

# Visual Impairment Simulation for Inclusive Interface Design

Veljko B. Petrović

University of Novi Sad—Faculty of Technical Sciences

Dositeja Obradovića 6  
21000, Novi Sad, Serbia

pveljko@uns.ac.rs

Dragan Ivetić

University of Novi Sad—Faculty of Technical Sciences

Dositeja Obradovića 6  
21000, Novi Sad, Serbia

ivetic@uns.ac.rs

## ABSTRACT

This paper describes research in developing disability simulation used for inclusive design of user interfaces. It presents a medium-fidelity prototype which simulates visual impairment caused by Age-Related Macular Degeneration in real time on arbitrary user interfaces, and describes how the prototype design was arrived at. It does so by surveying previous work in the field, identifying broad trends, and systematizing problems visual impairment simulation systems must solve. This systematization focuses on issues of simulator portability, the importance of eye-tracking, the vital nature of real-time performance, the flexibility of the solution, and veracity of the simulator to actual AMD symptoms.

## Keywords

HCI, AMD, macular, impairment, inclusive

## 1. INTRODUCTION

This paper describes the development of a disability simulation framework focusing on visual impairment, specifically visual impairment caused by maculopathy, common in disorders such as Age-Related Macular Degeneration (AMD). This framework is developed in order to support inclusive design, a term used to denote design focused on universal usability—allowing for maximal possible variability in users targeted by an interface. The importance of this is underlined by Shneiderman's eight golden rules of interface design being updated to include universal usability[Shn09a]

AMD was chosen as a subject of particular study because of how common it is and how much more common it is likely to become, given that it is a gerontological disorder, and the human lifespan is increasing[Uni02a]. It also—as shall be presented later—presents with a wide variety of symptoms, making it a useful test-case for considering the modeling and simulation of visual impairment in general.

The paper first presents the case that there is a problem with user interfaces (UI) and people with AMD, then that the problem is not worth dismissing, and finally that disability simulation is a valid approach to ameliorating that problem. The paper then presents previous work in this field, identifies certain trends and suggests, based on those trends,

the need for a more comprehensive framework for disability simulation, using the AMD focus as both illustrative and alone worth the effort. A systematization of problems facing a visual impairment system some of which are addressed in previous work and some not, is presented, and then used to outline a visual impairment system a medium-fidelity prototype of which is then presented.

This paper is divided into seven sections: the first is the introduction, the second discusses the validity of the approach and previous work in the field, the third outlines the problems a general visual impairment framework must solve, the fourth outlines the software prototype implemented, the fifth outlines the conclusion and further avenues of research, the sixth contains the acknowledgements, and the seventh contains the references.

## 2. IMPAIRMENT AND SIMULATION

The first question that arises when considering this research is whether it represents a suitable investment of time and attention: is AMD (and by extension visual impairment) a big enough problem? The answer to this question follows from the nature of AMD as a disorder and its epidemiology.

### AMD—Nature, Epidemiology, and Risk

A detailed aetiology of Age-Related macular degeneration is outside the scope of this paper, however, for purposes of orientation it suffices to say that AMD is a progressive degenerative disease of the macula, a region of the retina responsible for central (as opposed to peripheral) vision. It is divided

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Short Papers Proceedings

into ‘dry’ and ‘wet’ varieties, but for the purposes of designing interfaces there is no significant difference between the two. Its cause is unknown, and there is no effective treatment (though certain forms of surgery cure certain forms of ‘wet’ AMD by reversing malign neoplasia of retinal blood vessels).

Are there likely to be many users with AMD? The total number of computer users is increasing. This is especially true if all devices with interfaces are counted, such as phones, tablets, consoles, and smart TVs. As the number of users increases, it is expected that a significant part of the increase will be from the elderly, partially from increased market penetration and partially from the existing user-base growing older. This is inevitable as for some key technologies the market penetration is rapidly approaching 100%. According to the ITU[Int13a] the penetration rate for mobile phones is 96% world-wide, and the penetration rate for an online presence is 39%. These numbers become 100% and 77% respectively if only Europe is considered. As these numbers increase—as trends indicate—they will invariably include the elderly as well, especially since, according to the UN[Uni02a], the population of Europe is aging significantly. It is expected that the population of over-sixties will reach 28.8% by 2025.

