



## Visual Saliency and Reference Resolution in Simulated 3-D Environments

J. KELLEHER<sup>1</sup> & J. VAN GENABITH<sup>2</sup>

<sup>1</sup>*Media Lab Europe, Sugar House Lane, Bellevue, Dublin 8, Ireland (E-mail: john.kelleher@medialabeurope.org)*; <sup>2</sup>*National Center for Language Technology, School of Computing, Dublin City University, Dublin, Ireland (E-mail: josef@computing.dcu.ie)*

**Abstract.** In this paper we present a novel false colouring-based visual saliency algorithm and illustrate how it is used in the situated language interpreter (SLI) system to ground a reference resolution framework for natural language interfaces to 3-D simulated environments. The visual saliency algorithm allows us to dynamically maintain a model of the evolving visual context. The visual saliency scores associated with the elements in the context model can be used to resolve underspecified references.

**Keywords:** natural language interfaces, reference resolution, virtual environments, visual saliency

**Abbreviations:** SLI – situated language interpreter

### 1. Introduction

Many modern computer applications share a visualised virtual space with the user: graphics design programs, computer games, navigation aids etc.<sup>1</sup> Such applications are often ideal candidates for natural language interfaces that allow users to refer to and manipulate objects in the shared application domain. In these applications the human user interacts with the system using *situated language*. Situated language is spoken from a particular point of view within a physical or simulated context (Byron, 2003). The goal of the situated language interpreter (SLI)<sup>2</sup> project is to develop a natural language interpretive framework for natural language virtual reality (NLVR) systems. An NLVR system is a computer system that allows a user to interact with simulated 3-D environments through a natural language interface.

The interpretation of referring expressions against a changing context is one of the most important tasks in NLVR systems. Referring expressions come in a variety of surface forms including: definite descriptions, indefinites, pronouns, demonstratives. Each referring

expression introduces a representation into the semantics of its utterance and this representation must be bound to an element in the context in order for the utterance's interpretation to be fully resolved. From a computational perspective reference resolution involves two main tasks:

1. Creating and maintaining a model of the evolving discourse context (DC) (this model should contain representations of all the objects that are available for reference and their properties).
2. Matching the representation introduced by a given referring expression to an element (or elements) in the set of possible referents provided by the DC.

'The DC has traditionally been thought of as a discourse history, and most computational processes accumulate items into this set only using linguistic events as input' (Byron, 2003, p. 3). However, for visually grounded discourse a purely linguistically driven DC model is not adequate. Psycholinguistic studies (Spivey-Knowlton et al., 1998) have demonstrated that the interpretation of language in a shared visual domain is dependent on the visual context:

Given these results, approaches to language comprehension that assign a central role to encapsulating linguistic subsystems are unlikely to prove fruitful. More promising are theories in which grammatical constraints are integrated into processing systems that coordinate linguistic and non-linguistic information as the linguistic input is processed. (Spivey-Knowlton et al., 1998, pp. 211–212).

Furthermore, in NLVR contexts it has been found that:

Users expect the system to have full perceptual knowledge of any graphical elements produced by it ... [consequently] a visual history, analogous to the discourse history, must be accumulated (Byron, 2003, p. 6).

Following these results, we argue that the ability of NLVR systems to interpret referring expressions is greatly improved if they maintain a DC that models the evolving visual context as well as the linguistic context. In order to model the flow of information to the user from the visual context, we have developed and implemented a visual saliency algorithm that works in real-time and across different and changing simulated environments. Unlike previous NLVR systems (Winograd, 1973; Andre et al., 1988; Herzog, 1997; Smith et al., 1997; Fuhr et al., 1998; Klipple and Gurney, 1999; Goldwater et al., 2000; Kelleher et al., 2000; Kievit

et al., 2001; Jording and Wachsmuth, 2002) visual salience is a crucial component in reference resolution in the SLI system. This paper describes this algorithm and illustrates how it is used to resolve references.

