reference angle* and is formed by two white lines that were 1m in length. A third, yellow line of the same length was also displayed with one end anchored to the corner of the reference angle. This line is referred to as the *target line*. The participant could control the orientation of the target line using the scroll wheel of a mouse. The entire setup is visually similar to the face of an analog clock where the reference angle is formed

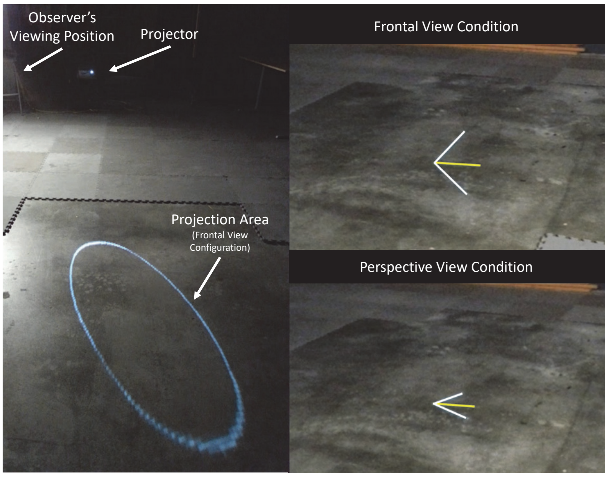


Fig. 10. A view of the experimental environment (left) and stimuli (right) presented in Experiment IV. The lines of the stimuli have been highlighted for clarity.

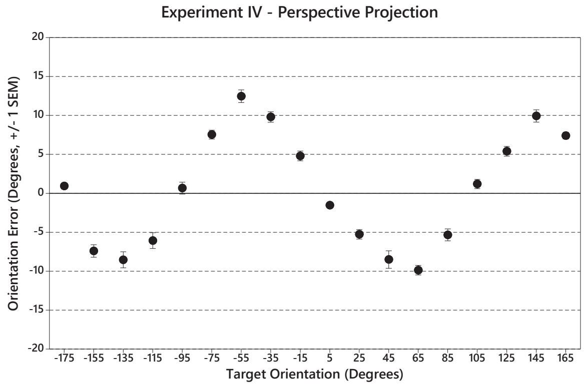


Fig. 11. Bisection error in degrees for the *Perspective* viewing condition. Error bars represent the Standard Error of the Mean (SEM) for each target orientation.

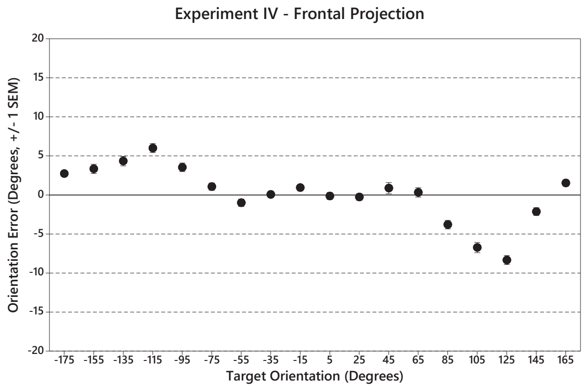


Fig. 12. Bisection error in degrees for the *Frontal* viewing condition. Error bars represent the Standard Error of the Mean (SEM) for each target orientation.

by the hour and minute hands and the target line is the second hand. This method allowed both the stimulus and the judgment to be given in the same plane. The participants were asked to adjust the target line until it bisected the reference angle into two equal sub-angles. Participants were also given the analogy of cutting a large slice of pie (the reference angle) into two smaller, equally sized pieces of pie. Once the participants had adjusted the target line, they would submit their judgment by clicking the left button of the mouse. This experiment took place in the same room described in Experiment III. A digital projector was used to draw the reference and target lines on the floor. The image shown by the projector was calibrated to adjust for the floor's slope and the participant's viewing position using the method described by Jones et al. [20].

As with the previous experiments, bisections were performed at orientations ranging from $-160°$ to $180°$ in $20°$ increments, with each orientation repeated three times for each viewing condition. Presentations were blocked such that one full set of orientation stimuli was presented for one viewing condition, either Perspective or Frontal, and followed by another full set or stimuli from the opposing condition. The initial viewing condition was randomly selected. Presentation order within each block was shuffled using the same method as previously described. Otherwise, presentations occurred in the same manner as described in the previous experiments. An important difference for the bisection task was that the angle the participants were trying to reproduce is the *midpoint* within a $90°$ angle. Thus, the judgment should be offset by $45°$ from the orientation of the L-shaped stimulus. For instance, when the reference angle was at an orientation of $0°$, the target line would bisect it at $45°$.

For this experiment 10 participants (2 females, 8 males, mean age: 28.1) were recruited from the student, staff, and surrounding community of the University of Southern California. All participants were paid \$20 for their participation in this experiment. If the pattern of error seen in the previous experiment was, in fact, due to observing and reporting in two different planes, then the current results should show judgment errors very close to zero. If this is not the case, a similar pattern of sinusoidal errors should appear here as well. This task has an important difference from those in the previous experiments. The adjustment task in Experiments I - III required the participant to reproduce the position of one object relative to another. The bisection task in this experiment, however, requires the participant to instead identify the point of subjective equality between two extents. As such, one would expect to see the similar pattern of judgments in both task types but with a reversal in sign. This relationship is depicted in Figure 9.

3.4.2 Results

As can be seen in Figure 11, a strong pattern of sinusoidal error is apparent in the Perspective condition. A Watson U^2 test indicated that mean judgments for all orientations significantly fell to one side or the other of veridical ($p < 0.001$). Mean judgments also fell outside the 99% confidence angles for each tested orientation excepting $-175°$, $-95°$, $5°$, and $105°$. This finding is quite similar to the previous experiments where errors approached zero near the cardinal directions and peaked along the diagonals. As predicted, the sign of the errors were in the opposite direction seen in the previous experiments.

The errors seen in the Frontal condition, however, cease to adhere to a sinusoidal pattern, see Figure 12. Though not veridical, an overall decrease in error can be seen across all orientations. Mean errors in this condition did not exceed $10°$ and generally stayed well below $5°$. Targets for 12 of the 20 target orientations fell within their respective 99% confidence angles. Interestingly, judgments only significantly differed from their target orientations between $85°$ and $145°$ in both the negative and positive directions. A possible explanation for these errors could be the manner in which this condition was displayed. Since the rotational center was the same for both the Perspective and Frontal conditions, this meant that targets falling beyond $\pm 90°$ had the appearance of being below the ground plane. It has been observed that when virtual geometry is presented behind a real-world surface, its perceived position is biased toward the real surface's position [12, 47]. This would explain why judgments for stimuli beyond $\pm 90°$ were positively biased for negative orientations and negatively biased for positive orientations. However, further experimentation is necessary to test this hypothesis.

4 Discussion

Overall, the main finding of this series of studies is that a strong pattern of sinusoidal error can be seen in both real and virtual environments when judging allocentric orientations. We feel it is important to note that sinusoidal patterns are not uncommon in nature. As such, we would like to frame the results presented here in the context of related work.

Dick and Hockstein [7] found oscillating patterns of error when observers judged the tilt of a luminous line. The periodicity of the errors they observed varied depending on the method used to report the line's tilt. Interestingly, much of the errors observed in this study were not strictly sinusoidal in the typical sense. Instead, they resemble a rectified sine wave where errors oscillate, but share a single sign (all positive errors or all negative errors). However, the judgments for this particular pattern were made by numerically reporting the angle of tilt in degrees on a keyboard. Fully sinusoidal errors, however, did not appear until observers estimated tilt using "minutes" as if looking at a clock face. The angles tested ranged from 0° to 90°, one quarter of the orientations tested in the present studies. If the oscillating pattern of errors continued beyond this range we would find that this pattern exhibited periodicity of 4 oscillations per 360° of orientation. This is a very short period compared to the present results. However, when their participants were asked to estimate line tilt using a contrived unit not directly associated with either degrees or minutes, they found a pattern of error that is nearly identical to the present results for the range of tested orientations. However, it is unknown whether this pattern would continue to be sinusoidal beyond the 0° to 90° range. Perhaps the most important point that can be made by Dick and Hockstein [7] is that the judgment method has a potent influence on performance. As noted above, this could offer a potential explanation for the two distinct patterns seen in Experiment III.

Appelle [1] describes an *oblique effect* where visual task performance is degraded when stimuli are presented at oblique angles. One can argue that the results found in the present studies can simply be described as a result of this oblique effect. However, it is important to note that the oblique effect describes a *type* of performance error and not a *cause* of error. Attributing the present results to the oblique effect is appropriate since the phenomenon can accurately be described as a function of stimulus obliqueness. Unfortunately, this is not explanatory of the mechanisms underlying the effect nor does it ensure that one oblique effect is fundamentally related to other oblique effects. We can draw parallels between the oblique effect and the so-called *distance compression problem* seen in virtual environments. The distance compression problem is used to describe a general tendency toward distance underestimation but is not in itself an explanation for the tendency. Instead, it is well-known that multiple factors can lead to distance compression in virtual environments [4, 5, 22, 26, 29, 32, 35, 36]. Similarly, the oblique effect provides a very good way to classify the type of errors observed here, but it is not sufficient to serve as an explanation.

Goldmeier [15] found errors in the judgment of right angles formed by intersecting lines, leading to the notion that these were a special case. The present results show that this effect is not restricted to right angles and does not require a literal angle (intersecting lines) to be present. This was also observed by Snippe and Koenderink [48]. However, it appears that the effect can be elicited with as little as two reference points, as opposed to four. As such, only two explicit points of reference are necessary as long as there is an implied reference axis, such as a bearing or ahead direction.

Ferrante et al. [14] contend that perception of a right angle is dependent on the visual horizontal- and vertical-ness of its constituent lines. However, the results of Experiments I, II, & III contradict this assumption as there are no explicit linear references. The angles presented to the observer are based on two reference points on the ground plane. This indicates that the observed angular anisotropy is likely not mediated by the horizontal and vertical orientation of two discrete lines, but the position of discrete references within the observer's visual environment. However, this assumes that Ferrante et al. [14]'s findings and the present results are due to the same underlying processes. Given that their results would be predicted by the patterns observed here, but not vice versa, it is reasonable to speculate that they are the result of similar, if not the same, processes.

The present results are also in agreement with Ferrante et al. [14]'s assertion that allocentric orientation judgments may be primarily based on a retinal as opposed to a primarily environmental frame of reference. It is important to emphasize that other references, such as gravity, are known to influence orientation judgments. The notion of a primarily retinal frame of reference is supported by the findings of Experiment IV showing that errors decreased significantly when reference lines subtended the same visual angle. These results are also consistent with those reported by Snippe and Koenderink [48]. Though their work was specifically examining threshold sensitivity when judging the similarity of two angles, their work indicated a strong sensitivity to stimuli at 90° and 180° angles. This is consistent with the observations of Appelle [1] which predict decreasing judgment errors as sensitivity increases for orientations approaching vertical and horizontal. This was seen in the results of all experiments in the current work as well. A fully, as opposed to partially, retina-based frame of reference is unlikely to be the sole factor modulating these errors as others have shown that many aspects of visual perception are also influenced by environmental and gravitational references [2, 3, 8, 16].

There exists some neurological evidence to support the notion that angles of certain orientations are processed differently in the early stages of vision. Shevelev et al. [45] demonstrated that there is heightened activity in neurons within the primary visual cortex of cats when presented with both lines and crosses of various orientations. There was strong orientation dependent changes in neuronal activity as stimuli were rotated about the center of vision. Strongest activity was found approximately along diagonals within the visual field. It would be premature to indicate that this phenomenon and the observations made here and elsewhere are directly related to each other. However, it is important to keep in mind that percepts are not independent of their neural representations.

5 Limitations

There are several limitations to the experiments presented here. Firstly, the number of participants per experiment was relatively small. However, the measured effects were sufficiently strong as to manifest with a high degree of statistical confidence. Nonetheless, additional participants would undoubtedly improve the quality of the analysis. This is especially true for Experiment III where participants were separated into two groups. The patterns seen there would almost certainly be clearer had more participants been tested.

Second, the procedures used in Experiment III were somewhat different than those used in Experiments I and II. Specifically, participants were not required to close their eyes between trails in the previous experiments. This decreases the comparability between these experiments. Given the odd results in this experiment, it may be the case that the differing patterns of error may have been due in some way to this procedural difference. Further investigations would be necessary to address this weakness.

Another limitation deals with eye height. In all experiments, participants sat in a somewhat tall chair, see Figure 1. During Experiments I and II, eye height was controlled via tracking the position of the HMD. In Experiment IV, however, the projection of the Frontal condition was calculated based on the approximate average seated eye height of the participants. It has been shown that eye height may have an effect on spatial judgments [29]. It is possible that the fixed eye height could have influenced the results for this condition. However, the results of Experiment IV exhibited smaller variability than those seen in the previous experiments. As such, we suspect that the influence of this uncontrolled variable was likely small.

6 Conclusions

This series of experiments began as part of an effort to see what effect, if any, immersive virtual environments had on the judgment of allocentric orientation as compared to real world performance. Experiment I demonstrated that a pattern of sinusoidal errors quite similar to those described in related studies also occurred in head-mounted virtual environments [7, 50]. Experiment II examined whether or not these errors could be attributed to observing the virtual world through a small FOV. To test this hypothesis, we presented the virtual environment through the full FOV of our display ($136^\circ \times 88^\circ$). The sinusoidal errors persisted in this experiment as well.

Experiment III aimed to determine if these errors persisted in the real world. We attempted to mimic the task performed in the virtual environment as closely as possible. An important difference was that participants were asked to close their eyes between trials since the real stimuli had to be manually placed. The results from this experiment were puzzling and seemed to indicate that participants fell into two distinct groups: those whose judgment errors adhered to a short period sine wave and those who adhered to a long period sine wave. The participants in the long period group seemed to exhibit a tendency for their judgments to be pushed forward in depth. The participants who exhibited the short period pattern, on the other hand, exhibited similar errors to those seen in the previous experiments. The existence of two groups in this experiment could imply that observers may have multiple strategies for performing orientation judgments with one more greatly preferred in the context of the virtual environment.

A possible explanation for the two groups observed in Experiment III can be found in Dick and Hockstein [7]. They found oscillating but not fully sinusoidal errors when asking observers to report the orientation in degrees for a tilted line. However, when participants were asked to report the orientation of the line in "minutes" as though it were the minute hand of a clock, adopted a sinusoidal profile. The pattern of errors that they observed changed further when participants judged orientation using a contrived unit of measurement. Though the stimuli and procedures from their study are quite different from ours, it is possible that similar strategies were used by our subjects when estimating orientations.

Experiment IV aimed to improve our measurements in two specific ways by using a bisection task. First, this removed a possible confound in our prior study designs: participants observing and reporting judgments in two different planes. Second, this removed the need for participants to implicitly reference their gaze direction when making judgments. For this new task, participants viewed a right angle on the ground plane and then proceeded to bisect that angle into equal halves thereby keeping observations and judgments within the same space. The pattern of errors predicted by the prior experiment was observed in this experiment, but only when stimuli were presented in a perspective view. When the stimuli were presented parallel to the frontal visual plane, errors significantly decreased for all target orientations.

Ultimately, we did not find that an immersive virtual environment altered the judgment of allocentric orientation, but we did find evidence of a robust visual anisotropy when judging orientations from a perspective viewpoint. This effect presented regardless of FOV, real or virtual viewing context, or judgment task. Another interesting finding that requires additional study is the possibility that observers may have multiple strategies for estimating orientations, one resulting in short period and another resulting in long period sinusoidal errors. This is somewhat implicated by the results of Experiment III. If multiple strategies exist, why did they only manifest themselves in the orientation judgment task in the real world? Why did they not also present in the real world bisection task? These are open questions and necessitate further study.

We believe this work makes three main contributions. First, the present results describe a visual anisotropy between allocentric orientations observed along the ground plane and their actual orientations. The pattern of errors was robust and, under most viewing conditions, universal among subjects. Though somewhat similar patterns have been seen in by others [1–3, 7, 8, 16, 50], this specific pattern, with regard to allocentric orientation judgments along the ground plane, appears to be unique.

Second, these results have implications for applications where the judgment of allocentric orientations are needed. Since these errors seem to be very consistent from one observer to the next, it is likely that existing tasks, such as air traffic control or ground force coordination, may be affected. However, knowing that these biases exist may aid in the training of people who perform such tasks thus, reducing the influence of these errors.

Finally, we believe that these results have broad applications in the area of spatial user interfaces. As these become more and more common, especially in the form of virtual and augmented environments, it will be necessary for us to better understand the biases and limitations inherent in both the interfaces and their users. For interfaces that require some form of angular judgment, we have two recommendations. First, since the errors seen through this series of studies adhere to a systematic pattern, they can possibly be predicted and corrected in the interface itself. Another approach would be to use the addition of directional indicators such as those seen in Ellis and Hacisalihzade [10]. A third approach can be to present such interfaces frontally oriented. The frontally presented stimuli in Experiment IV resulted in significantly reduced errors. However, it is still an open question whether or not the errors that did occur in this condition were the result of a "x-ray vision"-like view. If so, this implies that errors can be introduced by an interface's proximity to other surfaces within an environment. A similar observation has been made with regard to distance judgments in augmented environments [47]. This is of particular importance for augmented reality interfaces where graphics may appear to be overlapping, intersecting, or behind real world surfaces.

Acknowledgments

This material is based upon work supported by the Office of Naval Research under Grant No. N00014-13-1-0237. Any opinions, findings, and conclusions expressed in this material are those of the authors and do not necessarily reflect the views of ONR. The authors are also grateful to the USC-ICT Mixed Reality Lab (MxR) for the use of facilities and equipment needed for these studies.

References

[1] S. Appelle. Perception and discrimination as a function of stimulus orientation: the" oblique effect" in man and animals. *Psychological bulletin*, 78(4):266, 1972.

[2] M. Barnett-Cowan, R. W. Fleming, M. Singh, and H. H. Bülthoff. Perceived object stability depends on multisensory estimates of gravity. *PloS one*, 6(4):e19289, 2011.

[3] M. Barnett-Cowan and L. R. Harris. Perceived self-orientation in allocentric and egocentric space: effects of visual and physical tilt on saccadic and tactile measures. *Brain research*, 1242:231–243, 2008.

[4] G. Bruder, F. Steinicke, and P. Wieland. Self-motion illusions in immersive virtual reality environments. In *Virtual Reality Conference (VR), 2011 IEEE*, pp. 39–46, march 2011. doi: 10.1109/VR.2011.5759434

[5] S. H. Creem-Regehr, J. K. Stefanucci, and W. B. Thompson. Chapter six - perceiving absolute scale in virtual environments: How theory and application have mutually informed the role of body-based perception. In *Psychology of Learning and Motivation*, p. in press. Academic Press, 2015. doi: 10.1016/bs.plm.2014.09.006

[6] J. C. Davis. *Statistics and Data Analysis in Geology*. John Wiley & Sons, New York, 2002.

[7] M. Dick and S. Hochstein. Visual orientation estimation. *Perception & Psychophysics*, 46(3):227–234, 1989.

[8] R. T. Dyde, M. R. Jenkin, and L. R. Harris. The subjective visual vertical and the perceptual upright. *Experimental Brain Research*, 173(4):612–622, 2006.

[9] D. Eby and M. Braunstein. The perceptual flattening of 3-dimensional scenes enclosed by a frame. *Perception*, 24(9):981–993, 1995.

[10] S. R. Ellis and S. S. Hacisalihzade. Symbolic enhancement of perspective displays. *Proceedings of the Human Factors Society Annual Meeting*, 34(19):1465–1469, 1990. doi: 10.1177/154193129003401910

[11] S. R. Ellis, M. W. Mcgreevy, and R. J. Hitchcock. Perspective traffic display format and airline pilot traffic avoidance. *Human Factors*, 29(4):371–382, 1987. doi: 10.1177/001872088702900401

[12] S. R. Ellis and B. M. Menges. Localization of virtual objects in the near visual field. *Human Factors*, 40(3):415–431, 1998.

[13] S. R. Ellis, G. K. Tharp, A. J. Grunwald, and S. Smith. Exocentric judgements in real environments and stereoscopic displays. *Proceedings of the Human Factors Society Annual Meeting*, 35(20):1442–1446, 1991. doi: 10.1177/154193129103502005

[14] D. Ferrante, W. Gerbino, and I. Rock. Retinal vs. environmental orientation in the perception of the right angle. *Acta Psychologica*, 88(1):25 – 32, 1995. doi: 10.1016/0001-6918(93)E0057-9

[15] E. Goldmeier. Similarity in visually perceived forms. *Psychological Issues*, 8(1):1–136, 1972.

[16] L. R. Harris and C. Mander. Perceived distance depends on the orientation of both the body and the visual environment. *Journal of Vision*, 14(12):17–17, 2014.

[17] V. Interrante, B. Ries, and L. Anderson. Distance perception in immersive virtual environments, revisited. *Virtual Reality Conference, IEEE*, 0:3–10, 2006. doi: 10.1109/VR.2006.52

[18] H. Intraub, C. V. Gottesman, E. V. Willey, and I. J. Zuk. Boundary extension for briefly glimpsed photographs: Do common perceptual processes result in unexpected memory distortions? *Journal of Memory and Language*, 35(2):118–134, 1996.

[19] H. Intraub and M. Richardson. Wide-angle memories of close-up scenes. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 15(2):179, 1989.

[20] J. A. Jones, L. C. Dukes, D. M. Krum, M. T. Bolas, and L. F. Hodges. Correction of geometric distortions and the impact of eye position in virtual reality displays. In *2015 International Conference on Collaboration Technologies and Systems (CTS)*, pp. 77–83, June 2015. doi: 10.1109/CTS.2015.7210403

[21] J. A. Jones, D. M. Krum, and M. T. Bolas. Vertical field-of-view extension and walking characteristics in head-worn virtual environments. *ACM Trans. Appl. Percept.*, 14(2):9:1–9:17, Oct. 2016. doi: 10.1145/2983631

[22] J. A. Jones, J. E. Swan II, and M. Bolas. Peripheral stimulation and its effect on perceived spatial scale in virtual environments. *Visualization and Computer Graphics, IEEE Transactions on*, 19(4):701–710, 2013. doi: 10.1109/TVCG.2013.37

[23] J. A. Jones, J. E. Swan II, G. Singh, and S. R. Ellis. Peripheral visual information and its effect on distance judgments in virtual and augmented environments. In *Symposium on Applied Perception in Graphics and Visualization*, pp. 29–35. Toulouse, France, 2011.

[24] W. L. Keeney-Kennicutt and Z. H. Merchant. *Using Virtual Worlds in the General Chemistry Classroom*, chap. 8, pp. 181–204. ACS Symposium Series, 2013. doi: 10.1021/bk-2013-1142.ch008

[25] J. Kelly, W. Hammel, Z. Siegel, and L. Sjolund. Recalibration of perceived distance in virtual environments occurs rapidly and transfers asymmetrically across scale. *Visualization and Computer Graphics, IEEE Transactions on*, 20(4):588–595, April 2014. doi: 10.1109/TVCG.2014.36

[26] P. B. Kline and B. G. Witmer. Distance perception in virtual environments: Effects of field of view and surface texture at near distances. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 40, pp. 1112–1116, 1996.

[27] T. M. Knnapas. Influence of head inclination on the vertical-horizontal illusion. *The Journal of Psychology*, 46(2):179–185, 1958. doi: 10.1080/00223980.1958.9916283

[28] S. Lee, X. Hu, and H. Hua. Effects of optical combiner and ipd change for convergence on near-field depth perception in an optical see-through hmd. *Visualization and Computer Graphics, IEEE Transactions on*, PP(99):1–1, 2015. doi: 10.1109/TVCG.2015.2440272

[29] M. Leyrer, S. A. Linkenauger, H. H. Bülthoff, U. Kloos, and B. Mohler. The influence of eye height and avatars on egocentric distance estimates in immersive virtual environments. In *Proceedings of the ACM SIGGRAPH Symposium on Applied Perception in Graphics and Visualization*, APGV '11, pp. 67–74. ACM, New York, NY, USA, 2011. doi: 10.1145/2077451.2077464

[30] B. Li, J. Walker, and S. A. Kuhl. The effects of peripheral vision and light stimulation on distance judgments through hmds. *ACM Trans. Appl. Percept.*, 15(2):12:1–12:14, Apr. 2018. doi: 10.1145/3165286

[31] B. J. Mohler, J. L. Campos, M. B. Weyel, and H. H. Bülthoff. Gait parameters while walking in a head-mounted display virtual environment and the real world. In B. Fröhlich, R. Blach, and R. van Liere, eds., *Eurographics Symposium on Virtual Environments, Short Papers and Posters*, pp. 85–88. Eurographics Association, Weimar, Germany, 2007. doi: 10.2312/PE/VE2007Short/085-088

[32] B. J. Mohler, S. H. Creem-Regehr, and W. B. Thompson. The influence of feedback on egocentric distance judgments in real and virtual environments. In *Proceedings of the 3rd Symposium on Applied Perception in Graphics and Visualization*, APGV '06, pp. 9–14. ACM, New York, NY, USA, 2006. doi: 10.1145/1140491.1140493

[33] B. J. Mohler, W. B. Thompson, S. H. Creem-Regehr, P. Willemsen, H. L. Pick, Jr., and J. J. Rieser. Calibration of locomotion resulting from visual motion in a treadmill-based virtual environment. *ACM Trans. Appl. Percept.*, 4(1), Jan. 2007. doi: 10.1145/1227134.1227138

[34] N. Nilsson, S. Serafin, and R. Nordahl. Establishing the range of perceptually natural visual walking speeds for virtual walking-in-place locomotion. *Visualization and Computer Graphics, IEEE Transactions on*, 20(4):569–578, April 2014. doi: 10.1109/TVCG.2014.21

[35] L. Phillips and V. Interrante. A little unreality in a realistic replica environment degrades distance estimation accuracy. In *Virtual Reality Conference (VR), 2011 IEEE*, pp. 235 –236, march 2011. doi: 10.1109/VR.2011.5759485

[36] L. Phillips, V. Interrante, M. Kaeding, B. Ries, and L. Anderson. Correlations between physiological response, gait, personality, and presence in immersive virtual environments. *Presence: Teleoperators and Virtual Environments*, 21(2):119–141, 4 2012.

