

Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

SCIENCE @ DIRECT®

Artificial Intelligence 167 (2005) 62–102

**Artificial  
Intelligence**[www.elsevier.com/locate/artint](http://www.elsevier.com/locate/artint)

## Dynamically structuring, updating and interrelating representations of visual and linguistic discourse context <sup>☆</sup>

J. Kelleher <sup>a,\*</sup>, F. Costello <sup>b</sup>, J. van Genabith <sup>c</sup><sup>a</sup> *Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), Saarbrücken, Germany*<sup>b</sup> *University College Dublin, Ireland*<sup>c</sup> *Dublin City University, Ireland*

Received 22 July 2004; received in revised form 21 February 2005; accepted 14 April 2005

Available online 19 August 2005

---

### Abstract

The fundamental claim of this paper is that salience—both visual and linguistic—is an important overarching semantic category structuring visually situated discourse. Based on this we argue that computer systems attempting to model the evolving context of a visually situated discourse should integrate models of visual and linguistic salience within their natural language processing (NLP) framework. The paper highlights the importance of dynamically updating and interrelating visual and linguistic discourse context representations. To support our approach, we have developed a real-time, natural language virtual reality (NLVR) system (called LIVE, for Linguistic Interaction with Virtual Environments) that implements an NLP framework based on both visual and linguistic salience. Within this framework saliency information underpins two of the core subtasks of NLP: reference resolution and the generation of referring expressions. We describe the theoretical basis and architecture of the LIVE NLP framework and present extensive evaluation results comparing the system's performance with that of human participants in a number of experiments.

© 2005 Elsevier B.V. All rights reserved.

---

<sup>☆</sup> The authors would like to thank Hans Kamp, Laurent Romary and our three anonymous reviewers for valuable comments and feedback.

\* Corresponding author.

*E-mail address:* [kelleher@dfki.de](mailto:kelleher@dfki.de) (J. Kelleher).

<sup>1</sup> This work was partly supported by the EU FP6 IST Cognitive Systems Integrated project Cognitive Systems for Cognitive Assistants (CoSy) FP6-004250- IP.

*Keywords:* Visual salience; Reference resolution; Generating referring expressions; Discourse context; Cross-modal representations; Synthetic vision

---

## 1. Introduction

Many current natural language processing (NLP) systems consider language in a vacuum. These systems manipulate words and other linguistic constructs, but they do not have access to the non-linguistic objects to which these words refer. However, psycholinguistic experiments have shown that perceptual information, in particular visual context, effects how humans process language [1–3]. Motivated by these results we have adopted an approach to NLP that seeks to integrate language processing with other forms of information by exploiting the relationship between natural language semantics and visual context.

Modern computer applications (graphic design programs, computer games, navigation aids, etc.) often share a visualised virtual space with the user. A natural language virtual reality (NLVR) system is a computer system that allows a user to interact with these virtual environments using natural language (NL). In these applications the user interacts with the system using *situated language*. Situated language is uttered from a particular point of view within a physical or simulated context [4]. From theoretical linguistic and cognitive perspectives NLVR systems are interesting as they provide ideal testbeds for investigating the interaction between language and vision. From a human-computer interaction (HCI) viewpoint, NLVR systems promise many advantages to the user interacting with these virtual environments.

In this paper we describe an NLP framework for NLVR systems that grounds linguistic discourse in visual perception, modelled in terms of visual salience. To support our approach, we have developed a real-time, natural language virtual reality (NLVR) system (called LIVE, for Linguistic Interaction with Virtual Environments) that implements the framework. Fig. 1 is an example of the virtual environment provided by the LIVE system. In this environment, the system uses visual and linguistic contextual information to understand user commands and to generate linguistic descriptions of the visual environment.



Fig. 1. A scene from the LIVE system's domain.