Section 2 looks at the core results regarding the distributional properties of passive perception acuity and active attention. Section 3 reviews previous computational work including connectionist, ray-casting and false colouring-based approaches to modelling vision. Section 4 presents the SLI false colouring-based visual saliency algorithm and contrasts it with previous false colouring-based approaches (Noser et al., 1995; Kuffner and Latombe, 1999) and (Peter and O'Sullivan, 2002). Section 5 shows how the SLI system uses visual salience to resolve ambiguous or underspecified references. Section 6 concludes.

## 2. Perception and Attention

Although visual perception seems effortless, at any given moment, the visual environment presents far more information than can be processed. To cope with this potential overload the brain is equipped with a set of attentional mechanisms that regulate the processing of visual stimuli by selecting regions within the visual buffer for detailed processing. Landragin et al. (2001) lists some of these mechanisms: visual familiarity, intentionality, an object's physical characteristics, and the structure of the scene. At the most basic level, these mechanisms can be categorised as being active or passive selectors.

The eye acts as a passive selector: high-resolution information about the retinal image is preserved only at the centre of gaze. The fovea is a shallow pit in the retina which is located directly opposite the pupil, consisting of cones and is the site of highest visual acuity, the ability to recognise detail. Visual acuity 'drops 50% when an object is located only 1° from the centre of the fovea and an additional 35% when it is 8° from the centre' (Forgus and Melamed, 1976, p. 228).

However, even with the filtering of information achieved through passive attentional processes there is still far more information in the visual field than can be processed by the brain (Chum and Wolfe, 2001). Perceivers are active seekers and processors of information. Posner et al. described attention as a 'spotlight that enhances the efficiency of the detection of events within its beam' (1980, p. 172).

Although the spotlight metaphor is useful for describing how active attention is deployed across space, it has some drawbacks. For one, it

implies an even distribution of attention at every point within the area the spotlight falls upon when in fact, similar to visual acuity, 'the spatial distribution of attention follows a gradient with decreased effects of attention with increased eccentricity from its focus' (Chum and Wolfe, 2001, p. 276). Attention is greatest at a single point in the visual buffer and drops off gradually from that point.

### 3. Previous Computational Work

Section 2 examined some of the aspects of perception that pertain to modelling vision, in particular how visual attention affects awareness of what is perceived and how the amount of attention paid to a particular location in the visual buffer is dependent on the distance between that location and the focus of attention.

Many computational models of vision attention have been developed, see Koch and Itti (2001) and Heinke and Humphreys (2004) for recent reviews. However, most of these models are not suitable for NLVR systems as they have a connectionist or neural net architecture and consequently require training. As a result, these models are restricted to the domains described by or sufficiently similar to the training set given to the system. For example, connectionist navigational systems trained with images from the inside of a factory would need to be retrained to handle a forest environment. A system that requires retraining when shifting from one visual domain to another is not suitable as a model of rendered environments which may change drastically from program to program or even within the one application.

Alternative models of visual perception use 3-D graphics techniques. These models can be classified based on the graphics techniques they use: ray casting and false colouring. Tu and Terzopoulos (1994a, b) implemented a realistic virtual marine world inhabited by autonomous artificial fish. The model used a graphics technique called ray casting to determine if an object met the visibility conditions. Ray casting can be functionally described as drawing an invisible line from one point in a 3-D simulation in a certain direction, and then reporting back all the 3-D object meshes this line intersected and the coordinates of these intersections. It is widely used in offline rendering of graphics; however, it is computationally expensive and for this reason is not used in real-time rendering.

Another graphics-based approach to modelling vision was proposed in (Noser et al., 1995). This model was used as a navigation system for animated characters. The vision module consists of scanning the image that results from a modified version of the world fed into the system's graphics engine. Briefly, each object in the world is assigned a unique colour or 'vision-id' (Noser et al., 1995, p. 149). This colour differs from the normal colours used to render the object in the world; hence the term false colouring. An object's false colour is only used when rendering the object in the visibility image off-screen, and does not affect the renderings of the object seen by the user, which may be multi-coloured and fully textured. At specified time intervals, a model of the character's view of the world using the false colours is rendered. Once this rendering is finished, the viewport<sup>3</sup> is copied into a 2-D array along with the *z*-buffer<sup>4</sup> values. By scanning the array and extracting the pixel colour information, a list of the objects currently visible to the animated character can be obtained. Kuffner and Latombe (1999) proposed a navigation behavioural system that used false colouring synthetic vision. Peter and O'Sullivan (2004) also used a false-colouring approach to modelling vision; however they integrated their vision model as part of a goal driven memory and attention model which directed the gaze of autonomous virtual humans.