[37] W. T. Pollock and A. Chapanis. The apparent length of a line as a function of its inclination. *Quarterly Journal of Experimental Psychology*, 4(4):170–178, 1952. doi: 10.1080/17470215208416615

[38] E. D. Ragan, S. Scerbo, F. Bacim, and D. A. Bowman. Amplified head rotation in virtual reality and the effects on 3d search, training transfer, and spatial orientation. *IEEE Transactions on Visualization and Computer Graphics*, 23(8):1880–1895, Aug 2017. doi: 10.1109/TVCG.2016.2601607

[39] A. R. Richardson and D. Waller. Interaction with an immersive virtual environment corrects users' distance estimates. *Human Factors*, 49(3):507–517, 2007.

[40] B. E. Riecke, D. W. Cunningham, and H. H. Bülthoff. Spatial updating in virtual reality: the sufficiency of visual information. *Psychological research*, 71(3):298–313, 2007.

[41] B. E. Riecke, M. V. D. Heyde, and H. H. Bülthoff. Visual cues can be sufficient for triggering automatic, reflexlike spatial updating. *ACM Transactions on Applied Perception (TAP)*, 2(3):183–215, 2005.

[42] B. E. Riecke, H. A. H. C. v. Veen, and H. H. Bülthoff. Visual homing is possible without landmarks: A path integration study in virtual reality. *Presence: Teleoperators and Virtual Environments*, 11(5):443–473, 2002. doi: 10.1162/105474602320935810

[43] S. M. Ritter. The vertical-horizontal illusion: An experimental study of meridional disparities in the visual field. *The Psychological Monographs*, 23(4):1–110, 1917.

[44] R. N. Shepard. *Mind Sights: Original Visual Illusions, Ambiguities, and other Anomalies*. WH Freeman and Company, New York, 1990.

[45] I. Shevelev, N. Lazareva, G. Sharaev, R. Novikova, and A. Tikhomirov. Selective and invariant sensitivity to crosses and corners in cat striate neurons. *Neuroscience*, 84(3):713 – 721, 1998. doi: 10.1016/S0306-4522(97)00393-X

[46] W. C. Shipley, B. M. Mann, and M. J. Penfield. The apparent length of tilted lines. *Journal of Experimental Psychology*, 39(4):548–551, 1949.

[47] G. Singh, J. E. Swan II, J. A. Jones, and S. R. Ellis. Depth judgment measures and occluding surfaces in near-field augmented reality. In *Symposium on Applied Perception in Graphics and Visualization*, pp. 149–156. Los Angeles, California, USA, 2010.

[48] H. P. Snippe and J. J. Koenderink. Discrimination of geometric angle in the fronto-parallel plane. *Spatial Vision*, 8(3):309–328, 1994. doi: 10.1163/156856894X00017

[49] J. E. Swan, A. Jones, E. Kolstad, M. A. Livingston, and H. S. Smallman. Egocentric depth judgments in optical, see-through augmented reality. *Visualization and Computer Graphics, IEEE Transactions on*, 13(3):429–442, may-june 2007. doi: 10.1109/TVCG.2007.1035

[50] G. K. Tharp and S. R. Ellis. The effects of training on errors of perceived direction in perspective displays. Technical Report NASA-TM-102792, A-90081, NAS 1.15:102792, NASA Ames Research Center, July 1990.

[51] H. von Helmholtz. *Treatise on Physiological Optics*, vol. 3. Dover Publications, New York, 1962.

algorithms

MDPI

Article

Following the Writer's Path to the Dynamically Coalescing Reactive Chains Design Pattern

João Paulo Oliveira Marum [1,*], H. Conrad Cunningham [2,*], J. Adam Jones [3] and Yi Liu [4]

[1] Department of Electrical Engineering and Computer Science, Syracuse University, 3-127 CST, Syracuse, NY 13244, USA
[2] Department of Computer and Information Science, University of Mississippi, 201 Weir Hall, University, MS 38677, USA
[3] Department of Computer Science and Engineering, Mississippi State University, 143 Rice Hall, Mississippi State, MS 39762, USA; jadamj@acm.org
[4] Department of Computer and Information Science, University of Massachusetts Dartmouth, 302E Dion Building, Dartmouth, MA 02747, USA; yliu11@umassd.edu
* Correspondence: jomarum@syr.edu (J.P.O.M.); hcc@cs.olemiss.edu (H.C.C.); Tel.: +1-315-443-1135 (J.P.O.M.)

Abstract: Two recent studies addressed the problem of reducing transitional turbulence in applications developed in C# on .NET. The first study investigated this problem in desktop and Web GUI applications and the second in virtual and augmented reality applications using the Unity3D game engine. The studies used similar solution approaches, but both were somewhat embedded in the details of their applications and implementation platforms. This paper examines these two families of applications and seeks to extract the common aspects of their problem definitions and solution approaches and codify the problem-solution pair as a new software design pattern. To do so, the paper adopts Wellhausen and Fiesser's writer's path methodology and follows it systematically to discover and write the pattern, recording the reasoning at each step. To evaluate the pattern, the paper applies it to an arbitrary C#/.NET GUI application. The resulting design pattern is named DYNAMICALLY COALESCING REACTIVE CHAINS (DCRC). It enables the approach to transitional turbulence reduction to be reused across a range of related applications, languages, and user interface technologies. The detailed example of the writer's path can assist future pattern writers in navigating through the complications and subtleties of the pattern-writing process.

Keywords: design pattern; writer's path; event-based architecture; implicit invocation; reactive programming; transitional turbulence; dependency graph

check for updates

Citation: Oliveira Marum, J.P.; Cunningham, H.C.; Jones, J.A.; Liu, Y. Following the Writer's Path to the Dynamically Coalescing Reactive Chains Design Pattern. *Algorithms* **2024**, *17*, 56. https://doi.org/10.3390/a17020056

Academic Editors: Simone Fontana, Silvia Corchs and Aurora Saibene

Received: 22 October 2023
Revised: 17 January 2024
Accepted: 22 January 2024
Published: 25 January 2024

1. Introduction

A visual user interface must respond quickly to user actions and display their effects accurately. This is especially important for virtual and augmented reality applications, but it is also important for desktop and Web applications. Each of these applications "interacts with its environment on an ongoing basis" [1]. It reacts to a stream of *events*, where an event may be a stimulus from the external environment (such as a user movement) or from the computational environment (such as a notification that some software component changes its state).

When the handling of an event affects the state of one component of a visual user interface, that component may cause events that affect several other components (i.e., *update* the other components). Each of these components may, in turn, cause events that affect additional components, and so forth, as the component updates ripple throughout the user interface. We call the processing of all the component updates resulting from some initial event an *update cycle*. When an update cycle is completed, the user interface can potentially enter a *stable state* with no updates pending. For convenience, we define *latency* as the period of "time" it takes for all components of a system to reach a stable state after some stimulus (such as processing an external event).

The display system operates independently of the event handling system. Therefore, it may take several cycles of the display system for the states of all components to be updated and the user interface to reach a stable state. This period of *transitional turbulence* [2] (or glitchiness [3,4]) can result in displays in which the visible states of the components do not meet the users' expectations of the user interface's behavior. Due to these inconsistent displays, users can (at least temporarily) perceive the system to be unreliable and inaccurate. For convenience, we often refer to the occurrence of a visible inconsistency on a display as an *error*.

If some component `C` causes an event that directly affects some other component `D`, then `D` *depends on* `C`. To alleviate the transitional turbulence problem, Marum et al. [5,6] developed a reactive programming approach [7] that encodes these dependency relationships between components in a *dependency graph* and then uses the graph to rearrange the processing of the updates from an update cycle in any order consistent with the dependency relationships. This enables the processing of all the events from one update cycle as if the update cycle were a single large-grained event that updates all the components. This enables faster updates and more accurate visualizations, potentially providing users with a more satisfying experience.

For Web and desktop graphical user interfaces (GUIs) implemented with C#, the approach builds the dependency graph by analyzing the relationships between the components of the GUI (i.e., its graphical controls such as radio buttons) [6,8]. Many effects that had previously been spread across multiple display cycles now occur within a single cycle. By conducting a set of experiments, the Marum et al. case study [6,8] shows that the approach can perform better (i.e., decrease latency) and exhibit more accurate behavior (i.e., display fewer errors) than similar applications using the standard C#/.NET GUI and the Sodium [9] and Rx.NET [10] reactive programming libraries.

For virtual and augmented reality applications implemented using the Unity3D game engine and C# [11], the approach is similar, except that it takes advantage of Unity3D's existing object hierarchy [5]. The approach builds the dependency graph by analyzing the relationships among the Unity3D game components. If Unity3D's object hierarchy changes, the approach recomputes the dependency graph. By reordering the events based on the dependencies, the approach eliminates many of the inconsistent displays without degrading the performance of the system. By dynamically reacting to changes in the object hierarchy, the approach can smoothly handle relatively complex applications. By conducting a set of experiments, Marum et al. [5] shows that the approach can also perform better and exhibit more accurate behavior than both an unmodified Unity3D application and a similar application developed using the reactive library UniRx [12].

The Marum et al. case studies [5,6] address two different but related problems and devise two similar solutions using different user interface technologies. Both solutions work by augmenting the normal event processing mechanisms used in the applications. In this research, we examine the two case studies and seek to isolate the essence of the solution approach so that it can potentially be applied by others to similar problems using similar programming languages and user interface technologies. To do so, we address the following research questions:

(RQ1) Can we codify the solution approach as a new software design pattern [13,14]?
(RQ2) Can we follow Wellhausen and Fiesser's writer's path [15] to write the new pattern step by step?

Section 2 describes the writer's path and the other methods that we use. Sections 3–12 then record how we use these methods systematically to write the desired new software design pattern, which we name DYNAMICALLY COALESCING REACTIVE CHAINS (DCRC). Appendix B shows the complete DCRC pattern. Section 13 then demonstrates the technical feasibility and efficacy of the DCRC pattern by applying it to an arbitrary C#/.NET GUI application. Section 14 discusses the evolution of the pattern and related and future work.

2. Writing Software Patterns

A *software design pattern* is defined in the classic "Gang of Four" patterns book as a "general and reusable solution to a set of problems with common characteristics within a given context" [14]. A pattern is not invented; it is distilled from practical experience [13]. Patterns codify "best practices" for software architecture and design [16]. Patterns are written and published to document these best practices and enable others to apply them in their own work.

The "Siemens" book [13] groups software patterns into three categories:

- An *architectural pattern*—also called an *architectural style* [17,18]—is a high-level, language-independent abstraction that guides the design of the system-wide structure.
- A *design pattern* is a mid-level, (mostly) language-independent abstraction that guides the design of a subsystem.
- An *idiom* is a low-level language-specific abstraction that guides some aspects of both design and implementation.

Among several existing formats for describing patterns [13–16,19], we choose the simple format described by Wellhausen and Fiesser [15], which presents the following structural elements in the given order:

Pattern Name gives an evocative name for the pattern.
Context describes the circumstances in which the problem occurs.
Problem describes the specific problem to be solved.
Forces describe why the problem is difficult to solve, identifying the often contradictory considerations that must be balanced to solve the problem.
Solution describes how the solution to the problem works at an appropriate level of detail.
Consequences describe what happens when a software designer applies the pattern. It gives both the possible benefits and liabilities of using the pattern.

All of the above elements except Consequences are also prescribed by the MANDATORY ELEMENTS PRESENT pattern from Meszaros and Doble's Pattern Language for Pattern Writing [16]. In its OPTIONAL ELEMENTS WHEN HELPFUL pattern, the Consequences element is called the Resulting Context. Following their READABLE REFERENCES TO PATTERNS pattern, we show the pattern names using small capitals in this paper.

Although a fully specified software pattern should be published in the order given above, the elaboration of the pattern's elements usually does not proceed in that order. Instead, it spirals through the elements and may require multiple iterations over a period of time. In this paper, we adopt the Wellhausen and Fiesser *writer's path* [15] to guide us in writing the pattern because it is a simple methodology that enables us to explore the problem domain systematically and refine the description of the pattern incrementally. We enhance its steps by using other established methods such as Scope, Commonality, and Variability (SCV) analysis [20] and the pattern-writing patterns from Meszaros and Doble's [16] and Harrison's [21,22] pattern languages. We carefully record our steps to help others use this methodology to write patterns for other problems.

1. Explore the new pattern's rationale and scope.
   We consider questions such as: Why should we write a new pattern? What is included in and excluded from its scope? What concrete examples do we have that we can examine? We then state a crisp definition for the scope.
2. Examine existing solutions.
   We consider the answers to the questions from the previous step and discuss the solutions with others. We seek to determine what is common across all the solutions and what is variable among the solutions (i.e., holds for only some of the solutions).We briefly summarize the general solution, focusing on its essence. We collect a list of possible names for the pattern. We also list any clever ideas identified in the solutions for later consideration, even if they are not essential to the solution.

3. Describe the problem that leads to the solution.
   We strive to state this description in one sentence. We must be careful to separate the problem from its solution and make sure that the solution actually solves the problem.
4. Consider the consequences of the solution, both its benefits and its liabilities.
   We consider any "clever ideas" identified in Step 2. These may help us identify the consequences of applying the pattern. To identify the benefits, we consider the desirable outcomes that result from applying the pattern. (That is, we consider the difference in the result when the pattern is applied versus when the pattern is not applied.) To identify liabilities, we consider the complications that result from applying the pattern and what the possible undesirable outcomes are.
5. Identify the forces that make the problem difficult to solve.
   The forces usually conflict with one another, pushing in contradictory directions. We consider what differentiates the chosen solution from other possible solutions to the problem to help identify the different forces at work. We give each force a meaningful name.
6. Match each force with the corresponding consequences.
   A force makes the problem difficult to solve. How the solution resolves this difficulty leads to the corresponding consequences. Each force must be resolved and may have both benefits and liabilities. Each consequence must be matched by a force. The matching of forces and consequences helps guide us from the problem to the solution.
7. Elaborate the context in which the problem exists.
   We carefully consider all the assumptions made by the problem and its solution. The problem might not even exist outside of this context. The context cannot be changed by the solution.
8. Choose a pattern name.
   A good name should evoke the core idea of the solution. It should be easy to remember.
9. Reexamine and rewrite the six elements of the pattern.
   We use the Context to describe the background and assumptions. We focus on devising a short, crisp Problem description. We put what makes the Problem difficult in the Forces and ensure the Solution solves the Problem and balances the Forces. We link the Forces with the Consequences.
10. Put the pattern elements in the standard order.
    We restate the Solution and Consequences appropriately to match the other elements, writing the pattern so that it flows smoothly from Context to Consequences.
11. Evolve the pattern based on feedback and experience.
    When writing the pattern, we seek feedback from experts in the technical area and in pattern writing. After a period of time, we reexamine and rewrite the pattern description. We continue to evolve the pattern as we gain deeper experience with its use. Patience is necessary because it takes time to ensure that the pattern description is accurate.

3. Exploring Rationale and Scope

In Step 1 on the writer's path, we explore the rationale and scope of the new pattern. Given the dynamic .NET GUI [6] and VR [5] case studies described in Section 1, there appears to exist "a recurring solution to a problem" that can potentially "be reused" by others. As suggested by Meszaros and Doble's PATTERN pattern [16], we seek to document "the solution using the pattern form". We begin by asking: What is the scope (i.e., the context) of the new pattern?

3.1. *Implicit Invocation Architectural Pattern*

For the research reported in this paper, we find that the IMPLICIT INVOCATION (II) architectural pattern [17,23–25] is useful to help us define the scope of the DCRC design pattern. Using typical software architecture terminology [18,26,27], Shaw describes the *system model* as a graph with software components at the nodes and connectors along the edges [24]. The *components* are high-level computational and data storage entities and the *connectors* are the interactions among the components. Furthermore, there is a *control structure* that governs how the system executes.

Figure 1 depicts the IMPLICIT INVOCATION architectural pattern. According to Shaw [24], an IMPLICIT INVOCATION system consists of a "loosely coupled collection" of "independent reactive processes" (i.e., "modules" [17]). The components are these modules, which can "signal significant events without knowing the recipients of the signals". The connectors are the implicit (or automatic) invocations of procedures in the modules' interfaces "that have registered interest in events". The control structure is "decentralized" and asynchronous, so that the individual components are unaware of the recipients of their signals.

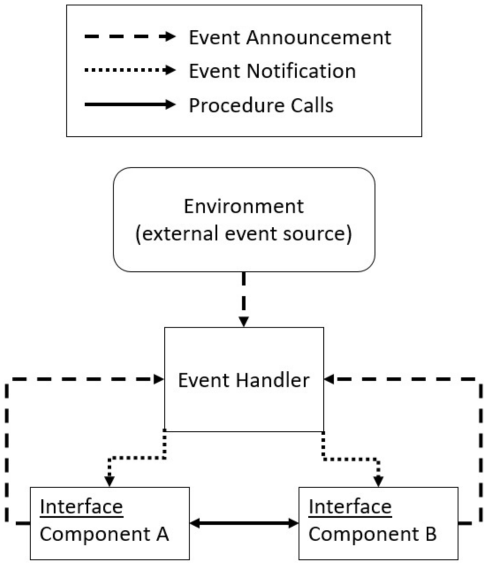


Figure 1. Implicit Invocation architectural pattern.

Implementing the IMPLICIT INVOCATION pattern usually requires some kind of "event handler that registers components' interest in receiving events and notifies them that events" have been signaled [24]. When a component registers interest in an event, it associates a procedure with that event. To notify the component that the event has been signaled, the event handler implicitly invokes the associated procedure [17]. We assume that the event handler is nondeterministic but fair. That is, once an event is signaled by a component, all the listeners' associated procedures will eventually be invoked, but there is no guarantee in what order the events will be handled.

An implicit invocation system has advantages and disadvantages [17,24,28,29]. Among the advantages is support for software reuse and dynamic reconfiguration. Among the disadvantages are the nondeterminism of processing order and the difficulty in reasoning about correctness.

There are, of course, many different variations of the implicit invocation concept, such as the classic OBSERVER [14] and PUBLISHER-SUBSCRIBER [13,28] design patterns. In this paper, we use the general term IMPLICIT INVOCATION, which seems to describe the overall concept and operation of the event-driven programming mechanisms in Marum's case studies and many other user interface platforms.

3.2. Identifying the Context

In both the .NET GUI [6,8] and VR [5] case studies, the built-in event handling systems follow the implicit invocation architectural pattern as described above. In both, we also observe transitional turbulence as described in Section 1. Both case studies also layer the solution to the transitional turbulence problem on top of the built-in event handling systems. Thus, to define the Context for our new pattern, we focus on the following characteristic:

(C1) The application is constructed according to the IMPLICIT INVOCATION architectural pattern, assuming nondeterministic but fair handling of events.

As we continue to write the pattern, we identify other assumptions about the Context in which the pattern is relevant. In writing the new pattern, we also constrain it in the following ways:

- As suggested by the CLEAR TARGET AUDIENCE [16] and CONSISTENT-"WHO" [22] patterns, we focus our attention on developers who are working within a software architecture described by the IMPLICIT INVOCATION pattern. We do not assume any particular programming language or user interface platform in the general description.
- As suggested by the TERMINOLOGY TAILORED TO AUDIENCE and UNDERSTOOD NOTATIONS patterns [16], we use terminology, concepts, and notations that should be familiar to the identified target audience. We also relate the terminology we use in the pattern description to that we use in the IMPLICIT INVOCATION architectural pattern description.
- As suggested by the DEAD WEASELS pattern [22], we seek to identify any "weasel words"—words that "imply meaning but have no real substance" or are too ambiguous or imprecise to guide the reader in applying the pattern effectively. We try to replace a "weasel word with a phrase or paragraph that is more specific". For example, we use a word such as "system" with care because it might have many different meanings in the discussion.

4. Examining Existing Solutions

In Step 2 on the pattern writer's path, we examine existing solutions. Our primary objective in writing a new pattern is to unify the solutions that emerged from two related case studies: the dynamic .NET GUI [6,8] and VR [5] families of applications.

There is, of course, a wealth of other research on reactive programming languages and systems [3,7,9,10,30–35] that we could profitably examine. However, in this paper, we focus our attention on solutions that follow the IMPLICIT INVOCATION architectural pattern and work by augmenting the normal event handling mechanisms of the user interface technologies on which they are built. The solutions in both Marum et al. case studies satisfy these criteria. The new design pattern seeks to document how a developer should analyze an existing application, develop appropriate new software mechanisms to reduce transitional turbulence, and incorporate the mechanisms into a modified application.

As illustrated in Appendix A, both case studies develop a reactive programming approach that encodes the complex relationships between the components of a specialized IMPLICIT INVOCATION system in a dependency graph and then uses the graph to rearrange the updates of the components in any order consistent with the dependency constraints. As the case studies demonstrate experimentally [5,6,8], this approach enables faster updates and more accurate visualizations, potentially providing users with a more satisfying experience. Harrison's "WHAT"-SOLUTIONS pattern [22] suggests writing the core idea of a solution in a one- or two-sentence summary. We thus state the Solution element's summary as follows:

> A solution encodes the complex relationships among the application's components in a dependency graph, and then uses the graph to order the updates of the components without violating the dependency constraints. The goal is to reorder the updates of the components so that the new order reduces transitional turbulence without degrading the performance of the system.

This summary will form a prominent part of the full description of the new pattern's Solution element.

As a result of our analysis, we identify at least three possible names for our pattern: Reactive Dependency Graph, Transitional Turbulence Reduction, and Dynamically Coalescing Reactive Chains. We choose among these names in Section 10.

5. Describing the Problem

In Step 3 on the pattern writer's path, we describe the problem that leads to the solution. The core of a pattern is the pairing of a Problem with the corresponding Solution. However, Harrison [22] observes that often "the problem and solution are basically restatements of one another" during the early phases of writing a pattern. To help differentiate these, the "WHY"-PROBLEMS pattern [22] suggests that pattern writers ask themselves "how the world would be worse" if the new pattern is not used. Of course, the pattern writers can make "the world" as specific as it needs to be by how they define the Context.

The core issue addressed by the example applications in Section 4 is reducing transitional turbulence. Transitional turbulence can result in an external presentation that does not accurately represent the expected behavior of the system. This leads to the following statement of the Problem element for the new pattern:

> We want to eliminate or reduce the length of the periods of transitional turbulence during which the external presentation does not accurately reflect the state of the application. We need to do this without sacrificing performance. The goal is to better satisfy the observers' expectations by increasing the accuracy of the external presentation.

Consider how the problem can be specifically observed in the case studies. In the example .NET GUI applications, when a user enters data in the form, it may reconfigure itself. If the interconnections among controls are complex, then it may take several display cycles for all the changes to propagate throughout the form. During this period, the form may show invalid options or may redraw itself while the user is entering data. It is understandable that both situations would be frustrating to the user.

Although the Context and Solution must be refined further, the proposed Solution seems to solve the stated Problem in the given Context—as Harrison's BIG PICTURE pattern [21] suggests it should. The Problem specifies *what* must be performed. The Solution proposes *how* that can be accomplished. The Context describes the environment in which the Problem and its Solution exist.

6. Considering the Consequences

In Step 4 on the pattern writer's path, we consider the consequences of applying the Solution to the Problem, both its benefits and liabilities. To identify these Consequences, we reexamine aspects of the example applications in Section 4.

6.1. Benefits

To identify the benefits, we consider what the desirable outcomes are from applying the pattern to the Problem to construct a Solution.

The example applications analyze the structure of the user interface and optimize its event processing by combining the state changes associated with sequences of related events into larger units. In doing so, they seek to mitigate the effects of transitional turbulence. Therefore, we state the first benefit:

- A solution coalesces sets of dependent internal events into "large-grained" events such that the handling of a large-grained event causes the same overall state change as the corresponding set. This can decrease latency and increase accuracy.

In the example applications, the structure of the user interface may change at run time. That is, components and events can be added, deleted, or modified. The applications dynamically adapt to these changes. They seek to preserve the benefits of the event processing optimizations that mitigate the effects of transitional turbulence. We state the second benefit:

- A solution dynamically adapts to changes in an application's component architecture at run time.

The example applications augment the standard (.NET or Unity3D) event processing system, but do not replace it. They use libraries, the C# reflection facilities, and other lightweight programming techniques to optimize event processing. In other situations, a solution might need to use other mechanisms, such as preprocessing tools. We state the third benefit:

- An application can be readily adapted to use the mechanisms that implement the solution.

6.2. Liabilities

To identify the liabilities, we consider the complications resulting from applying the pattern and the possible undesirable results.

In the example applications, the structure of the user interface can change at run time. These changes in the user interface's underlying structure may, in themselves, degrade the event processing performance, and thus increase latency and decrease accuracy. In addition, these changes may degrade the effectiveness of optimizations that are based on the user interface's structure. The application may need to undertake a costly reanalysis of that structure to incorporate different optimizations—as the example applications do. A solution should minimize the cost of adapting to structural changes at run time. We state the first liability:

- Changes to an application's component architecture at run time can increase latency and decrease accuracy.

Any Solution that reduces transitional turbulence likely requires that the structure of the user interface be analyzed and modified before its normal operation begins. This can be a costly operation, particularly if performed at run time—as the example applications do. A solution should minimize this startup overhead. We state the second liability:

- Implementing a solution often causes additional processing overhead at startup and shutdown of the application.

