Perception Model for people with Visual Impairments

Pradipta Biswas, Tevfik Metin Sezgin, Peter Robinson

Computer Laboratory, 15 JJ Thomson Avenue, Cambridge CB3 0FD, University of Cambridge, United Kingdom
{pb400, mts33, pr} @ cl.cam.ac.uk

Abstract. Scientists from many different disciplines (including physiology, psychology, and engineering) have worked on modelling visual perception. However this field has been less extensively studied in the context of computer science, as most of the existing perception models work only for very specific domains such as menu searching or icon searching tasks. We are developing a perception model that works for any application. It takes a list of mouse events, a sequence of bitmap images of an interface and locations of different objects in the interface as input, and produces a sequence of eye-movements as output. We have identified a set of features to differentiate among different screen objects and using those features, our model has also reproduced the results of previous experiments on visual perception in the context of HCI. It can also simulate the effects of different visual impairments on interaction. In this paper we discuss the design, implementation and two pilot studies to demonstrate the model.

1 Introduction

Usability evaluation is an important step for successful design of any product. However user trials are often expensive and time consuming. Additionally for users with special needs, it is particularly difficult to get a representative population for a user trial. These difficulties with user trials led us to design a simulator that can model human computer interactions for people with a wide range of physical abilities and skills. In this paper we describe a particular component of this simulator - the visual perception model.

Computer Scientists have studied theories of perception extensively for graphics and more recently, for Human-Computer Interaction (HCI). A good interface should contain unambiguous control objects (like buttons, menus, icons etc.) that are easily distinguishable from each other and reduce visual search time. In this work, we have identified a set of features to differentiate among different screen objects and this set of features also reproduces the results of previous experiments on visual perception in the context of HCI. We have developed a prototype of the model. It can also simulate the effects of different visual impairments on interaction. Unlike previous works, our model not only shows how a computer interface is perceived to a visually impaired person, but also it can simulate the dynamics of interactions with a computer.

2 Related Work

How do we see? This question has been addressed in many ways over the years. The Gestalt psychologists in early 19th century pioneered an interpretation of the processing mechanisms for sensory information [8]. Later the Gestalt principles gave birth to the top-down or constructivist theories of visual perception. According to this theory, the processing of sensory information is governed by our existing knowledge and expectations. On the other hand, bottom-up theorists suggest that perception occurs by automatic and direct processing of stimuli [8]. Considering both approaches, recent models of visual perception incorporate both top-down and bottom-up mechanisms [14]. This is also reflected in recent experimental results in neurophysiology [12, 17].

Knowledge about theories of perception has helped researchers to develop computational models of visual perception. Marr's model of perception is the pioneer in this field [14] and most of the other models follow its organization. However it was never implemented in a practical system [18]. In recent years, a plethora of models have been developed (e.g. ACRONYM, PARVO, CAMERA etc. [18]), which have also been implemented in computer systems. The working principles of these models are based on the general framework proposed in the analysis-by-synthesis model of Neisser [14] and mainly consist of the following three steps:

1. **Feature extraction:** As the name suggests, in this step the image is analysed to extract different features such as colour, edge, shape, curvature etc. This step mimics neural processing at the V1 region of brain.
2. **Perceptual grouping:** The extracted features are grouped together mainly based on different heuristics or rules (e.g. the proximity and containment rule in CAMERA system, rules of collinearity, parallelism and terminations in ACRONYM system [18]). In brain, similar type of perceptual grouping occurs in V2 and V3 regions.
3. **Object recognition:** The grouped features are compared to known objects and the closest match is chosen as the output.

In these three steps, the first step models the bottom-up theory of attention while the last two steps are guided by top-down theories. All of these models aim to recognize objects from a background picture and some of them have proved successful at recognizing simple objects (like mechanical instruments). However they have not demonstrated such good performance at recognizing arbitrary objects [18]. These early models do not operate at a detailed neurological level. Itti and Koch [10] present a review of some computational models, which try to explain vision at the neurological level. Itti's pure bottom-up model [10] even worked in some natural environments, but most of these models are used to explain the underlying phenomena of vision (mainly the bottom-up theories) rather than prediction.

