Modeling Visual Attention for Rule-Based Usability Simulations of Elderly Citizen

Aaron Ruß

DFKI (German Research Center for Artificial Intelligence) GmbH,
Alt-Moabit 91c, 10559 Berlin, Germany
aaron.russ@dfki.de

Abstract. Designing systems for the special interests and needs of older user has become an important subject. However, necessary usability evaluations are time and resource consuming. One way of automation lies in simulating UI use. Since substantial sensory and cognitive age-related effects on the human visual system have been observed, mechanisms of *Visual Attention (VA)* are promising candidates for simulating GUI interactions specific for older users. This article discusses VA mechanisms relevant for simulating age-related effects in GUI interactions. An integration of such mechanisms is discussed on basis of the MeMo workbench, a rule-based approach that uses UI interaction simulations for uncovering usability problems. In the end, simulation of GUI interactions cannot replace human-based usability evaluation, but can provide early feedback for GUI designs, reducing time and resource demands for evaluations. In that, VA provides an instrumental framework for considering age-related effects in simulations of GUI interactions by older users.

Keywords: visual attention, user model, usability simulation, deficit, impairment, rule-based, Monte Carlo simulation.

1 Introduction

In recent years, designing *User Interfaces (UI)* for elderly users has attracted increasing interest, not only within the scientific community but also commercially. The UI design for this user group poses many challenges. Not only are there numerous age-related effects that have to be considered for successful UI design, but the group also shows far greater diversity in their needs and preferences than younger groups. This diversity concerns sensor-motor functions as well as cognitive functions. The group of older users consists of the full spectrum of high functioning users who well into their seventh decade show no or very few signs of cognitive decline, as well as users who very early deteriorate physically and mentally.

Many studies have shown the aging process to negatively affect sensor-, motor- and cognitive functions. For instance, sensor acuity generally decreases as well as strength and accuracy in motor functions. And even when cognitive function can be maintained, its particulars change: for instance, *fluid intelligence* (processing speed, working memory, etc.) generally decreases while *crystalline* increases (knowledge, verbal fluency, etc.).

D. Harris (Ed.): Engin. Psychol. and Cog. Ergonomics, HCII 2011, LNAI 6781, pp. 72–81, 2011.

Age-related effects on *visual perception* are of special interest when considering the usability of *Graphical User Interfaces (GUI)*. Generally, evaluating usability is a time and resource consuming process and numerous approaches for automation exist [1]. While model-based evaluations [2] mitigate the problem of the time and resource consuming process of recruiting and conducting user-based evaluations by *simulations*, they, instead, require considerable effort for constructing appropriate *user, system,* and *task models*. Essential differences between the various approaches concern the level of detail of the simulation and, usually depending on that, the effort necessary for creating the models. Similarly, the "balance" is affected, i.e. if more effort has to be invested e.g. in constructing the user versus the system model.

MeMo [3] is a workbench for semi-automatically conducting usability evaluations by simulations, i.e. model-based UI evaluations. This article describes a model for *Visual Attention (VA)* that is partially implemented in the workbench for simulating mechanisms of visual perception. The workbench uses models to simulate *users* solving a *task* by interacting with an *UI*. Additionally, the workbench supports constructing and configuring the models for the *UI*, the *task*, and the *user*. The goal is to provide *Information Technology (IT)* professionals with a tool for usability simulations. According to the knowledge that IT professionals usually possess, the workbench requires high effort for creating the *system model* and moderate to low effort for creating and configuring the *task* and *User Models (UM)*. Basic configuration of the UM can be achieved by specifying attributes (e.g. *visual acuity*) and requires no expert knowledge in cognitive science.

Tasks are assumed to be governed by an information-exchange pattern between user and system. Within this limitation, the UM is designed as task-independent as possible in order to allow its application in usability simulations of different UIs with as little effort as possible. Successful task completion is specified by *conditions*, which allows simulating different solution paths and errors, also referred to as simulation of *beginners-, novice-, exploratory behavior* or *generative* approach [2].