Similarly, any Solution that reduces transitional turbulence likely adds overhead to the normal processing of events. This overhead may be especially significant when the solution must adapt to changes in the user interface's structure. The example applications introduce this kind of overhead because they use dependency graphs to optimize the event processing and must rebuild the graph when the structure of the user interface changes. A solution should minimize this operational overhead. We state the third liability:

- Implementing a solution often causes additional run-time processing overhead, especially when the component architecture changes.

In addition, any Solution that reduces transitional turbulence likely makes the programs more complex and, hence, more costly to design, implement, test, and maintain. The added software mechanisms should be kept lightweight. The example applications use special libraries to handle most of the additional processing needed; the libraries work on top of the standard (.NET or Unity3D) event handler.

To enable the dependency graph to be built, the application developer must adapt some objects in the user interface to allow the library to manipulate them. The mechanisms are relatively lightweight but do make the application's code more complicated. We state the fourth liability:

- An application must be adapted to use the mechanisms that implement the solution. Modifying the application often complicates its design, implementation, testing, or use.

7. Identifying the Forces

In Step 5 on the pattern writer's path, we identify the forces. The Forces are the often contradictory aspects of the stated problem and its context that make it difficult to select and devise a solution [16]. Following the suggestion of the VISIBLE FORCES pattern [16], we assign each force in the new pattern a meaningful name and display the set of forces as a list. Following the suggestion of the FORCES HINT AT SOLUTION pattern [22], we order the Forces in the list from Problem-oriented issues toward Solution-oriented issues. To identify the Forces, we reexamine aspects of the example applications from Section 4.

Decreasing transitional turbulence is the primary motivation for attempting to solve the Problem. This gives rise to the first force we identify.

> *Transitional Turbulence Reduction*: We want to decrease the transitional turbulence in the application's execution to better satisfy the observers' expectations.

We adopt the same criteria as Marum et al. [5,6,8] to quantify transitional turbulence: latency (perhaps measured in update cycles) and error (i.e., inaccuracy) counts. In an implicit invocation architecture such as the applications we examine in Section 4, decreasing transitional turbulence probably requires a solution that optimizes event processing.

The structures of the example applications' user interfaces can change at run time. We want to handle this situation in any Solution to the Problem. This is the second force.

> *Run-time Reconfiguration:* We want to adapt to changes in an application's component architecture at run time.

Dynamically changing the structure may complicate any solution that optimizes the event processing based on the user interface's structure. For example, if the solution builds and uses a dependency graph of the controls in a .NET GUI, then changes in the the GUI's structure invalidates the graph. This requires that the dependency graph be updated whenever the structure changes, which likely makes the code more complex and degrades performance.

In the example applications (from Section 4), any Solution to the Problem likely requires that the user interface be analyzed and modified before its normal operation begins. Both steps probably require that new software mechanisms (i.e., code) be developed and executed. Additionally, the modified user interface probably has more complex code and a longer execution time. We want to avoid significantly increasing execution time. This gives rise to the third force.

> *Startup Cost Inflation*: We want to avoid adding significant startup or shutdown costs.

Consider the example .NET GUI applications. The analysis may construct a dependency graph of the GUI's controls, and the modification may augment the GUI to use the dependency graph in optimizing the event processing.

- If the analysis and modification can be performed statically, then they can be conducted in a preprocessing phase and will thus have a limited impact on the startup and shutdown of the GUI's execution.
- If the analysis and modification must be performed dynamically, then they must be conducted at run time and can thus have a more significant impact on the startup and shutdown of the GUI's execution. We want these costs to be small.

Because of the requirement to support dynamic changes to the GUI (as discussed above), the example applications do the analysis and modification completely at run time. Some of the initial analysis and modification could have been performed in a preprocessing step, but that would require the mechanisms to be implemented in two completely different ways.

As noted above, the modified user interface has more overhead and more complex code for event processing. The costs of supporting Run-time Reconfiguration also adds processing overhead and code complexity. However, we want to keep the execution costs of event processing small. This is the fourth force.

Operational Overhead Creep: We want to avoid adding significant processing overhead during the application's normal operation.

All the mechanisms introduced in the discussion of the other forces above increase the complexity of the programs. This increases the cost of designing, implementing, testing, and maintaining the application. We want to keep this cost small. This is the fifth force.

Code Cluttering: We want to avoid significantly complicating the application's design, implementation, testing, or use.

We want any modifications of the programs to be simple and supported by libraries and/or tools. We also want the modifications to the event processing to work on top of the standard event processing mechanisms. The example applications designed and implemented libraries to handle most of the additional processing needed; the libraries work on top of the standard event handlers.

Each force is potentially in conflict with other forces, as shown in Figure 2. A solution must balance these forces.

- The *Transitional Turbulence Reduction* force is in conflict with all the other forces. Seeking to reduce transitional turbulence tends to increase the costs due to the other forces. Seeking to keep the costs due to the other forces low tends to make it difficult to reduce transitional turbulence.
- The *Code Cluttering* force is in conflict with all the other forces. They represent factors that can make the design and implementation of the code more complex. If no code can be added (i.e., the program is kept uncluttered), then the other forces cannot be satisfied.
- The *Run-Time Reconfiguration* force is in conflict with *Operational Overhead Creep*. Dynamically adapting to changes in the application's component architecture increases the operational overhead cost. If no increase in overhead cost is allowed, then the *Run-time Reconfiguration* force likely cannot be satisfied.

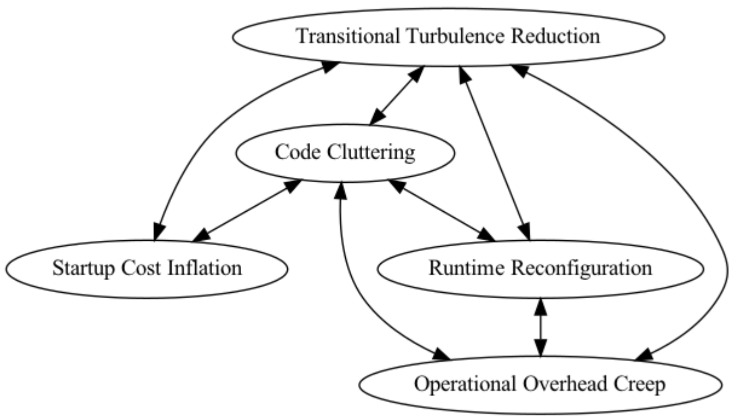


Figure 2. Conflicts among the pattern's forces.

8. Matching Forces with Consequences

In Step 6 on the pattern writer's path, we match each force with the corresponding consequences. Figure 3 shows how we map the Forces to the benefits and liabilities in the new pattern.

A force makes the problem difficult to solve. How the solution resolves this difficulty leads to the corresponding consequence. The matching of forces and consequences helps guide us from the problem to the solution.

- Each force must be resolved; thus the force must be matched with at least one consequence.
- Each consequence must be matched with exactly one force. That is, it must be the unique result of the resolution of some force.

- A force may match both a benefit and a liability. A solution must seek to realize the benefit without incurring the liability. In Figure 3 note that the *Code Cluttering* and *Run-time Reconfiguration* forces each match with both a benefit and a liability.

If a force cannot be matched to a consequence, then it is likely part of the context of the problem that is not resolved by the solution.

Matching Forces with Consequences

Transitional Turbulence Reduction: We want to decrease the transitional turbulence in the application's execution to better satisfy the observers' expectations.

- Benefit: A solution coalesces sets of dependent internal events into large-grained events such that the handling of a large-grained event causes the same overall state change as the corresponding set. This can decrease latency and increase accuracy (i.e., decrease the number of errors).

Run-time Reconfiguration: We want to adapt to changes in an application's component architecture at run time.

- Benefit: A solution dynamically adapts to changes in an application's component architecture at run time.
- Liability: Changes to an application's component architecture at run time can increase latency and decrease accuracy.

Startup Cost Inflation: We want to avoid adding significant startup or shutdown costs.

- Liability: An implementation of a solution often causes additional processing overhead at startup and shutdown of the application.

Operational Overhead Creep: We want to avoid adding significant processing overhead during the application's normal operation.

- Liability: An implementation of a solution often causes additional run-time processing overhead, especially when the component architecture changes.

Code Cluttering: We want to avoid significantly complicating the application's design, implementation, testing, or use.

- Benefit: An application can be readily adapted to use the mechanisms that implement the solution.
- Liability: An application must be adapted to use the mechanisms implementing the solution. Modifying the application often complicates its design, implementation, testing, or use.

Figure 3. Matching the Forces with the Consequences.

9. Elaborating the Context

In Step 7 on the pattern writer's path, we elaborate the context in which the problem exists. The context defines "aspects and requirements that are so important that the problem may not exist outside the context but that are, at the same time, not modified by the solution" [15]. The context "imposes constraints on the solution" [16].

Section 3.2 states the basic Context as follows:

(C1) The application is constructed according to the IMPLICIT INVOCATION architectural pattern, assuming nondeterministic but fair handling of events.

Beyond that, we examine what additional assumptions the example applications from Section 4 make about contexts in which they execute.

In the previous discussion of the *Run-time Reconfiguration* force and the corresponding benefit, we assume that the following characteristic holds for the component architectures. We add this to the Context:

(C2) The application's component architecture may change at run time. The application organizes the components into a hierarchical structure. This structure may change dynamically at run time as a result of external stimuli or the actions of components.

In the previous discussion of transitional turbulence, we assumed that the display system operates independently from the application, but accesses the application's data structures before rendering a representation to the display screen. This situation means that periods of transitional turbulence can exist. Thus, we add the following characteristics to the Context:

(C3) The application presents some aspects of its state that can be observed periodically from outside the system. The timing of this presentation is not under the control of the application.

(C4) Because of the asynchronous nature of the application's operation, the externally observable presentation may exhibit periods of transitional turbulence.

The example applications assume that the components encapsulate their states behind interfaces and restrict all accesses to the states to functions defined in the interface [36–38]. A lack of encapsulation would make it more difficult to determine the relationships among the controls. Hence, we add the following to the Context:

(C5) Each component is an information-hiding module with a well-defined interface. The only way to change or access its state explicitly is by calling one of its accessor or mutator procedures (e.g., properties in some object-oriented languages).

The example applications build dependency graphs that record relationships among the controls. They do this dynamically at run time, so we assume that the implementation environment or the application itself allows a program to extract metadata about the components and their interfaces. Thus, we refine the Context to require that some kind of reflection capability be available:

(C6) The application supports reflection capabilities. That is, application-level code can examine the application's features (such as its components, events, event handlers, and hierarchical structure) at run time and extract metadata (such as names, types, and the type signatures of the procedures in component interfaces).

10. Choosing the Pattern Name

In Step 8 on the pattern writer's path, we choose a pattern name. We can use several patterns from Meszaros and Doble [16] to guide us in this task.

- The EVOCATIVE NAME pattern suggests choosing a name that evokes an image that conveys "the essence of the pattern solution to the target audience" [16]. The name should be memorable and suitable for adding to the technical vocabulary of software developers.
- The NOUN PHRASE NAME pattern suggests naming the pattern for the result it creates.
- The MEANINGFUL METAPHOR NAME pattern further suggests choosing a name based on a metaphor that is familiar to the target audience.

We adopt the Pattern Name

DYNAMICALLY COALESCING REACTIVE CHAINS

because it seems to best meet these criteria. It is a noun phrase that metaphorically evokes how the solution achieves transitional turbulence reduction by coalescing a chain (sequence) of events into a single large-grained event at run time. For convenience, we sometimes use the acronym DCRC.

11. Rewriting the Pattern Elements

In Step 9 on the pattern writer's path, we reexamine and rewrite the six pattern elements. At this point in our process, the primary element that needs attention is the Solution, including how it relates to the Problem and the Forces. We need to provide sufficient detail for the reader to use the pattern effectively to design and implement a concrete solution. However, we want to keep the new pattern independent of specific programming language and user interface technologies and do not want to overwhelm the reader with arcane details of particular implementations and implementation technologies.

11.1. Solution-Writing Guidelines

Several of Harrison's pattern-writing patterns [21,22] give us guidance on how to refine the Solution:

- As discussed in Section 5, the BIG PICTURE pattern [21] suggests that the Problem and Solution should "by themselves" convey the key idea—"the big picture"—of the new pattern.
- The MATCHING PROBLEM TO SOLUTION pattern [21] suggests that the Solution should solve the "whole" Problem "but not more".
- The CONVINCING SOLUTION pattern [21] suggests that pattern writers seek to make the Solution "compelling". Often, this means making it "narrower and deeper".
- As discussed in Section 4, the "WHAT"-SOLUTIONS pattern [22] suggests writing the core idea of the Solution in a one- or two-sentence summary placed at the beginning of the Solution description. The "HOW"-PROCESS pattern [22] suggests extending the summary with more detail about "what to do, how to do it, and why to do it that way," including providing any appropriate illustrations. In particular, it should describe how the Solution balances the Forces and identify any Forces that are not considered.
- The FORCES HINT AT SOLUTION pattern [22] suggests that the Forces should guide the reader from the Problem to the Solution.

Because of our goal of keeping the new pattern technology independent, we found satisfying Harrison's CONVINCING SOLUTION and MATCHING PROBLEM TO SOLUTION patterns [21] challenging. The latter required us to tweak the statement of the Context to include subtle assumptions the Solution makes about the environment.

How do the forces hint at the solution? The primary purpose of a solution is to realize the benefit of the *Transitional Turbulence Reduction* force. It does so by using a dependency graph to reorder the updates of the components. To realize the benefit and avoid the liability of the *Code Cluttering* force, the solution works by layering lightweight software mechanisms on top of (i.e., by augmenting) the application's normal event processing system. To avoid the liability of the *Startup Cost Inflation* force, the solution must be able to build the dependency graph efficiently. To avoid the liability of the *Operational Overhead Creep* force, the solution must be able to reorder the events according to their dependencies and process them efficiently. To realize the benefit and avoid the liability of the *Runtime Reconfiguration* force, the solution must be able to detect a change in the component architecture and rebuild the dependency graph efficiently.

11.2. Solution: Summary (from Section 4)

A solution encodes the complex relationships among the application's components in a dependency graph, and then uses the graph to order the updates of the components without violating the dependency constraints. The goal is to reorder the updates of the components so that the new order reduces transitional turbulence without degrading the performance of the system.

11.3. Solution: Definitions

What do we mean by a "dependency graph" in this context?

- If the execution of some component `X` of an application can directly affect a subsequent execution of some other component `Y` in any way, then `Y` *depends on* `X`. For example, `X` might trigger an event for which `Y` listens; change the value of some attribute of its state that `Y` accesses; directly call one of `Y`'s mutator procedures; or create, delete, or modify `Y`.

- A *dependency graph* is a directed acyclic graph formed by placing the components at the nodes and adding a directed edge from some component `Y` to some component `X` only if `Y` depends on `X`.

Figure 4 shows a dependency graph for an application with ten components named with the upper case letters `A` through `J` and directed edges from every component to every other component on which it is directly dependent. The shaded area of the figure includes the six other components that are directly or indirectly dependent on component `A`. Any change in the state of component `A` may require changes in all other components in the shaded area. If each edge is implemented as an event, then six independent events must be processed to propagate the changes to all dependent components. This is the *update cycle*. The event handling system processes these events in a nondeterministic order, interleaved with any other pending events.

To apply the DCRC pattern, we are primarily interested in recording the dependencies related to the implicit invocations—between components that listen for an event and those that trigger the event. Of course, being able to record other kinds of dependencies may also be helpful.

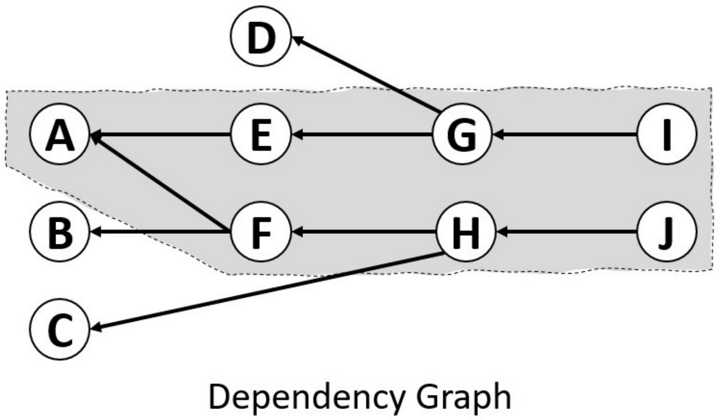


Figure 4. Dependency graph for an application with ten components, illustrating an update cycle.

11.4. Solution: Augmenting the Application

To apply the DCRC pattern to an application that satisfies the Context, we can augment the application with appropriate *software mechanisms*. For example, Figure 5 illustrates how a solution can augment an application's event handling to coalesce dependent events into larger-grained events without modifying the underlying event-handling mechanisms. Beginning with the application's II architecture (shown in panel 1), a solution first determines the dependency relationships between the components (panel 2 and also Figure 4) and then builds the corresponding dependency graph (panel 3). Then it can use the dependency graph to rearrange the component updates in any order that satisfies the dependency constraints (panel 4). In particular, the solution seeks to optimize the processing of an update cycle by performing all the updates in the cycle (the shaded area in Figure 4) directly as part of the processing of the first event.

The software mechanisms may include some combination of libraries, frameworks, tools, and design and programming techniques. The various mechanisms should be *lightweight*. That is, they should execute efficiently and should not require extensive modifications to the existing application. The "software mechanisms" needed and the meaning of "lightweight" depend on the application's specific implementation technologies and performance requirements.

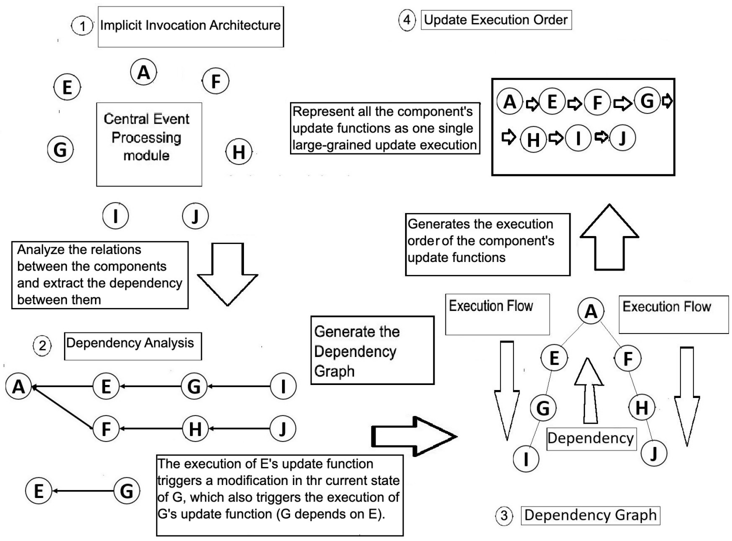


Figure 5. Three-step process to coalesce dependent events by augmenting the event processing.

For applications that satisfy the Context, a developer can augment the application's event-handling mechanisms to solve the Problem. In general, to construct a Solution, the developer needs to design, implement, and install three primary software mechanisms: one to build the dependency graph at startup of the application, one to rebuild the dependency graph when needed during the application's execution, and one to coalesce all the component updates in an update cycle into a single event. The construction of these software mechanisms involves an *augmentation workflow* with three phases:

1. *Augmentation analysis*, which requires analyzing the application to identify how to add the necessary mechanisms;
2. *Augmentation development*, which requires developing (i.e., designing and implementing) the mechanisms;
3. *Augmentation incorporation*, which requires incorporating the mechanisms into the operation of the application.

Figure 6 summarizes the augmentation workflow, restating the tasks as questions to answer.

Augmentation Workflow

1. Augmentation Analysis

(AA1) How to iterate through components?
(AA2) How to extract dependency relationships between components?
(AA3) How to select which components to include in dependency graph?

2. Augmentation Development

(AD1) How to design and implement mechanism to differentiate between components included/excluded from dependency graph?
(AD2) How to design and implement mechanisms to detect whether component architecture or dependencies changed?
(AD3) How to design and implement mechanisms to construct/reconstruct dependency graph?

3. Augmentation Incorporation

(AN1) How to augment application to construct dependency graph at startup?
(AN2) How to augment application to use dependency graph to coalesce processing of event chains?
(AN3) How to augment application to update dependency graph when component architecture changes?

Figure 6. Summary questions for the augmentation workflow.

11.4.1. Solution: Augmentation Analysis

In the augmentation analysis phase, the solution developer must perform the three tasks AA1, AA2, and AA3 to analyze the original application and define the requirements for the new software mechanisms.

(AA1) Examine the hierarchical structure to identify how a program can iterate through the components (i.e., accessing each component exactly once).

(AA2) Examine the design and implementation of the components and the features of the implementation language to identify how a program can extract the dependency relationships between the components at run time.

Task AA2 may involve the use of the existing features of the components or the reflection capabilities of the implementation language. If sufficient capabilities do not exist, we can design lightweight modifications that implement sufficient application-specific capabilities.

(AA3) Examine the components and events to determine which of the relationships between the components to include in the dependency graph and which to exclude. To reduce transitional turbulence, the augmented application program can manipulate the components and relationships included, but cannot manipulate those excluded.

Generally speaking, in task AA3, we include the component relationships arising from the application's custom code (which we can modify if needed) and exclude those in the supporting framework (which we cannot modify). We may also want to exclude any component relationship if that relationship represents an expensive computation or an arbitrary delay.

11.4.2. Solution: Augmentation Development

In the augmentation development phase, the solution developer must perform the three tasks AD1, AD2, and AD3 to design and implement the new software mechanisms according to the requirements specified in the augmentation analysis phase.

(AD1) Design and implement a lightweight run-time mechanism that enables the program to differentiate between the components that are to be included in the dependency graph and those that are not.

Task AD1 involves features already present in the application (e.g., types, value of some property, metadata) or may involve modifying the application to add appropriate features. For example, in an object-oriented system in which the components are objects, we could modify the included components to implement a "marker interface" that can be checked by reflection. The developer should establish a criterion to determine what to include in the dependency graph and what to exclude. In general, this criterion can be defined as a function that is called by the dependency graph-building procedure. It must return a boolean value `true` if its argument should be inserted into the dependency graph and otherwise return `false`.

(AD2) Design and implement a lightweight run-time mechanism that enables the program to detect whether the component architecture or the dependencies among the components have changed since the previous check (or since the beginning of operation).

In this Context, the task AD2 assumes that a change to the hierarchical structure holding the components likely means a change to the component architecture.

(AD3) Design and implement lightweight mechanisms to construct the dependency graph initially and to reconstruct it when needed.

To build a dependency graph in task AD3, the program can traverse the hierarchical structure (e.g., perform a breadth-first traversal of the Document Object Model), placing each component at a node and adding edges to other nodes according to the depends-on relationships between components. However, it must prune the graph appropriately to remove any cycles.

11.4.3. Solution: Augmentation Incorporation

In the augmentation incorporation phase, the solution developer must perform the three tasks AN1, AN2, and AN3 to incorporate the new software mechanisms into the original application. This phase builds on the results of the augmentation development phase. Figure 7 shows how the augmented application can incorporate the three primary software mechanisms into a typical object-oriented GUI application at run time.

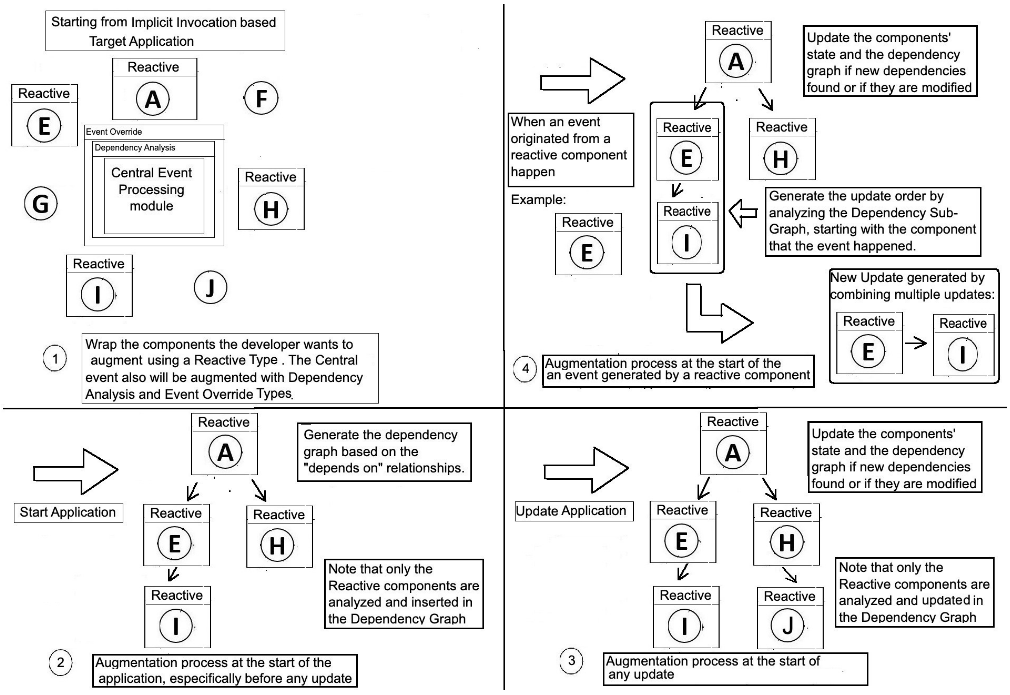


Figure 7. Three-step process to incorporate the augmentation into an existing application.

(AN1) The application must construct the dependency graph at or before startup.

As shown in panel 1 of Figure 7, the augmentation process begins with an IMPLICIT INVOCATION application modified with the basic mechanisms developed in augmentation development task AD1. As shown in panel 2, it then uses the mechanisms developed in task AD3 to build the initial dependency graph at startup.