#### 4. The SLI Visual Saliency Algorithm

The basic assumption underpinning the SLI visual saliency algorithm is that an object's prominence in a scene is dependent on both its centrality within the scene and its size. The algorithm is based on the false colouring approach introduced in Section 3. Each object is assigned a unique ID. In the current implementation, the ID number given to an object is simply 1 + the number of elements in the world when the object is created. A colour table is initialised to represent a one-to-one mapping between object IDs and colours.<sup>5</sup> Each frame is rendered twice: firstly using the objects' normal colours, textures and shading. This is the version that the user sees. The second rendering is off-screen. This rendering uses the unique false colours and flat shading. The size of the second rendering does not need to match the first. Indeed, scaling the image down increases the speed of the algorithm as it reduces the number of pixels that are scanned. In the SLI system the false colour rendering is 200 × 50 pixels, a size that yields sufficient detail (by comparison, the fully coloured, textured and shaded on-screen rendering is

$400 \times 300$  pixels). After each frame is rendered, a bitmap image of the false colour rendering is created. The bitmap image is then scanned and the visual salience information extracted. Figure 1 illustrates the normal rendering of a sample scene from the SLI system and Figure 2 illustrates the  $200 \times 150$  false colour rendering of the same scene.

To model the size and centrality of the objects in the scene, the SLI system assigns a weighting to each pixel using equation (1). In this equation,  $P$  equals the distance between the pixel being weighted and the centre of the image, and  $M$  equals the maximum distance between the centre of the image and the point on the border of the image furthest



Figure 1. A normal rendering of a scene in the SLI domain.

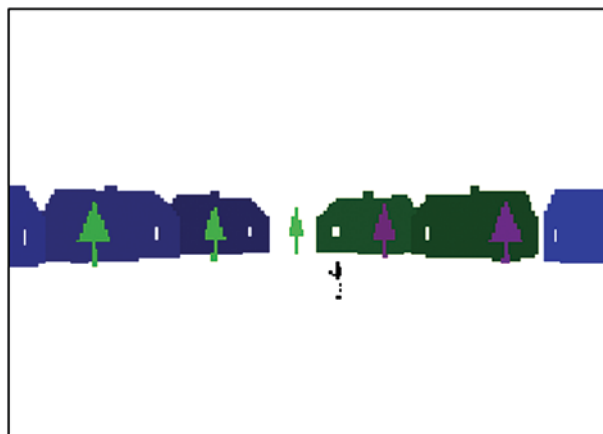


Figure 2. The false colour rendering of the scene in Figure 1.

from the centre; i.e., in a rectangular or square image,  $M$  is equal to the distance between the centre of the image and one of the corners of the image.

$$\text{Weighting} = 1 - \left( \frac{P}{M + 1} \right) \quad (1)$$

This equation normalises the pixel weightings between 0 and 1. Figure 3 illustrates the distribution of pixel weightings assigned using equation (1). It is evident that the closer a pixel is to the centre of the image the higher its salience.

After weighting the pixels, the SLI system scans the image and, for each object in the scene, sums the weightings of all pixels that are coloured using an object's unique colour ID. Finally, the summed pixel weighting for each object in the scene is normalised between 0 and 1 by dividing it by the maximum summed pixel weight ascribed to an object in the scene. This normalised value is the relative visual saliency of the object in the scene. Figure 4 gives the SLI visual saliency algorithm.

Everything else being equal, this algorithm ascribes larger objects a higher saliency than smaller objects since they cover more pixels and objects which are more central to the view will be rated higher than objects at the periphery of the scene as the pixels the former cover will have a higher weighting. The algorithm results in a list of the currently visible objects, each with an associated saliency rating. Figure 5 illustrates the relationship between the false colour rendering of a scene and the weightings ascribed to the pixels in the viewport.