In the field of Human Computer Interaction, the EPIC [11] and ACT-R [1] cognitive architectures have been used to develop perception models for menu searching and icon searching tasks. Both the EPIC and ACT-R models [4, 9] are used to explain the results of Nielsen's experiment on searching menu items [15] and found that users search through a menu list both in systematic and random ways. The ACT-R model has also been used to find out the characteristics of a good icon in the context of an icon-searching task [6, 7]. However the cognitive architectures emphasize modeling human cognition, so their perception and motor modules are not as well developed as others. The working principles of the perception models in EPIC and ACT-R/PM are quite simpler from the earlier general-purpose computational models of vision. These models do not use any image processing algorithms. The features of the target objects are manually fed into the system and they are manipulated by handcrafted rules in a rule-based system. As a result, these models do not scale well to general-purpose interaction tasks. Modelling of visual impairment is particularly difficult using these models. An object seems blurred in a continuous scale for different degrees of visual acuity loss and this continuous scale is hard to model using propositional clauses in ACT-R or EPIC. Shah et. al. [20] have proposed the use of image processing algorithms in a cognitive model, but they have not published any results about the predictive power of their model yet.

3. Design

We have developed a perception model as part of a simulator for HCI. The simulator takes a task definition and locations of different objects in an interface as input and then predicts the cursor trace, probable eye movements across the screen and task completion time, for different input device configurations (e.g. mouse or single switch scanning systems) and undertaken by persons with different levels of skill and physical disabilities. The architecture of the simulator is shown in Figure 1. It consists of the following three components:

The Application model represents the task currently undertaken by the user by breaking it up into a set of simple atomic tasks using the KLM model [5].
The Interface Model decides the type of input and output devices to be used by a particular user and sets parameters for an interface.
The User Model simulates the interaction patterns of users for undertaking a task analysed by the task model under the configuration set by the interface model. It uses the sequence of phases defined by the Model Human Processor [5]. The perception model simulates the visual perception of interface objects. The cognitive model determines an action to accomplish the current task. The motor-behaviour model predicts the completion time and possible interaction patterns for performing an action. The details of the simulator and the cognitive and motor-behaviour models can be

found in two separate papers [2, 3]. In the following sections we present the perception model in detail.

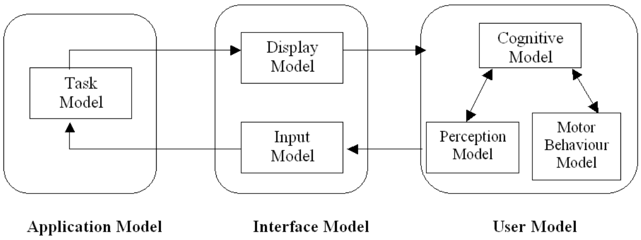


Figure 1. Architecture of the Simulator

Modelling perception

Our perception model takes a list of mouse events, a sequence of bitmap images of an interface and locations of different objects in the interface as input, and produces a sequence of eye-movements as output. The model is controlled by four free parameters: distance of the user from the screen, foeveal angle, parafoveal angle and periphery angle (Figure 2). The default values of these parameters are set according to the EPIC architecture [11]. The model can also be used to simulate the effect of different visual impairments.

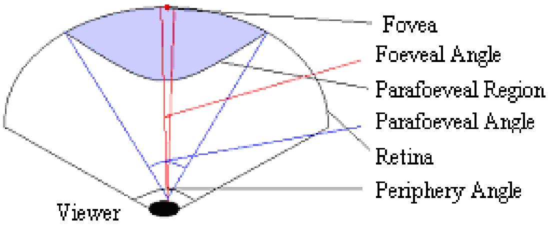


Figure 2. Foveal, parafoveal and peripheral vision

We perceive something on a computer screen by focusing attention at a portion of the screen and then searching for the desired object within that area. If the target object is not found we look at other portions of the screen until the object is found or the whole screen is scanned. Our model simulates this process in three steps (Figure 3).