The natural language understanding (NLU) component of the LIVE system can interpret commands to navigate through the simulation (e.g., *walk/run forward, turn left/right*), commands that cause changes in the user's view of the simulation (e.g., *look at the blue house, look at the tree to the right of it*) and commands that cause changes in the simulation (e.g., *make the red house bigger, make the yellow one smaller, move this + [gesture] back*). A central focus of the LIVE project is on the interpretation of *referring expressions*.<sup>2</sup> A referring expression is a natural language expression that denotes an entity called the *referent*. The interpretation of referring expressions against a changing context is one of the most important tasks for an NLVR system. Referring expressions come in a variety of surface forms including: definite descriptions, indefinites, pronouns, demonstratives. Each type of referring expression encodes different signals about the status the speaker believes the referent occupies within the hearer's set of beliefs. For example, the use of a pronominal reference signals that the speaker assumes that the intended referent is easily identifiable in the hearer's current mental model of the discourse context because it has a high degree of *salience*.

The natural language generation (NLG) component of the LIVE system can generate scene descriptions. NLG is the process of constructing natural language outputs from non-linguistic input. At an abstract level, it can be viewed as the inverse of NLU: NLU maps from text to meaning while NLG maps from meaning to text. However, notwithstanding the fact that both processes visit many of the same linguistic issues, there are marked differences between their internal operations and concerns. NLU is primarily concerned with ambiguity, underspecification, and ill-formed input. For NLG, however, these issues do not arise to the same extent, as the input to an NLG system tends to be relatively unambiguous, well-specified, and well-formed; instead, the dominant concern in NLG is choice. [6] list some of the choices NLG systems must make: *content selection*, the system must choose the appropriate content to express from a potentially over-specified input; *lexical selection*, the system must choose the lexical items most appropriate for expressing particular concepts (this choice deals with the issue of surface realisation); *sentence structure*, the system must appon the selected content into phrases, clauses, and sentence-size chunks; *discourse structure*, NLG systems frequently deal with multi-sentence discourse, which must have a coherent, discernible structure.

The core focus of the generation component of the LIVE system is the generation of referring expressions. The main novelty of the LIVE generation algorithm is the integration of visual salience with one of the standard reference generation algorithms: Dale and Reiter's Incremental Algorithm [7].

Both the interpretation and generation of referring expressions presuppose a discourse context. The term *mutual knowledge* is often used to describe the set of things that are taken as shared knowledge by the participants in a discourse [8, p. 355]. Following Grice's [9] cooperative principle, a cooperative speaker will only use a referring expression to denote an object they assume the hearer has knowledge of; i.e., an object that they take to be in the mutual knowledge set. The set of objects in the mutual knowledge set accumulates

---

<sup>2</sup> We will restrict our discussion to reference to entities, although discourses include reference to many other types of referents; e.g., propositions, events, etc. [5].

over the course of a discourse. 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 semantics to be fully resolved. From a computational perspective reference resolution involves two main tasks:

- (1) creating and maintaining a model of what the interlocutors consider as mutual knowledge (this model should contain 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.

Symmetric processes are necessary for reference generation:

- (1) constructing a context set containing a single target object, for which a description is to be generated, and a set of distractor objects, containing the relevant objects within the set of all objects that are available for reference from which the target object is to be distinguished,
- (2) determining which set of properties is needed to single out the target object from the distractors.

Most previous discussions on reference in discourse have been purely linguistically focused. In these discussions the set of possible referents is called the *Discourse Context* (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” [4, p. 3]. There are two fundamental referring operations on the DC model: *evoking* and *accessing*.

“When a referent is first mentioned in a discourse, we say that a representation for it is evoked into the model. Upon subsequent mention, this representation is accessed from the model”. [5, p. 672].

The term *evoking* is used to describe a reference to an entity that is known to the interpreter through their conceptual knowledge but is new to the discourse. Indefinite noun phrases are normally used to evoke a representation for a new entity that satisfies the given description into the DC. Once a representation of an entity has been evoked into the DC it licenses the use of other types of reference to access the representation of the entity. Referring expressions that access a representation in the DC are called *anaphoric* and the referring expression that introduced the accessed representation is called the *antecedent*. Pronouns are the paradigmatic example of anaphoric reference.