2 Related Work

Most simulation-based automation tools [1] either require a concrete task solution to be specified (*expert simulation*; e.g. CogTool) or are based on a cognitive architecture (e.g. SOAR, EPIC, ACT-R) [2]. Expert simulations allow investigating efficiency (*"how fast is the task solved"*) and effort (e.g. the *learning* effort for tasks). In difference, the focus of MeMo lies on investigating the efficacy (*"[how well] could the task be solved?"*).

While cognitive architectures enable constructing detailed high-fidelity cognitive models that allow rigorous validation against experimental data, their UMs are also highly task-dependent. With MeMo, lower cognitive fidelity is traded for a more task-independent UM within the domain of UI usability evaluation.

In [4], novice user behavior is modeled by two interacting probabilistic state graphs (Markov processes) as models for the view of the (novice) user on the system and the (expert) designer's view. Simulated task errors can occur for mismatching states. However, this requires modelers to specify explicitly both state models and the corresponding probability matrices. *Image Processing Algorithms (IPA)* are used to model visual perception and impairments [5]. VA is simulated using IPA by comparing features of potential focus areas to the target stimulus, selecting the most similar.

There are several semantic approaches for simulating *information seeking behavior* using some form of attention mechanism. Not strictly a VA mechanism, SNIF-ACT uses *satisficing* [2] for determining how to process *information patches* (web pages), i.e. deciding when to continue with the current *patch*, when to follow a link to a new *patch*, or when to return to a previous *patch* [6]. Chanceaux and colleagues model the reading of text-boxes in a web page using *font-size*, *locality*, and an *inhibition of return* mechanism [7]. Several approaches simulate web-browsing using methods of semantic analysis (e.g. [2, 6, 7]). Some use VA methods for calculating the order in which a webpage is evaluated (e.g. [7-10]). Often heuristics for VA are used, reducing UM complexity and the need for resource consuming computations (e.g. [7, 11]).

3 The Simulation Workbench MeMo

MeMo is a workbench supporting semi-automatic usability evaluations by means of simulations [3]. The next paragraphs describe relevant parts for the UM simulation.

System Model. The system model represents the interaction logic of the UI that is under inspection. Generally, for each software application, a new system model has to be constructed. The basic objects of a system model are *UI elements* (e.g. buttons, text fields) that offer interactions to the simulated user (e.g. left click on a button). The process model for the system follows a *state machine* approach: UI elements form *system states* (nodes), which are connected by *transitions* (edges). The transitions represent interactions that the UM can select in the current UI state (e.g. clicking a button).

UI elements can to be annotated with relevant attribute values (e.g. *font size* for labels, *contrast* against the background). The rule-based simulation draws on these attributes for calculating probability distributions which the UM uses to select the next interaction.

Rule-based Simulation. The simulations follow a Monte Carlo approach by repeatedly simulating the UM solving a task using probability distributions for deciding the UM's next actions: for each iteration, a task definition specifies the *starting state* in the UI model as well as *termination conditions* for successful task completion. Beginning with the starting state, the UM calculates *probability distributions* for selecting an *interaction* that is available in this UI state. A simulation step comprises three phases, *perception, information processing, and interaction execution* wherein *perception* and *processing* may be reiterated several times before an interaction is finally selected. The UM selects and executes interactions based on calculated probability distributions, causing the system model to change states. This selection process continues with new UI states until the task's termination conditions are met or the UM "gives up" – e.g. because of a lack of viable interaction options.

When calculating the distributions, the probabilities are manipulated by *rules*. Rules follow a typical IF-THEN schema of *condition* and *consequence*. Notably, multiple rules are applied, if their conditions are satisfied. This allows modeling a UM by iteratively extending a set of rules until the UM is represented by a large set of relatively simple rules. For example, the current rule set comprises about 600 rules, derived from literature analysis, experiments, and consulting usability experts [3].