- Scanning the screen and decompose it into primitive features
- Finding the probable points of attention fixation
- Deducing a trajectory of eye movement

The perception model represents a user's area of attention by defining a focus rectangle within a certain portion of the screen. The area of the focus rectangle is calculated from the distance of the user from the screen and the periphery angle (Figure 2). However it has already been found that we can see objects even which are out of attention (obviously with less accuracy [10]) and so the size of the focus rectangle varies with the number of probable targets in its vicinity. If the focus rectangle contains more than one probable target (whose locations are input to the system) then it shrinks in size to investigate each individual item. Similarly in a sparse area of the screen, the focus rectangle increases in size to reduce the number of attention shifts.

The model scans the whole screen by dividing it into several focus rectangles, one of which should contain the actual target. The probable points of attention fixation are calculated by evaluating the similarity of other focus rectangles to the one containing the target. We know which focus rectangle contains the target from the list of mouse events that was input to the system. The similarity is measured by decomposing each focus rectangle into a set of features (colour, edge, shape etc.) and then comparing the values of these features. The focus rectangles are aligned with respect to the objects within them [do you mean a 2D spatial alignment or a 1D ranking?].

Finally, the model shifts attention by combining three different strategies,
Nearest strategy [6,7]: At each instant, the model shifts attention to the nearest probable point of attention fixation from the current position.
Random Strategy: Attention randomly shifts to any probable point of fixation.
Cluster Strategy: The probable points of attention fixation are clustered according to their position and attention shifts to the cluster centre of one of these clusters.
We choose any one of these strategies probabilistically [probabilistic selection implies a probability distribution. How is the cluster really decided?].

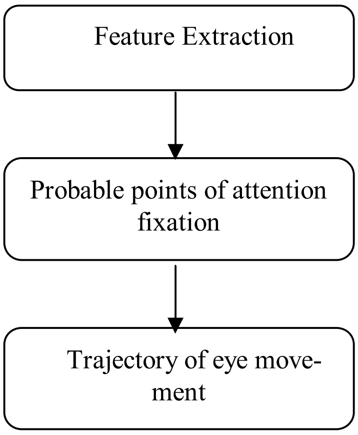

Feature Extraction
Probable points of attention fixation
Trajectory of eye movement

Figure 3. Simulating visual perception

Pilot Studies

Study 1- Comparing performances for colour and shape recognition

In a computer screen, any target can be characterised by two properties – its colour and shape. In this study, we have investigated which of the features is easier to detect for impaired vision. We compared the reaction times people take to recognize a target from distractors of same colour and different shape and vice versa (Figure 4). Prior to each session, the participants were told about the target (e.g. a red circle) and then instructed to point to the target as soon as they could find it. We measured the reaction time between target display and recognition. We used nine types of targets of different colours and shapes. We recruited 10 participants (6 male, 4 female and average 25.4), who did not have any colour-blindness and had no visual impairment that could impede their vision after correction. We simulated visual impairment by using translucent filters from the Inclusive Design Toolkit [22] and considered four conditions (normal vision, mild acuity loss, severe acuity loss and central vision loss). The reaction times are shown in Figures 5. As can be seen from the Figures 5, shape recognition takes more time in general and especially for severe acuity loss and central vision loss. With the filters (simulating vision loss), participants took more time to differentiate between target and distractors of same colour and different shapes than the other case and some of them even reported that they could not detect the corners of the shapes.