However, in NLVR systems the discourse is situated in a visual environment shared between the user and the system. The user maintains a spatially-located embodiment within the environment and, consequently, their utterances are spoken from a particular point of view within a simulated context. Psycholinguistic studies [3] have demonstrated that in visually grounded discourse the interpretation of language 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”. [3, pp. 211–212]

Furthermore, in NLVR contexts it has been found that users often refer to objects that they have seen in the 3-D simulation:

“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”. [4, p. 6]

This interaction between visual perception and linguistic reference introduces a third form of operation on the DC model. Following [10], we use the term *exophoric* to describe a referring expression that denotes an object or objects that have entered the DC model through visual perception, but have not been previously evoked into the model by a referring expression drawing on the interlocutors’s conceptual knowledge.

In brief, in a visually situated discourse an entity can enter the DC model from at least 3 contextual sources: the interlocutors’ conceptual knowledge, the spatio-temporal context they visually perceive, and the previous utterances in the discourse. Furthermore, when we describe a referring expression based on the contextual source it draws its referent from (as opposed to its lexical form), we will speak of three different types of reference: evoking, exophoric and anaphoric.

For this paper, an important aspect of exophoric references is that they can be underspecified<sup>3</sup> with respect to the linguistic description of their referent. It has been shown that people use perceptual salience to resolve linguistically underspecified exophoric references:

“In order to identify the intended referent under these circumstances [*where there is more than one entity in the discourse domain whose properties fulfil the linguistic description of the referent*], subjects rely on perceptual salience as well as pragmatics”. [2, p. 6]

The ability of participants to resolve underspecified exophoric references using visual salience points to a saliency-based ordering of the set of possible referents within the visual context, similar to the saliency based ordering of possible antecedent referents within the linguistic context model that was noted above. Following this, this paper argues that in order to capture the flow of information from the visual perceptual context into what the user considers mutual knowledge, an NLVR system’s DC model (or any DC model of visually situated discourse) should maintain both a model of the evolving linguistic context and a model of what the user has seen. Moreover, the DC model should define a

---

<sup>3</sup> An underspecified reference is one for which there is more than one candidate referent available.

mechanism for ordering the elements in the visual perceptual context based on their visual saliency. In order to meet these requirements, we propose that DC models of visually situated discourse should integrate a model of visual saliency. To support this claim we will present a simple algorithm for modelling visual saliency and a DC model that integrates the information captured by this algorithm with a model of an evolving linguistic context. Furthermore, we will illustrate how the visual saliency information stored in this DC model can be used to underpin two of the fundamental sub-tasks of NLP: reference resolution and generation.

The structure of this paper is as follows. In Section 2 we review previous related work. In Section 3 an overview of the LIVE architecture is provided. In Section 4 the LIVE visual saliency algorithm is presented. In Section 5 the architecture of and data structures used in LIVE's DC model are described. In Section 6 the LIVE reference resolution algorithm is presented. In Section 7 the LIVE generation component is introduced and the algorithm it uses for reference generation is presented. In Section 8 the results from the evaluation of the framework are presented. Finally, in Section 9 we summarise and conclude.

## 2. Related work

Considering the advantages and opportunities that NLVR systems promise, it is not surprising that they have a rich tradition in HCI, AI and NLP research. Winograd's SHRDLU [11] is one of the earliest and best-known of these systems. SHRDLU carried on a dialogue with a user concerning the activity of a simulated robot arm in a simple blocks world. [12–14] provide overviews of more recent systems. Surprisingly, most of these systems use the visual context to provide a general domain of discourse but neglect the contextual information available from the visual scene when interpreting and generating references. Two of the most recent systems, DENK [15] and VIENA [16], do make limited direct use of the visual context for NLU. However, these systems only distinguish between what is currently visible and what is not, using this binary distinction as a filter on the set of candidate referents for a given reference. The Bishop system [17] is closest to our approach in that it grounds NLP in a context model constructed from the system's model of visual perception.