The simulation result is a set of task solutions. In difference to a single solution, multiple solutions can also reveal unlikely but interesting solutions. *Interesting* in this context means solutions, that are non-optimal or even unsuccessful. The frequency of specific solutions can be interpreted as indicator for their importance. Analyzing which rules have fired and lead to non-optimal task solutions, can readily provide semantic explanations for UM decisions (e.g. rule with condition *"if button label X has small font size …"*) and in consequence offer critique on how to improve the UI.

User Group Model. The UM represents a *user group* and is exposed in different degrees to the workbench user. The workbench GUI allows direct manipulation of a set of mostly intuitive UM attributes (e.g. *age*, *visual acuity*). A considerable part of the UM is comprised of rules, that are defined using a *XML Schema Definition (XSD).* During simulation, the rules inform probability distributions. For this, the rules draw on UI *features* (UI element attributes that represent their perception by the UM) and UM attributes. Accordingly, different rule sets define different UMs.

Lastly, part of the UM is "hard-wired", implemented as software-modules. For instance, the UM follows the *Model Human Processor (MHP)* approach [2] where each simulation step is comprised of perception, information processing (*cognition*) and interaction execution (*motor*). The ontological commitment to these three phases is implemented in form of software modules.

Task Definition. In task definitions the *conditions* are specified, that determine when a task is successfully completed. Additionally, the *starting state* is specified, i.e. the system state in which the UM starts solving the task. The definition also contains task specifics for the UM, mainly *task knowledge*; the UM employs the specified task knowledge in an information-exchange strategy [3] similar to the *label following* approach [2, 4].

4 Visual Attention for Usability

Visual Attention (VA) is an integral aspect of usability evaluations – be it explicitly or implicitly. In methods considering VA directly, this helps to answer questions about the *If,* the *When,* and *How Easily* users may find task-relevant GUI elements (e.g. [6, 12, 13]). Implicitly, VA plays a role when considering properties that concern visual saliency, as for example *contrast of luminance and color, size, layout, composition, readability,* etc.

In context of automated usability evaluations, considering VA enables the simulation of various related user behaviors for revealing usability problems. For instance, VA allows taking *sequence effects* concerning GUI displays into account: a UM scanning a GUI selects a *sufficiently* fitting GUI element (i.e. matching the task goal), instead of the optimal element that would appear later in the UM's scan path. Such sequence effects can be caused by misperceptions as for example reading errors, or by some misleading (semantic) similarity to the optimal choice. For simulating elderly users, such usage errors become especially interesting, since declining sensory functions may increase perception errors.

4.1 Visual Attention

Currently prevalent, space-based theories of *visual attention (VA)* employ a *spot light* metaphor describing the attention process [2, 14, 15]. Conceptually, at least two important components drive VA: a bottom-up, signal-driven process and a top-down, cognitive process [16]. Their individual impact on VA is highly variable, depending on the visual signal (e.g. if it is a purely random pattern, has some structure, or is even meaningful) as well as the context (e.g. which task the viewer is currently pursuing or if expectations are involved, induced by prior knowledge). Many of the relatively fast bottom-up processes can be approximated as *parallel* working and *pre-attentive* [15].

Saliency maps are a well-known concept in vision models for determining first and successive fixations (i.e. the *scan path*), derived from saliency values computed for the visual scene (e.g. [17]). Biologically inspired, the saliency map architecture is based on *feature integration theory* and has been implemented in several VA models. Commonly, these models analyze *features* of the visual image (e.g. color, direction, movement), resulting in separate feature (or *conspicuity*) maps which then are combined into a conjoint saliency map for the image – some models also explicitly consider top-down influences on saliency (e.g. [14]).

Most models employ a winner-takes-all strategy for determining the point of first fixation, using the most salient region. A scan path is derived by selecting the next most salient areas, where previously fixated areas receive reduced saliency in order to facilitate focusing new regions (*inhibition of return* mechanism, e.g. [7, 18]).