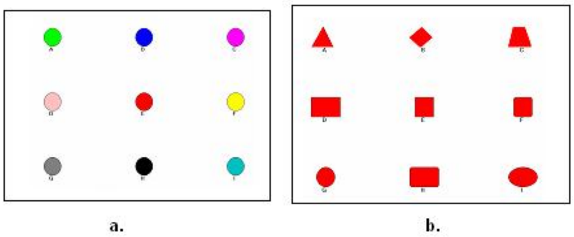


Figure 4. a. Screen to test colour recognition   b. Screen to test shape recognition

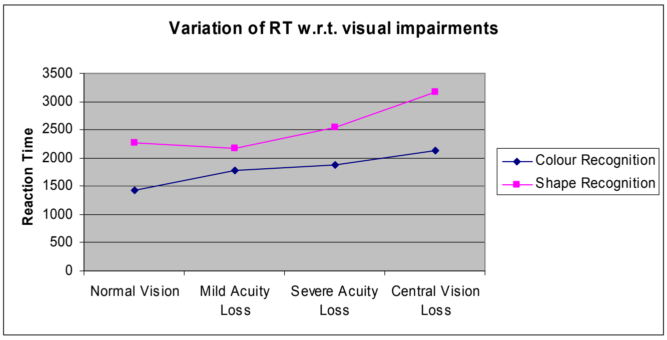


Figure 5. Variations of reaction time (in msec) for different impairments.

Guided by this study, we developed algorithms to simulate the process of colour and shape recognition. We used colour histogram matching algorithm [16] to measure and compare the colours, the Sobel operator [16] for edge detection and the shape context algorithm [21] for shape measurement. We simulated severe acuity loss by a low pass Gaussian filter. We found that the colour histogram matching algorithm works well even for a blurred screen; however the shape context matching algorithm does not. In particular, the edge detection algorithm, which is run before the shape context algorithm, fails to detect edges in a blurred screen. This is also consistent with the result we found in the study: with blurred vision people take more time to detect edges and thus to differentiate shapes from one another. However the colour information is not lost by blurring (as long as the colours contrast with background) and the colour-histogram matching algorithm finds it easier to recognize colour in the same way as the human participants. These results can be extended in future to predict reaction time from the colour histogram and shape context matching coefficients.

Study 2- Defining the best set of features to predict the probable points of fixation
The second study considered the best set of features to predict the probable points of fixation. For the pilot study, we assumed that in a screen, users' attention would fix on icons, which were same as the target icons instead of other types of icons. For example, if the target was a pdf file then attention would mostly be fixed on the pdf icons in the screen. We considered seven different types of icons (Figure 6) and looked for the best classification performance for different feature subsets. We used a backpropagation neural network as classifier. Figure 7 shows the classification performance for 15 different subsets of the Colour in RGB, Colour in YUV, shape and edge features. The error bars show the standard deviation for 30 runs for the best classifier parameters. As can be seen from Figure 7 the best results are obtained for the Colour (YUV), shape and edge features.