(AN2) When some component `C` included in the dependency graph signals an event `E`, the application must *intercept* `E` and directly call the procedures associated with event `E` on all listening components as recorded in the dependency graph. Then it must recursively apply the process to all events signaled by the listening components. This continues as long as there are dependencies indicated in the graph (which cannot have cycles). This process dynamically coalesces the processing of chains of events into what is processed as one "large-grained" event. The meaning of "intercept" depends on the specific application's implementation technologies.

As shown in panel 4 of Figure 7, the augmented system combines the updates of all the components in the update cycle into a sequence of direct procedure calls. This coalesces the processing of a whole chain of events into a single event.

(AN3) After processing each "large-grained" event in the previous step, the application must check whether the application's component architecture has changed (e.g., the addition, modification, or deletion of any component in the hierarchical structure) or the dependencies among components have changed. If so, then the dependency graph must be updated appropriately to reflect the new component architecture.

As shown in panel 3 of Figure 7, at the beginning of any update, the augmentation process uses the mechanisms developed in augmentation development task AD2 to determine whether the dependency graph needs to be rebuilt. If so, it uses the mechanisms developed in task AD3 to rebuild the graph.

11.4.4. Solution: Balancing the Forces

In the Solution described above, we handle all the identified Forces. How do we balance the various Forces to achieve this Solution?

Transitional Turbulence Reduction

For a state change in any component, the augmented application must propagate the effects to all its directly or indirectly dependent components without the delays and nondeterminism introduced by the normal event-handling system—as if all were part of the processing of one large-grained event. This can decrease latency and increase accuracy (i.e., decrease the number of errors).

Run-Time Reconfiguration

Frequently during normal operation of the application, the augmented application checks if its component architecture has changed. If it detects a change, it then reconstructs the dependency graph to reflect the new architecture. The extra costs incurred in reconstructing the dependency graph must not itself worsen the solution's overall effect on the latency and accuracy.

Changes to an application's component architecture during normal operation can increase latency and decrease accuracy. However, a good solution must dynamically adapt to such changes and seek to mitigate the effects on latency and accuracy.

Startup Cost Inflation

When applying the pattern, developers should seek to keep the cost of initially constructing the dependency graph low. The developers should carefully select the components to include in the analysis and use efficient methods to determine dependency relationships and build the graph.

The augmented application likely incurs additional processing overhead at startup and shutdown. In particular, the extra costs for constructing the initial dependency graph should be small in proportion to the potential accuracy and performance gain in an application that runs sufficiently long.

Operational Overhead Creep

The augmented application likely incurs additional processing overhead during normal operation, especially when the component architecture changes. In particular, the extra costs for checking for changes in the component architecture and reconstructing the dependency graph should be small in proportion to the potential accuracy and performance gain in an application that runs sufficiently long. In cases in which the component architecture changes infrequently, the augmented application should incur minimal costs.

Code Cluttering

To implement a solution, the developer must augment the existing application by incorporating a set of software mechanisms as described above. Unfortunately, modifying the application often complicates its design, implementation, testing, and use.

However, in a good design and implementation of the solution's new software mechanisms, it should be possible to readily augment the existing solution. Therefore, the new software mechanisms must be carefully designed, implemented, and documented so that the solution can work well with typical application designs.

For example, for a typical GUI application, it should be possible to implement the solution approach as a software framework with wrapper classes for the controls and a library that implements the algorithms to build/reconstruct the dependency graph and uses it to coalesce chains into “large-grained” events.

12. Putting the Elements in Standard Order

In Step 10 on the pattern writer’s path, we put the pattern elements in the standard order as defined in Section 2. We seek to organize the DCRC pattern according to the SINGLE-PASS READABLE [16], SKIPPABLE SECTIONS [16], and FINDABLE SECTIONS [16] patterns. That is, we seek to write the pattern so that it flows smoothly from Context to Consequences, capable of being read sequentially and understood in one pass. We also seek to indicate the pattern’s elements clearly and enable readers to skip past elements or detailed descriptions when they are trying to understand and use the pattern.

As shown in Appendix B, the full pattern description consists of the Pattern Name from Section 10, the refined Context from Section 9, the Forces from Section 7, the refined Solution from Section 11, and the Consequences from Section 6 restructured to show the mapping from the Forces in Figure 3. For a presentation of the pattern separate from this paper, it might be helpful to provide additional elements such as those suggested by the GLOSSARY [16] and OPTIONAL ELEMENTS WHEN HELPFUL [16] patterns.

13. Applying the Pattern

In Sections 3–12, we elaborated the DCRC software pattern by following the steps on the writer’s path, applying relevant pattern-writing patterns along the way. In this section, we first demonstrate the technical feasibility and efficacy [4,39] of the DCRC pattern by applying it to an arbitrary GUI application developed using C# and the built-in .NET GUI framework.

A software pattern corresponds to a design science technological rule [40] of the form:

To solve an instance of the Problem in the Context apply the Solution.

Thus, to apply the DCRC pattern, we must show that the beginning application satisfies its Context and Problem, then we modify the application as described by its Solution.

13.1. Satisfying the Context and Problem

Section 9 presented the DCRC pattern’s Context, which captures six characteristics that must be satisfied by the problem’s environment. We list those below and identify the features of a C#/.NET GUI application that satisfy each characteristic.

(C1) The application is constructed according to the IMPLICIT INVOCATION architectural pattern, assuming nondeterministic but fair handling of events.

(C2) The application’s component architecture may change at run time. The application organizes the components into a hierarchical structure. This structure may change dynamically at run time as a result of external stimuli or the actions of components.

The structure and operation of the built-in event handling system of the C#/.NET GUI follows the IMPLICIT INVOCATION architectural pattern. The GUI’s “components” are its controls, each of which is represented by an object. The GUI arranges the objects representing the controls into a hierarchical data structure internally, for example, the Document Object Model (DOM) in a Web application. This data structure forms the “component architecture”. It may change as the result of some action from outside the GUI or by execution of the GUI’s controls themselves. Thus, a C#/.NET GUI application satisfies the characteristics C1 and C2.

(C3) The application presents some aspects of its state that can be observed periodically from outside the system. The timing of this presentation is not under the control of the application.

(C4) Because of the asynchronous nature of the application's operation, the externally observable presentation may exhibit periods of transitional turbulence.

A C#/.NET GUI application consists of controls that are executed asynchronously and communicate through the event handling mechanism. Due to the fine-grained nature of the events, it may be necessary to process many events to propagate the changes at one control to all other controls. However, the display system operates independently from the GUI and directly accesses the GUI's data structures. Thus, a C#/.NET GUI application can exhibit transitional turbulence and therefore satisfies characteristics C3 and C4.

(C5) Each component is an information-hiding module with a well-defined interface. The only way to change or access its state explicitly is by calling one of its accessor or mutator procedures (e.g., properties in some object-oriented languages).

In a C#/.NET GUI application, each control is an object that instantiates a class from the `Control` class hierarchy. This object implements its class's interface and encapsulates (i.e., hides) all its attributes. Thus, the only way for another object to access or alter a control's internal state is to call a method on its interface. Some of the control's methods are associated with the operation of the event-handling system. Therefore, a control is an "information-hiding module" with a well-defined interface [36,37,41]. Thus a C#/.NET GUI application satisfies characteristic C5.

(C6) The application supports *reflection* capabilities. That is, application-level code can examine the application's features (such as its components, events, event handlers, and hierarchical structure) at run time and extract metadata (such as names, types, and the type signatures of the procedures in component interfaces).

The primary programming language of the .NET framework is the object-oriented language C#. The language's extensive reflection facilities enable a program to examine its objects at run time and extract metadata about their features (e.g., the names, types, and values of attributes, the names and type signatures of methods, the types of objects, and the classes and interfaces extended by classes). Thus a C#/.NET GUI satisfies characteristic C6.

Therefore, a .NET GUI application satisfies the Context of the DCRC software pattern. Now let us consider the pattern's Problem element, which states:

> We want to eliminate or reduce the length of the periods of transitional turbulence during which the external presentation does not accurately reflect the state of the application. We need to do this without sacrificing performance. The goal is to better satisfy observers' expectations by increasing the accuracy of the external presentation.

As we noted above, a C#/.NET GUI application can exhibit transitional turbulence. This can cause the GUI display to inaccurately reflect the state of the application for periods of time. In some circumstances, we may want to eliminate or reduce the length of these periods without sacrificing performance. Therefore, the DCRC pattern addresses a problem that is relevant for C#/.NET GUI applications.

13.2. Constructing a Solution

Given an arbitrary GUI application that uses the built-in C#/.NET GUI framework and satisfies the DCRC pattern's Context and Problem elements, we now show how to modify the application to achieve a Solution. Section 11 states the basic requirement in the pattern's Solution summary as follows:

> A solution encodes the complex relationships among the application's components in a dependency graph and then uses the graph to order the updates of the components without violating the dependency constraints. The goal is to reorder the updates of the components so that the new order reduces transitional turbulence without degrading the performance of the system.

For a C#/.NET GUI application, we say that a control `B` "depends on" a control `A` if the execution of `A` can directly affect a subsequent execution of `B` in any way. We form the "dependency graph" by placing the controls at the nodes and adding a directed edge from one node to another if the corresponding components have a depends-on relationship.

The pattern's Solution calls for us to augment the C#/.NET GUI with "lightweight software mechanisms" to reduce transitional turbulence without extensive modification of the built-in event handling mechanism. The required mechanisms may include C# interfaces, classes, methods, functions, data structures, or combinations thereof. The pattern's Solution element describes the construction of these software mechanisms using an augmentation workflow consisting of three phases:

1. augmentation analysis, which requires analyzing the application to identify how to add the necessary mechanisms
2. augmentation development, which requires developing (i.e., designing and implementing) the mechanisms
3. augmentation incorporation, which requires incorporating the mechanisms into the operation of the application

To show that we can construct the desired software mechanisms, we answer the questions given in Figure 6.

13.2.1. Augmentation Analysis

In the augmentation analysis phase, we must perform the three analysis tasks AA1, AA2, and AA3 given in Section 11.4.

(AA1) How can we enable a C#/.NET GUI application to iterate through its components?

A GUI application provides a hierarchical collection of its controls. For a Web application, this collection holds the Document Object Model (DOM). For a desktop application, the `Designer` class holds a collection of controls as objects of class `Control` or one of its subclasses. We can augment the application to iterate through this collection and examine each of its controls, as task AA1 requires.

(AA2) How can we enable a C#/.NET GUI application to extract the dependency relationships between its components?

C#'s `Type` class enables a program to examine any of its objects and extract metadata about their features, including the names and type signatures of its methods and the names, types, and values of its attributes. We can augment the application to examine its control objects to determine the dependency relationships among them, as task AA2 requires.

Suppose `A` and `B` are two controls in the GUI. If one of control `A`'s attributes holds a reference to control `B` or one of `A`'s methods has a formal parameter of type `B`, then control `B` depends on the control `A`.

(AA3) How can we enable a C#/.NET GUI application to select which components to include in and exclude from its dependency graph?

We likely should exclude any control from the event processing optimization that we cannot modify, such as a control that is part of of the .NET system or a third-party library. We should also exclude any control that requires excessive execution time. We should include all other controls, which we call the "reactive" controls. These controls and their interrelationships must be included in the dependency graph.

To allow us to designate a control as reactive, we define a C# `interface` named `iReactive` that declares a special event handling method `reactiveUpdate()`. We require that any class that implements `iReactive` defines an appropriate method body for `reactiveUpdate()`. If we need to mark an existing control as reactive, we can "wrap" the object with an instance of a class that implements `iReactive`. When the augmented application builds the dependency graph, we can include all controls that implement `iReactive` and exclude all those that do not.

13.2.2. Augmentation Development

In the augmentation development phase, we must perform the three design and implementation tasks AD1, AD2, and AD3 given in Section 11.4. In this phase, we use the results of the augmentation analysis above to design and implement the mechanisms. We then incorporate the resulting mechanisms into the GUI application as described in Section 13.2.3.

(AD1) How can we design and implement a mechanism for a C#/.NET GUI application to differentiate between the components to be included in and excluded from the dependency graph?

We require that all reactive control objects implement the `iReactive` interface as described in Section 13.2.1 above. We then develop a mechanism (e.g., a function) that uses the C# reflection facilities to determine whether or not an object implements the `iReactive` interface. (If an object does, it must be included in the dependency graph; if it does not, it must be excluded from the dependency graph.)

(AD2) How can we design and implement a mechanism to detect whether a C#/.NET GUI application's component architecture or the dependencies among the components have changed?

To determine whether the GUI has changed, we develop a mechanism to store a snapshot of the GUI's structure at the beginning of an update cycle (as defined in Section 13.2.3 below). The snapshot consists of the dependency graph, where each node has a reference to its associated control object. At the end of the update cycle, the mechanism must check whether the GUI structure has changed since the beginning of the cycle. In particular, it must detect GUI changes that modify existing dependencies or add new ones.

To determine whether there are changes in the dependencies, the mechanism must examine each reactive control. If that control did not appear in the previous snapshot, then the dependency graph is no longer valid. (To compare two control objects, a C#/.NET program checks if they have the same name and type.) If that control did appear in the previous snapshot and any of its dependencies have changed, then the dependency graph is no longer valid. To check whether a control's dependencies have changed, the mechanism examines the control's attributes and methods using C#'s reflection facilities. If any control appears in the previous snapshot but not in the current GUI, then the dependency graph is no longer valid. If nothing has changed from the previous snapshot, then the dependency graph remains valid. If the dependency graph is no longer valid, then it needs to be rebuilt.

(AD3) How can we design and implement the mechanism to construct the dependency graph initially and to reconstruct it when needed?

As discussed in Section 13.2.1 above, a .NET GUI consists of a hierarchical collection of control objects. A C# program can iterate through this collection and examine each control using C#'s reflection facilities. If some control object `C` implements the `iReactive` interface (meaning it should be included in the dependency graph), then the mechanism inserts a new node for `C` into the dependency graph. For every other control that depends on `C`, the mechanism inserts a directed edge from that node to `C`'s node.

The algorithms are essentially the same for the initial construction of the dependency graph and for the graph's reconstruction because of a change in the GUI's structure. The reconstruction is different in that it only needs to iterate through the controls referenced by the previous dependency graph.

13.2.3. Augmentation Incorporation

In the augmentation incorporation phase, we must perform the three application augmentation tasks AN1, AN2, and AN3 given in Section 11.4. In this phase, we take the software mechanisms developed in Section 13.2.2 above and incorporate them into the GUI application.

(AN1) How can we augment the C#/.NET GUI application to construct the dependency graph at or before startup?

A C# GUI is an instance of the class `Form` or one of its subclasses. This class defines an event handler method `form_start()` that executes during the `Form`'s instantiation—after it instantiates all its controls but before it renders the form to the display. We modify this method to incorporate the construction of the dependency graph into the event handling system.

To designate a `Form` as reactive, we define a C# `interface` named `iUpdatable` that declares a `form_start()` method. We require that any class implementing `iUpdatable` defines an appropriate method body for `form_start()`. Using the mechanisms developed in Section 13.2.2 above, this method must examine the GUI and construct the initial dependency graph as an object in the `Form` subclass.

(AN2) How can we augment the C#/.NET GUI application to use the dependency graph to coalesce the processing of chains of events into a "large-grained event"?

As discussed in Section 13.2.1, we define an interface `iReactive` that declares a special event handler method `reactiveUpdate()`. Any `Control` subclass that implements `iReactive` must define `reactiveUpdate()` to have appropriate behavior (which may include the behavior of the built-in event handler method `Update()`). The reactive control class must also override the built-in `Update()` method so that it calls the `reactiveUpdate()` method instead of the control's standard event handling code.

If a reactive control responds to an external (e.g., user interaction) event, then the built-in event handler method `Update()` must detect the external event and redirect its handling to the augmented event handler method `reactiveUpdate()`. Based on the constraints in the dependency graph, this method constructs a sequence of updates of the dependent controls. Then it explicitly invokes the `reactiveUpdate()` methods of each control in the sequence. This process thus propagates the effects of one external event throughout the GUI. From the standpoint of the built-in event handling system, this whole sequence of updates is executed as one "large-grained" event in the augmented GUI application.

(AN3) How can we augment the C#/.NET GUI application to update the dependency graph when the component architecture changes during operation?

We must modify the application so that, at the end of the update cycle, it checks whether the GUI structure has changed since the beginning of the cycle (as described in Section 11.4.2 above). If the GUI has changed, we must rebuild the dependency graph (using mechanisms from Section 11.4.2). We add this check to the augmented event handler `reactiveUpdate` (as described above). It must do this immediately after inferring the execution order from the dependency graph and calling the `reactiveUpdate()` methods of the dependent controls.

13.3. Evaluating the Pattern

Section 13.1 argued that an arbitrary GUI application developed using C# and its built-in .NET GUI framework satisfies the DCRC pattern's Context and Problem elements. Section 13.2 then demonstrated how to modify that application to achieve a Solution. Furthermore, the original Marum et al. .NET GUI case study [6,8] discussed in Section 1 implemented and tested several such applications. Thus, the DCRC pattern is technically feasible.

To investigate the efficacy of the pattern's Solution in reducing transitional turbulence, we can examine the results of the Marum et al. .NET GUI case study [8]. The case study developed a library that embodied the solution approach. The library carries out the dependency graph construction at startup, the graph's reconstruction when needed, and the coalescing of events at run time.

Using the library, the case study developed three different application scenarios (i.e., three different, relatively complex self-completing forms) for the two different user interface platforms (i.e., desktop and Web) and conducted a set of experiments comparing the unmodified .NET GUI applications against the augmented .NET GUI applications. The experiments investigated how transitional turbulence and performance were affected. That is, they evaluated how well the augmented applications balanced the *Transitional Turbulence Reduction* force against the conflicting *Startup Cost Inflation*, *Run-time Reconfiguration*, and *Operational Overhead Creep* forces.

The experiments measured the startup costs for each application [8]. The average startup costs for the augmented .NET GUI applications was 2.6 times the average startup costs of the corresponding unmodified applications (i.e., 55 ms versus 21 ms). Therefore, as expected, there was *Startup Cost Inflation* that had to be mitigated by overall improvements in performance and transitional turbulence reduction.

The experiments also measured the overall execution time for each application, which included times for start-up, operational overhead, run-time reconfiguration, and component execution [8]. The augmented .NET GUI applications executed in about half of the total time required by the corresponding unmodified .NET GUI applications. Thus, for these experiments, the *Startup Cost Inflation*, *Run-time Reconfiguration*, and *Operational Overhead Creep* forces were appropriately balanced.

The experiments characterize transitional turbulence by determining the average number of errors per cycle and the number of cycles required for transitional turbulence to subside [6,8]. One of the application scenarios shows a decrease in the average errors per cycle by 80% from the corresponding unmodified .NET GUI application to the augmented .NET GUI application (i.e., from five errors to one). The other two application scenarios showed only one error per cycle for both the unmodified or augmented .NET GUI applications. The three application scenarios showed that it took an average of 75% fewer cycles for the transitional turbulence to subside in the augmented .NET GUI applications than it did in the unmodified.NET GUI applications (i.e., one cycle instead of four). Therefore, these experiments show that when transitional turbulence exists in an unmodified application, the augmented application exhibits a reduction in transitional turbulence, which shows that the three performance-related forces are in balance. The augmented applications decreased both the transitional turbulence and the overall execution time.

The experiments did not measure to what extent the design and implementation of the augmented applications became more complex [6,8]. However, by expressing the solution as a separate library and a set of wrapper classes for the C#/.NET GUI components, the augmented applications seem to have kept the added complexity small compared to the significant improvements in transitional turbulence and overall performance.

These experiments indicate that the Solution provided by the DCRC pattern is efficacious in a variety of circumstances. That is, it can reduce transitional turbulence in situations where it exists. In a separate case study, Marum et al. [5] demonstrated the feasibility and efficacy of the solution approach for a variety of VR applications using Unity3D. Of course, more experiments should be conducted on the C#/.NET GUI and other applications of the pattern to explore the efficacy of the pattern more fully.

This research extracted the DCRC pattern from two specific Marum et al. case studies and elaborated it systematically using the writer's path. We sought to capture all the assumptions the case studies made about the user interfaces and event-handling systems in the Context element. As a result, we expect the pattern to be applicable to any application that satisfies the Context.

In this section, we have argued that the DCRC pattern is applicable to an arbitrary C#/.NET GUI application. We expect that it is also applicable in other situations (e.g., other applications, languages, and user interface technologies) that satisfy the context. Additional experimentation and evaluation will be needed to determine whether that is indeed the case. If not, it may be necessary to evolve the pattern's context and other elements to handle the additional situations.

14. Discussion

In this section, we reflect on this pattern-writing research, examine related work, and suggest future research on the DCRC pattern and writer's path methodology.

14.1. Evolving the Pattern

The final step on the pattern writer's path is to evolve the pattern based on feedback and experience. The patterns community [42] often uses a process called *shepherding* to assist pattern writers [21,22]. This is a "process in which a pattern author receives feedback from another, experienced pattern author" [15]. It is an iterative process in which the experienced writer—the "shepherd"—gives feedback to the pattern's author—the "sheep". Harrison's THREE ITERATIONS [21] pattern suggests that approximately three rounds of feedback and revision are required. Often, this coaching is associated with a conference such as Pattern Languages of Programs (PLoP) [42].

We incorporated a feedback mechanism into our pattern writing process. The DCRC pattern is being written by a diverse four-person team. Two members of the team have expertise in software architecture and two in the application areas and technologies underlying the two case studies. The pattern authors include one member from each group, and the pattern reviewers include the other member from each group. Three members of the team were involved in the initial case studies [5,6] and one was unfamiliar with the case studies before joining the team. The team seeks to further revise the DCRC pattern as it gains more experience using the pattern and writing other patterns.

In the future, we plan to continue to evaluate the generality of the DCRC pattern by conducting new case studies or replicating previous case studies using different programming languages and user interface technologies. For example, we plan to replicate the .NET GUI case study using Java and JavaFX [43,44] and the VR case study using C++ and the CryEngine game engine [45].

In the DCRC pattern description, we seek to specify a design pattern with a relatively broad context. It would have been easier for us to specify idioms for the narrower contexts of C#/.NET and Unity3D by drawing on our understanding of the work in the Marum et al. case studies [5,6,8]. As work on the pattern continues in the future, it may be useful to specialize the general pattern to specific technologies, which may enable the definition of related idioms that are much simpler and more straightforward to apply than the current description of the DCRC pattern.

Software patterns are, in some sense, always works in progress that can incorporate "deeper experience gained when applying patterns in new and interesting ways" [46]. In particular, they may be refined to form part of a *pattern language*—"a network of interrelated patterns that defines a process for resolving software development problems systematically". A pattern language combines a *vocabulary*—a set of evocatively named patterns—with a *grammar*—the rules for combining individual patterns into valid sequences in which they can be applied.

A better understanding of the DCRC pattern may enable the definition of a network of more specific patterns that can be woven into a pattern language [46–48]. For example, the relatively complex Solution element suggests that refactoring into several fine-grained patterns would be helpful. In addition, studying common variations of the event-driven, IMPLICIT INVOCATION architecture and implementation platforms can potentially lead to a family of related patterns.

14.2. Reflecting on the Writer's Path Methodology

As our primary methodology for writing patterns, we adopted the writer's path from Wellhausen and Fiesser's tutorial [15], integrating other established methods [16,20–22,46] where helpful. The writer's path is promoted as accessible to novice pattern writers. However, we could not find detailed examples of its use, so we chose to record our steps along the path systematically for the possible benefit of other pattern writers. This record also gives us the opportunity to identify challenges and issues for possible future research related to the methodology.

- As we began to write the DCRC pattern, we observed that the Context, Problem, and Solution elements were entangled with each other and with the incidental details of the implementation technologies, the nature of the application domains, the specific program implementations, and the history of their development. In future research, we suggest that steps 1–3 on the writer's path be refined further to help writers articulate clear, precise Context, Problem, and Solutions descriptions at an appropriate level of abstraction.
- To match the Forces with the Consequences, we found it necessary to refactor both the Forces and the Consequences to ensure that the issues were covered in compatible ways. In future research, we suggest that writer's path steps 4–6 be enhanced to help pattern writers identify the Forces and Consequences and state them compatibly.
- As stated in Section 11, the DCRC pattern's Solution element is complex and, thus, difficult to understand. In future research, we suggest that the writer's path be refined further to guide pattern writers in extracting essential information from existing solutions, narrowing the scope, and simplifying the pattern description.
- As we were rewriting the Solution (in writer's path Step 9), we found it necessary to revise the Context to capture several subtle assumptions made by the full Solution description (e.g., support for reflection). In future research, we suggest that the earlier steps of the writer's path be enhanced to help pattern writers identify the Solution's assumptions about the application's environment.
- The writer's path methodology does not currently address how to collect feedback from users and incorporate changes into the pattern, except by repeating the relevant steps. In future research, we suggest extending the methodology to guide pattern writers during this maintenance phase of the pattern life cycle, in particular, on when and how to evolve a pattern into a pattern language [46–48].

Although some researchers have begun efforts to better ground pattern writing in the scientific method [49,50], our purpose in this paper is pragmatic. We use the pattern-writing process to help us systematically deconstruct a set of related applications to reveal their hidden common structure, separating the essential features of the solution from the incidental features of the implementations. For the cases we studied, the writer's path enabled us to capture this structure and draft an appropriate software pattern. In the future, it may be useful to revise the writer's path methodology to incorporate the insights of Riehle et al. [50], Iba [19], design science methodologists [4,40], and others.

14.3. Leveraging Related Research

There are many different variations of the IMPLICIT INVOCATION software pattern. These include the simple OBSERVER [14], advanced OBSERVER [14], revisited OBSERVER [51], EVENT NOTIFICATION [52], and PROPAGATOR [53] design patterns. Mijač et al. [29] evaluates these patterns extensively and finds "a great similarity between considered design patterns, especially in their overall idea and intent", but identifies "features that should be considered when dealing with complex propagation scenarios." In subsequent work, Mijač et al. [4] proposes the REACTOR design pattern to include these improved features. Mijač et al. [35] then incorporates these ideas into an application framework named REFRAME, which "provides built-in abstractions, mechanisms and tools for handling reactive dependencies" in the C#/.NET context.