However, the influence of bottom-up processes on visual saliency has been shown to be highly dependent on the task pursued, with predictions most accurate for non-specific viewing of artificially generated displays. For instance, in search tasks, bottom-up saliency can be increasingly overridden by top-down processes or even counteracted [19, 20]. Similarly, meaningful content of an image usually informs top-down influences on saliency. Generally, bottom-up saliency takes more precedence, if the viewer is less familiar with the image content. For instance, [18] describes an experiment, where domain specific images were shown to domain experts and non-experts. Comparing eye movement data with a VA model revealed that non-experts were influenced more by bottom-up saliency, whereas experts focused more on semantically relevant regions. In addition, prior knowledge or task demands can prime saliency of visual features (e.g. search for red objects) [15] as well as determine preference to search by specific strategies or concentrating the search on promising image areas [18].

4.2 Visual Attention and Effects of Ageing

VA is strongly influenced by bottom-up as well as by top-down components. Thus, when considering the effects of ageing on VA, the impact on sensory capacities as well as on cognitive functions are of interest. In this, memory is not only relevant for considering top-down effects on VA, but also for how perceived stimuli are processed. For instance, a known strategy for dealing with the restricted processing capacities is *re-coding* (*chunking*) or *grouping* [21].

Physiologically, visual acuity mainly decreases due to changes to the lens [22]. Additionally, opacity of the lens increases, decreasing the intensity of light passing through and causing reduced contrast perception [23]. Since contrast – i.e. the perception of differences – allows to structure a scene and to distinguish objects, it can be considered the most important bottom-up saliency feature.

Generally, corrected-to-normal eyesight can be maintained well into the sixth decade after which visual acuity declines rapidly [22]. In combination with loss of *contrast sensitivity*, visual acuity is disproportionally exacerbated under conditions of low luminance (and low contrast) [22]. This may have effects on the readability and discrimination of elements in GUI designs.

Visual perception in old age is worse than would be expected from sensory decline alone, which can be explained by exacerbating neuronal and cognitive age-related developments [22-26]. A general explanatory construct for age-related effects is *inhibition control*, i.e. the ability to resist interference. In the visual context, this shows in the form of a decreased resistance for salient distractors [22, 27], i.e. older users may be more easily distracted and misled by visually salient GUI objects that are not task-relevant.

With regard to *memory*, similar developments concerning *speed reduction* and *inhibitory control* (*interference*) have been observed [22]. In general, most *short-term* and *working memory* systems show considerable age-related degradation with the exception of *verbal memory* (i.e. understanding of word meaning) [28]. For GUI design, this means that older users may increasingly face problems when the amount of steps for solving a task increases – especially when combined with the need to memorize information between steps.

Additionally, age-related effects have been identified for different ways of accessing declarative[1] memory (i.e. memory for *facts*): *recollection* exhibits strong age-related effects, i.e. the access of memory that is specific with regard to a certain context. Mostly unaffected by age is memory access by *familiarity*, i.e. the non-specific recall of memory (e.g. general knowledge, such as word meaning, without relating it to some specific occurrence or context). With regard to GUI design, this suggests that older users may have more difficulties to learn the use of GUI elements that function and behave substantially different, depending on context. This may also affect perception and expectations about design and layout regarding *consistency*.

In summary, with increasing age, the sensory capacities for vision are negatively affected – this is exacerbated by cognitive factors. Affected are visual acuity, contrast, and color sensitivity, accompanied by the need for higher light intensities. Cognitive processes are slowed and temporal resolution of perception is decreased [22]. In addition, attention focus is more easily intruded by interfering stimuli, i.e. inhibitory control is compromised. These age-related effects can have a substantial influence on GUI usage and are of special interest when evaluating the usability of GUIs for older users.

[1] Non-declarative memory (e.g. habitualized strategies) exhibits only minor effects [22].