One can establish a lower bound for the number of interface users in this demographic by looking at interface use proxies: the prevalence of social network use for people over 65 is 32%[Dug13a]The upper bound trends towards 100% as technological progress mandates the use of interfaces in order to have an independent life.

The prevalence of AMD is difficult to determine since diagnosing AMD is a nontrivial task, and it often goes unreported in its earlier stages. However, a number of studies have been done on the epidemiology of AMD, with markedly varied results. Less conservative estimates give such results as 64 % of people over eighty[DeJ06a] modulated by certain risk factors[Mar11a]. More conservative estimates broadly agree on lower but still worrying levels of prevalence, with results such as 11.90% for men over 80, and 16.39% for women over 80[Gro04a], or 3.7% for people between 75 and 84, and 11.0% for the total population over 85[Vin95a]. Either way, with the aging of the population being what it is, this prevalence is expected to double in the future and to increase by at least 50% by 2020[Gro04a].

If the more conservative figures are applied to the USA—serving here as a model nation due to easy access to detailed census data[Bur15a]—we find that according to [Gro04a] the estimated number of people with AMD is 1,658,000. The Rotterdam study, one the other hand, indicates results of an approximate total of 1,087,000. These approximate results indicate that just under 1% of the adult

population of the United States has AMD, not counting earlier cases of the disease in the 55-70 range. Of course, as the population ages and life-extending medical care becomes more sophisticated, this number will increase.

It is evident, therefore, that there does exist a population of users with AMD. Does this population have a significant amount of difficulty when using interfaces? A systematization of the symptoms of AMD can be seen in section 3.1, but briefly, AMD leads to general loss of acuity, loss of color and contrast sensitivity, gaps in the visual field first visible in text, the loss of a central (foveal) sight (in whole or in part), and unpredictable shifting deformation of the visual field (metamorphopsia). This is a considerable amount of impairment and previous research in this field[SCO02a] shows conclusively that conventional interfaces are not suitable for people with AMD.

### Disability Simulation and Interfaces

Disability simulation is the practice of creating some sort of apparatus which simulates the experience of having some sort of disability. Its original purpose was as an aid of empathy[Wil69a], but careful analysis shows that it is flawed in achieving this[Flo07a]. However, it can still be used to foster a rather more practical form of empathy—simulating a disability is a great way for a designer to gauge how a design will be perceived and used by people with disabilities. This can be the virtual modeling of users for the purposes of ergonomic design[Kak12a], the purposes of rehabilitation and accessibility design[Har14a], or for the purposes of UI design as in the Cambridge Impairment Simulator[Bis13a][Bis12a].

Of course, when doing usability testing nothing can possibly replace testing using people who actually have AMD (or other disabilities and disorders), but the use of disability simulation—occasionally also called user modeling—is crucial in allowing for the iterative testing of an interface. This iterative testing using simulation and approximation is crucial for what’s referred to as ‘inclusive design’ as opposed to designing the interface exclusively for the able-bodied and then adding accessibility features later. The utility of disability simulation is such that it was the focus of a Horizon 2020 FP7 EU project[Ver15a], which included visual impairment as well[Sul13a].

Thus, clearly, there does exist a significant problem and it is very likely that disability simulation is the way it can be at least ameliorated.

### Previous Work

The idea of disability/impairment simulation is not new and has been explored in various settings for various applications. A survey of the literature has shown that previous work can be reasonably divided

into either application-specific simulators or universal attempts to simulate impairment. Application-specific simulators focus on one specific application either because they focus on researching one activity to the exclusion of others or because they deliberately reduce their focus to a specific platform to increase their ability to accurately simulate impairment.

The activities researched with application-specific simulators of the first kind are mostly those tasks that impact most heavily on independent life: reading and driving. Driving studies evaluate how well affected people can drive, and how much help visual aids are [Pel05a]. In the matter of reading there's been research on the eye-movements of the impaired [Pid06a] with applications in rehabilitation [Var04a] or in visual aid development [Har14a]. Likewise, application-specific simulators sometimes focus on the application platform such as Swing/NetBeans [Vot09a].