Both the REACTOR and the DCRC design patterns place the dependency graph in a central role, but the contexts of the two patterns differ. The DCRC pattern seeks to augment an existing user interface implementation (such as one that uses the built-in C#/.NET GUI) by adding software mechanisms that build the dependency graph and use it to reduce transitional turbulence. Future enhancements to the DCRC can take advantage of insights from the REACTOR pattern. Similarly, if restricted to specific languages and user interface technologies, the ideas of the DCRC pattern can potentially form the basis of an application framework, as REFRAME does with the REACTOR pattern.

The Marum et al. case studies [5,6] included comparisons of (what we now call) the DCRC solution approach with similar applications developed using the reactive programming packages Sodium [9], Rx.NET [10], and UniRx [12]. In the future, these and other reactive programming approaches [7] should be examined more closely to determine what new ideas can be incorporated into a future revision of the DCRC pattern. Of interest are approaches such as FrTime [3], functional reactive programming [32], FlapJax [33], Elm [30], Distributed REScala [31], and DOM-based functional reactive programming [34].

Our primary focus in this paper has been on defining a pragmatic design pattern that is useful to both practitioners and researchers. To date, we have not paid attention to formulating a formal specification or model. However, the ongoing work to evolve the pattern's Solution could benefit from a better formal understanding of the IMPLICIT INVOCATION architectural pattern and its variants. These have been the focus of formal methods research using a variety of different formalisms, including Z notation [23], process algebra and trace semantics [54], temporal logic [55], model checking [56,57], aspect-oriented programming concepts [58], pattern contracts [59,60], category theory [61], and colored Petri nets [62].

15. Conclusions

Two recent studies addressed the problem of reducing transitional turbulence in applications developed in C# on .NET. The first investigated this problem in desktop and Web GUI applications [6,8] and the second in virtual and augmented reality applications using the Unity3D game engine [5]. The studies used similar solution approaches, but both were somewhat embedded in the details of their applications and implementation platforms.

To answer question RQ1 posed in Section 1, we examined these two families of applications, extracted the common aspects of their problem definitions and solution approaches, and codified this problem-solution pair as a new software design pattern named DYNAMICALLY COALESCING REACTIVE CHAINS (DCRC). We developed the pattern incrementally in Sections 3–12 and then demonstrated its technical feasibility and efficacy in Section 13. In Section 14, we discuss related work and how the DCRC pattern might evolve in the future. This pattern enables the problem-solving approach to be reused in a range of related applications and implementation technologies. This work lays a foundation for further research on transitional turbulence and related software architecture issues.

To answer question RQ2 posed in Section 1, we adopted the writer's path methodology from Wellhausen and Fiesser's tutorial [15] to write new software patterns in a step-by-step manner. We outlined the writer's path in Section 2 and then, in Sections 3–12, followed the path systematically to write the DCRC pattern, carefully recording our reasoning at each step. In Section 14, we discuss related work and possible future enhancements to the writer's path methodology. There are many published patterns, but few well-documented examples of how those patterns were written. The writer's path methodology and detailed example in this paper can assist future pattern writers in navigating through the complications and subtleties of the pattern-writing process. By examining the use of the methodology in this example, we also identified ways in which the methodology itself can be improved.

Writing software patterns is a pragmatic art that has been practiced successfully for more than three decades. It is not possible to capture all the useful processes for pattern writing in one simple software engineering methodology. Even if that were possible, it probably would not match the thinking styles of all software engineers. However, carefully worked examples with thoughtful reflection on the thinking processes involved can be quite useful to others who seek to write or update patterns. That was a motivation for this paper's attention to detail.

Author Contributions: Conceptualization (seminal VR case study) J.P.O.M. and J.A.J.; conceptualization (subsequent GUI case study) J.P.O.M. and H.C.C.; conceptualization (DCRC design pattern) J.P.O.M., H.C.C. and Y.L.; software J.P.O.M.; methodology (case studies) J.P.O.M., J.A.J. and H.C.C.; methodology (enhanced writer's path) H.C.C., Y.L. and J.P.O.M.; writing (dissertation chapter) J.P.O.M.; writing (journal paper draft) H.C.C.; writing (revising and editing) all authors; and supervision H.C.C. and J.A.J. All authors have read and agreed to the published version of the manuscript.

Funding: The first author's (J.P.O.M.) work was supported by CAPES, Coordination for Enhancement of Academic Level Individuals—Brazil and by the University of Mississippi Department of Computer and Information Science.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Acknowledgments: A preliminary version of this work appeared as Chapter 5 of the first author's dissertation [63], which was supervised by the second and third authors. The authors are grateful for constructive feedback from reviewers who have evaluated previous drafts of this paper. We especially thank Cesar Rego of the University of Mississippi for his extensive comments.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AR	Augmented Reality
DCRC	Dynamically Coalescing Reactive Chains design pattern
GUI	Graphical User Interface
II	Implicit Invocation architectural pattern
MVC	Model-View-Controller design pattern
VR	Virtual Reality

Appendix A. Examining Two Existing Solutions

In Step 2 on the pattern writer's path, we examine existing solutions. Our primary objective in writing a new pattern is to unify the solutions that emerged from two related case studies:

- Dynamic Web and desktop graphical user interface (GUI) applications implemented with C# on the .NET platform [6,8]
- Dynamic virtual reality (VR) and augmented reality (AR) applications implemented in the Unity3D game engine using C# [5].

Appendix A.1. Dynamic GUI Application

In the first case study, Marum et al. [6,8] explored the issue of transitional turbulence occurring in Web and desktop GUI applications implemented with C# on the .NET platform.

In this environment, a GUI consists of a loosely coupled collection of controls (i.e., the components in the IMPLICIT INVOCATION architectural pattern). Each control responds to events in which it is "interested". A response to an event may result in the control changing its state and triggering new events that notify other controls of the state change. Thus, one control responding to one event may trigger chains of events affecting several other controls in the GUI. In complex cases, these event chains may be long; reaching a stable state may require the processing of many events. The propagation of events is performed by an event-handling layer of the system, not by the controls themselves. Therefore, from the perspective of an application developer, the order in which events are handled is nondeterministic.

Although a GUI's controls are loosely coupled from a communication perspective, an implementation usually arranges them into some hierarchical data structure. For example, the controls within a Web-based GUI are organized by the Document Object Model (DOM) within a browser. Similarly, controls within a C# desktop GUI are organized by a separate class named `Designer`; this class abstracts the user interface's visual representation and contains a hierarchical set of controls. The display system uses these data structures when it periodically renders the GUI on the screen.

This is where transitional turbulence can arise. The processing of a long chain of events may span several cycles of the display system. A control may be rendered with a state that is inconsistent with the states of other controls. This may result in displays that are temporarily inaccurate or misleading from the perspective of a human user's expectations of the user interface's behavior. To combat this problem, this case study developed a reactive programming [7] approach that analyzes the complex relationships among the GUI controls, encodes these dependencies into a dependency graph, and then uses the graph to rearrange the updates of the controls in an order consistent with the dependency constraints. It builds the graph when the GUI starts up and then rebuilds it whenever it detects that the dependencies might have changed. The approach thus coalesces the processing of a chain of what may be several events in the unmodified system into a single large-grained event that updates the states of many controls at once.

Due to the nature of the display system, the approach cannot totally eliminate the transitional turbulence that can cause inaccurate or misleading displays. However, coalescing multiple events into large-grained events does potentially decrease the number of inaccuracies displayed for the rendered state of the system by simplifying the isolated updates of each individual component of the system and agglutinating them into larger execution flows consisting of several components linked into a chain. So, even though the approach makes the code more complex, it flattens the multiple execution flows into a single flow while maintaining the overall performance of the system in terms of starting and update processing times.

To evaluate the approach, this case study developed a prototype library and used it to perform several experiments [6,8]. The experiments involved both desktop and Web versions of three different forms that self-complete (i.e., compute the values in some fields from values supplied in other fields). The experiments performed an automated test a large number of times on each form and measured the startup time, the total time, and the total number of inaccuracies (i.e., errors) when compared against the predicted visual and overall state of the system after the chain of events occurred. Marum et al. [6,8] compared the approach with other approaches that used the .NET GUI library. Figure A1 shows, in general, how the case study's approach constructs the dependency graph for this GUI application and modifies the GUI's event handling mechanism accordingly.

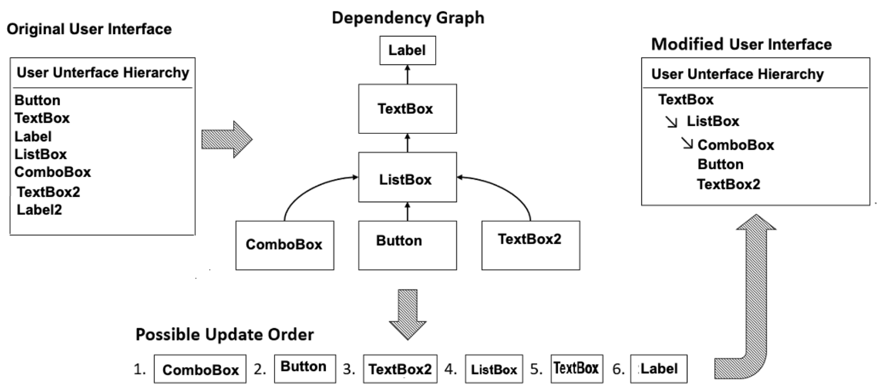


Figure A1. Constructing a dependency graph for the dynamic GUI and determining a new update order.

The experiments indicate that, on average, the approach requires less total time and exhibits fewer visual inaccuracies, at the cost of a modest increase in startup time compared to the three alternatives. Each application developed with the prototype library required approximately 2.6 times as much time to start up as the corresponding unmodified .NET application required. However, it was able to complete the entire chain of form updates in about half of the time the corresponding unmodified .NET application required. Furthermore, it exhibited significantly fewer visual inaccuracies for complex self-completion forms than the corresponding unmodified .NET application exhibited. Based on the results of their experiments, they concluded that their approach improves performance and results in a more accurate behavior in many situations.

The Marum et al. solution approach seems to work well for the kind of problems envisioned and the technologies used in the first case study [6,8]. Given how the approach works, it seems reasonable that it will work for other applications requiring similar solutions. In the next subsection, we examine a related case study that uses somewhat different technologies, seeking a more precise understanding of the general solution and the problem it solves.

Appendix A.2. Dynamic VR Applications

In the second case study, Marum et al. [5] extended and systematized the research from a preliminary study [64] conducted a few months earlier. This preliminary work also motivated the case study that we examined in the previous subsection.

The case study sought to eliminate the instability corresponding to the transitional turbulence that occurs in virtual reality (VR) and augmented reality (AR) applications implemented with C# in the popular Unity3D game engine [11]. These applications are inherently *reactive* and *nondeterministic* with respect to how and when the internal mechanism will execute such events and eventually deliver the resulting state.

Whenever multiple game objects in the simulated scene interact with each other, it may take several cycles for the VR/AR application to update the states of all components and reach stability. As we discuss for the first case study, this is called transitional turbulence or, sometimes, the "ripple effect". Transitional turbulence can result in inconsistencies in what is displayed for the user, which may lead to inconsistent and misleading states within the VR/AR application, making the entire application seem unreliable and unpredictable. The approach focuses on reordering the execution of events so that the "ripple effect" can often be resolved within one update cycle.

Much of the nondeterminism is due to the unpredictable nature of the user interactions, but some of it is due to the lack of the application developer's control over some aspects of the execution, especially those aspects affecting the order in which events and the responses to those events occur in the system. The removal of this type of nondeterminism yields a more accurate system.

This study shows that Unity3D does not provide a mechanism for controlling the order of execution. Marum et al. [5] argues that the ability to change the execution order of components—and, consequently, to enable the correct ordering of the components' executions in a scene graph—is key to achieving highly accurate systems. To be perceived as accurate, simulated interactions must occur in the same order as the interactions would occur in the corresponding real-world situation. If they do not, then the simulation does not seem realistic to the user. Consider a domino chain. When the first domino falls, the second should fall when the weight of the first domino causes it to fall. The third falls similarly, and so forth throughout the chain. If any one of these falls is shown incorrectly, the whole simulated sequence is likely to be perceived as unrealistic.

As in the dynamic GUI case study, this case study developed a reactive programming approach to mitigate the transitional turbulence problems. This approach analyzes the complex relationships among the game objects present in the scene hierarchy, encodes these dependencies in a dependency graph, and then uses the graph to rearrange the updates in an order consistent with the dependency constraints. It builds the graph when the application starts up and then rebuilds it whenever it detects that the dependencies might have changed. The approach thus coalesces the processing of a chain of what may be several events in the unmodified system into a single, large-grained event that updates the states of many controls at once. As in the previous case study, because of the nature of the display system, the approach cannot totally eliminate the transitional turbulence that can cause inaccurate or misleading displays, but coalescing multiple events into large-grained events does potentially decrease the number of inaccuracies and increase the system's performance.

To evaluate the approach, this case study developed a prototype library and used it to perform several experiments [5]. The experiments involved a three-way comparison among Unity3D applications using their approach, the built-in Unity3D event system, and UniRx, the Reactive Extensions library for the Unity3D platform [12]. Figure A2 shows, in general, how the case study constructs a dependency graph for this VR application and modifies the game scene accordingly.

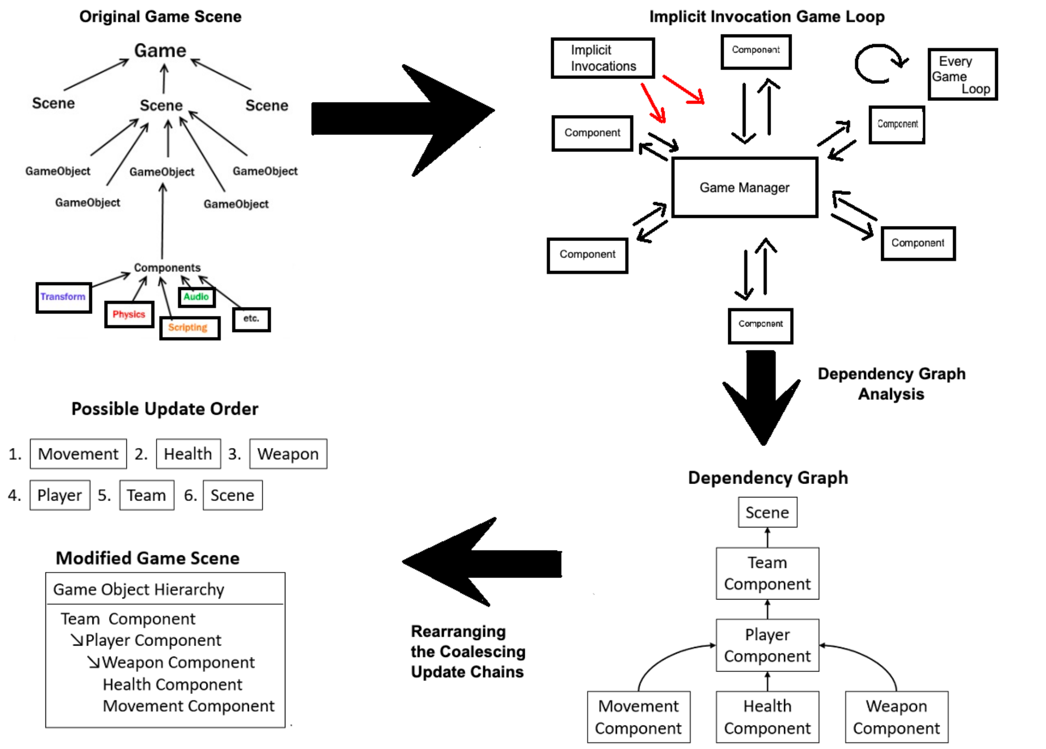


Figure A2. Three-step augmentation of a dynamic VR application using the design pattern's model.

The experiments used a test scenario built around an expression evaluator to demonstrate how the update order of the game objects affects the interactions among the game objects. The test scenario represented the expression as a game tree with an operator at each internal node with its operands as its subtrees. The values are at the leaf nodes. For the correct value of the expression to be calculated, the operand subtrees must be evaluated before the corresponding operator. To determine the effects of reconfiguration of Unity3D's game tree, the experiments include (a) tests that kept the expression tree stable throughout the run and (b) tests that randomly introduced changes in the tree's structure during the test run. For each test run, the experiment measured the startup time, latency, and total errors during the run and computed the average startup time, latency, errors per cycle, errors that resulted in a visibly inaccurate state, and the number of miscalculations that occurred in a test that failed. The experiments indicate that, on average, the approach exhibited a shorter latency and fewer errors, at the cost of a modest increase in the startup time compared to the other two alternatives. Marum et al. [5] concluded that the approach improves performance and results in more accurate behavior.

Thus, the Marum et al. solution approach also seems to work well for the types of problem envisioned and the technologies used in the second case study [5]. In Sections 4–15, Appendixs A.1 and A.2, the task is to identify the commonalities of the two specific solutions and codify a general, technology-independent approach as a new design pattern.

Appendix B. Final Design Pattern

Appendix B.1. Pattern Name

DYNAMICALLY COALESCING REACTIVE CHAINS

Appendix B.2. Context

(C1) The application is constructed according to the IMPLICIT INVOCATION architectural pattern, assuming nondeterministic but fair handling of events.

(C2) The application's component architecture may change at run time. The application organizes the components into a hierarchical structure. This structure may change dynamically at run time as a result of external stimuli or the actions of components.

(C3) The application presents some aspects of its state that can be observed periodically from outside the system. The timing of this presentation is not under the control of the application.

(C4) Because of the asynchronous nature of the application's operation, the externally observable presentation may exhibit periods of transitional turbulence.

(C5) Each component is an information-hiding module with a well-defined interface. The only way to change or access its state explicitly is by calling one of its accessor or mutator procedures (e.g., properties in some object-oriented languages).

(C6) The application supports *reflection* capabilities. That is, application-level code can examine the application's features (such as its components, events, event handlers, and hierarchical structure) at run time and extract metadata (such as names, types, and the type signatures of the procedures in component interfaces).

Appendix B.3. Problem

We want to eliminate or reduce the length of the periods of transitional turbulence during which the external presentation does not accurately reflect the state of the application. We need to do this without sacrificing performance. The goal is to better satisfy the observers' expectations by increasing the accuracy of the external presentation.

Appendix B.4. Forces

Transitional Turbulence Reduction: We want to decrease the transitional turbulence in the application's execution to better satisfy the observers' expectations.

Run-time Reconfiguration: We want to adapt to changes in an application's component architecture at run time.

Startup Cost Inflation: We want to avoid adding significant startup or shutdown costs.

Operational Overhead Creep: We want to avoid adding significant processing overhead during the application's normal operation.

Code Cluttering: We want to avoid significantly complicating the application's design, implementation, testing, or use.

Appendix B.5. Solution

Appendix B.5.1. Solution: Summary

A solution encodes the complex relationships among the application's components in a dependency graph, and then uses the graph to order the updates of the components without violating the dependency constraints. The goal is to reorder the updates of the components so that the new order reduces transitional turbulence without degrading the performance of the system.

Appendix B.5.2. Solution: Definitions

What do we mean by a "dependency graph" in this context?

- If the execution of some component `X` of an application can directly affect a subsequent execution of some other component `Y` in any way, then `Y` *depends on* `X`. For example, `X` might trigger an event for which `Y` listens; change the value of some attribute of its state that `Y` accesses; directly call one of `Y`'s mutator procedures; or create, delete, or modify `Y`.
- A *dependency graph* is a directed acyclic graph formed by placing the components at the nodes and adding a directed edge from some component `Y` to some component `X` only if `Y` depends on `X`.

Figure 4 shows a dependency graph for an application with ten components named with the upper case letters `A` through `J` and directed edges from every component to every other component on which it is directly dependent. The shaded area of the figure includes the six other components that are directly or indirectly dependent on component `A`. Any change in the state of component `A` may require changes in all other components in the shaded area. If each edge is implemented as an event, then six independent events must be processed to propagate the changes to all dependent components. This is the *update cycle*. The event handling system processes these events in a nondeterministic order, interleaved with any other pending events.

To apply the DCRC pattern, we are primarily interested in recording the dependencies related to the implicit invocations—between components that listen for an event and those that trigger the event. Of course, being able to record other kinds of dependencies may also be helpful.

Appendix B.5.3. Solution: Augmenting the Application

To apply the DCRC pattern to an application that satisfies the Context, we can augment the application with appropriate *software mechanisms*. For example, Figure 5 illustrates how a solution can augment an application's event handling to coalesce dependent events into larger-grained events without modifying the underlying event-handling mechanisms. Beginning with the application's II architecture (shown in panel 1), a solution first determines the dependency relationships between the components (panel 2 and also Figure 4) and then builds the corresponding dependency graph (panel 3). Then it can use the dependency graph to rearrange the component updates in any order that satisfies the dependency constraints (panel 4). In particular, the solution seeks to optimize the processing of an update cycle by performing all the updates in the cycle (the shaded area in Figure 4) directly as part of the processing of the first event.

These software mechanisms may include some combination of libraries, frameworks, tools, and design and programming techniques. The various mechanisms should be *lightweight*. That is, they should execute efficiently and should not require extensive modifications to the existing application. The "software mechanisms" needed and the meaning of "lightweight" depend on the application's specific implementation technologies and performance requirements.

For applications that satisfy the Context, a developer can augment the application's event-handling mechanisms to solve the Problem. In general, to construct a Solution, the developer needs to design, implement, and deploy three primary software mechanisms: one to build the dependency graph at startup of the application, one to rebuild the dependency graph when needed during the application's execution, and one to coalesce all the component updates in an update cycle into a single event. The construction of these software mechanisms involves an *augmentation workflow* with three phases:

1. *Augmentation analysis*, which requires *analyzing* the application to identify how to add the necessary mechanisms;
2. *Augmentation development*, which requires *developing* (i.e., designing and implementing) the mechanisms;

3. *Augmentation incorporation*, which requires *incorporating* the mechanisms into the operation of the application.

Figure 6 summarizes the augmentation workflow, restating the tasks as questions to answer.

Solution: Augmentation Analysis

In the augmentation analysis phase, the solution developer must perform the three tasks AA1, AA2, and AA3 to analyze the original application and define the requirements for the new software mechanisms.

(AA1) Examine the hierarchical structure to identify how a program can iterate through the components (i.e., accessing each component exactly once).

(AA2) Examine the design and implementation of the components and the features of the implementation language to identify how a program can extract the dependency relationships between the components at run time.

Task AA2 may involve the use of the existing features of the components or the reflection capabilities of the implementation language. If sufficient capabilities do not exist, we can design lightweight modifications that implement sufficient application-specific capabilities.

(AA3) Examine the components and events to determine which of the relationships between the components to include in the dependency graph and which to exclude. To reduce transitional turbulence, the augmented application program can manipulate the components and relationships included, but cannot manipulate those excluded.

Generally speaking, in task AA3 we include the component relationships arising from the application's custom code (which we can modify if needed) and exclude those in the supporting framework (which we cannot modify). We may also want to exclude any component relationship if that relationship represents an expensive computation or an arbitrary delay.

Solution: Augmentation Development

In the augmentation development phase, the solution developer must perform the three tasks AD1, AD2, and AD3 to design and implement the new software mechanisms according to the requirements specified in the augmentation analysis phase.

(AD1) Design and implement a lightweight run-time mechanism that enables the program to differentiate between the components that are to be included in the dependency graph and those that are not.

Task AD1 involves features already present in the application (e.g., types, value of some property, metadata) or may involve modifying the application to add appropriate features. For example, in an object-oriented system in which the components are objects, we could modify the included components to implement a "marker interface" that can be checked by reflection. The developer should establish a criterion to determine what to include in the dependency graph and what to exclude. In general, this criterion can be defined as a function that is called by the dependency graph-building procedure. It must return a boolean value `true` if its argument should be inserted into the dependency graph and otherwise return `false`.

(AD2) Design and implement a lightweight run-time mechanism that enables the program to detect whether the component architecture or the dependencies among the components have changed since the previous check (or since the beginning of operation).

In this context, task AD2 assumes that a change to the hierarchical structure holding the components likely means a change to the component architecture.

(AD3) Design and implement lightweight mechanisms to construct the dependency graph initially and to reconstruct it when needed.

To build a dependency graph in task AD3, the program can traverse the hierarchical structure (e.g., do a breadth-first traversal of the Document Object Model), placing each component at a node and adding edges to other nodes according to the *depends-on* relationships between components. However, it must prune the graph appropriately to remove any cycles.

Solution: Augmentation Incorporation

In the augmentation incorporation phase, the solution developer must perform the three tasks AN1, AN2, and AN3 to incorporate the new software mechanisms into the original application. This phase builds on the results of the augmentation development phase. Figure 7 shows how the augmented application can incorporate the three primary software mechanisms into a typical object-oriented GUI application at run time.

(AN1) The application must construct the dependency graph at or before startup.

As shown in panel 1 of Figure 7, the augmentation process begins with an IMPLICIT INVOCATION application modified with the basic mechanisms developed in augmentation development task AD1. As shown in panel 2, it then uses the mechanisms developed in task AD3 to build the initial dependency graph at startup.

(AN2) When some component `C` included in the dependency graph signals an event `E`, the application must *intercept* `E` and directly call the procedures associated with event `E` on all listening components as recorded in the dependency graph. Then it must recursively apply the process to all events signalled by the listening components. This continues as long as there are dependencies indicated in the graph (which cannot have cycles). This process dynamically coalesces the processing of chains of events into what is processed as one "large-grained" event. The meaning of "intercept" depends on the specific application's implementation technologies.

