A False Colouring Real Time Visual Saliency Algorithm for Reference Resolution in Simulated 3-D Environments

John Kelleher[1] and **Josef van Genabith**[2]

Abstract. In this paper we present a novel false colouring visual saliency algorithm and illustrate how it is used in the Situated Language Interpreter system to resolve natural language references.

1 Introduction

The focus of the Situated Language Interpreter (SLI)[3] project is to develop a natural language interpretive framework to underpin the development of 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 central tenet of this work is that the interpretation of natural language (NL) in 3-D simulated environments should be based on a model of the user's knowledge of the environment. In the context of an NLVR system, one of the user's primary information sources is the visual context supplied by the 3-D simulation. 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 simulated environments. Unlike previous NLVR systems [19, 1, 6, 4, 5, 8] salience in particular visual salience is a crucial component in reference resolutions in the SLI system. This paper describes this algorithm and illustrates how it is used to resolve references

2 Perception and Attention

Although visual perception seems effortless, "psychophysical experiments show that the brain is severely limited in the amount of visual information it can process at any moment in time" [15, pg. 1]. In effect, there is more information perceived than can be processed. The human faculty of attention is the "selective aspect of processing" [9, pg. 84]. Attention regulates the processing of perceived visual stimuli by selecting a region within the visual buffer for detailed processing. Our knowledge of the human attention process is not complete, "but it appears to consist of a set of mechanisms that exhibit different, sometimes opposing effects" [7, pg. 9]. For example, [11] lists: visual familiarity, intentionality, an object's physical characteristics, and the structure of the scene. This multiplicity makes the modelling of visual perception difficult. A priori, one of the major functions of visual attention is object identification. With this in mind, an important factor when considering modelling visual attention is the difference between foveal and peripheral vision. 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. It "drops 50 percent when an object is located only 1° from the centre of the fovea and an additional 35 percent when it is 8° from the centre" [3, pg. 228]. Identifying an object requires the use of foveal vision, occurring when a person looks directly at the object, causing the image of the object falling on the retina to be centred on the fovea. The dependence of object identification on foveal vision implies a relationship between foveal vision and attention. Moreover, this gradation across visual acuity is congruent with the gradation of attention theory. This theory posits that "attention is greatest at a single point, and drops of gradually from that point" [9, pg. 90]. Following this, the more central a location is with respect to the centre of an eye fixation the higher the location's salience. Indeed, the most common computational mechanism for modelling visual attention is a filtering of visual data by removing portions of the input located outside a spatial focus of attention [7].

3 Previous Computational Work

Section 2 examined some of the aspects of perception that pertain to modelling vision, in particular how attention affects the human awareness of what people perceive. It was noted that spatial attention is the most commonly used visual filtering mechanism. There are many computational models of vision that use this abstraction; most have been developed for robot navigation.

[7] reviews several of the robotic attention systems. However, there are two reasons why the models of vision created for robotic systems are not suitable for NLVR systems. First, nearly all of these systems have a connectionist or neural net architecture. This form of system requires 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. Second, the major difficulties facing robotic vision (pattern recognition, distance detection, and the binding problem [14]) do not impact on NLVR systems because the visual scene is already modeled.

There have been several models of visual perception developed that use 3-D graphics techniques. These models can be classified based on the graphics techniques they use: ray casting and false colouring. [17, 18] 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

[1] Media Lab Europe, email: john.kelleher@medialabeurope.org
[2] School of Computing, Dublin City University
[3] See: http://www.medialabeurope.org/ kelleherj

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 [12]. This model was used as a navigation system for animated characters. The vision module was comprised of a modified version of the world being fed into the system's graphics engine and scanning the resulting image. In brief, each object in the world is assigned a unique colour or "vision-id" [12, pg. 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. Then, at a specified time interval, a model of the character's view of the world, using the false colours, is rendered. Once this rendering is finished, the viewport[4] is copied into a 2-D array along with the z-buffer[5] values. By scanning the array and extracting the pixel colour information, a list of the objects currently visible to the actor can be obtained. [12] used this vision model as part of a navigation system for animated characters. Another navigation behavioral system that used false colouring synthetic vision was proposed by [10]. [13] 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. Currently, in the implementation this 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. Each frame is rendered twice: firstly using the objects' normal colours, textures and normal shading. This is the version that the user sees. The second rendering is off-screen. This rendering uses the unique false colours for each object 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 x 150 pixels, a size that yields sufficient detail. 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.