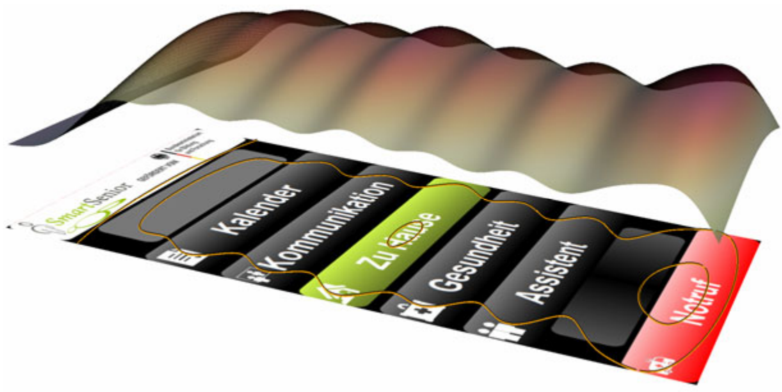

Fig. 1. Bottom-up saliency for GUI elements. The *contour line* renderings implicate two salient areas. The implemented saliency model only calculates "interaction objects" (e.g. buttons), by using Gaussian functions that consider annotated attributes of GUI elements; this includes attributes e.g. for *luminance contrast* (note that *color contrast* is currently not included).

4.3 Visual Attention Model – Current State of Implementation and Next Steps

The VA model is applied in the *perception* phase of a simulation step in the MeMo workbench. In essence, a "spot" in the current GUI state is selected and the corresponding GUI elements made available to the UM's *information processing*. [2]

During this perception-phase a location-based map of the GUI display for bottom-up saliency is calculated (see Fig. 1). Saliency values in this map correspond to (perceivable) UI elements and are calculated by a rule-based approach (see sect. 3), using features of GUI elements as well as user attributes.

For example, during the perception phase, a high contrast of a button increases its saliency; whereas a rule for considering (age-related) visual acuity problems may decrease saliency disproportionally more for UI elements with lower contrast. An advantage of the rule-based approach is that it can provide explanations for the formation of saliency values: after the simulation, an analysis of executed rules can help identify the reasons for the UM decisions and the course of the UM's task solution.

This rule-based approach uses a simplified representation of the GUI (i.e. a *model*) for calculating the visual saliency. This allows simulations even for only roughly sketched GUI drafts in early development stages; but it also requires the construction of a UI model. According to the development stage, the UI model may at first only contain rough layout and type information (e.g. *"button in the upper left corner"*) and in the course of the development process gain more details (e.g. font size of labels, contrast of GUI elements).

---

[2] The size of the "spot" area depends on parameters, common in usability evaluations: *distance* to the display, *resolution* of the display; as parameter for the visual angle (*fovea*), the EPIC default value 2° is used (e.g. 60 cm *distance* and a 20'' display with 1600x1200 px [100 ppi] would approximately result to a 2x2 cm "spot"). Due to *inhibition of return*, a selected area receives a decreasing saliency reduction in following simulation steps.

The next steps for the implementation are to use spatial information to simulate grouping effects of neighboring elements (*chunking*, see sect. 4.2). This also provides the prerequisite for simulating memory concerning location sensitive information[3]. In the *CODE Theory of Visual Attention (CTVA)*, Laplace distributions are used for modeling proximity effects [9]: nearby elements merge and amplify their saliency in order to explain grouping effects (visual chunking). The effort of attentional focus is represented as a *perception threshold* cutting the "height" of Laplace distributions (e.g. *contour lines* in saliency visualizations, see Fig. 2 and Fig. 1): with low thresholds, elements tend to be perceived as a group, whereas high thresholds allow perceiving individual elements. This provides a basic mechanism for *visual grouping* by proximity. With regard to low-detail-level models ("GUI sketches"), proximity may be the only information available for deriving visual groups. With more design information available, the grouping mechanism can also consider more features.