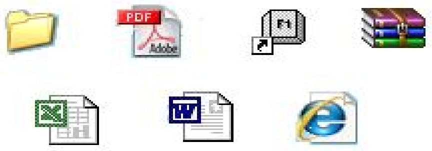


Figure 6. Icons used in pilot study

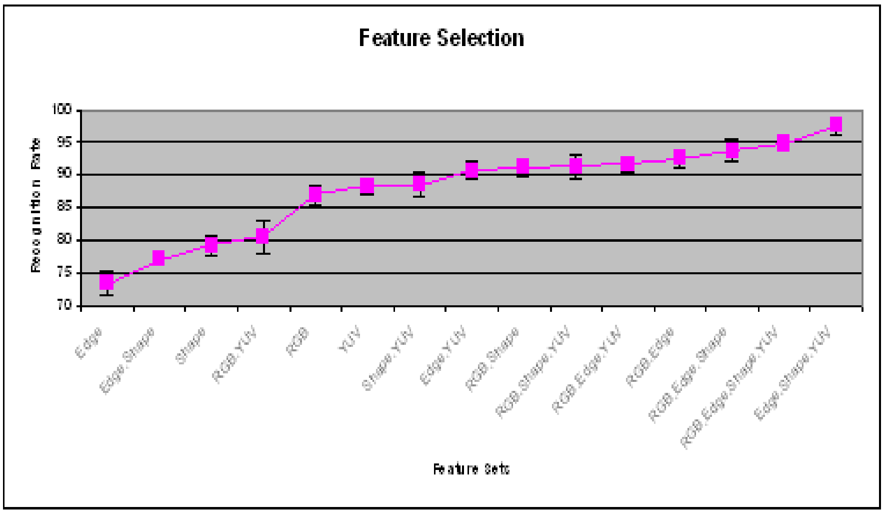
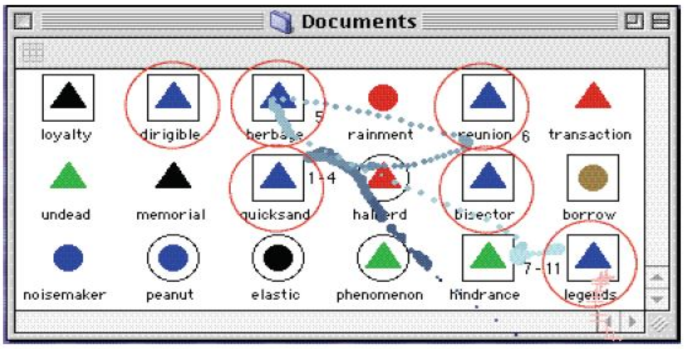


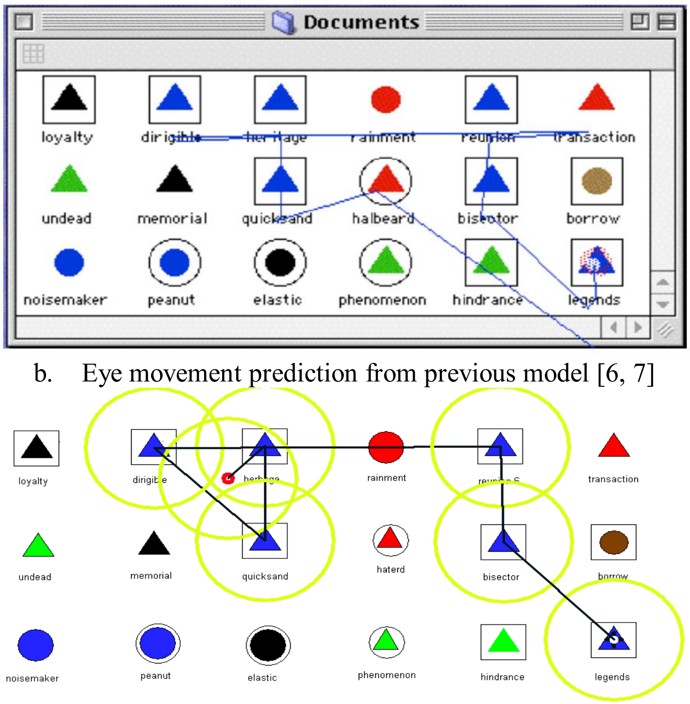
Figure 7. Classifier performance for different feature sets

Validation

We do not yet have eye-tracking data of our own, so we compared the performance of our result to some previous eye-tracking data [6,7]. Figure 8 shows the actual eye-tracking data of a previous experiment (Figure 8a), performance of the previous model (Figure 8b) and the performance of our model (Figure 8c). It can be seen that our model successfully identified all the probable points of fixation.



a. Eye tracking data [from 6, 7]