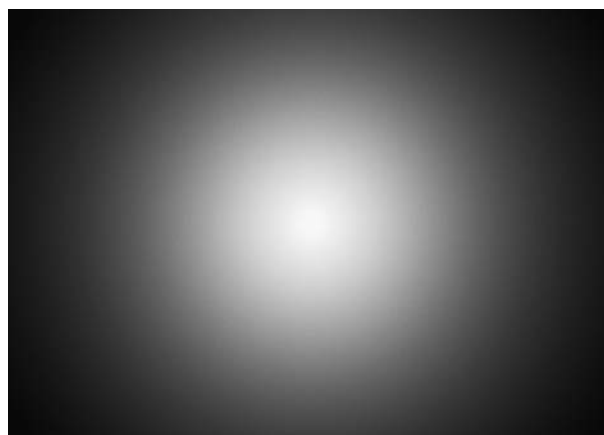


Figure 3. The weighting assigned to the pixels in the viewport using Equation (1). The darker the pixels the lower the weighting. Weightings range from 0 to 1.

**Input:** A bitmap of a false colour rendering of a scene and a table (colourtable) listing the one-to-one mapping of false colours and object ID's.

**Output:** A list of the objects rendered in the input scene each with an associated value representing its relative visual saliency in the scene.

1. **Let** objectlist = 2D array of length colourtable
2. **Let** MAX = 0
3. Weight each pixel using Equation 1.
4. **foreach**  $p \in$  Pixels
  - a) **Let** objectID = getObjectID(colourtable,  $p$ .RGB)
  - b) objectlist[objectID][0] = objectlist[objectID][0] +  $p$ .weighting
  - c) **if** (objectlist[objectID][0] > MAX) **then** MAX = objectlist[objectID][0]
5. **foreach**  $e \in$  objectlist
  - a) objectlist[e][0] = objectlist[e][0] / MAX
6. **return** objectlist

Figure 4. The SLI visual saliency algorithm.

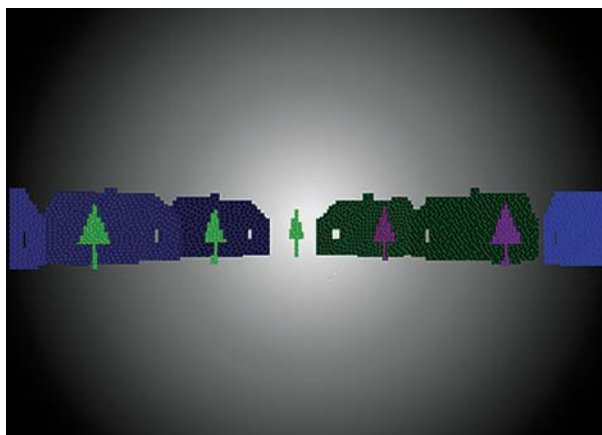


Figure 5. An overlay of the false colour rendering of Figure 1 on the distribution of pixel weightings.



It is important to note that the scanning process in the SLI visual salience algorithm differs from those in the previous false colour based synthetic vision models (Noser et al., 1995; Kuffner and Latombe, 1999; Peter and O’Sullivan, 2002). The previous false colouring algorithms simply recorded whether the object had been rendered or not. The SLI algorithm records whether an object has been rendered and ascribes each object a relative prominence within the scene. It is this difference that allows the SLI system to rank the objects based on their visual salience. We do not claim that this algorithm accommodates all the perceptual factors that impact on visual salience (cf. the list identified by Landragin et al. (2001)). However, it defines a reasonable model of visual saliency that operates fast enough for real-time systems with changing environments. Furthermore, by using a false colouring algorithm to model visual salience our algorithm naturally accounts for the effects of partial object occlusion.