As shown in panel 4 of Figure 7, the augmented system combines the updates of all the components in the update cycle into a sequence of direct procedure calls. This coalesces the processing of a whole chain of events into a single event.

(AN3) After processing each "large-grained" event in the previous step, the application must check whether the application's component architecture has changed (e.g., the addition, modification, or deletion of any component in the hierarchical structure) or the dependencies among components have changed. If so, then the dependency graph must be updated appropriately to reflect the new component architecture.

As shown in panel 3 of Figure 7, at the beginning of any update, the augmentation process uses the mechanisms developed in augmentation development task AD2 to determine whether the dependency graph needs to be rebuilt. If so, it uses the mechanisms developed in task AD3 to rebuild the graph.

Appendix B.5.4. Solution: Balancing the Forces

In the Solution described above, we handle all the identified Forces. How do we balance the various Forces to achieve this Solution?

Transitional Turbulence Reduction

For a state change in any component, the augmented application must propagate the effects to all its directly or indirectly dependent components without the delays and nondeterminism introduced by the normal event-handling system—as if all were part of the processing of one large-grained event. This can decrease latency and increase accuracy (i.e., decrease the number of errors).

Run-Time Reconfiguration

Frequently during normal operation of the application, the augmented application checks if its component architecture has changed. If it detects a change, it then reconstructs the dependency graph to reflect the new architecture. The extra costs incurred in reconstructing the dependency graph must not itself worsen the solution's overall effect on the latency and accuracy.

Changes to an application's component architecture during normal operation can increase latency and decrease accuracy. However, a good solution must dynamically adapt to such changes and seek to mitigate the effects on latency and accuracy.

Startup Cost Inflation

When applying the pattern, developers should seek to keep the cost of initially constructing the dependency graph low. The developers should carefully select the components to include in the analysis and use efficient methods to determine dependency relationships and build the graph.

The augmented application likely incurs additional processing overhead at startup and shutdown. In particular, the extra costs for constructing the initial dependency graph should be small in proportion to the potential accuracy and performance gain in an application that runs sufficiently long.

Operational Overhead Creep

The augmented application likely incurs additional processing overhead during normal operation, especially when the component architecture changes. In particular, the extra costs for checking for changes in the component architecture and reconstructing the dependency graph should be small in proportion to the potential accuracy and performance gain in an application that runs sufficiently long. In cases in which the component architecture changes infrequently, the augmented application should incur minimal costs.

Code Cluttering

To implement a solution, the developer must augment the existing application by incorporating a set of software mechanisms as described above. Unfortunately, modifying the application often complicates its design, implementation, testing, and use.

However, in a good design and implementation of the solution's new software mechanisms, it should be possible to readily augment the existing solution. Therefore, the new software mechanisms must be carefully designed, implemented, and documented so that the solution can work well with typical application designs.

For example, for a typical GUI application, it should be possible to implement the solution approach as a software framework with wrapper classes for the controls and a library that implements the algorithms to build/reconstruct the dependency graph and uses it to coalesce chains into "large-grained" events.

Appendix B.6. Consequences

Appendix B.6.1. Benefits

- *Transitional Turbulence Reduction:* A solution coalesces sets of dependent internal events into "large-grained" events such that the handling of a large-grained event causes the same overall state change as the corresponding set. This can decrease latency and increase accuracy (i.e., decrease the number of errors).
- *Run-time Configuration:* A solution dynamically adapts to changes in an application's component architecture at run time.
- *Code Cluttering:* An application can be readily adapted to use the mechanisms that implement the solution.

Appendix B.6.2. Liabilities

- *Run-time Reconfiguration:* Changes to an application's component architecture at run time can increase latency and decrease accuracy.
- *Startup Cost Inflation:* Implementing a solution often causes additional processing overhead at startup and shutdown of the application.
- *Operational Overhead Creep:* Implementing a solution often causes additional run-time processing overhead, especially when the component architecture changes.
- *Code Cluttering:* An application must be adapted to use the mechanisms that implement the solution. Modifying the application often complicates its design, implementation, testing, or use.

References

1. Chandy, M.K.; Misra, J. *Parallel Program Design: A Foundation*; Addison Wesley: Boston, MA, USA, 1988.
2. Lorenz, E.N. Deterministic Nonperiodic Flow. *J. Atmos. Sci.* **1963**, *20*, 130–141. [CrossRef]
3. Cooper, G.H.; Krishnamurthi, S. Embedding Dynamic Dataflow in a Call-by-Value Language. In Proceedings of the Programming Languages and Systems, 15th European Symposium on Programming, Vienna, Austria, 27–28 March 2006; pp. 294–308.
4. Mijač, M.; García-Cabot, A.; Strahonja, V. Reactor Design Pattern. *TEM J. Technol. Educ. Inform.* **2021**, *10*, 18–30. [CrossRef]
5. Marum, J.P.O.; Jones, J.A.; Cunningham, H.C. Dependency Graph-based Reactivity for Virtual Environments. In Proceedings of the IEEE VR 2020 Workshop on Software Engineering and Architectures for Interactive Systems (SEARIS), Atlanta, GA, USA, 22–26 March 2020; pp. 246–253.
6. Marum, J.P.O.; Cunningham, H.C.; Jones, J.A. Unified Library for Dependency Graph Reactivity on Web and Desktop User Interfaces. In Proceedings of the ACM Southeast Conference, ACMSE 2020, Tampa, FL, USA, 2–4 April 2020; pp. 26–33.
7. Bainomugisha, E.; Carreton, A.L.; van Cutsem, T.; Mostinckx, S.; de Meuter, W. A Survey on Reactive Programming. *ACM Comput. Surv.* **2013**, *45*, 1–34. [CrossRef]
8. Marum, J.P.O.; Cunningham, H.C.; Jones, J.A. *Unified Library for Dependency Graph Reactivity on Web and Desktop User Interfaces: ADDENDUM*; Technical Report; University of Mississippi, Department of Computer and Information Science: Oxford, MS, USA, 2020. Available online: https://john.cs.olemiss.edu/~hcc/papers/Addendum_ACMSE_2020.pdf (accessed on 16 January 2024).
9. Blackheath, S.; Jones, A. *Functional Reactive Programming*; Manning: Shelter Island, NY, USA, 2016.
10. ReactiveX Project. ReactiveX: An API for Asynchronous Programming with Observable Streams. 2023. Available online: http://reactivex.io (accessed on 16 January 2024).
11. Unity Technologies. *Unity User Manual 2020.3*; Unity Technologies: San Francisko, CA, USA, 2023. Available online: https://docs.unity3d.com/Manual (accessed on 16 January 2024).
12. Kawai, Y. UniRx: Reactive Extensions for Unity3D. GitHub. 2024. Available online: https://github.com/neuecc/UniRx (accessed on 16 January 2024).
13. Buschmann, F.; Meunier, R.; Rohnert, H.; Sommerlad, P.; Stal, M. *Pattern-Oriented Software Architecture: A System of Patterns*; Wiley: Chichester, UK, 1996; Volume 1.
14. Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*; Addison Wesley: Boston, MA, USA, 1995.
15. Wellhausen, T.; Fiesser, A. How to Write a Pattern? A Rough Guide for First-Time Pattern Authors. In Proceedings of the 16th European Conference on Pattern Languages of Programs, EuroPLOP '11, Irsee, Germany, 11–15 July 2011; pp. 1–9.
16. Meszaros, G.; Doble, J. A Pattern Language for Pattern Writing. In *Pattern Languages of Program Design 3*; Addison Wesley: Boston, MA, USA, 1998; pp. 529–574.
17. Garlan, D.; Shaw, M. An Introduction to Software Architecture. In *Advances in Software Engineering and Knowledge Engineering*; Ambriola, V., Tortora, G., Eds.; World Scientific: Singapore, 1993; pp. 1–39.
18. Qian, K.; Fu, X.; Tao, L.; Xu, C.W.; Diaz-Herrera, J.L. *Software Architecture and Design Illuminated*; Jones & Bartlett Learning: Burlington, MA, USA, 2010.
19. Iba, T. How to Write Patterns: A Practical Guide for Creating a Pattern Language on Human Actions. 2021 Pattern Languages of Programs Conference, PloPourri, a Methodological, Philosophical, and Educational Study on Pattern Languages. 2022. Available online: https://hillside.net/plop/2021/plopourri/PLoP21_PLOPOURRI_Iba_Methodology4.pdf (accessed on 16 January 2024).
20. Coplien, J.; Hoffman, D.; Weiss, D. Commonality and Variability in Software Engineering. *IEEE Softw.* **1998**, *15*, 37–45. [CrossRef]
21. Harrison, N.B. The Language of Shepherding: A Pattern Language for Shepherds and Sheep. In *Pattern Languages of Program Design*; Harrison, N., Foote, B., Rohnert, H., Eds.; Addison Wesley: Boston, MA, USA, 1999; Volume 4, pp. 507–530.
22. Harrison, N.B. Advanced Pattern Writing: Patterns for Experienced Writers. In *Pattern Languages of Program Design*; Manolescu, D.A., Voelter, M., Noble, J., Eds.; Chapter 16; Addison Wesley: Boston, MA, USA, 2006; Volume 5, pp. 433–451.
23. Garlan, D.; Notkin, D. Formalizing Design Spaces: Implicit Invocation Mechanisms. In Proceedings of the VDM'91, Formal Software Development Methods: Proceedings of the International Symposium of VDM Europe, Noordwijkerhout, The Netherlands, 21–25 October 1991; pp. 31–44.

24. Shaw, M. Some Patterns for Software Architectures. In *Pattern Languages of Program Design*; Vlissides, J., Coplien, J.O., Kerth, N.L., Eds.; Addison Wesley: Boston, MA, USA, 1996; Volume 2, pp. 255–269.
25. Qian, K.; Fu, X.; Tao, L.; Xu, C.; Diaz-Herrera, J.L. Implicit Asynchronous Communication Software Architecture. In *Software Architecture and Design Illuminated*; Chapter 8; Jones & Bartlett Learning: Burlington, MA, USA, 2010; pp. 177–198.
26. Shaw, M.; DeLine, R.; Klein, D.V.; Ross, T.L.; Young, D.M.; Zelesnik, G. Abstractions for Software Architecture and Tools to Support Them. *IEEE Trans. Softw. Eng.* **1995**, *21*, 314–335. [CrossRef]
27. Garlan, D. Software Architecture. In *Wiley Encyclopedia of Computer and Science Engineering*; Wah, B.W., Ed.; Wiley Online Library: Hoboken, NJ, USA, 2007.
28. Eugster, P.T.; Felber, P.A.; Guerraoui, R.; Kermarrec, A. The Many Faces of Publish/Subscribe. *ACM Comput. Surv.* **2003**, *35*, 114–131. [CrossRef]
29. Mijač, M.; Kermek, D.; Zlatko, S. Complex Propagation of Events: Design Patterns Comparison. In *Information Systems Development: Transforming Organisations and Society (ISD2014 Proceedings)*; Strahonja, V., Vrček, N., Plantak Vukovac, D., Barry, C., Lang, M., Linger, H., Schneider, C., Eds.; University of Zagreb, Faculty of Organization and Informatics: Varaždin, Croatia, 2014; pp. 306–316.
30. Czaplicki, E.; Chong, S. Asynchronous Functional Reactive Programming for GUIs. In Proceedings of the 34th SIGPLAN Conference on Programming Language Design and Implementation, PLDI '13, Seattle, WA, USA, 16–19 June 2013; pp. 411–422.
31. Drechsler, J.; Salvaneschid, G.; Mogk, R.; Mezini, M. Distributed REScala: An Update Algorithm for Distributed Reactive Programming. In Proceedings of the 2014 ACM International Conference on Object Oriented Programming Systems Languages and Applications, OOPSLA 2014, Portland, OR, USA, 20–24 October 2014; pp. 361–376.
32. Elliott, C.M. Push-Pull Functional Reactive Programming. In Proceedings of the 2nd SIGPLAN Symposium on Haskell, Haskell '09, Edinburgh, UK, 3 September 2009; pp. 25–36.
33. Meyerovich, L.A.; Guha, A.; Baskin, J.; Cooper, G.H.; Greenberg, M.; Bromfield, A.; Krishnamurthi, S. Flapjax: A Programming Language for Ajax Applications. In Proceedings of the 24th ACM SIGPLAN Conference on Object Oriented Programming Systems Languages and Applications, OOPSLA 2009, Orlando, FL, USA, 25–29 October 2009; pp. 1–20.
34. Reynders, B.; Devriese, D.; Piessens, F. Experience Report: Functional Reactive Programming and the DOM. In Proceedings of the Companion to the First International Conference on the Art, Science and Engineering of Programming, Programming '17, Brussels, Belgium, 3–6 April 2017; pp. 23:1–23:6.
35. Mijač, M.; Garcia-Cabot, A.; Strahonja, V. REFRAME—A Software Framework for Managing Reactive Dependencies in Object-oriented Applications. *SoftwareX* **2023**, *24*, 101571. [CrossRef]
36. Parnas, D.L. On the Criteria to Be Used in Decomposing Systems into Modules. *Commun. ACM* **1972**, *15*, 1053–1058. [CrossRef]
37. Britton, K.H.; Parker, R.A.; Parnas, D.L. A Procedure for Designing Abstract Interfaces for Device Interface Modules. In Proceedings of the 5th International Conference on Software Engineering, San Diego, CA, USA, 9–12 March 1981; pp. 195–204.
38. Parnas, D.L. The Secret History of Information Hiding. In *Software Pioneers: Contributions to Software Engineering*; Broy, M., Denert, E., Eds.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 398–409.
39. Mijač, M. Evaluation of Design Science Instantiation Artifacts in Software Engineering Research. In Proceedings of the Central European Conference on Information and Intelligent Systems, Varaždin, Croatia, 2–4 October 2019; pp. 313–321.
40. Engström, E.; Storey, M.; Runeson, P.; Höst, M.; Baldassarre, M.T. How Software Engineering Research Aligns with Design Science: A Review. *Empir. Softw. Eng.* **2020**, *25*, 2630–2660. [CrossRef]
41. Cunningham, H.C.; Zhang, C.; Liu, Y. Keeping Secrets within a Family: Rediscovering Parnas. In Proceedings of the International Conference on Software Engineering Research and Practice (SERP), Las Vegas, NV, USA, 21–24 June 2004; pp. 712–718.
42. The Hillside Group. A Group Dedicated to Design Patterns. 2023. Available online: https://hillside.net (accessed on 16 January 2024).
43. Chin, S.; Vos, J.; Weaver, J. *The Definitive Guide to Modern Java Clients with JavaFX*; Apress: New York, NY, USA, 2019.
44. OpenJFX Project. JavaFX. Gluon. 2024. Available online: https://openjfx.io (accessed on 16 January 2024).
45. Cleary, A.; Vandenbergh, L.; Peterson, J. Reactive Game Engine Programming for STEM Outreach. In Proceedings of the 46th ACM Technical Symposium on Computer Science Education, SIGCSE '15, Kansas City, MO, USA, 4–7 March 2015; pp. 628–632.
46. Buschmann, F.; Henney, K.; Schmidt, D.C. *Pattern-Oriented Software Architecture, On Patterns and Pattern Languages*; Wiley: Chichester, UK, 2007; Volume 5.
47. Iba, T.; Isaku, T. A Pattern Language for Creating Pattern Languages: 364 Patterns for Pattern Mining, Writing, and Symbolizing. In Proceedings of the 23rd Conference on Pattern Languages of Programs, PLoP '16, Monticello, IL, USA, 23 October 2016; pp. 1–63.
48. Iba, T.; Kanai, T. Systematization of Patterns for Weaving a Pattern Language as a Whole. 2021 Pattern Languages of Programs Conference, PloPourri, a Methodological, Philosophical, and Educational Study on Pattern Languages. 2022. Available online: https://hillside.net/plop/2021/plopourri/PLoP21_PLOPOURRI_Iba_Methodology2.pdf (accessed on 16 January 2024).
49. Kohls, C.; Panke, S. Is That True...? Thoughts on the Epistemology of Patterns. In Proceedings of the 16th Conference on Pattern Languages of Programs, PLoP '09, Chicago, IL, USA, 28–30 August 2009; pp. 1–14.
50. Riehle, D.; Harutyunyan, N.; Barcomb, A. Pattern Discovery and Validation Using Scientific Research Methods. In *Transactions on Pattern Languages of Programming V (TPLoP)*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2021; pp. 1–25. Available online: https://arxiv.org/abs/2107.06065 (accessed on 16 January 2024).

51. Eales, A. The Observer Pattern Revisited. In *Educating, Innovating & Transforming: Educators in IT: Concise Paper*; The Pennsylvania State University: State College, PA, USA, 2005.
52. Riehle, D. The Event Notification Pattern—Integrating Implicit Invocation with Object-Orientation. *Theory Pract. Object Syst.* **1996**, *2*, 43–52. [CrossRef]
53. Feiler, P.H.; Tichy, W.F. Propagator: A Family of Patterns. In Proceedings of the TOOLS USA 97. International Conference on Technology of Object Oriented Systems and Languages, IEEE, Santa Barbara, CA, USA, 1 August 1997; pp. 355–366.
54. Garlan, D.; Jha, S.; Notkin, D.; Dingel, J. Reasoning about Implicit Invocation. In Proceedings of the 6th ACM SIGSOFT International Symposium on Foundations of Software Engineering, Lake Buena, SIGSOFT '98/FSE-6, Vista, FL, USA, 3–5 November 1998; pp. 209–221.
55. Mikkonen, T. Formalizing Design Patterns. In Proceedings of the 20th International Conference on Software Engineering, Kyoto, Japan, 19–25 April 1998; pp. 115–124. [CrossRef]
56. Garlan, D.; Khersonsky, S. Model Checking Implicit-Invocation Systems. In Proceedings of the Tenth International Workshop on Software Specification and Design, IWSSD, San Diego, CA, USA, 5–7 November 2000; pp. 23–30.
57. Garlan, D.; Khersonsky, S.; Kim, J.S. Model Checking Publish-Subscribe Systems. In Proceedings of the International SPIN Workshop on Model Checking of Software, Portland, OR, USA, 9–10 May 2003; pp. 166–180.
58. Xu, J.; Rajan, H.; Sullivan, K. Aspect Reasoning by Reduction to Implicit Invocation. In Proceedings of the Foundations of Aspect-Oriented Languages Workshop (FOAL), Lancaster, UK, 23 March 2004; pp. 31–36.
59. Soundarajan, N.; Hallstrom, J.O. Responsibilities and Rewards: Specifying Design Patterns. In Proceedings of the 26th International Conference on Software Engineering, IEEE, Edinburgh, UK, 28 May 2004; pp. 666–675.
60. Soundarajan, N.; Hallstrom, J.O.; Shu, G.; Delibas, A. Patterns: From System Design to Software Testing. *Innov. Syst. Softw. Eng.* **2008**, *4*, 71–85. [CrossRef]
61. Fiadeiro, J.L.; Lopes, A. An Algebraic Semantics of Event-based Architectures. *Math. Struct. Comput. Sci.* **2007**, *17*, 1029–1073. [CrossRef]
62. Valero, V.; Macia, H.; Díaz, G.; Cambronero, M.E. Colored Petri Net Modeling of the Publish/Subscribe Paradigm in the Context of Web Services Resources. In Proceedings of the Formal Methods for Industrial Critical Systems: Proceedings of the 20th International Workshop, FMICS 2015, Oslo, Norway, 22–23 June 2015; pp. 81–95.
63. Marum, J.P.O. Dependency-based Reactive Change Propagation Design Pattern Applied to Environments with High Unpredictability. Ph.D. Thesis, University of Mississippi, Department of Computer and Information Science, University, MS, USA, 2021. Available online: https://egrove.olemiss.edu/etd/2122 (accessed on 16 January 2024).
64. Marum, J.P.O.; Jones, J.A.; Cunningham, H.C. Towards a Reactive Game Engine. In Proceedings of the 50th IEEE SouthEastCon, Huntsville, AL, USA, 11–14 April 2019; pp. 1–8.

US009645395B2

(12) **United States Patent**
Bolas et al.

(10) **Patent No.: US 9,645,395 B2**
(45) **Date of Patent: May 9, 2017**

(54) **DYNAMIC FIELD OF VIEW THROTTLING AS A MEANS OF IMPROVING USER EXPERIENCE IN HEAD MOUNTED VIRTUAL ENVIRONMENTS**

(71) Applicants: **Mark Bolas**, Los Angeles, CA (US); **J. Adam Jones**, Torrance, CA (US); **Ian McDowall**, Woodside, CA (US); **Evan Suma**, Culver City, CA (US)

(72) Inventors: **Mark Bolas**, Los Angeles, CA (US); **J. Adam Jones**, Torrance, CA (US); **Ian McDowall**, Woodside, CA (US); **Evan Suma**, Culver City, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **14/216,220**

(22) Filed: **Mar. 17, 2014**

(65) **Prior Publication Data**

US 2014/0268356 A1 Sep. 18, 2014

Related U.S. Application Data

(60) Provisional application No. 61/799,758, filed on Mar. 15, 2013.

(51) **Int. Cl.**

G02B 27/14 (2006.01)
G02B 27/01 (2006.01)
G02B 27/00 (2006.01)

(52) **U.S. Cl.**

CPC ***G02B 27/017*** (2013.01); ***G02B 27/0093*** (2013.01); *G02B 2027/014* (2013.01);

(Continued)

(58) **Field of Classification Search**

CPC G02B 27/0172; G02B 27/017; G02B 27/0093; G02B 2027/0178;

(Continued)

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,406,532	A	9/1983	Howlett
5,508,849	A	4/1996	Goodell

(Continued)

FOREIGN PATENT DOCUMENTS

EP	1379082	B1	6/2005
WO	WO2004006570	A1	1/2004

(Continued)

OTHER PUBLICATIONS

Pig Nose Disclosure by Third Party. possibly before Mar. 17, 2014.
LEEP Camera, possibly demonstrated before Mar. 17, 2014.

(Continued)

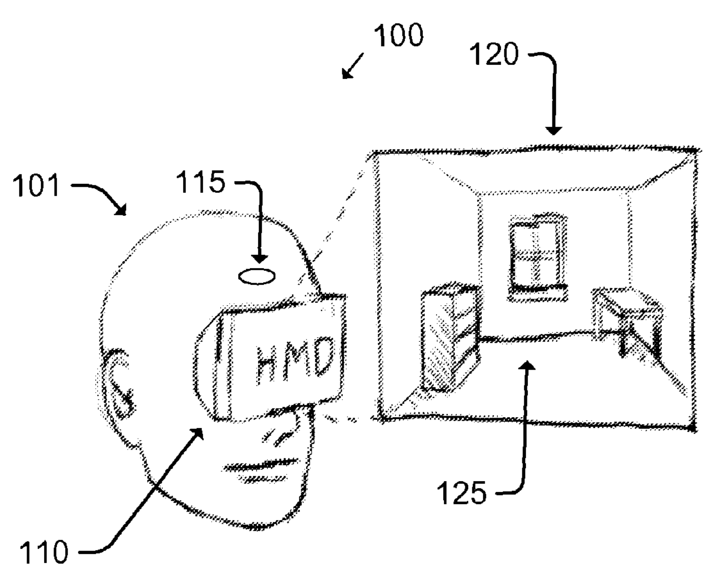
Primary Examiner — Tuyen Tra

(74) *Attorney, Agent, or Firm* — McDermott Will & Emery LLP

(57) **ABSTRACT**

A head-mounted display (HMD) allows a user to view a virtual environment. The HMD displays a field of view to the user. However, the user may experience simulator sickness or motion sickness from viewing the field of view. The HMD is connected to a sensor which can monitor the user. By monitoring the user's physiological state, the user's simulator sickness can be detected or predicted. To reduce the negative effects, the field of view can be throttled. The field of view can also be throttled in order to provide a better user experience even if the user does not experience sickness.

17 Claims, 5 Drawing Sheets



(52) **U.S. Cl.**
CPC *G02B 2027/0123* (2013.01); *G02B 2027/0178* (2013.01); *G02B 2027/0187* (2013.01)

(58) **Field of Classification Search**
CPC G02B 2027/0123; G02B 2027/014; G02B 2027/0187
USPC 359/630, 629, 618, 13, 376, 458, 462; 345/8, 7
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

8,212,859 B2	7/2012	Tang et al.	
8,708,884 B1 *	4/2014	Smyth	A61M 21/00 600/27
8,957,835 B2	2/2015	Hoellwarth	
9,293,079 B2	3/2016	Bolas et al.	
2002/0019296 A1 *	2/2002	Freeman	A63F 13/02 482/4
2002/0099257 A1 *	7/2002	Parker et al.	600/27
2010/0079356 A1	4/2010	Hoellwarth	
2012/0094754 A1	4/2012	Suzuki et al.	
2012/0135803 A1	5/2012	Nonaka et al.	
2013/0113973 A1	5/2013	Miao	
2013/0141360 A1	6/2013	Compton et al.	
2014/0139551 A1 *	5/2014	McCulloch	G09G 5/377 345/633
2014/0267637 A1	9/2014	Hoberman et al.	
2015/0355465 A1	12/2015	Jones et al.	