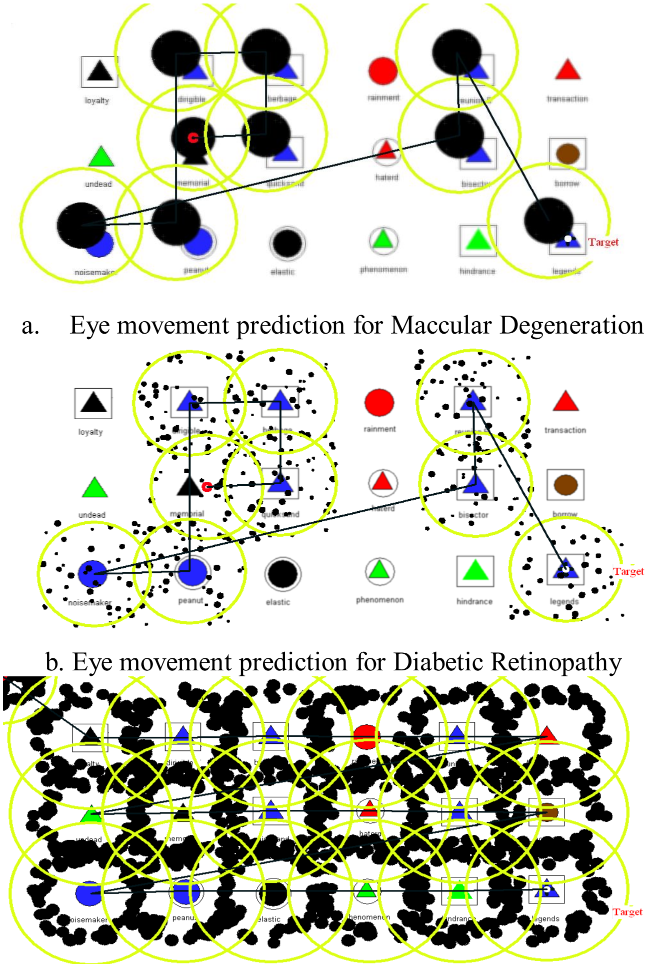
b. Eye movement prediction from previous model [6, 7]

c. Eye movement prediction from our model

Figure 8. Validating the model

Modelling visual impairment

Our model can also simulate the effects of different visual impairments on interaction. To cover a wide range of visual impairments, we had three-level model. The first level simulates different diseases (currently Macular Degeneration, Diabetic Retinopathy, Tunnel vision and Colour-Blindness). The next level simulates the effect of change in different visual functions (e.g. Visual acuity, Contrast sensitivity, Visual field loss etc.). The last level allows different image processing algorithms to be run (e.g. Filtering, Smoothing etc.) on input images to manually simulate the effect of a particular impairment. This approach also makes it easier to model the progress of an impairment. The previous simulations on visual impairments model the progress of impairment by a single parameter [22, 23] or using too much parameter [24] [I couldn't understand what you mean by too much parameter]. In our system, the progress of any impairment can be modelled either by a single parameter or by changing the values of different visual functions. For example, the extent of a particular case of Maccular [Is this with double c or single c?] Degeneration can be modeled either by a single scale or by using different scales for visual acuity and central visual field loss. Additionally, most previous work (like the Visual Simulator Project [23] or the Inclusive Toolkit project [22]) simulate visual impairment on still images for a fixed position of eyes. Unlike this previous work, our model not only shows how a computer interface is perceived to a visually impaired person, but it can also simulate the dynamics of interactions with a computer. Figure 9 shows a few demonstrations of our simulator. In all these figures, the desired target is marked with the text 'Target' written in red. The black line indicates the trajectory of eye movements through a series of intermediate points of attention fixation marked with yellow rings.



a. Eye movement prediction for Maccular Degeneration

b. Eye movement prediction for Diabetic Retinopathy

c. Eye movement prediction for Tunnel Vision

Figure 9. Eye movement prediction for different visual impairments

Figure 9a shows a sequence of eye movements for Maccular Degeneration. As can been seen from the figure, the whole screen becomes blurred since the patient is using peripheral vision and black spots appear in the centre of point of fixation due to central field loss. In case of Diabetic Retinopathy (Figure 9b), some random black spots appear at the region of attention fixation due to blockage of blood vessels inside the eyes. In both of these cases the number of points of fixation is greater than in normal vision (Figure 8) since patients need to investigate all blue targets due to blurring of the screen. For tunnel vision (Figure 9c), the patient cannot use any peripheral vision, so he never sees the screen as a whole and can only see a small portion of it. So all targets need to be examined and eyes have to move systematically from left to right and top to bottom until the target is reached.