The Bishop system focuses exclusively on reference resolution. The main components of the system are: a set of visual feature extraction algorithms, a lexicon that is grounded in terms of these visual features, a parser, and a compositional engine driven by the parser that combines the lexical units. The visual features extracted by the system are: the average RGB colour of all the pixels attributed to an object; the centre of mass of an object; the distance between pairs of objects' centres of mass; groups of objects in the scene as determined by finding all the sets of objects that contain more than one object, and in which each object is less than a threshold away from another object in the group; pairs and triplets (same as groups, but filtered to produce only groups of two or three objects respectively); the attentional vector sum which is the spatial relation measure between two objects. The domain of discourse provided by the visual simulation is a flat surface containing up to 30 objects with random positions. The objects are all of identical shape (cones in the examples given) and size, and are either green or purple. This domain of discourse was designed to

lead the users to make reference to spatial aspects of the scene. Given a user input consisting of a reference to one of the objects in the scene (e.g., *the purple on the left*) the system tries to resolve the reference using the visual features extracted from the scene.

Although Bishop and LIVE share an approach to grounding NLP in a visual context, there are important differences between the two systems. Firstly, Bishop focuses on resolving references containing spatial terms rather than on the effect of visual salience on reference. When dealing with these types of reference, if there is an unresolved ambiguity at the end of the interpretation process, Bishop chooses the candidate referent with the maximum reference strength (i.e., the candidate referent whose location is best described by the spatial term). However, as the system has no model of visual salience, Bishop cannot resolve underspecified references which do not contain spatial terms. Secondly, in the Bishop system the user has a fixed view of the world from which all the objects in the simulation are visible. As a result the system has no visual memory. However, in many VR simulations, including LIVE, the user can move around the world and, consequently, may make a first reference to an object that they saw earlier but is not currently visible. Finally, although the Bishop system can handle some anaphoric references (essentially certain words, such as *that* as in *to the left of that one*, are taken to refer to the referent of the preceding input), its model of discourse is less sophisticated than the one proposed in this work.

### 3. The LIVE system architecture

Most previous NLVR systems give their NLP modules complete access to all the objects in the simulated world. We argue that this is phenomenologically unrealistic and impractical in large simulated environments where such an approach can result in a multitude of (situationally irrelevant) possible referents in the case of reference resolution, or distractors for reference generation. In contrast, the LIVE NLP module's knowledge of the simulation is mediated through a dynamically evolving context model which is founded on the information captured by a visual saliency algorithm as the user navigates the simulation.<sup>4</sup> Consequently, the system uses a model of what the user has seen as the basis for its NLP. The advantage of this is that the system can reduce the number of objects it considers as possible referents for a referring expression to those that have been rendered during the course of the user-system interaction. Similarly, visual saliency information associated with the objects that are in the dynamically evolving context model enables the LIVE reference generation algorithm to generate sufficiently detailed but underspecified references in some of the contexts where previous reference generation algorithms that do not accommodate visual saliency would fail.

Fig. 2 is a schematic overview of the LIVE system architecture. Each of the boxes names a component in the system. The arrows between the boxes represent the flow of information between the components: (1) represents the user entering commands to the system; (2) represents the parsed input being passed to the interpretive module; (3) rep-

---

<sup>4</sup> For repeat visits, the context model can be stored and reactivated for the next visit.

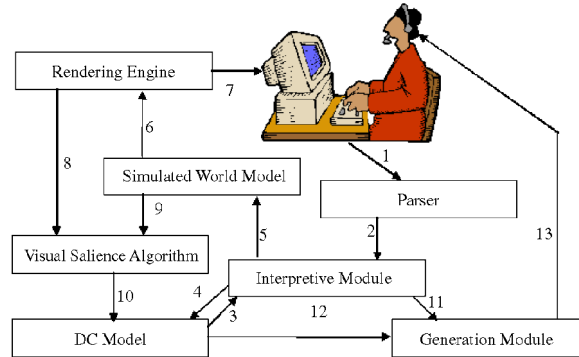


Fig. 2. Schematic of the LIVE system architecture and the data flow between the system components.

represents the flow of contextual information from the DC model to the interpretive module (this information provides a context for the interpretation of the user's input); (4) represents the updating of the DC model after the interpretation module has processed linguistic user input; (5) represents the updating of the world module after the interpretation module has processed a user input; (6) represents the rendering engine's use of the world model during the rendering process, represented by (7); (8) represents the interrogation of the rendering engine by the visual saliency algorithm after each frame has been rendered; (9) represents the visual saliency algorithm's use of false-colouring information stored in the world model; (10) represents the updating of the DC model by the visual saliency algorithm after a frame has been rendered; (11) represents the triggering of the generation module by the interpretation module in response to a generation command; (12) represents the flow of contextual information from the DC model to the generation module; and (13) represents the outputting of the generated description.