Attempts to simulate impairment for any sort of application are generally focused on acquiring the video output of GUI rendering and then modifying it in order to simulate the effects of impairment. Some of these are heavily hardware based, such as the case with parts of the Inclusive Design Toolkit [Inc15a] which relies on specially made glasses to simulate certain visual impairments, but most solutions are predominantly software-based. The most sustained work on this field is the work on the groundbreaking Cambridge Impairment Simulator [Bis13a][Bis12a][Goo07a] which is a vital part of the Inclusive Design Toolkit and the relevant perceptual model [Bis08a]. A number of tools have also been developed partially or fully outside of academia. These tools purport to help with inclusive design by simulating visual impairments. The most interesting of such projects are the Visual Impairment Simulator [Vis15a] and WebAIM Low Vision Simulator [Web15a].

Lastly, a few solutions do not fit these categories: Some research has been done in using simulations to evaluate the severity of various impairments from a medical point of view [Fin99a], and there is also work on visual field simulation which touches on the subject of visual impairment simulation but focuses, instead, on optimal resolution for gaze-contingent displays [Per02a].

The solutions analyzed are equally heterogeneous in their means and their ends. One quarter use an analogue system for vision alteration, relying on specialized lenses that deform the user's visual field. The rest rely on active simulation, either using software tools (66.67%) or specialized hardware (8.33%) of those, 33.33% are gaze-contingent, and the rest (36.36%) either ignore gaze or use a gaze-proxy. Also heterogeneous are the fields and ultimate

goals of the solutions: 41.67% are fundamentally ophthalmological in purpose, half are intended to aid inclusive design, and 8.33% are special purpose.

Each solution succeeds on its own terms, resolving those problems the authors intended to tackle. However, as is the case in any research there are still open questions to be addressed. One of the key things to consider with all of these solutions is that they pick and choose which symptoms they simulate and to which extent. In certain cases, such as in [Har14a] or [Per02a] this is clearly a deliberate choice because only some factors were of interest to the authors. In other cases the choice is not deliberate, but is instead a unwanted but necessary compromise with technological limitations. Either way, it is necessary to acknowledge the limits of what was simulated in order to be able to ascertain the applicability of the simulation to actual design work.

### 3. OPEN QUESTIONS

This section deals with the open questions left after the previous work, especially those whose answer pertains to the development of a general framework for visual impairment simulation. AMD is used as a test-case because it is significant, sufficiently frequent, and presents with a wide array of symptoms. The questions to be answered can be organized into questions of:

- veracity,
- performance,
- universal applicability, and
- scalability.

It should be pointed out that these are not questions *entirely* unaddressed in previous work. Rather, their central nature is such that, even when they have been addressed, further work is necessary. In brief, veracity means that the framework must replicate the impairment as accurately as it is possible, performance means the framework must allow simulators to run in real-time, universal applicability means that the framework must allow for a wide selection of target interfaces, and scalability means that the framework must be accessible as simply as possible to as large an amount of interface designers as possible regardless of budget.

#### Veracity

It is not immediately obvious why veracity is important. It is quite reasonable to say that it is only necessary to simulate the 'important' symptoms of a visual impairment while leaving the others out. The difficulty, of course, is to determine what 'important' is for the purposes of interface design. To assume what is important to an interface is to ignore the perspective of the visually impaired—the exact same empathy deficiency disability simulation was created to solve [Wil69a].

Previous work clearly addresses this question, but does so in a haphazard fashion—not due to incompetence or oversight, but due to different focus. Not one of the surveyed solutions, for instance, implemented metamorphopsia, and most focused on the most obvious symptom: the central scotoma.

It is, however, easy to say that the simulation must be faithful to the impairment it seeks to emulate. It is quite more difficult to say how such a thing may be done. Using AMD as an example the first step is to gather the symptoms as they are described in the medical literature:

- a) The user may experience reduced general visual acuity[Sco02a]
- b) The user may experience reduced ability to perceive color correctly[Sco02a]
- c) The user may experience a difficulty[Sco02a] discriminating between similar light levels in a picture as measured by the Pelli-Robson Contrast Sensitivity Chart.
- d) The user may experience minor gaps in their visual field causing letters to drop out of text[Dej06a] or for lettering on densely-formatted documents to seem misaligned causing problems in, e.g., reading tables.
- e) The user may experience more significant foveal (central sight) scotoma (gaps in the visual field), blocking portions of the visual field[Dej06a]. These gaps may be visible as voids, spots, or deformations. Voids are filled in by the visual cortex in the same way the scotoma caused by the optic nerve is, spots are visibly dark or black, and deformations are visibly flickering as in the case of the scintillation scotoma or in some way distorted images which block part of the visual field.
- f) The user may experience a complete loss of central sight[Dej06a].
- g) The user may experience significant metamorphopsia—a deformation of the visual field which causes straight lines to appear curved and shifting[Dej06a][Rio08a] and causes visual elements to appear misaligned.
- h) The user may experience complete (for legal purposes) loss of vision[Dej06a][Rio08a].