Discussion

The first study qualitatively proves the credibility of colour histogram and shape context algorithms to model colour and shape recognition processes for both normal and impaired vision. The second study shows that they can also be used to identify icons besides primitive shapes (with more than 90% accuracy). Table 1 presents a comparative analysis of our model to that of ACT-R/PM and EPIC models. Our model seems to be more accurate, scalable and easier to use than the existing models. However, in real life situations the model also produces some false positives because it fails to take account of the domain knowledge of users. This knowledge can be either application specific or application independent. There is no way to simulate application specific domain knowledge without knowing the application beforehand. However there are certain types of domain knowledge that are application independent and apply to almost all applications. For example, the appearance of a pop-up window immediately shifts attention in real life, however the model still looks for probable targets in other parts of the screen. Similarly, when the target is a text box, users focus attention to the corresponding labels rather than other text boxes, which we do not yet model. There is also scope to model perceptual learning. Currently our neural network (used as a classifier) trains itself after each execution, but there is no way to remember a particular location, which would be used for the same purpose as before [I couldn't quite understand the argument in this sentence]. For that purpose, we could consider some high level features like the caption of a widget, handle of the application etc. to remember the utility of a location for a certain application. These issues did not arise in previous works since they used very specific and simple domains to model [4, 6, 7, 9].

We are still not in a position to assert that our model actually outperforms previous models. Currently we are working on an experiment to track users' gaze while they try to recognize a target from a real life application (besides primitive shapes). We will simulate impairment using filters as our first study. Then we will try to predict the points of attention fixation and eye movements using our model. We are also working to predict the visual search time using the EMMA model [19], which will also help in evaluating the model.

Table 1. Comparative analysis of our model

	ACT-R/PM or EPIC models	Our Model	Advantages of our model
Storing Stimuli	Propositional Clauses	Spatial Array	Easy to use and Scalable
Extracting Features	Manually	Automatically using Image Processing algorithms	Full automation
Matching Features	Rules with binary outcome	Image processing algorithms that give the minimum squared error	More accurate
Modelling top down knowledge	Not relevant as applied to very specific domain.	Considers the type of target (e.g. button, icon, combo box etc.).	More detailed and practical
Shifting Attention	Systematic/ Random and Nearest strategy	Clustering/ Nearest /Random strategy	Not worse than previous, probably more accurate

Conclusions

In this paper we have presented a perception model that can be used to evaluate and compare the visual feedback provided by different computer interfaces. The model is part of a larger system that is used to evaluate interfaces with respect to a wide range of skills and physical abilities [2, 3]. Our perception model takes a list of mouse events, a sequence of bitmap images of an interface and locations of different objects in the interface as input, and produces a sequence of eye-movements as output. The model supports existing theories on visual perception and it can also explain the results of most of the experiments done on visual perception in the field of Human-Computer Interaction. The model can also simulate the effect of different visual impairments on interactions. Unlike previous work, our model not only shows how a computer interface is perceived to a visually impaired person, but also it can simulate the dynamics of interactions with a computer. Currently we are in the process of calibrating the model using an eye-tracker.