An implicit assumption underpinning the pixel weighting distribution used by the algorithm is that the user’s attention is focused on the centre of the image. In the SLI system, a command such as *look at the green house* has the effect of updating the viewport such that the referent of *the green house* occupies centre position in the updated viewport. An alternative, more sophisticated approach is to leverage the tight coupling between gaze and visual attentional focus using eye tracking technology to compute the location of the user’s gaze at each scene rendering. Using such eye tracking information the distribution of the pixel weightings can be modified to reflect the user’s gaze position as the maximum of the salience distribution by setting  $M$  equal to the maximum distance between the coordinates of the user’s gaze and the edge of the viewport and measuring  $P$  as the distance between the pixel being weighted and the coordinates of the user’s gaze.

In the SLI system, we have integrated the information created by the visual salience algorithm with a model of user input discourse. Using this information the SLI system is able to define a local context for the interpretation of a given exophoric<sup>6</sup> reference. When a reference is made to an object in the visual environment the system is able to restrict the set of objects it considers as candidate referents to those that are currently in the view frustrum or that the user has seen. A further advantage of this approach is that the visual salience scores associated with the objects in the context model allows the system to adjudicate between candidate referents when resolving ambiguous references. In Section 5 we will discuss this application of the visual saliency algorithm in more detail.

## 5. Using Visual Saliency to Resolve Ambiguous References

Since Russell (1905), there has been a debate concerning the singularity constraint associated with definite descriptions. The constraint requires that for felicitous use of a definite description there should be one, and only one, candidate referent in the context of the utterance. An ambiguous or underdetermined reference is a reference that breaks the singularity constraint; i.e., there is more than one candidate referent. However, it has been shown in psycholinguistic experiments that subjects can easily resolve ambiguous or underdetermined references (Duwe and Strohner, 1997). ‘In order to identify the intended referent under these circumstances, subjects rely on perceptual saliency as well as on pragmatic assumptions about the speaker’s communicative goals’ (Duwe and Strohner, 1997, p. 6).

An advantage of using a visual saliency model as an input to an NLVR system’s context model is that the visual saliency scores associated with the objects in the context model allows the system, in many instances, to adjudicate between candidate referents when resolving underspecified or linguistically ambiguous references, as illustrated below. Given Figure 6 as the visual context, the referring expression *the house* in *make the house wider*, is an example of an ambiguous use of a definite description. This is because there is more than one object in the context that fulfills the linguistic description of the expression’s referent.

However, in this instance the SLI system can utilise the visual saliency score associated with each of the candidates as a probability of the

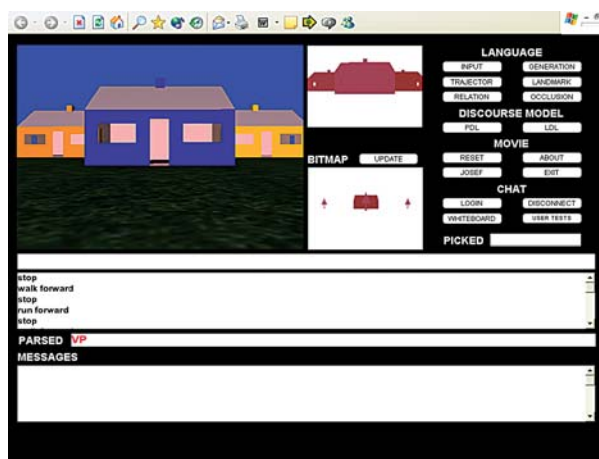


Figure 6. A scene containing three houses.

candidate being the referent for the expression. In this case, the SLI system ascribes the house in the foreground a normalised visual saliency of 1.0000 and each of the houses in the background a normalised visual saliency of 0.0117. Based on these visual saliency scores, the system decides that the user is referring to the house in the foreground and updates the simulation accordingly. Figure 7 illustrates the state of the system after this user input has been interpreted.

Clearly, however, not all ambiguous references can be resolved based on visual saliency. In some instances, the difference in the visual saliency scores associated with each of the candidate referents is not sufficient to allow the selection of a referent. Accordingly, as part of the interpretation process for resolving ambiguous references, the SLI system compares the saliency of the primary candidate referent and the other candidates. If the saliency difference does not exceed a predefined confidence interval, the system outputs a message to the user explaining that it is unable to resolve the reference. In SLI scenarios, it is found that when comparing normalised saliency scores, ranging from 0 to 1, a confidence interval of 0.4 works well. This of course can be adjusted to model a more or less stringent interpretation. Figure 8 illustrates a scene with two houses that have equal visual saliency scores. In this instance, both houses have a visual saliency rating of 1.0000 indicating that they are the two most salient objects in the scene.