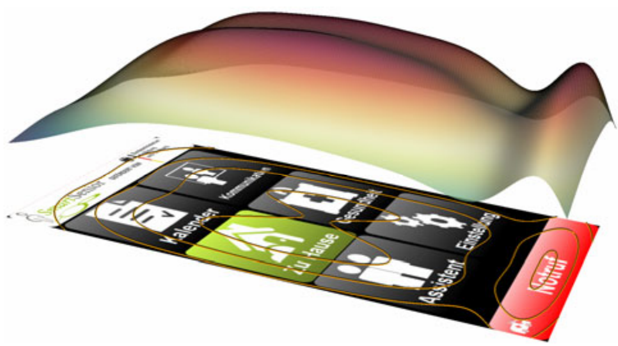

Fig. 2. Alternative layout to Fig. 1. Here, the contour lines suggest differing visual groupings for UI elements in the upper image region, depending on different "perception thresholds".

Furthermore, the grouping information will be exploited for simulating limited cognitive resources: first, the perception threshold is adapted, in order to allow only a limited amount of groups to be inspected at the same time.[4] Then, a selected group is "zoomed in" and the process is repeated as needed: threshold adaption, group selection, zooming in. Depending on "saliency contrast" between GUI elements, different search strategies may be employed, e.g. selecting the most salient (perceivable) group versus systematically scanning groups with similar saliency. This enables simulations of usability problems due to sequence effects (see sect. 4). Additionally, this method allows estimating cognitive workload in terms of attention and memory demands.

[3] E.g. simulating the expectation that a certain GUI element can be found at a specific position or in relation of other "nearby" elements.

[4] The amount of perceivable groups may be influenced by memory limitations (see sect. 4.2).

5 Conclusion

In this article, we considered *Visual Attention (VA)* and its importance for evaluating GUIs. In this regard, *age* has notable effects on VA mechanisms, e.g. loss of sensory acuity, slowing and loss of several cognitive capacities. Accordingly, VA mechanisms provide an expedient framework for incorporating age-related effects in usability simulations.

Using the MeMo workbench, we examined a partially implemented model for bottom-up VA mechanisms, focusing on aspects that are relevant for age-related effects. In this approach, analyzing the formation of rule-based saliency maps can readily provide (semantically relevant) explanations for simulated usability problems.

In conclusion, it is important to note that we are a long way from simulations that can replace human-based usability evaluations. However, they can provide an early and cost effective feedback for UI designs while alleviating the need for extensive usability and cognitive science knowledge on part of the "conductors".

Acknowledgements. Work presented in this article was funded within the SmartSenior project by the German Federal Ministry of Education and Research (BMBF, FKZ 16KT0902); GUI screenshots with courtesy of Ehsan Gholamsaghaee.

References

[1] Ivory, M., Hearst, M.: The state of the art in automating usability evaluation of user interfaces. ACM Comput. Surv. 33, 470–516 (2001)

[2] Sears, A., Jacko, J.A.: The Human-Computer Interaction Handbook, 2nd edn. Lawrence Erlbaum Associates, Mahwah (2007)

[3] Engelbrecht, K., Kruppa, M., Möller, S., Quade, M.: MeMo workbench for semi-automated usability testing. In: Proc. 9th Interspeech, Australia, pp. 1662–1665 (2008)

[4] Biswas, P., Robinson, P.: Automatic evaluation of assistive interfaces. In: Proc. 13th International IUI, pp. 247–256. ACM, New York (2008)

[5] Biswas, P., Robinson, P.: Modelling perception using image processing algorithms. In: Proc. Human-Computer Interaction, pp. 494–503. British Computer Society (2009)

[6] Fu, W., Pirolli, P.: SNIF-ACT: A cognitive model of user navigation on the World Wide Web. Hum. Comput. Interact. 22, 355–412 (2007)

[7] Chanceaux, M., Guérin-Dugué, A., Lemaire, B., Baccino, T.: A Model to Simulate Web Users' Eye Movements. In: Gross, T., Gulliksen, J., Kotzé, P., Oestreicher, L., Palanque, P., Prates, R.O., Winckler, M. (eds.) INTERACT 2009. LNCS, vol. 5726, pp. 288–300. Springer, Heidelberg (2009)