#### 4. Modelling visual perception

The LIVE system uses information about the visual saliency of different objects in a scene when interpreting or generating referring expressions. To compute visual saliency the system draws on previous computational models of visual 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. [18] 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 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” [19, 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 [20]. Perceivers are active seekers and processors of information. [21] described attention as a “spotlight that enhances the efficiency of the detection of events within its beam” [21, 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” [20, p. 276]. Attention is greatest at a single point in the visual buffer and drops off gradually from that point.

#### 4.1. Modelling visual perception: Previous work

Visual attention affects awareness of what is perceived and 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 visual attention have been developed, see [22] and [23] for recent reviews. However, most of these models are not suitable for NLVR systems as they have connectionist architectures 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 for 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 techniques they use: ray casting and false colouring. [24] implemented a 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. Ray casting is widely used in off-line rendering of graphics; however, it is computationally expensive and for this reason is not often used in real-time rendering.

Another graphics-based approach to modelling vision was proposed in [25]. 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* [25, 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 inter-

vals, a model of the character’s view of the world using the false colours is rendered. Once this rendering is finished, the viewport<sup>5</sup> is copied into a 2-D array along with the z-buffer<sup>6</sup> values. By scanning the array and extracting the pixel colour information, a list of the objects currently visible to the actor can be obtained. [26] proposed a navigation behavioural system that used false colouring synthetic vision. [27] 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.

#### 4.2. Modelling visual perception: The LIVE visual saliency algorithm

The basic assumption underpinning the LIVE 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 technique. 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. 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. For each object 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 LIVE system the false colour rendering is  $200 \times 150$  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. Fig. 3 illustrates the normal rendering of a scene from the LIVE system and Fig. 4 illustrates the  $200 \times 150$  false colour rendering of the same scene. (For colours in the figures see the web version of this article.)

To model the size and centrality of the objects in the scene, the LIVE system assigns a weighting to each pixel using Eq. (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. This equation normalises the pixel weightings between 0 and 1. Fig. 5 illustrates the distribution of pixel weightings assigned using Eq. (1). It is evident that the closer a pixel is to the centre of the image, the higher its salience.

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

<sup>5</sup> A viewport is the rectangular area of the display window. It can be conceptualised as a window onto the 3-D simulation.

<sup>6</sup> The z-buffer stores for each pixel in the viewport the depth value of the object rendered at that pixel.



Fig. 3. A scene in the LIVE domain.

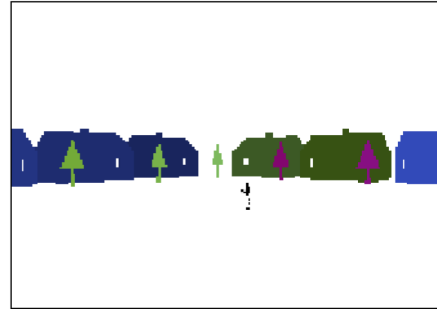


Fig. 4. The false colour rendering of the scene in Fig. 3.

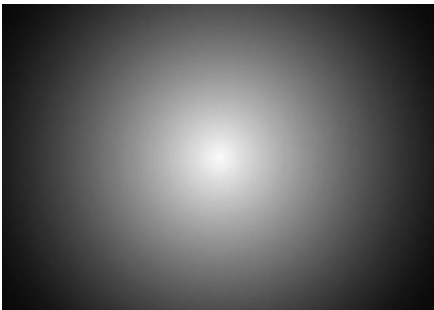


Fig. 5. The weighting assigned to the pixels in the viewport using Eq. (1). The darker the pixels the lower the weighting. Weightings range from 0–1.