References

[1] Anderson, J. R., and Lebiere, C., The Atomic Components of Thought. Hillsdale, NJ: Erlbaum, 1998

[2] Biswas P. and Robinson P., Automatic Evaluation of Assistive Interfaces, In Proc. of the ACM Intl. Conf. on Intelligent User Interfaces (IUI), 247-256, 2008

[3] Biswas P. and Robinson P., Simulation to Predict Performance of Assistive Interfaces, In Proc. of the 9th Intl. ACM SIGACCESS Conf. on Computers and Accessibility (ASSETS '07), 827-828, 2007

[4] Byrne M. D., ACT-R/PM And Menu Selection: Applying A Cognitive Architecture To HCI, International Journal of Human Computer Studies,vol. 55, 2001

[5] Card, S., Moran, T., and Newell, A. The Psychology of Human-Computer Interaction, Lawrence Erlbaum Associates, Hillsdale, NJ, 1983

[6] Fleetwood, M. F. & Byrne, M. D. (2002) Modeling icon search in ACT-R/PM.Cognitive Systems Research, Vol. 3 (1), 25-33

[7] Fleetwood M. F. and Byrne M. D., Modeling the Visual Search of Displays: A Revised ACT-R Model of Icon Search Based on Eye-Tracking Data, Human-Computer Interaction, 2006, Vol. 21, No. 2, 153-197

[8] Hampson P. J. and Moris P. E., Understanding Cognition, Blackwell Publishers Ltd., Oxford, UK, 1996

[9] Hornof, A. J. and Kieras, D. E., Cognitive Modeling Reveals Menu Search Is Both Random And Systematic. In Proc. of the ACM/SIGCHI Conference on Human Factors in Computing Systems, pp. 107-115, 1997

[10] Itti L. and Koch C., Computational Modelling of Visual Attention, Nature Reviews, Neuroscience, Vol. 2, March 2001, 1-10.

[11] Kieras, D. and Meyer, D.E.. An Overview of The EPIC Architecture For Cognition And Performance With Application To Human-Computer Interaction, Human-Computer Interaction, vol. 14,pp. 391-438, 1990

[12] Luck S. J. et. al., Neural Mechanisms of Spatial Selective Attention In Areas V1, V2, And V4 of Macaque Visual Cortex, Journal of Neurophysiology, vol. 77, pp. 24-42, 1997

[13] Marr, D. C., Visual Information Processing: the structure and creation of visual representations. Philosophical Transactions of the Royal Society of London B, 290, 199-218

[14] Neisser, U., Cognition and Reality, San Francisco, Freeman, 1976

[15] Nilsen E. L., Perceptual-motor Control in Human-Computer Interaction (Technical Report No. 37), Ann Arbor, MI: The Cognitive Science and Machine Intelligence Laboratory, the University of Michigan

[16] Nixon M. and Aguado A., Feature Extraction and Image Processing, Elsevier, Oxford, First Edition, 2002

[17] Reynolds J. H. and Desimone R., The Role of Neural Mechanisms of Attention In Solving The Binding Problem, Neuron 24: 19-29, pp.111-145, 1999

[18] Rosandich, R. G., Intelligent Visual Inspection using artificial neural networks, Chapman and Hall, London, First Edition, 1997

[19] Salvucci D. D., An integrated model of eye movements and visual encoding, Cognitive Systems Research, January, 2001

[20] Shah K. et. al., Connecting a Cognitive Model to Dynamic Gaming Environments: Architectural and Image Processing Issues, In Proc. of the 5th Intl. Conf. on Cognitive Modeling,189-194, 2003

[21] Belongie S., Malik J., and Puzicha J., Shape Matching and Object Recognition Using Shape Contexts, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (24): 509-521, 2002

[22] Inclusive Design Toolkit, Available at: http://www-edc.eng.cam.ac.uk/betterdesign/downloads/visualsim.html, Accessed n 27th March, 2008

[23] Vision Simulator, Available at: http://www.my-vision-simulator.com/, Accessed n 27th March, 2008

[24] Visual Impairment Simulator, Available at: http://cita.rehab.uiuc.edu/software/vis/index.php, Accessed n 27th February, 2008