[8] Kitajima, M., Polson, P., Blackmon, M.: CoLiDeS and SNIF-ACT: Complementary models for searching and sensemaking on the Web. In: HCIC Winter Workshop (2007)

[9] Pirolli, P., Card, S., Van Der Wege, M.: Visual information foraging in a focus+ context visualization. In: Proc. SIGCHI, pp. 506–513. ACM, New York (2001)

[10] Stone, B., Dennis, S.: Using LSA Semantic Fields to Predict Eye Movement on Web Pages. In: McNamara, D.S., Trafton, J.G. (eds.) Proc. 29th Cognitive Science Society Conference, pp. 665–670. Lawrence Erlbaum Associates, Mahwah (2007)

[11] Halverson, T., Hornof, A.: A minimal model for predicting visual search in human-computer interaction. In: Proc. SIGCHI, pp. 431–434. ACM, New York (2007)

[12] Faraday, P.: Visually critiquing web pages. In: Proc. 6th Conference on Human Factors & the Web, Texas (2000)

[13] Kitajima, M., Blackmon, M., Polson, P.: A comprehension-based model of Web navigation and its application to Web usability analysis. In: Proc. HCI, pp. 357–373 (2000)

[14] Navalpakkam, V., Itti, L.: Modeling the influence of task on attention. Vis. Res. 45, 205–231 (2005)

[15] Wolfe, J., Horowitz, T.: What attributes guide the deployment of visual attention and how do they do it? Nat. Rev. Neurosci. 5, 495–501 (2004)

[16] Treue, S.: Visual attention: the where, what, how and why of saliency. Curr. Opin. Neurobiol. 13, 428–432 (2003)

[17] Itti, L., Koch, C.: Computational modelling of visual attention. Nat. Rev. Neurosci. 2, 194–203 (2001)

[18] Underwood, G.: Cognitive processes in eye guidance: algorithms for attention in image processing. Cognit. Comput. 1, 64–76 (2009)

[19] Betz, T., Kietzmann, T., Wilming, N., König, P.: Investigating task-dependent top-down effects on overt visual attention. J. Vis. 10ax, 1–14 (2010)

[20] Einhäuser, W., Rutishauser, U., Koch, C.: Task-demands can immediately reverse the effects of sensory-driven saliency in complex visual stimuli. J. Vis. 8, 1–19 (2008)

[21] Miller, G.: The Magical Number Seven, Plus or Minus Two Some Limits on Our Capacity for Processing Information. Psychol. Rev. 63, 81–97 (1956)

[22] Birren, J.E., et al.: Handbook of the Psychology of Aging, 6th edn., vol. 564. Academic Press, Burlington (2006)

[23] Elliott, D., Whitaker, D., MacVeigh, D.: Neural contribution to spatiotemporal contrast sensitivity decline in healthy ageing eyes. Vis. Res. 30, 541–547 (1990)

[24] Higgins, K., Jaffe, M., Caruso, R., Demonasterio, F.: Spatial contrast sensitivity: effects of age, test-retest, and psychophysical method. J. Opt. Soc. Am. 5, 2173–2180 (1988)

[25] Salthouse, T., Hancock, H., Meinz, E., Hambrick, D.: Interrelations of age, visual acuity, and cognitive functioning. J. Gerontol. B Psychol. Sci. Soc. Sci. 51, P317–P330 (1996)

[26] Wright, C., Drasdo, N.: The influence of age on the spatial and temporal contrast sensitivity function. Doc. Ophthalmol. 59, 385–395 (1985)

[27] Munoz, D., Broughton, J., Goldring, J., Armstrong, I.: Age-related performance of human subjects on saccadic eye movement tasks. Exp. Brain. Res. 121, 391–400 (1998)

[28] Park, D.: Ageing and memory: mechanisms underlying age differences in performance. Aust. J. Ageing 17, 69–72 (1998)