Once the symptoms are gathered it is tempting to provide ad-hoc implementations for all of them. However principles of good design, not to mention the sheer number of possible impairments preclude this approach. While the development of a full visual impairment modeling language is beyond the scope of this paper—though one is being developed—the simplest way to understand symptoms of visual impairments from the point of view of the

simulator/framework designer is to divide them into the selector and effector components. Selectors determine which part of the visual field is affected and can be composited from such components as: the whole field, vision of the fovea, vision of the foveola, peripheral vision, random subsections, and text, where compositing is done using simple set intersection. Effectors control how the selected areas of the visual field are modified. One possible way to systematize such changes is to base them on visual variables.

Variable	Symptoms
Position	(a)(d)(e)(f)(g)
Size	(a)(g)
Shape	(a)(g)
Value	(c)
Color	(b)
Orientation	(g)
Texture	(a)(d)(c)(g)

**Table 1. Mapping symptoms to visual variables.**

Visual variables[Ber83a][Gar09a] are a system of describing various ways in which an image informs the viewer. Originally intended as a way of systematizing and discussing cartography, they were later adapted to various other problems including interfaces and visualization[Car03a]. The visual variables are: position, size, shape, value, color, orientation, and texture. Table 1 schematizes the connection between variables and AMD symptoms for purposes of illustration.

These connections are useful in the broader context of developing a universal approach to disability simulation and modeling, as the changes made by the impairment to certain subsections of the visual field can be explained in terms of effectors corresponding to visual variables, changing, say, position, or value, or texture or some combination thereof.

It should be noted that appropriately simulating most of these symptoms demands discriminating between central vision and peripheral vision which necessitates both some way of tracking the user's gaze and knowing the distance between the user and the display. Distance is necessary because the description of the visual field must be in terms of degrees of the visual field. Converting this into pixels demands the distance from the display. Not all of the previous proposed solutions consider this, with only those designed for ophthalmological purposes paying much attention. This issue is further discussed in the subsection on scalability.

## Performance

The first question regarding performance is to ask if it is necessary at all. Aside from the obvious rejoinder that no piece of software is better if it is slower, it should be said that real-time simulation allows for a piece of software to be used in a way that closely mimics the way a visually impaired person might use it. Offline simulation based on video might be useful, but will never allow for testing protocols or traditional usability evaluation. In previous work, some efforts were offline, some didn't use software processing at all, relying on optics to simulate disability, and others can be divided into those which used slow intercepts (100ms times have been reported in [Vot09a]) or those which operated quickly, but had limited capabilities, such as solutions based on hardware overlays.

When it comes to performance, only two significant problems present themselves. The first is the problem of text-based selectors. While it is possible to completely avoid its use, this is only feasible through very precise gaze-tracking with a very high sampling frequency—enough to fully capture and subvert saccade movements—which is not always practical, as is discussed in the subsection on scalability. In case text-based selectors are used, this necessitates some way to recognize text using computer vision algorithms. Current algorithms meant for real-time text recognition run in timeframes around 300ms[Neu12a], but is likely that using a simplified method—specifically stopping at stage one classification—some increase in performance could be possible. The goal, of course, is to have sub 30ms times in order to allow for 30fps functioning of the impairment simulator.

The second problem depends on how the simulator gets the image of the interface it plans to deform. There are three approaches: Toolkit level intercept, compositing level intercept, and raster level intercept. Toolkit level intercepts are out of the question because this will fail to answer the question of universal applicability. If the image capture is done by relying on the toolkit used to generate the GUI, then interfaces done using any other type of toolkit are impossible to simulate. Compositing level intercept is better—this attempts to capture DirectX or OpenGL commands a piece of software is sending to the driver, and uses those to capture footage. Normally this is used to record footage of 3D applications and video games. Unfortunately this approach is unlikely to work with normal 2D Windows applications and is not cross-platform at all.