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 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.

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

This equation normalises the pixel weightings between 0 and 1. The closer a pixel is to the centre of the image, the higher its weighting. 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 that object's unique colour. 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. This simple algorithm results in a list of the currently visible objects, each with an associated saliency rating.

It is important to note that the scanning process in the SLI visual salience algorithm differs from those in the previous false colour synthetic vision models [12, 10, 13]. 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. However, it does define a reasonable model of visual saliency that operates fast enough for real-time systems.

In the SLI system, we have integrated the information created by this 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 deictic reference; i.e., 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 the 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 some ambiguous references. In Section 5 we will discuss this application of the visual saliency algorithm in more detail.

5 Using Visual Salience to Resolve Ambiguous References

Since Russell [16], there has been a debate concerning the singularity constraint associated with definite descriptions. The singularity constraint is: given the use of a definite description there should be one, and only one, candidate referent in the context of the utterance. An ambiguous or undetermined reference is a reference that breaks the singularity constraint; i.e., there is more than one candidate referent. It has been shown, however, in psycholinguistic experiments that people can easily resolve ambiguous or underdetermined references [2]. "In order to identify the intended referent under these circumstances, subjects rely on perceptual salience as well as on pragmatic assumptions about the speaker's communicative goals" [2, pg. 6].

An advantage of using a visual saliency model as an input to an NLVR system's context model is that the visual salience scores associated with the objects in the context model allows the system, in some instances, to adjudicate between candidate referents when resolving underspecified or linguistically ambiguous references, as illustrated below. Given Figure 1 as the visual context, the referring expression *the house* in *make the house wider*, is an example of an

[4] A viewport is the rectangular area of the display window. It can be conceptualised as a window onto the 3-D simulation.

[5] The z-buffer stores for each pixel in the viewport the depth value of the object rendered at that pixel

ambiguous visible situation 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.

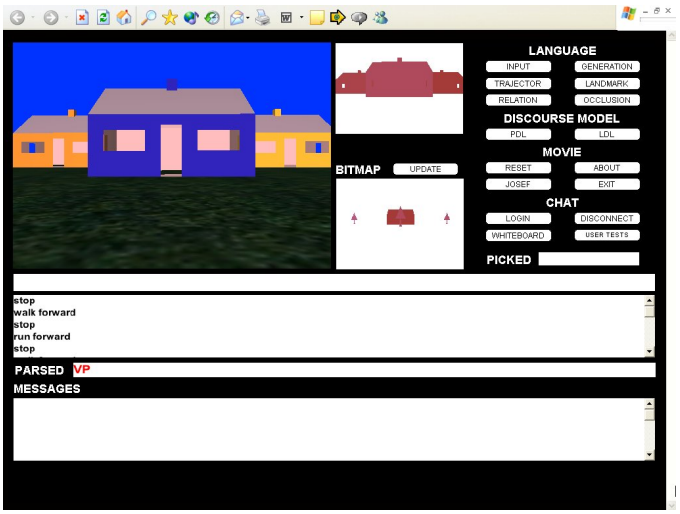

Figure 1. A scene containing three houses.

However, in this instance the SLI system can utilise the visual saliency scores associated with each of the candidates as a probability of the candidate being the referent for the expression. In this case, the SLI system ascribes the blue house in the foreground a normalised computed visual salience of 1.0000 and each of the houses in the background a normalised visual salience of 0.0117. Based on these visual saliency scores, the system decides that the user is referring to the blue house in the foreground and updates the simulation accordingly. Figure 2 illustrates the state of the system after this user input has been interpreted.