Taking Figure 8 as the visual context, if the user input involves an ambiguous referring expression, such as *make the house taller*, the sys-

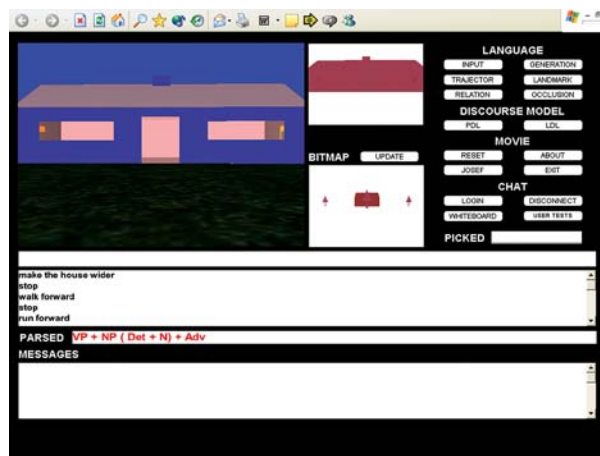


Figure 7. The state of the simulation after the SLI system has interpreted the under-determined reference *the house* and processed the input *make the house wider*.

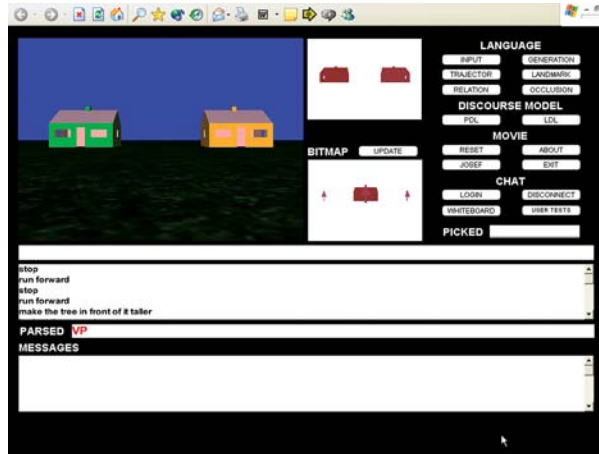


Figure 8. A scene with two houses that have equal visual saliency scores.

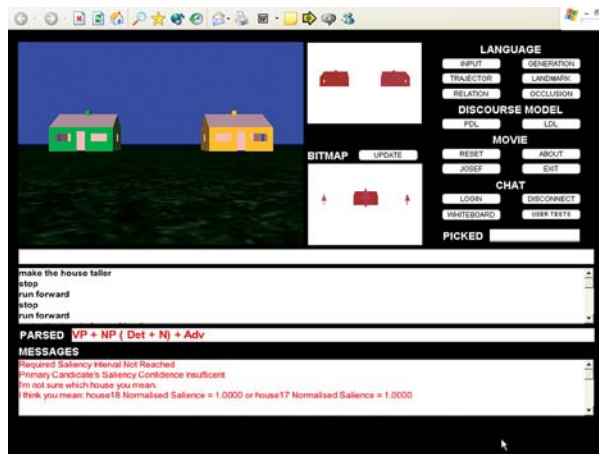


Figure 9. The state of the SLI system after the system has output a message to the user stating that the saliency differences between the candidate referents of an underdetermined expression did not permit the system to resolve the reference.

tem is unable to resolve the reference. Figure 9 illustrates the state of the system after this command has been interpreted.