In practice the best two approaches—on Microsoft Windows, which was chosen in order to support as many developers as possible—are DirectX front surface readback and direct read of the screen buffer

using the bitblit Win32 function. Of these two, the latter has shown to be slightly faster and delivers steady 30fps in most cases, though it struggles to go much past that that.

## Universal Applicability

While a simulator would be much easier to construct using laboratory grade equipment, high-end hardware, and precisely controlled circumstances, this rather defeats the purpose of inclusive design. Inclusive design is meant to be universal, as there's no telling which interface element of which software package will be used by a visually impaired person. To allow for this, the simulator must be accessible to everyone, no matter their software or hardware, and it must be such that it does not place any undue burden on the user who should focus on the interface design above all.

It is doubtless true that better hardware allows for better simulations: higher fidelity and higher efficiency leading to better results. However, such hardware is hard to come by and expensive, and accessibility and inclusive design are already a low priority in a lot of commercial software. Adding a hefty price tag does not help inclusive design becoming a universal in UI engineering, rather the opposite. Thus, it is crucial that the simulator be capable of running in situations with little to no specialized hardware, adapting to limits in accuracy as best it can. Naturally, in the presence of suitable hardware it can adapt to utilize those superior resources increasing its efficiency. However, it cannot demand such hardware be present without jeopardizing its goal of propagating inclusive design. This requirement for adapting to changing circumstances is outlined further as a part of scalability.

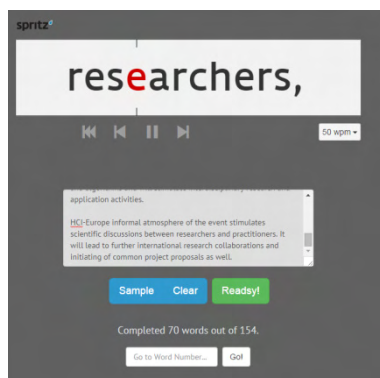
## Scalability

The scalability requirement combines affordability and ease of use—it represents what is required to allow the framework to reach ubiquitous use. The two key goals here are plug-and-play installation and no need for specialized hardware. This latter goal is the most difficult one because veracity demands gaze-tracking in order to differentiate between peripheral and central vision which need to be treated markedly differently even in healthy adults[Per02a]. Since the presence of purpose-built gaze-tracking hardware cannot be relied upon, some alternative solution needs to be found. The two approaches that present themselves are tracking a proxy for the user's gaze or implementing a gaze-tracking solution which uses hardware that *can* be relied upon, such as a webcam.

The proxy used for the user's gaze in the literature is naturally the position of the mouse cursor, however, this poses difficulties which may be impossible to

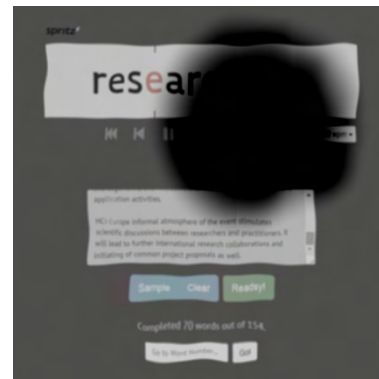
resolve. The idea is that the tester or developer will be instructed to keep his or her eyes focused on the position of the cursor throughout testing thus obviating the need for accurate gaze-tracking. The problem there is the scenario where the user has, say, a simulated foveal scotoma obscuring a vital part of the interface forcing the user to improvise using peripheral vision. Will the user be so disciplined to avoid a few quick—barely liminal—glances with his or her central vision?

While this problem could be studied separately, it is actually possible to get a good idea by consulting a field distant from HCI. Averted vision is a venerable technique of observation in astronomy[Bar77a] and it consists of doing just what is expected of the user in the gaze proxy solution: Keeping visual focus on some other object and using peripheral vision to observe the target. Considering that averted observation was—and is—considered something which requires training and which is easy to do wrong, it can be assumed that using essentially the same approach in visual impairment simulation is equally difficult. Further, off-center focusing is a skill that helps people adapt to scotomata and it has been determined that even people with an accurate simulation of a foveal scotoma can only be trained to avert their gaze correctly after five hours of training[Har14a].