FOREIGN PATENT DOCUMENTS

WO	WO2004006578 A1	1/2004
WO	WO2005041568 A1	5/2005
WO	WO2005069638 A1	7/2005
WO	WO2005069640 A1	7/2005
WO	WO2006003600 A1	1/2006
WO	WO2006003603 A1	1/2006
WO	WO2006003604 A1	1/2006
WO	WO2006003624 A1	1/2006
WO	WO2012035174 A1	3/2012

OTHER PUBLICATIONS

Sutherland, I.E. 1965 The Ultimate Display. In Proceedings of IFIP Congress, International Federation for Information Processing, London: Macmillan and Co., 1965, pp. 506-508.
Sutherland, I.E. 1968 A Head-Mounted Three Dimensional Display. In AFIPS '68 (Fall Joint Computer Conference), Dec. 9-11, 1968, New York, NY: ACM, pp. 757-765.
hasbroMY3D.com.2010. Product Guide to MY3D App. Hasbro, 3 pages.
USPTO. 2015. Non-final Office Action dated Dec. 4, 2015 for U.S. Appl. No. 14/211,459, entitled "Head-Mounted Display Frame for Improved Spatial Performance in Head Mounted Virtual Environments," filed Mar. 14, 2014, published as U.S. PG Pub 2015/0355465 A1.
USPTO. 2016. Non-final Office Action dated Apr. 8, 2016 for U.S. Appl. No. 14/216,143, entitled "Hybrid Stereoscopic Viewing Device," filed Mar. 17, 2014, published as U.S. PG Pub 2014/0267637 A1.
USPTO. 2016. Final Office Action dated Jun. 20, 2016 for U.S. Appl. No. 14/211,459, entitled "Head-Mounted Display Frame for Improved Spatial Performance in Head Mounted Virtual Environments," filed Mar. 14, 2014, published as U.S. PG Pub 2015/0355465 A1.
Non-Final Office Action dated Feb. 10, 2017, which issued in U.S. Appl. No. 14/211,459.
Notice of Allowance dated Jan. 31, 2017, which issued in U.S. Appl. No. 14/216,143.

* cited by examiner

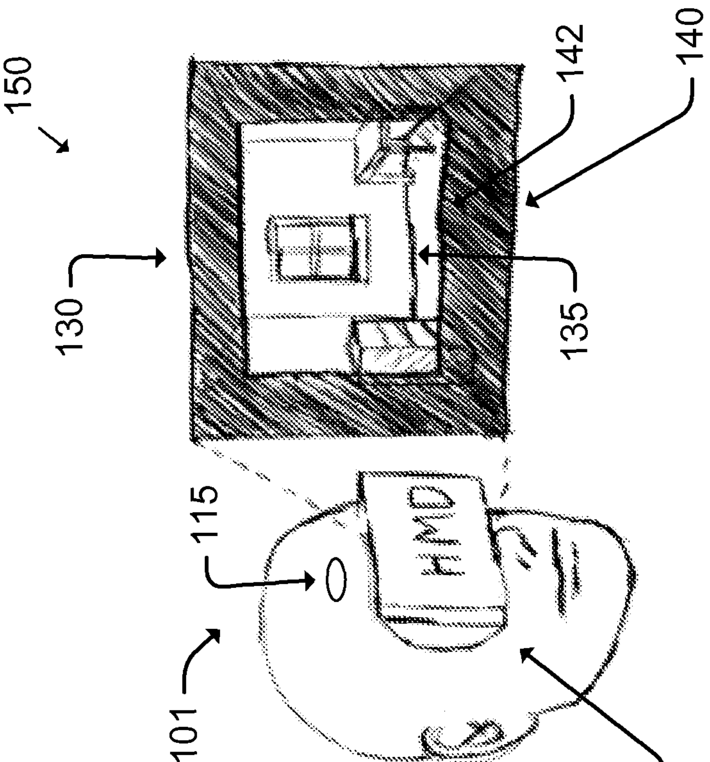


FIG. 1B

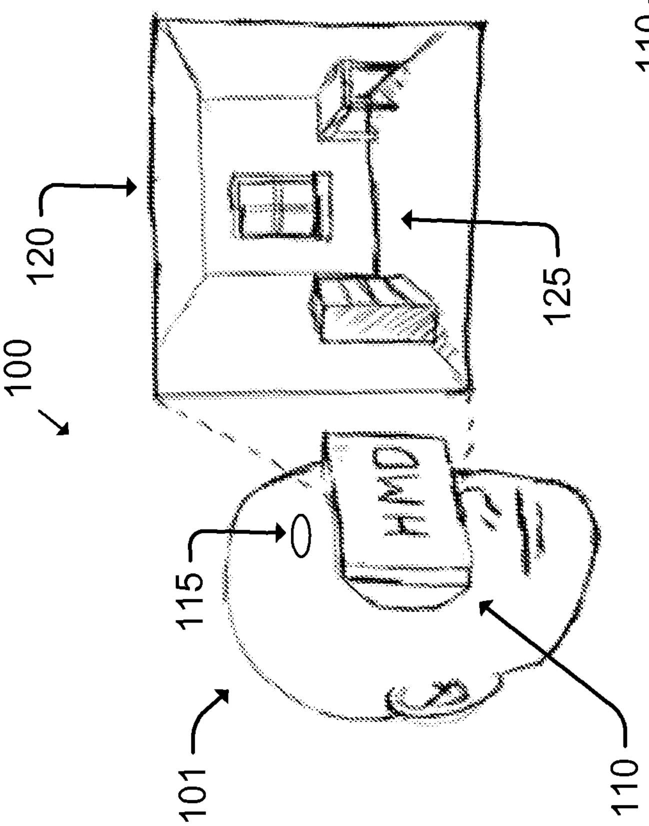


FIG. 1A

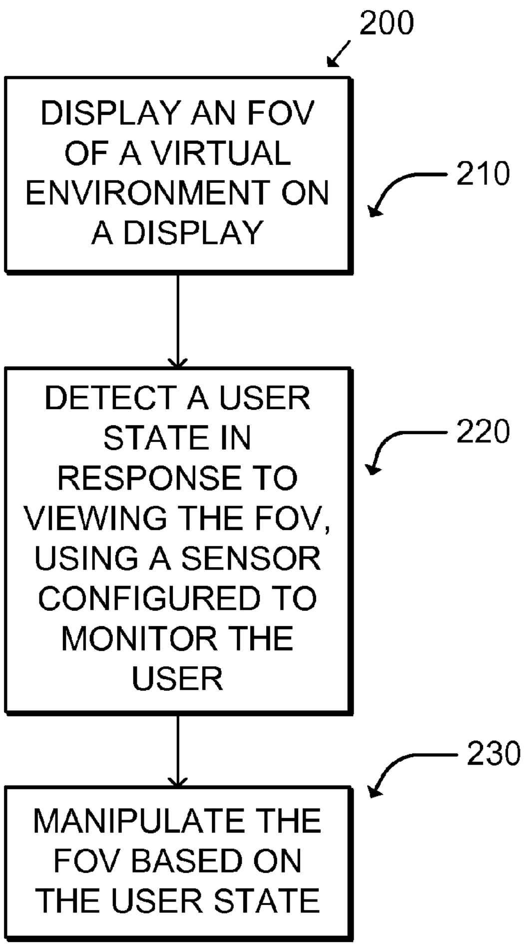

200
DISPLAY AN FOV OF A VIRTUAL ENVIRONMENT ON A DISPLAY
210
DETECT A USER STATE IN RESPONSE TO VIEWING THE FOV, USING A SENSOR CONFIGURED TO MONITOR THE USER
220
MANIPULATE THE FOV BASED ON THE USER STATE
230

FIG. 2A

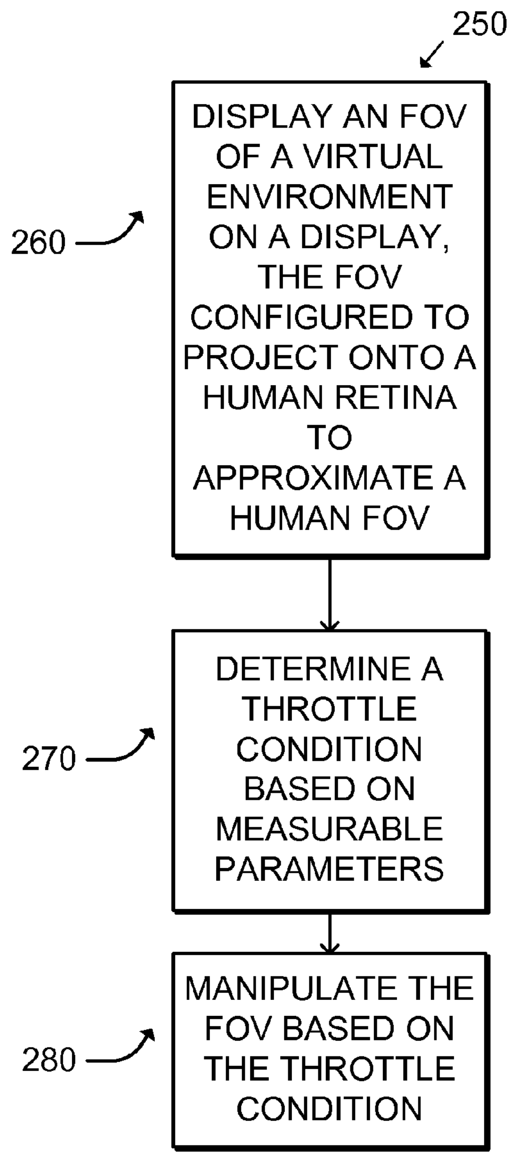

250
DISPLAY AN FOV OF A VIRTUAL ENVIRONMENT ON A DISPLAY, THE FOV CONFIGURED TO PROJECT ONTO A HUMAN RETINA TO APPROXIMATE A HUMAN FOV
260
DETERMINE A THROTTLE CONDITION BASED ON MEASURABLE PARAMETERS
270
MANIPULATE THE FOV BASED ON THE THROTTLE CONDITION
280

FIG. 2B

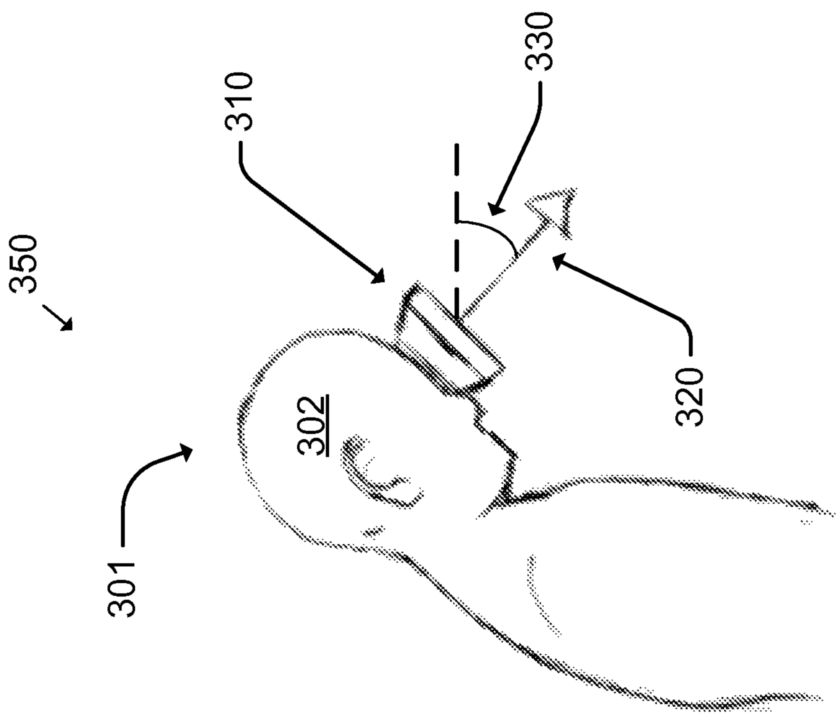


FIG. 3B

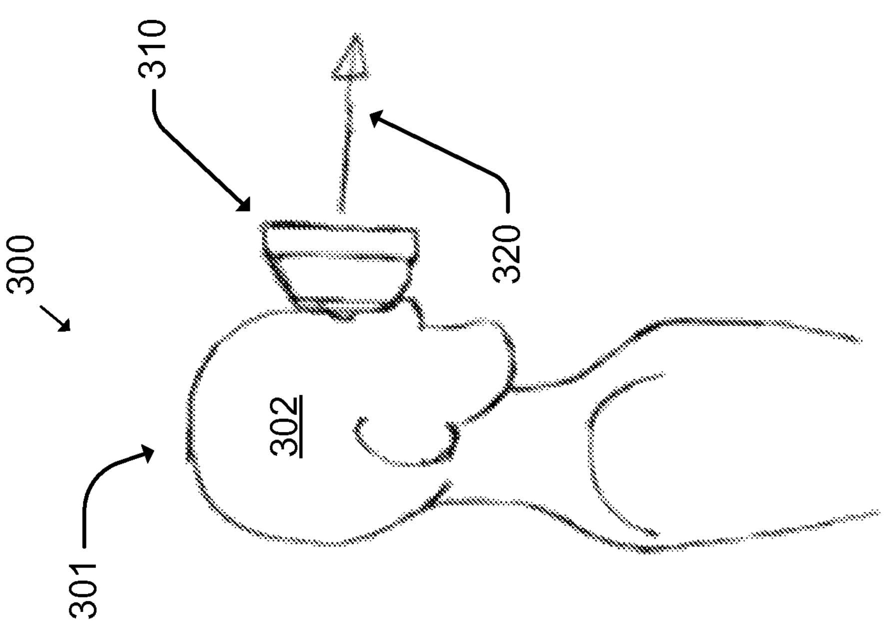


FIG. 3A

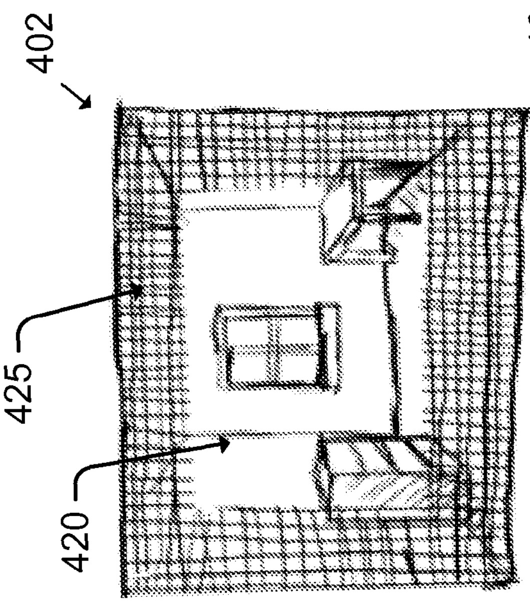


FIG. 4B

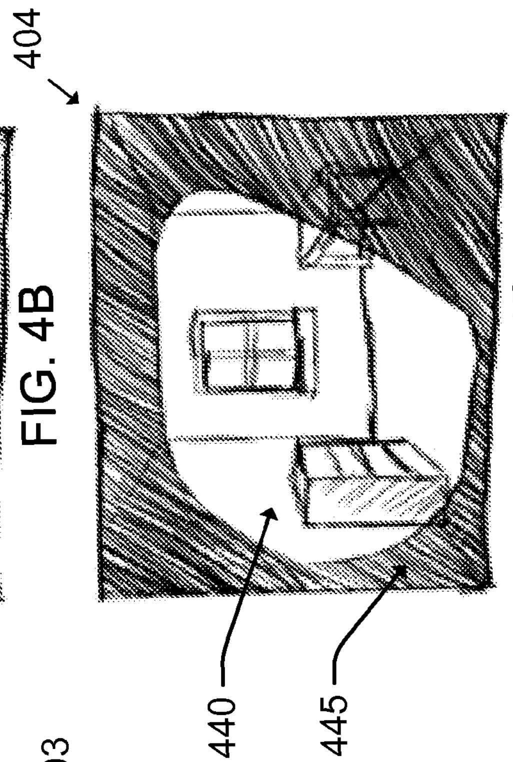


FIG. 4D

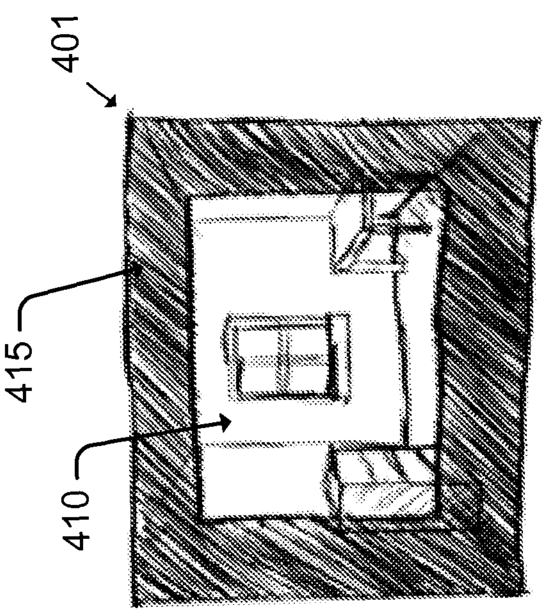


FIG. 4A

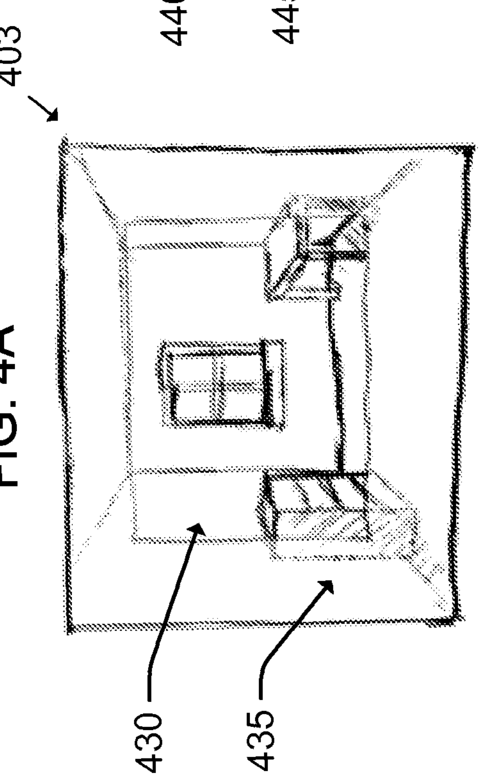


FIG. 4C

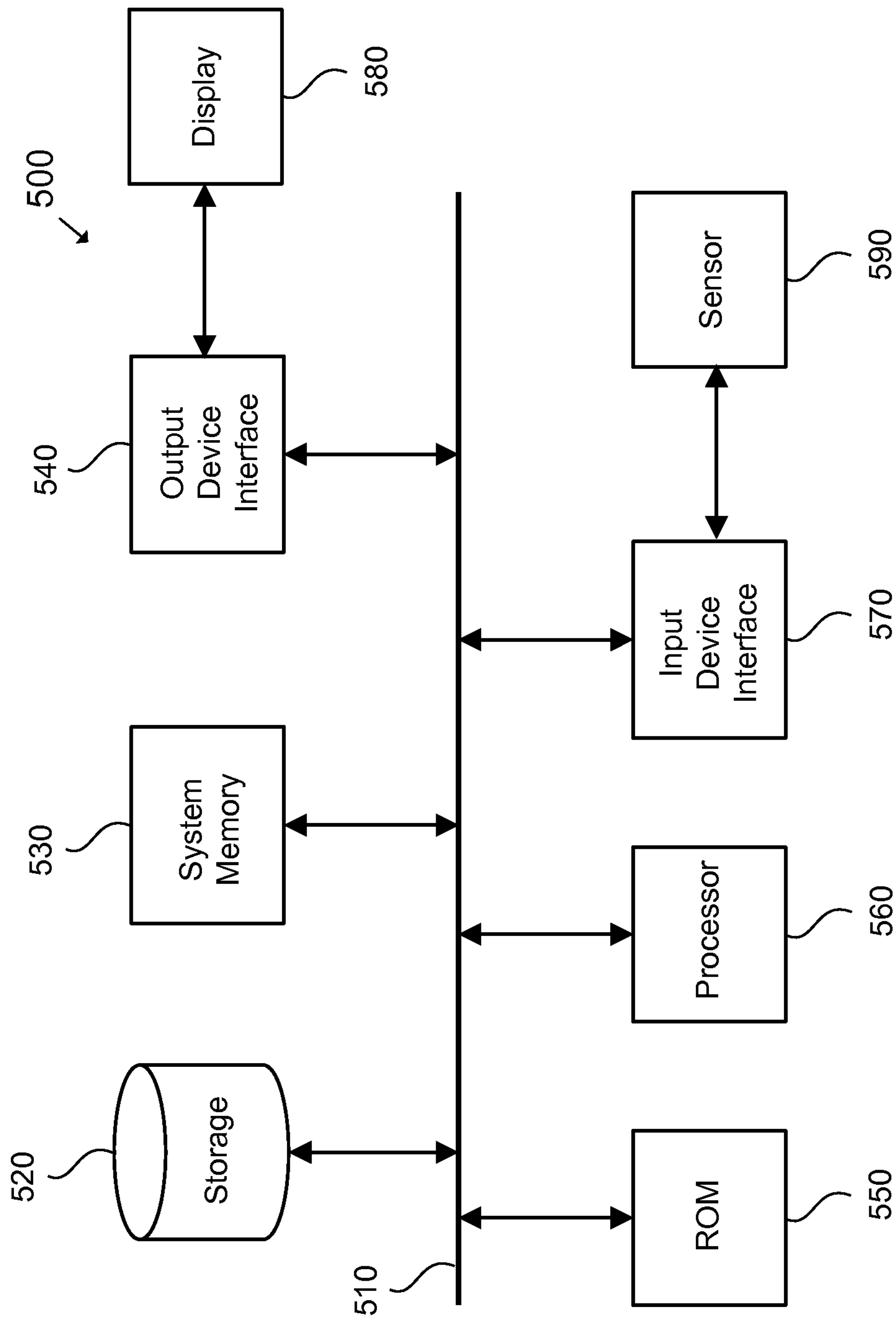

500
Display
580
Output Device Interface
540
System Memory
530
Storage
520
510
ROM
550
Processor
560
Input Device Interface
570
Sensor
590

FIG. 5

DYNAMIC FIELD OF VIEW THROTTLING AS A MEANS OF IMPROVING USER EXPERIENCE IN HEAD MOUNTED VIRTUAL ENVIRONMENTS

CROSS-REFERENCE TO RELATED APPLICATION

This application is based upon and claims priority to U.S. provisional patent application 61/799,758 entitled "Dynamic Field of View Throttling as a Means of Improving User Experience in Head Mounted Virtual Environments" filed Mar. 15, 2013, the entire content of which is incorporated herein by reference.

STATEMENT REGARDING FEDERALLY SPONSORED RESEARCH

This invention was made with government support under Grant No. W911 NF-04-D-0005, awarded by the Army Research Office (ARO). The government has certain rights in the invention.

BACKGROUND

Technical Field

This disclosure relates to head mounted displays for viewing virtual environments.

SUMMARY

Various factors within the field of view of a head-mounted display (HMD) may affect the user in ways that may cause one to experience simulator sickness or motion sickness. Some of these factors are related to virtual movement that may not be congruent to the user's actual movement. Others are related to the inaccurately rendered periphery, jitter in tracking, or delays in the refresh of the graphics. Many of these may be mitigated by restricting, expanding, or blurring regions of the visible area of the virtual environment being viewed in the HMD. Additionally, such manipulations can also influence the aesthetic qualities of the virtual scene or impact various aspects of the user's experience, performance, and behavior.

According to the present disclosure, the field of view as seen by a user of a virtual environment is dynamically throttled or restricted in real time in order to mitigate such simulator or motion sickness or to influence the user's experience or behavior in the virtual environment. The amount of restriction can be controlled based on a number of variables.

In one implementation, a non-transitory, tangible, computer-readable storage medium containing a program of instructions causes a computer system running the program of instructions to display, on a display of the computer system, a field of view (FOV) of a virtual environment. The computer system also detects, using a sensor configured to monitor a user state in response to viewing the FOV, the user state, and manipulate the FOV based on the user state.

In another implementation, a method of displaying a virtual environment comprises displaying, on a stereoscopic display, a field of view (FOV) of a virtual environment, determining, based on historical data and a current user physiological state, a visual alteration to the FOV, and rendering, on the stereoscopic display, the visual alteration to the FOV.

In yet another implementation, a virtual environment viewing device comprises a sensor for detecting physiological parameters, and a stereoscopic display connected to the sensor and configured to display a field of view (FOV) of a virtual environment, wherein the FOV is visually altered based on the physiological parameters.

These, as well as other components, steps, features, objects, benefits, and advantages, will now become clear from a review of the following detailed description of illustrative embodiments, the accompanying drawings, and the claims.

BRIEF DESCRIPTION OF DRAWINGS

The drawings are of illustrative embodiments. They do not illustrate all embodiments. Other embodiments may be used in addition or instead. Details that may be apparent or unnecessary may be omitted to save space or for more effective illustration. Some embodiments may be practiced with additional components or steps and/or without all of the components or steps that are illustrated. When the same numeral appears in different drawings, it refers to the same or like components or steps.

FIG. **1**A illustrates a user viewing a virtual environment through a head mounted display according to example aspects of the present disclosure.

FIG. **1**B illustrates the user of FIG. **1**A viewing the same virtual environment with the field of view restricted according to example aspects of the present disclosure.

FIG. **2**A-B illustrate a flow chart for field of view throttling according to example aspects of the present disclosure.

FIG. **3**A illustrates a user pose of a normal state when experiencing a virtual environment in a head-mounted display according to example aspects of the present disclosure.

FIG. **3**B illustrates the user of FIG. **3**A in a user pose indicating an abnormal state according to example aspects of the present disclosure.

FIG. **4**A illustrates field of view restriction using opaque polygons according to example aspects of the present disclosure.

FIG. **4**B illustrates field of view restriction using a screen mesh pattern according to example aspects of the present disclosure.

FIG. **4**C illustrates field of view restriction using a blur effect according to example aspects of the present disclosure.

FIG. **4**D illustrates field of view restriction using an asymmetric aperture according to example aspects of the present disclosure.

FIG. **5** illustrates a diagram of an electronic system according to example aspects of the present disclosure.

DETAILED DESCRIPTION OF ILLUSTRATIVE EMBODIMENTS

Illustrative embodiments are now described. Other embodiments may be used in addition or instead. Details that may be apparent or unnecessary may be omitted to save space or for a more effective presentation. Some embodiments may be practiced with additional components or steps and/or without all of the components or steps that are described.