Note that in Figure 9 the visual scene has not changed and the message text box contains a message to the user explaining why the system was unable to resolve the reference, as well as a listing of the candidate referents the system restricted its search to: *Required Saliency*

*Interval Not Reached, Primary Candidate's Saliency Confidence Insufficient, I'm not sure which house you mean, I think you mean: house 18 Normalised Saliency = 1.0000 or house 17 Normalised Saliency = 1.0000.*

## 6. Conclusions

In this paper, a novel algorithm for modelling the visual saliency of objects in the view volume was developed. This model of visual attention is a new application and extension of a synthetic model of vision that uses a graphics technique called false colouring (Noser et al., 1995). Unlike alternative connectionist approaches, it does not require training. In the SLI project, the function of this visual attention model is to capture the perceptual information flowing from the visual simulation to the user. For a real-time NLVR system, the advantages of using this visual saliency algorithm are: (1) the algorithm is a core component that allows the system to dynamically generate and update an evolving, local, visual, interpretive discourse context for referring expressions; (2) in many cases, the visual saliency scores associated with the objects in the context model allows the system to adjudicate between candidate referents when resolving underspecified or ambiguous references.

## Notes

<sup>1</sup> For an introduction to the early systems integrating language and vision see Maybury and Wahlster (1998) and McKeivitt (1995/1996).

<sup>2</sup> For more information on the SLI project see <http://www.mle.ie/~kelleherj>.

<sup>3</sup> A viewport is the rectangular area of the display window. It can be conceptualised as a window onto the 3-D simulation.

<sup>4</sup> The *z*-buffer stores for each pixel in the viewport the depth value of the object rendered at that pixel.

<sup>5</sup> In the current implementation the colour table contains 256 entries. Although this restricts the number of objects that can be added to the world, this restriction is more a matter of convenience than necessity as the colour table can be extended without affecting the rest of the system.

<sup>6</sup> An exophoric reference denotes an entity in the spatio-temporal surroundings that is new to the discourse.

## References

- Andre, E., Herzog, G. & Rist, T. (1988). On the Simultaneous Interpretation of Real World Image Sequences and their Natural Language Description: The System

- SOCCER. In *Proceedings of the 8th European Conference on Artificial Intelligence (ECAI-88)*, pp. 449–454, Pitmann.
- Byron, D. (2003). Understanding Referring Expressions in Situated Language: Some Challenges for Real-World Agents. In *Proceedings of the First International Workshop on Language Understanding and Agents for the Real World*. Hokkaido University.
- Chum, M. & Wolfe, J. (2001). Visual Attention. In Goldstein, E. B. (ed.) *Blackwell Handbook of Perception*, Handbooks of Experimental Psychology, 272–310. Blackwell (Chapter 9).
- Duwe, I. & Strohner, H. (1997). Towards a Cognitive Model of Linguistic Reference. Report: 97/1 – Situierete Kunstlicher Kommunikatoren 97/1, Universtat Bielefeld.
- Forgus, R. & Melamed, L. (1976). *Perception A Cognitive Stage Approach*. McGraw-Hill.
- Fuhr, T., Socher, G., Scheering, C. & Sagerer, G. (1998). A Three-Dimensional Spatial Model for the Interpretation of Image Data. In Olivier, P. & Gapp, K. (eds.) *Representation and Processing of Spatial Expressions*, 103–118. Lawrence Erlbaum Associates.
- Goldwater, S. J., Bratt, E., Gawron, J. & Dowding, J. (2000). Building a Robust Dialogue System with Limited Data. In *Proceedings of the Workshop on Conversational Systems at the First Meeting of the North American Chapter of the Association of Computational Linguistics*. Seattle, WA.
- Heinke, D. & Humphreys, G. (2004). Computational Models of Visual Selective Attention: A Review. In Houghton, G. (ed.) *Connectionist Models in Psychology*. Psychology Press.
- Herzog, G. (1997). Connecting Vision and Natural Language Systems. Technical Report SFB 314 Project VITRA, Universitt des Saarlandes.
- Jording, T. & Wachsmuth, I. (2002). An Anthropomorphic Agent for the Use of Spatial Language. In Coventry, K. & Olivier, P. (eds.) *Spatial Language: Cognitive and Computational Aspects*, 69–86 Dordrecht: Kluwer Academic Publishers.
- Kelleher, J., Doris, T., Hussain, Q. & ONullain, S. (2000). SONAS: Multimodal, Multi-user Interaction with a Modelled Environment. In Nuallin, S. (ed.) *Spatial Cognition – Foundation and Applications*, Advances in Consciousness Research, 171–18 Amsterdam/Philadelphia: John Benjamins Publishing.
- Kievit, L., Piwek, P., Beun, R. & Bunt, H. (2001). Multimodal Cooperative Resolution of Referential Expressions in the DenK System. In Bunt, H. & Beun, R. (eds.) *Cooperative Multimodal Communication*, Lecture notes in Artificial Intelligence, Vol. 2155, 197–214. Berlin Heidelberg: Springer-Verlag.
- Klippel, E. & Gurney, J. (1999). Deixis to Properties in the NLVR System. In Andre, E. Massimo, P. & Rieser, H. (eds.) *Proceedings of the Workshop on Deixis, Demonstration and Deictic Belief held on occasion of ESSLI XI*, 58–68. Utrecht, The Netherlands.
- Koch, C. & Itti, L. (2001). Computational Modelling of Visual Attention. *Nature Reviews Neuroscience* 2(3): 194–203.
- Kuffner, J. & Latombe, J. (1999). Fast synthetic vision, memory, and learning models for virtual humans. In *Proceedings of Computer Animation Conference (CA-99)*, 118–127, Geneva, Switzerland, IEEE Computer Society.
- Landragin, F., Bellalem, N. & Romary, L. (2001). Visual Saliency and Perceptual Grouping in Multimodal Interactivity. In *Proceeding of the International Workshop*