**Figure 1 Original unmodified interface for RVSP**

One alternative is to implement a webcam-based eye-tracking solution. This is difficult: commercial eye-tracking solutions generally use near-IR sources to illuminate the eye which is then recorded using a high-FPS camera. However, the problem is made easier when it is considered that the area of central vision is between 3° and 13° depending on which acuity threshold one wishes to adopt as the ‘edge’ of central vision—features of human being rarely yield to sharp distinctions. The size of 1-5° for a foveal scotoma is attested in the literature. Thus the system need only be accurate enough to capture a region of interest of that size, no smaller, which simplifies matters.



**Figure 2 RVSP interface modified by medium-fidelity prototype of impairment simulation**

While the technology to use webcams to track the user’s gaze does exist[Sew10a] it is not equal to IR-based systems[Bur14a]. Commercially available systems boast accuracy rates of around 1.7°[Sti15a], but suffer issues due to lack of lock and sensitivity to light.

Another possible solution is to use a gaze proxy like cursor position, but to track user distance and to rigorously simulate the visual field and, crucially, the difference in acuity between peripheral and central vision. This ameliorates the problem of quick barely liminal glances outlined above. This is less veracious than the webcam approach, but is maximally scalable, especially since a webcam based solution may cause technical glitches because of jitter and drift, while one using this enforced-proxy approach has no such issues.

#### 4. SOFTWARE PROTOTYPE

The methodology to tackle these problems is to develop a universal software simulator of visual impairment which seeks to better answer the four open questions outlined above.

As a testbed for further development a software prototype was built which simulates all the symptoms of AMD: scotoma, loss of central sight, metamorphopsia, loss of acuity, and loss of contrast and color perception. A complete loss of all vision was not simulated. This helped increase veracity: the ability to simulate acuity and contrast problems helped ‘hide’ the noncentral scotomata: creating an effect which corresponds to what patients report in the literature where the dropping out of parts of the visual field can be imperceptible while still creating problems. Further, the use of simulated metamorphopsia helped illuminate problems with relying on component alignment in UI design.

The prototype used DirectX front surface readback and bitblit-based raster read of the screen buffer, and DirectX 9.0c for the rendering of the changed image. It then used DirectX to apply all the changes to the image, using a SM4 pixel shader to implement all

effects except metamorphopsia which was implemented through render geometry deformation via a vertex shader. All of the effects are parameterized and can be tuned or entirely disabled depending on the severity of the AMD simulated. Eventually, this parameterization will come directly from an impairment model and allow for greater granularity. This approach was optimized until, with the use of bitblt (which proved faster in-implementation than buffer readback) and shader model 4 implementations of symptoms, a fixed framerate of approximately 30fps was achieved even during system load. Stress tests were performed using simulated CPU loads and Unreal Engine 4 authoring tools which served as a stand-in for graphically demanding applications.

No specialized hardware is required for the implementation of this testbed prototype, much in the same way that further improvements will not require any specialized hardware either. A simple development workstation is sufficient to run the software and benefit from its simulation. This makes it universally accessible: any developer can run it on the same machine used to design the interface in the first place and, thus, has little excuse not to do so. Scalability is achieved by adapting to any proxy the user's gaze available. In case of the presence of a gaze-tracking device, all that needs to change is that the symptoms are no longer calculated from the cursor position.

Figure 1 shows the unmodified original interface, and Figure 2 shows the simulator working. The prototype used a user gaze proxy based on the position of the mouse cursor while future versions will also support commercial eye-trackers and webcam eye tracking.

## 5. CONCLUSION

It is both possible and desirable to employ visual impairment simulation in interface design. Previous work in the field shows that there is a need for this sort of software, that such software can be made, and that such software can be made better. Or, rather, can be made in such a way as to incorporate various good features of several approaches in order to minimize wasted effort and allow the designers to easily come to understand the needs of all of their users.

Inclusive design can no longer remain an option, not when the ability to use an UI, whether fitted to a computer, a phone, a television, or a voting machine is a prerequisite for any level of participation in life and the economy. New tools and approaches will have to be developed in order to achieve this, and disability simulation is one step forward.

This paper demonstrated the need for visual impairment simulation, indicated and systematized the questions any framework for such simulation must answer, and offered tools for modeling such

solutions by using visual variables as language for describing alterations caused by impairment. It also provided a medium-fidelity prototype of such a solution.