The present disclosure describes graphical elements rendered in software to restrict or modify the rendered field of view (FOV) in a head mounted virtual environment. A head mounted display (HMD) includes a display that is seen through optics. The optics focuses the display onto the user's eyes such that the field of view is mapped to the user's natural field of view. As the user moves, trackers in the HMD allow detection of the yaw, pitch, and roll of the HMD, corresponding to the user's movements, to accordingly update the field of view of the virtual environment. In one implementation, the field of view is restricted using a graphically rendered occluding aperture consisting of opaque polygons that are drawn in front of the view of the remaining environment.

FIG. **1**A illustrates an implementation of a head mounted display environment **100**. A user **101** views a virtual environment **120** through an HMD **110**. The HMD **115** may be a virtual environment viewing device having a stereoscopic display, and may be attachable to the user **101**, or may be a head coupled display which can be placed near the user's eyes. The virtual environment **120** is graphically rendered with a certain field of view (FOV) **125**, which is determined by the specific hardware specifications of the HMD. A sensor **115** is attached to the user **101**, although in other implementations there may be additional sensors at additional locations, such as elsewhere along the user's body, or integrated with the HMD **110**. For example, one or more sensors **115** may be integrated within a portion of the HMD **110** which contacts the user **101**. The sensor **115** may wirelessly transmit data to the HMD **110** or other connected computer system, or alternatively can be connected through a cable or wire or otherwise configured to transmit data. The sensor **115** is configured to detect various parameters of the user's state, as described below. For example, a heart rate sensor may be used to detect fluctuations in the user's physiological state, which may be an early indicator of motion sickness. Alternatively, an electroencephalographic (EEG) sensor could be used to detect fluctuations in the user's cognitive state. These examples are illustrative, as other sensors may be used to monitor user state or environmental parameters, including both physical hardware devices and software measurements, depending on the goals of the application. The FOV **125** is not restricted or otherwise modified. To implement an occluding aperture, a series of connected polygons is rendered that form an occluding plane with a central window through which the virtual environment **120** can be seen. For example, FIG. **1**B illustrates a head mounted display environment **150**, including the user **101** viewing, through the HMD **110**, a virtual environment **130**, which may represent the same virtual environment. An FOV **135** appears restricted compared to the original or unrestricted FOV **125**, due to an occluding plane **140** with an aperture **142**. This window or aperture **142** can then be graphically scaled by performing geometric transforms standard to all computer graphics interfaces as a means of provide larger or small fields of view into the virtual environment. These parameters (aperture size, opacity, color, blurring, texturing, etc.) can be modulated based on a number of parameters, including measurements from the sensor **115**, as further described below. The method could also utilize semi-opaque, colored, blurred, or textured polygons as an alternative to complete opacity. The polygons or other techniques described here could also be used to represent virtual objects in front of, attached to, or following the user's head in the virtual environment.

The FOV may be modified through throttling, which refers to dynamically restricting the visible portion of the field of view as seen by a user of a virtual environment in real time. FIG. **2**A illustrates a flowchart **200** of an example method of throttling.

At **210**, an FOV, such as the FOV **125**, of a virtual environment, such as the virtual environment **120**, is displayed on a display, such as the HMD **110**. By default, the virtual environment is displayed using the full field of view supported by the head mounted display.

At **220**, a current user state is detected in response to viewing the FOV, using a sensor, such as the sensor **115**, configured to monitor the user. For example, the sensor may detect an abnormal physiological state, including various forms of sickness, or an expected abnormal physiological state, such as an expected or predicted sickness. At **230**, the FOV is manipulated based on the user state. For example, the FOV may be throttled or otherwise visually altered to mitigate or remedy the effects of the abnormal physiological state, or to prevent the expected abnormal physiological state. The throttling or visual alterations may be determined or selected based on the current user state, as detected by the sensor, and historical data which may correlate to physiological states or other parameters as described below. The HMD may continuously render the display, and the sensor may be continuously monitoring the user, such that the FOV may be dynamically and continuously manipulated. The HMD, or another communicatively connected computer device, may further have historical data to determine when an expected abnormal physiological state is detected.

FIG. **2**B illustrates a flowchart **250** of another example method of throttling. At **260**, an FOV of a virtual environment is displayed on a display. The FOV is configured to project onto a human retina to approximate a human FOV. In other words, the FOV maps onto the human FOV to approximate a 1-1 relationship. At **270**, a throttle condition based on measurable parameters is determined. The throttle condition may be based on sensors, as described herein, or other phenomena as described herein. At **280**, the FOV is manipulated based on the throttle condition. The FOV may be restricted, as described herein.

When a particular user state is detected (determined by a number of possible parameters), throttling is engaged, resulting in an artificially restricted field of view. When the state is no longer detected, the original field of view may be restored, either substantially instantaneously or gradually over time. The field of view can be throttled based on a number of parameters, such as the severity and manner of sickness experienced by the user, the likelihood of expected sickness in the future, the motions and behavior of the user, the user's physiological state, the user's cognitive or emotional state, the user's prior experiences, the user's performance of a specific task, the user's personal preferences, technical hardware or software performance, characteristics of the virtual scene, the amount of time in the virtual environment, and so forth, which may be detected by one or more sensors monitoring the user, user inputs, or other metrics and inputs.

Various factors within the field of view of a virtual environment affect the user in ways that may cause one to experience simulator sickness or motion sickness. Some of these factors are related to virtual movement that may not be congruent to the user's actual movement. Others are related to the inaccurately rendered periphery, jitter in tracking, or delays in the refresh of the graphics. Many of these can be mitigated by restricting, expanding, or blurring the visible area within a virtual environment. Therefore, the field of view as seen by a user of a virtual environment may be dynamically throttled in order to mitigate such simulator or motion sickness. In one implementation, throttling can be employed in response to observed symptoms, which may be determined by taking physiological measurements, observing changes in body posture or behavior, self-reports, or through other means using one or more sensors.

Biometric information such as breathing rate, sweating or perspiration measurement, skin conductance, galvanic skin response, or other physiological signals could be used as indicators of simulator sickness. Such objective measurements may be measured by a sensor and could therefore be used as a parameter for throttling the field of view.

In addition to responding directly to observed symptoms of simulator sickness, field-of-view throttling can also be used as a preventative measure to mitigate or reduce the likelihood of experiencing sickness in the future. For example, a throttled field-of-view may reduce exposure to known perceptual cues that are known to contribute to symptoms of simulator or motion sickness. Such data may be included in historical data.

For example, the likelihood of experiencing simulator sickness generally increases with longer amounts of time spent continuously immersed in a virtual environment. Therefore, in one implementation, the amount of time that has elapsed since the user entered the virtual environment is used to control the restriction. When a user first enters the virtual environment, the field of view is limited, for example, if the display's full field of view is 90 degrees, then the throttled field of view could be reduced to 40 degrees at the start of the experience. Over a period of time, the field of view is then gradually increased to the full 90 degrees. For example, the field of view may be expanded from the FOV **135** in FIG. **1**B to the FOV **125** in FIG. **1**A. This allows the user to acclimate to the virtual experience and could reduce the possibility of motion sickness.

The amount of time to acclimate and become comfortable with a virtual environment is likely to take less time for users that are experienced and familiar with interacting in virtual worlds, such as 3D games. Therefore, historical data including prior experience, such as self-reports on a pre-questionnaire, may be used as an input variable that controls field-of-view throttling. For example, the speed of throttling may be faster for experts and slower for novices, giving less experienced users greater time to acclimate to the experience.

Technical hardware and software performance are other parameters that could be used to control field of view throttling. For example, slow graphics rendering speeds and head tracking latency/noise are often associated with increased simulator sickness. Therefore, if rendering speeds are slow over a period of time, for example, averaging 10 frames per second over a one minute period, or any other threshold average framerate, then the field of view would gradually be reduced from the display's full field of view. If the rendering speeds later increased, for example averaging 60 frames per second, then the field of view would be increased to the full field of view. Similarly, if the tracking system that senses head orientation and/or position becomes noisy, or inaccurate, then the FOV could be reduced, until better tracking data is available.

Dynamic throttling may potentially reduce simulator sickness, but can also enhance or otherwise influence the user's experience of a virtual environment. For example, field of view throttling can be invoked as an aesthetic technique. For example, the field of view can be limited when the author of the environment wishes the user to be more focused on what is directly in front of them, such as in a battle scene. Afterwards, when the user transitions to an outdoor scene, this limited field of view could be expanded to make the virtual world feel more expansive. Alternatively, in an emotional scene, the field-of-view may be throttled to emphasize certain moments, in an attempt to increase its impact or provoke a greater response from the user.

It may be useful to dynamically throttle the field of view based on indicators of the user's cognitive state. For example, if the user appears to be confused or lost, it may be useful to temporarily expand the visible field of view to enhance spatial and situational awareness. This could be assessed by monitoring, through one or more sensors, movement through different areas of the virtual world, head motions (e.g. looking around frantically), brain or muscular activity, verbal feedback, or other factors.

The visible field of view could also be modified based upon user performance on specific tasks in a virtual environment. For example if a user is involved in competitive task (e.g. fight scene) and begins to do poorly, the field of view could be decreased to reduce distractions or enhance focus. Conversely, the field of view could be increased during tasks that require spatial or situational awareness, in order to allow greater visibility of the virtual scene.

Field of view throttling could also be employed as a reward mechanism. For example, the user's field of view may be increased as a result of successfully completing tasks in the virtual environment. In the context of a video game, this could be a visual representation of becoming stronger and gaining new abilities. Conversely, the field of view could be decreased to increase the difficulty of play or represent decreased situational awareness to represent the player becoming weaker.

While field of view throttling could be implemented entirely through automatic algorithms, it may also be useful to take the user's preferences into account. For example, the system may allow the user to select a "default" field of view that is used as a basis for automatic throttling based on other parameters. Alternatively, the user could be given direct control over the throttling parameters, either by gradually expanding/restricting the field of view at will, or by switching between multiple predefined field of view settings.

In one implementation, the field of view can be throttled based on changes in the user's motions and/or body posture. For example, one variable that could be used is a measurement of a user's average head declination, which could indicate that a user is experiencing motion sickness, as shown in FIGS. **1**A and **3**B.

FIG. **3**A shows a first user state **300**, which may correspond to a normal physiological state. A user **301** is using an HMD **310** attached to the user's head **302**. The HMD **310** includes trackers or other sensors (integrated with the HMD **310** but not visible in FIG. **3**A or **3**B), such as accelerometers and gyroscopes, which allows a display of the HMD **310** to dynamically render scenes of a virtual environment in response to the user's head movements. The trackers may detect a horizontal orientation **320** of the HMD **310**, which corresponds to an orientation of the head **302**. The head tracking data from the head mounted display's tracker may be averaged over time to determine that downward bias to the horizontal orientation **320**. A downward bias may indicate head declination that may be an indicator that the user **301** is experiencing simulator or motion sickness. FIG. **3**B shows the head **302** drooping, or be biased downward. The HMD **310** may detect that the horizontal orientation **320** is biased downward by an angle **330**. Specifically, if the tracking data from the display's head tracker indicated that the user's head **302** is consistently declined by a certain threshold, such as 10 degrees below horizontal, 20 degrees below horizontal, or other appropriate degree, the field of view may be throttled in response, such as decreasing the field of view to a smaller amount. Other measurements based on head or body tracking data could also be generated algorithmically.

FIG. **4**A illustrates a virtual environment screen **401** manipulated through opaque polygons **415** to fully occlude the periphery and restrict a field of view **410**. However, partial occlusion may also be used. FIG. **4**B illustrates an example of partial occlusion using a textured mask **425** with a pattern of occlusion, such as partial occlusion of the periphery of a virtual environment screen **402** in a screen mesh pattern to partially restrict a field of view **420**.

Visual effects that alter the appearance of certain areas of the field-of-view could also be used, such as blurring, reduction of brightness, or recoloring. For example, blur could be applied to the outer regions of the image, thus the central region can remain crisp, while the periphery would be softened. FIG. **4**C illustrates a field of view restriction using a blur effect **435** that renders the peripheral region of a virtual environment screen **403** out of focus, while the central region remains sharp, to partially restrict a field of view **430**. This could potentially reduce feelings of sickness or disguise unwanted visual artifacts while retaining the feeling of an expansive field of view.

A simple implementation of throttling would reduce the visible field of view equally in all directions, thereby creating a circular aperture. However, field of view restriction can also be applied unequally. For example, an oval, ellipse, rectangle, or any other arbitrary shape could be used. Additionally, it may also be advantageous to consider apertures derived from human anatomy, such as the shape created by scaling down the contour of an individual's visual field. FIG. **4**D illustrates field of view restriction using an asymmetric aperture derived by scaling down the counter of the human visual field. A virtual environment screen **404** is occluded by an asymmetric frame **445**, resulting in a field of view **440**. Although FIG. **4**D illustrates the asymmetric frame **445** as opaque, similar to the opaque polygons **415**, in other implementations the asymmetric frame **445** may be rendered through other visual effects as described above. Asymmetric field of view restriction could also potentially be used to guide the user to look in a certain direction. For example, to guide the user to look or move to the left, the field of view could be less restricted on the left side as opposed to the right side. Field of view restrictions could also be used for non-head-mounted or head-coupled virtual environments. For example, applying an aperture derived from human anatomy may provide greater engagement for imagery displayed on a standard computer monitor or cinema screen. In these cases, the imagery could be based on the user's head motion or presented entirely independent of head motion.

The field of view may be restricted by rendering a series of opaque polygons in front of the camera (viewpoint) of the virtual environment, creating an aperture or frame through which the virtual world is visible. This frame can be scaled upwards or downwards using standard graphics transformations to increase or decrease the visible field of view.

In an alternative implementation, the field of view restriction is achieved by rendering one or more polygons in front of the camera that represent the "window" through which the virtual world should be visible. A filter is applied through standard graphics libraries (e.g. OpenGL) so that the virtual environment is only visible through the window. Similar to the aperture described above, this window can be scaled upwards or downwards using standard graphics transformations to increase or decrease the visible field of view.

In some head-mounted displays, it may be necessary to predistort the images in software prior to displaying them on screen, in order to correct for optical distortion introduced by the physical lens. One method of achieving this involves rendering the scene to a textured polygon or series of polygons in front of a virtual camera, and then applying a series of graphical transformations to warp the geometry in a way that cancels out the lens distortion. In this implementation, the field of view can be reduced by rendering opaque polygons between the render-to-texture polygons and the virtual camera, thereby occluding particular regions of the visual field. Alternatively, the texture coordinates for the render-to-texture polygons could be modified to prevent the virtual scene from being rendered on the regions that should be restricted from view.

Additionally, in a standard graphics pipeline, 3D virtual environments are eventually rendered to one or more 2D viewports. To manipulate the visible field of view, the color values of the individual pixels in these viewports can be altered or blacked out using a variety of methods, such as a lookup table, a series of rules (if/then statements), mathematical equations, and so forth. In this implementation, the correspondences of individual pixels to specific areas of the field-of-view will vary based on the technical specifications of the particular head-mounted display chosen.

FIG. **5** conceptually illustrates an example computer system or electronic system with which some implementations of the subject technology can be implemented. Electronic system **500** can be a computer, phone, PDA, or any other sort of electronic device. The electronic system **500** may be integrated with a head mounted display, such as the HMD **110**, may be partially integrated with the head mounted display, or may be external to and in communication with the head mounted display. Such an electronic system includes various types of computer readable media and interfaces for various other types of computer readable media. Electronic system **500** includes a bus **510**, processing unit(s) **560**, a system memory **530**, a read-only memory (ROM) **550**, a permanent storage device **520**, an input device interface **570**, an output device interface **540**, a display **580**, and a sensor **590**.

The bus **510** collectively represents all system, peripheral, and chipset buses that communicatively connect the numerous internal devices of electronic system **500**. For instance, the bus **510** communicatively connects the processing unit(s) **560** with the ROM **550**, system memory **530**, and permanent storage device **520**. Portions of the bus **510** may be wireless.

From these various memory units, the processing unit(s) **560** retrieves instructions to execute and data to process in order to execute the processes of the present disclosure. The processing unit(s) **560** can be a single processor or a multi-core processor in different implementations.

The ROM **550** stores static data and instructions that are needed by the processing unit(s) **560** and other modules of the electronic system. The permanent storage device **520**, on the other hand, is a read-and-write memory device. This device is a non-volatile memory unit that stores instructions and data even when electronic system **500** is off. Some implementations of the subject disclosure use a mass-storage device (for example, a magnetic or optical disk and its corresponding disk drive) as the permanent storage device **520**.

Other implementations use a removable storage device (for example, a floppy disk, flash drive, and its corresponding disk drive) as the permanent storage device **520**. Like the permanent storage device **520**, the system memory **530** is a read-and-write memory device. However, unlike the storage device **520**, the system memory **530** is a volatile read-and-write memory, such as a random access memory. The system memory **530** stores some of the instructions and data that the processing unit(s) **560** needs at runtime. In some implementations, the processes of the present disclosure are stored in the system memory **530**, the permanent storage device **520**, or the ROM **550**. For example, the various memory units include instructions for receiving user activity data and updating dimensions in accordance with some implementations. From these various memory units, the processing unit(s) **560** retrieves instructions to execute and data to process in order to execute the processes of some implementations.

The bus **510** also connects to input device interface **570** and output device interface **540**. The input device interface **570** enables the user to communicate information and select commands to the electronic system. Input devices used with input device interface **570** include, for example, alphanumeric keyboards and pointing devices (also called "cursor control devices"). The sensor **590**, which may correspond to the sensor **115**, may be one or more sensors configured to detect a user state, such as physiological and other parameters as described above, and may be further configured as trackers for a head mounted display. The output device interface **540** enables, for example, the display of images generated by the electronic system **500**. Output devices used with the output device interface **540** include, for example, printers and display devices, for example, cathode ray tubes (CRT) or liquid crystal displays (LCD), which may be used as a display for the head mounted display. Some implementations include devices, for example, a touchscreen that functions as both input and output devices.

Finally, the bus **510** also couples the electronic system **500** to a wired or wireless network (not shown). In this manner, the computer can be a part of a network of computers (for example, a local area network (LAN), a wide area network (WAN), or an Intranet, or a network of networks, for example, the Internet. Any or all components of electronic system **500** can be used in conjunction with the present disclosure.

Unless otherwise indicated, the devices and processes that have been discussed herein are implemented with a computer system configured to perform the functions that have been described herein for the component. Each computer system includes one or more processors, tangible memories (e.g., random access memories (RAMs), read-only memories (ROMs), and/or programmable read only memories (PROMS)), tangible storage devices (e.g., hard disk drives, CD/DVD drives, and/or flash memories), system buses, video processing components, network communication components, input/output ports, and/or user interface devices (e.g., keyboards, pointing devices, displays, microphones, sound reproduction systems, and/or touch screens).

Each computer system may include software (e.g., one or more operating systems, device drivers, application programs, and/or communication programs). When software is included, the software includes programming instructions and may include associated data and libraries. When included, the programming instructions are configured to implement one or more algorithms that implement one or more of the functions of the computer system, as recited herein. The description of each function that is performed by each computer system also constitutes a description of the algorithm(s) that performs that function.

The software may be stored on or in one or more non-transitory, tangible storage devices, such as one or more hard disk drives, CDs, DVDs, and/or flash memories. The software may be in source code and/or object code format. Associated data may be stored in any type of volatile and/or non-volatile memory. The software may be loaded into a non-transitory memory and executed by one or more processors.

The components, steps, features, objects, benefits, and advantages that have been discussed are merely illustrative. None of them, nor the discussions relating to them, are intended to limit the scope of protection in any way. Numerous other embodiments are also contemplated. These include embodiments that have fewer, additional, and/or different components, steps, features, objects, benefits, and advantages. These also include embodiments in which the components and/or steps are arranged and/or ordered differently.

Unless otherwise stated, all measurements, values, ratings, positions, magnitudes, sizes, and other specifications that are set forth in this specification, including in the claims that follow, are approximate, not exact. They are intended to have a reasonable range that is consistent with the functions to which they relate and with what is customary in the art to which they pertain.

All articles, patents, patent applications, and other publications that have been cited in this disclosure are incorporated herein by reference.

The scope of protection is limited solely by the claims that now follow. That scope is intended and should be interpreted to be as broad as is consistent with the ordinary meaning of the language that is used in the claims when interpreted in light of this specification and the prosecution history that follows, except where specific meanings have been set forth, and to encompass all structural and functional equivalents.

Relational terms such as "first" and "second" and the like may be used solely to distinguish one entity or action from another, without necessarily requiring or implying any actual relationship or order between them. The terms "comprises," "comprising," and any other variation thereof when used in connection with a list of elements in the specification or claims are intended to indicate that the list is not exclusive and that other elements may be included. Similarly, an element preceded by an "a" or an "an" does not, without further constraints, preclude the existence of additional elements of the identical type.

None of the claims are intended to embrace subject matter that fails to satisfy the requirement of Sections 101, 102, or 103 of the Patent Act, nor should they be interpreted in such a way. Any unintended coverage of such subject matter is hereby disclaimed. Except as just stated in this paragraph, nothing that has been stated or illustrated is intended or should be interpreted to cause a dedication of any component, step, feature, object, benefit, advantage, or equivalent to the public, regardless of whether it is or is not recited in the claims.

The abstract is provided to help the reader quickly ascertain the nature of the technical disclosure. It is submitted with the understanding that it will not be used to interpret or limit the scope or meaning of the claims. In addition, various features in the foregoing detailed description are grouped together in various implementations to streamline the disclosure. This method of disclosure should not be interpreted as requiring claimed implementations to require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive subject matter lies in less than all features of a single disclosed implementation. Thus, the following claims are hereby incorporated into the detailed description, with each claim standing on its own as separately claimed subject matter.

The invention claimed is:

1. A non-transitory, tangible, computer-readable storage medium containing a program of instructions that cause a computer system running the program of instructions to

display an incoming image on a head-mounted display with a perimeter portion of the image being occluded on the display by the program of instructions; and

one or more of the following:

detect, using a sensor configured to monitor a user state in response to viewing the image, the user state, and dynamically throttle the amount of the occlusion in response to the user state being in an abnormal physiological state or an expected abnormal physiological state;

wherein the amount of the occlusion is manipulated by rendering the occluded portion to a textured polygon and manipulating the textured polygon;

wherein the amount of occlusion is based on one or more measurable parameters that include a virtual environment content, a system parameter, or a system performance;

wherein the amount of occlusion is based on an amount of elapsed time;

wherein the amount of occlusion is based on a user's previous virtual environment experience; or

wherein the amount of occlusion is based on influencing a user's experience with the virtual environment, influencing a task performance of the user, or completed tasks within the virtual environment.

2. The non-transitory, tangible, computer-readable storage medium of claim **1**, wherein the program of instructions further causes the computer system to:

detect, using a sensor configured to monitor a user state in response to viewing the image, the user state; and

dynamically throttle the amount of the occlusion in response to the user state being in an abnormal physiological state or an expected abnormal physiological state.

3. The non-transitory, tangible, computer-readable storage medium of claim **1**, wherein the occlusion includes an opaque polygon.

4. The non-transitory, tangible, computer-readable storage medium of claim **1**, wherein the occlusion includes a screen mesh pattern.

5. The non-transitory, tangible, computer-readable storage medium of claim **1**, wherein occlusion is symmetric.

6. The non-transitory, tangible, computer-readable storage medium of claim **1**, wherein the occlusion is asymmetric.

7. The non-transitory, tangible, computer-readable storage medium of claim **1**, wherein the amount of the occlusion is manipulated by rendering the occluded portion to a textured polygon and manipulating the textured polygon.

8. The non-transitory, tangible, computer-readable storage medium of claim **1**, wherein the instructions detect, using a sensor configured to monitor a user state in response to viewing the image, the user state, and dynamically throttle the amount of the occlusion in response to the user state being in an abnormal physiological state or an expected abnormal physiological state, and the detecting the user state further comprises detecting, using the sensor, physiological measurements of the user.

9. The non-transitory, tangible, computer-readable storage medium of claim **1**, wherein the instructions detect, using a sensor configured to monitor a user state in response to viewing the image, the user state, and dynamically throttle the amount of the occlusion in response to the user state being in an abnormal physiological state or an expected abnormal physiological state, and the sensor is further configured to detect at least one of a behavior of the user, a head and body posture of the user, and a cognitive state of the user.

10. The non-transitory, tangible, computer-readable storage medium of claim **1**, wherein the amount of the occlusion is automatically throttled to prevent expected motion sickness.

11. The non-transitory, tangible, computer-readable storage medium of claim **1**, wherein the amount of the occlusion is based on one or more measurable parameters that include a virtual environment content, a system parameter, or a system performance.

12. The non-transitory, tangible, computer-readable storage medium of claim **1**, wherein the amount of the occlusion is based on an amount of time elapsed.

13. The non-transitory, tangible, computer-readable storage medium of claim **1**, wherein the amount of the occlusion is based on a user's previous virtual environment experience.

14. The non-transitory, tangible, computer-readable storage medium of claim **1**, wherein the amount of the occlusion is based on influencing a user's experience with the virtual environment, influencing a task performance of the user, or completed tasks within the virtual environment.

15. The non-transitory, tangible, computer-readable storage medium of claim **1**, wherein the amount of occlusion is based on a user preference.

16. A method for displaying a virtual environment, the method comprising:

displaying, on a stereoscopic display, a field of view (FOV) of a virtual environment;

determining, based on historical data and a current user physiological state, a visual alteration to the extent of the FOV; and

rendering, on the stereoscopic display, the visual alteration to the extent of the FOV.

17. A virtual environment viewing device comprising:

a sensor for detecting physiological parameters; and

a stereoscopic display connected to the sensor and configured to display a field of view (FOV) of a virtual environment, wherein the extent of the FOV is visually altered based on the physiological parameters.

* * * * *