- on Information Presentation and Natural Multimodal Dialogue (IPNMD)*. Verona, Italy.
- Maybury, M. & Wahlster, W. (eds.) (1998). *Readings in Intelligent User Interfaces*. San Francisco, CA: Morgan Kaufman Publishers, Inc.
- McKevitt, P. (ed.) (1995/1996). *Integration of Natural Language and Vision Processing* (Vols. I–IV). Dordrecht, The Netherlands: Kluwer Academic Publishers.
- Noser, H., Renault, O., Thalmann, D. & Magnenat-Thalmann, N. (1995). Navigation for Digital Actors based on Synthetic Vision, Memory and Learning. *Computer Graphics* **19**(1): 7–9.
- Peter, C. & O’Sullivan, C. (2002). A Memory Model for Autonomous Virtual Humans. In *Proceedings of Eurographics Irish Chapter Workshop (EGIreland-02)*, 21–26. Dublin.
- Posner, M. I., Snyder, C. R. & Davidson, B. J. (1980). Attention and the Detection of Signals. *Journal of Experimental Psychology: General* **109**(2): 160–174.
- Russell, B. 1905. On Denoting. *Mind* **14**: 479–493. Reprinted *Logic and Knowledge* (1956), pp. 39–56, R.C. Marsh ed.
- Smith, A., Farley, B. & O’Nuallain, S. (1997). Visualization of Natural Language. In Dybjaer, L. (ed.) *Third Spoken Dialogue and Discourse Workshop: Topics in Natural Interactive Systems 1*, 80–86. Odense University.
- Spivey-Knowlton, M., Tanenhaus, M., Eberhard, K. & Sedivy, J. (1998). Integration of Visuospatial and Linguistic Information: Language Comprehension in Real Time and Real Space. In Olivier, P. & Gapp, K. (eds.) *Representation and Processing of Spatial Expressions*, 201–214, Lawrence Erlbaum Associates.
- Tu, X. & Terzopoulos, D. (1994a). Artificial Fishes: Physics, Locomotion, Perception, Behaviour. In *Proceedings of ACM SIGGRAPH*, 43–50, Orlando, FL.
- Tu, X. & Terzopoulos, D. (1994b). Perceptual Modelling for Behavioural Animation of Fishes. In *Proceedings of the Second Pacific Conference on Computer Graphics and Applications*, 185–200. Beijing, China.
- Winograd, T. (1973). A Procedural Model of Language Understanding. In Schank, R. & Colby, K. (eds.) *Computer Models of Thought and Language*, 152–186. W.H. Freeman and Company.