This research opened up several possible future avenues of research including the full design of a framework partially specified in this paper, and a design for a language for specifying visual impairments. Further, the precise efficacy of webcam-based gaze-tracking will have to be established for this particular application and an approach that's maximally tolerant to changes in lighting conditions, motions of the head, and poor calibration will have to be developed. The presence of a stable light-source and of a trained operator cannot be relied upon if the goal is, as it should be, the universal acceptance of inclusive design as the 'new normal.' Thus, the current state of the art for webcam based eye-tracking is insufficient for the needs of visual impairment simulation. Either the state of the art will have to be improved, or a greater tolerance to problems will have to be built into the simulator solution.

## 6. ACKNOWLEDGMENTS

This work is financially supported by Ministry of Science and Technological Development, Republic of Serbia; under the project number TR32044. "Development of software tools for the analysis and improvement of business processes", 2011-2014.

## 7. REFERENCES

- [Bar77a] Barrett, AA. Notes-Aristotle and Averted Vision. *Journal of the Royal Astronomical Society of Canada* 71, pp.327, 1977.
- [Ber83a] Bertin, Jacques. *Semiology of Graphics: Diagrams, Networks, Maps*, 1983.
- [Bis13a] Biswas, Pradipta, and Pat Langdon. *Inclusive User Modeling and Simulation. A Multimodal End-2-End Approach to Accessible Computing*, pp.71–89, 2013.
- [Bis12a] Biswas, Pradipta, Peter Robinson, and Patrick Langdon. *Designing Inclusive Interfaces Through User Modeling and Simulation. International Journal of Human-Computer Interaction* 28. pp1–33, 2012.
- [Bis08a] Biswas, Pradipta, Tefik Metin Sezgin, and Peter Robinson. 2008. *Perception Model for People with Visual Impairments. Visual Information Systems, Web-Based Visual Information Search and Management*, pp.279–90, 2008.
- [Bur15a] Bureau, U. S. Census. *American FactFinder – Results*, 2015.

- [Bur14a] Burton, Liz, William Albert, and Mark Flynn. A Comparison of the Performance of Webcam vs. Infrared Eye Tracking Technology. Proceedings of the Human Factors and Ergonomics Society Annual Meeting 58 pp.1437–41, 2014.
- [Car03a] Carpendale, MST. Considering Visual Variables as a Basis for Information Visualisation. Computer Science TR# pp.2001-693, 2003.
- [Dej06a] de Jong, Paulus T.V.M. Age-Related Macular Degeneration. New England Journal of Medicine 355 pp.1474–85. doi:10, 2006.
- [Dug13a] Duggan, Maeve, and Joanna Brenner. The Demographics of Social Media Users, 2012. Vol. 14. Pew Research Center's Internet & American Life Project, 2013.
- [Fin99a] Fine, Elisabeth M, and Gary S Rubin. Effects of Cataract and Scotoma on Visual Acuity. Optometry and Vision Science 76, 1999.
- [Flo07a] Flower, Ashley, Matthew K. Burns, and Nicole A. Bottsford-Miller. Meta-Analysis of Disability Simulation Research. Remedial and Special Education 28 pp72–79, 2007.
- [Gar09a] Garlandini, Simone, and Sara Irina Fabrikant. Evaluating the Effectiveness and Efficiency of Visual Variables for Geographic Information Visualization. Spatial Information Theory, pp. 195–211. Springer, 2009.
- [Goo07a] Goodman-Deane, Joy, Patrick M. Langdon, P. John Clarkson, Nicholas HM Caldwell, and Ahmed M. Sarhan. Equipping Designers by Simulating the Effects of Visual and Hearing Impairments. Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility. ACM. pp241–42, 2007.
- [Gro04a] Group, Eye Diseases Prevalence Research, and others. Prevalence of Age-Related Macular Degeneration in the United States. Archives of Ophthalmology 122. pp. 564, 2004.
- [Har14a] Harvey, Hannah, and Robin Walker. Reading with Peripheral Vision: A Comparison of Reading Dynamic Scrolling and Static Text with a Simulated Central Scotoma. Vision Research 98. pp54–60, 2014.
- [Incl5a] Inclusive Design Toolkit Home: <http://www.inclusivedesigntoolkit.com/betterdesign2/>, 2015.
- [Int13a] International Telecommunications Union. World Telecommunication/ICT Indicators Database 17th Edition, 2013.
- [Kak12a] Kaklanis, Nikolaos, Panagiotis Moschonas, Konstantinos Moustakas, and Dimitrios Tzovaras. Virtual User Models for the Elderly and Disabled for Automatic Simulated Accessibility and Ergonomy Evaluation of Designs. Universal Access in the Information Society 12. pp.403–25, 2012.
- [Mar11a] Mares, Julie A, Rick P Voland, Sherie A Sondel, Amy E Millen, Tara LaRowe, Suzen M Moeller, Mike L Klein, et al. Healthy Lifestyles Related to Subsequent Prevalence of Age-Related Macular Degeneration. Healthy Lifestyles and Prevalence of AMD. Archives of Ophthalmology 129. pp.470–80, 2011.
- [Neu12a] Neumann, L., and J. Matas. Real-Time Scene Text Localization and Recognition. 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp.3538–45, 2012.
- [Pel05a] Peli, E., A. Bowers, A. Mandel, K. Higgins, R. Goldstein, and L. Bobrow. Design for Simulator Performance Evaluations of Driving with Vision Impairments and Visual Aids. Transportation Research Record: Journal of the Transportation Research Board 1937. pp.128–35. 2005.
- [Per02a] Perry, Jeffrey S., and Wilson S. Geisler. Gaze-Contingent Real-Time Simulation of Arbitrary Visual Fields, 2002.
- [Pid06a] Pidcoe, P. E. Oculomotor Tracking Strategy in Normal Subjects with and without Simulated Scotoma. Investigative Ophthalmology & Visual Science 47. pp.169–78, 2006.
- [Rio08a] Riordan-Eva, Paul, and John Whitcher. Vaughan & Asbury's General Ophthalmology, 2008.
- [Sco02a] Scott, Ingrid U, William J Feuer, and Julie A Jacko. Impact of Graphical User Interface Screen Features on Computer Task Accuracy and Speed in a Cohort of Patients with Age-Related Macular Degeneration. American Journal of Ophthalmology 134. pp.857–62, 2002.
- [Sew10a] Sewell, Weston, and Oleg Komogortsev. Real-Time Eye Gaze Tracking with an Unmodified Commodity Webcam Employing a Neural Network. CHI '10 Extended Abstracts on Human Factors in Computing Systems. pp.3739–44, 2010.