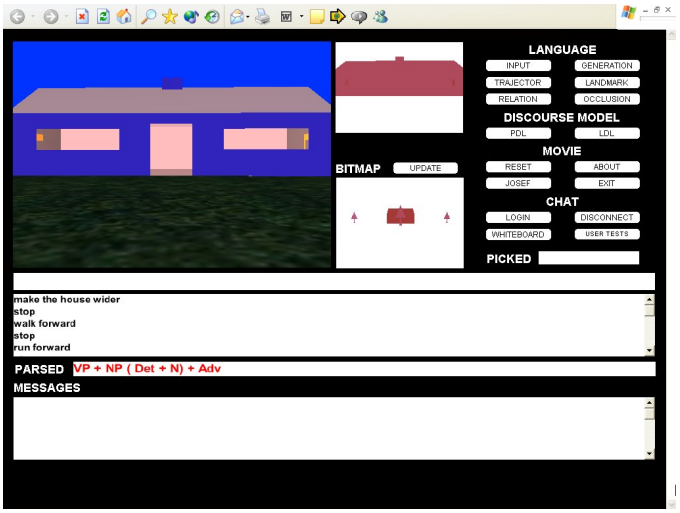

Figure 2. The state of the simulation after the SLI system has interpreted the underdetermined reference *the house* and processed the input *make the house wider*.

Clearly, however, not all ambiguous references can be resolved based on visual salience. 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 .4 works well. This of course can be adjusted to model a more or less stringent interpretation. Figure 3 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.

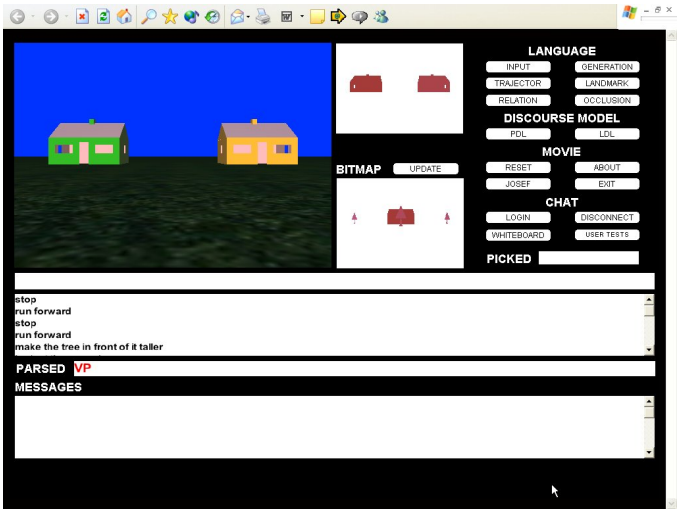

Figure 3. A scene with two houses that have equal visual saliency scores.

Taking Figure 3 as the visual context, if the user inputs an ambiguous referring expression, make the house taller, the system would be unable to resolve the reference. Figure 4 illustrates the state of the system after this command has been interpreted.

Note that in Figure 4 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 listing the candidate referents the system restricted its search to: *Required Saliency Interval Not Reached, Primary Candidates Saliency Confidence Insufficient, I'm not sure which house you mean, I think you mean: house 18 Normalised Salience = 1.0000 or house 17 Normalised Salience = 1.0000.*

6 Conclusions

In this paper, a computational algorithm for modelling the visual salience of objects in the view volume was developed. This model of visual attention is a novel application and extension of a synthetic model of vision that uses a graphics technique called false colouring [12]. In the SLI project, the function of this visual attention model is to try to capture the perceptual information flowing from the visual simulation to the user. For an NLVR system, the advantages of using this visual salience algorithm are that the information created by the algorithm can be used to define a local interpretive context for a given referring expression and the visual salience scores associated with the objects in the context model allows the system, in some

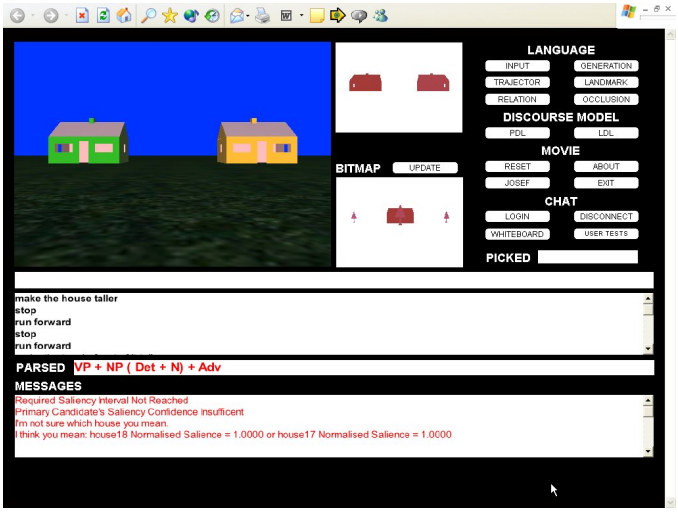

Figure 4. The state of the SLI system after the system has output a message to the user stating that the saliency differences between the candidate referents of an undetermined expression did not permit the system to resolve the reference.

instances, to adjudicate between candidate referents when resolving underspecified or linguistically ambiguous references.