- [Shn09a] Shneiderman, Ben, Catherine Plaisant, Maxine Cohen, and Steven Jacobs. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. 5th ed. Pearson, 2009.
- [Sti15a] Sticky: <http://www.sticky.ad/>.
- [Sul13a] Sulzmann, Frank, Roland Blach, and Manfred Dangelmaier. An Integration Framework for Motion and Visually Impaired Virtual Humans in Interactive Immersive Environments. *Universal Access in Human-Computer Interaction. Applications and Services for Quality of Life*. pp.107–15, 2013.
- [Uni02a] United Nations Department of Economic and Social Affairs Population Division. *World Population Ageing: 1950-2050*, 2002.
- [Var04a] Varsori, Michael, Angelica Perez-Fornos, Avinoam B. Safran, and Andrew R. Whatham. Development of a Viewing Strategy during Adaptation to an Artificial Central Scotoma. *Vision Research* 44. pp.2691–2705, 2004.
- [Ver15a] VERITAS FP7 IP: <http://veritas-project.eu/index.html>.
- [Vin95a] Vingerling, Johannes R, Ida Dielemans, Albert Hofman, Diederick E Grobbee, Michel Hijmering, Constantijn FL Kramer, and Paulus TVM de Jong. The Prevalence of Age-Related Maculopathy in the Rotterdam Study. *Ophthalmology* 102. pp.205–10, 1995.
- [Vis15a] Visual Impairment Simulator for Microsoft Windows: Visual Impairment Simulator for Microsoft Windows: <http://vis.cita.uiuc.edu/>.
- [Vot09a] Votis, K., T. Oikonomou, P. Korn, D. Tzovaras, and S. Likothanassis. A Visual Impaired Simulator to Achieve Embedded Accessibility Designs. *IEEE International Conference on Intelligent Computing and Intelligent Systems*, 2009. pp.368–72, 2009.
- [Web15a] WebAIM: Low Vision Simulation: <http://webaim.org/simulations/lowvision>.
- [Wil69a] Wilson, Earl D., and Dewaine Alcorn. Disability Simulation and Development of Attitudes toward the Exceptional. *The Journal of Special Education* 3. pp.303–7, 1969.