REFERENCES

[1] E. Andre, G. Herzog, and T. Rist, 'On the simultaneous interpretation of real world image sequences and their natural language description: The system soccer', in *In Proceedings of the 8th European Conference on Artificial Intelligence (ECAI-88)*, pp. 449–454. Pitmann, (1988).

[2] I. Duwe and H. Strohner, 'Towards a cognitive model of linguistic reference', Report: 97/1 - Situierte Kunstlicher Kommunikatoren 97/1, Univeristat Bielefeld, (1997).

[3] R.H. Forgus and L.E. Melamed, *Perception A Cognitive Stage Approach*, McGraw-Hill, 1976.

[4] T. Fuhr, G. Socher, C. Scheering, and G. Sagerer, 'A three-dimensional spatial model for the interpretation of image data', in *Representation and Processing of Spatial Expressions*, eds., P. Olivier and K.P. Gapp, 103–118, Lawrence Erlbaum Associates, (1998).

[5] S. J. Goldwater, E.O. Bratt, J.M. Gawron, and J Dowding, '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, (2000).

[6] G Herzog, 'Connecting vision and natural language systems', Technical Report SFB 314 Project VITRA, Universitt des Saarlandes, (1997).

[7] M.S. Hewitt, *Computational Perceptual Attention*, Ph.D. dissertation, University of Texas, Texas, 2001.

[8] T. Jording and I. Wachsmuth, 'An anthropomorphic agent for the use of spatial language', in *Spatial Language: Cognitive and Computational Aspects*, eds., K.R. Coventry and P. Olivier, 69–86, Kluwer Academic Publishers, Dordrecht, (2002).

[9] S.M. Kosslyn, *Image and Brain*, The MIT Press, 1994.

[10] J. Kuffner and J.C. Latombe, 'Fast synthetic vision, memory, and learning models for virtual humans.', in *Proceedings of Computer Animation Conference (CA-99)*, pp. 118–127, Geneva, Switzerland, (1999). IEEE Computer Society.

[11] F Landragin, N Bellalem, and L Romary, 'Visual salience and perceptual grouping in multimodal interactivity', in *Proceeding of the International Workshop on Information Presentation and Natural Multimodal Dialogue (IPNMD)*, Verona, Italy, (2001).

[12] H. Noser, O. Renault, D. Thalmann, and N Magnenat-Thalmann, 'Navigation for digital actors based on synthetic vision, memory and learning', *Computer Graphics*, **19**(1), 7–9, (1995).

[13] C. Peter and C. O'Sullivan, 'A memory model for autonomous virtual humans', in *Proceedings of Europraphics Irish Chapter Workshop (EGIreland-02)*, pp. 21–26, Dublin, (2002).

[14] O Renault, N Magnenat-Thalmann, and D Thalmann, 'A vision-based approach to behavioural animation', *Visualization and Computer Animation*, **1**(1), 18–21, (1990).

[15] J. Reynolds, 'Visual salience, competition, neuronal responce synchrony and selective attention', in *Sloan/Swartz Centers for Theoretical Neurobiology Annual Summer meeting*. The Swartz Foundation, (2001).

[16] B. Russell, 'On denoting', *Mind*, **14**, 479–493, (1905). Reprinted Logix and Knowledge (1956), pp. 39-56, R.C. Marsh ed.

[17] X Tu and D. Terzopoulos, 'Artificial fishes: Physics, locomotion, perception, behaviour', in *Proceedings of ACM SIGGRAPH*, pp. 43–50, Orlando, FL, (1994).

[18] X. Tu and D. Terzopoulos, 'Perceptual modelling for behavioural animation of fishes', in *Proceedings of the Second Pacific Conference on Computer Graphics and Applications*, pp. 185–200, Beijing, China, (1994).

[19] T. Winograd, 'A procedural model of language understanding', in *Computer Models of Thought and Language*, eds., R.C. Schank and K.M. Colby, 152–186, W. H. Freeman and Company, (1973).