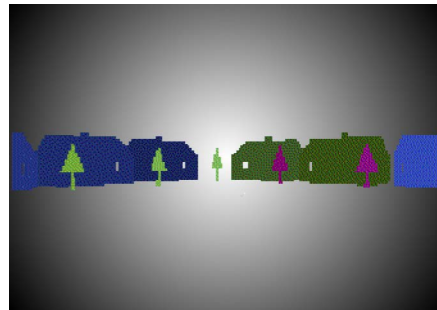


Fig. 6. An overlay of the false colour rendering of Fig. 3 on the distribution of pixel weightings.

After weighting the pixels, the LIVE 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 ID. The summed pixel weighting for each object in the scene is then normalised by dividing it by the overall maximum summed pixel weight ascribed to an object in the scene (Algorithm 1). This normalised value is the relative visual saliency of the object in the scene.

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. Fig. 6 illustrates the relationship between the false colour rendering of a scene and the weightings ascribed to the pixels in the viewport.

It is important to note that the scanning process in the LIVE visual saliency algorithm differs from those in the previous false colour based synthetic vision models [25–27]. Previous algorithms simply recorded whether an object had been rendered or not. The LIVE 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 LIVE system to rank the objects based on their visual saliency. We do not claim that this algorithm accommodates

**Algorithm 1** (*The LIVE visual saliency algorithm*).

*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 = array of length colourtable
- (2) **Let** MAX = 0
- (3) **foreach**  $p \in \text{Pixels}$ 
  - (a) **Let** *pixelWeight* = result of Eq. (1) applied to  $p$
  - (b) **Let** *pixelRGB* = the RGB value (i.e., the colour) of  $p$
  - (c) **Let** *objectID* = the object ID the *colourtable* maps *pixelRGB* to
  - (d) **Let** objectlist[objectID] = objectlist[objectID] + *pixelWeight*
  - (e) **if** (objectlist[objectID] > MAX)  
**then** MAX = objectlist[objectID]
- (4) **foreach**  $e \in \text{objectlist}$ 
  - (a) objectlist[ $e$ ] = objectlist[ $e$ ] / MAX
- (5) **return** objectlist

all the perceptual factors that impact on visual saliency (cf. the list identified by [18]). However, it defines a reasonable model of visual saliency that operates fast enough for real-time systems in rapidly changing environments. Furthermore, by using a false colouring algorithm to model visual saliency 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 LIVE 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 dynamically to reflect the user's gaze position as the maximum of the saliency 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 LIVE system, we have integrated the visual context information created by the visual saliency algorithm with a linguistic context model of user input. Using this information the LIVE system is able to define a local context for the interpretation of a given exophoric 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 viewport and those that the user has already seen. A further advantage of this approach is that the visual saliency scores associated with the objects in the context model allows the system to adjudicate between candidate referents when interpreting am-

biguous references (Section 6.4). Similarly, visual saliency scores allows LIVE to generate underspecified referring expressions where other systems would fail (Section 7.1).

## 5. Discourse context models

There is a large body of work on discourse structure. Two of the better known frameworks are DRT [28] and focus stacks [29]. These frameworks are representative of the majority of this work in that they are primarily concerned with linguistic phenomena and consequently neglect the effect of visually accessible referents on the context of a visually situated discourse, such as the discourse between a user and an NLVR system. In this section we address the question: *what should a basic NLVR DC model represent?* Having developed a basic set of design criteria, we present the architecture and data structures of the LIVE DC model.

In the introduction it was noted that in a visually situated discourse an interlocutor may select the referent of a referring expression from several information sources including: their conceptual knowledge, these types of referring expressions are called *evoking*; the spatio-temporal context they visually perceive, these types of referring expressions are called *exophoric*; the previous utterances in the discourse, these types of referring expressions are called *anaphoric*. Each of these types of reference introduces constraints on the design of an NLVR DC model. If an NLVR system is going to interpret evoking, exophoric and anaphoric references, the context model it uses must be updated as a result of both visual and linguistic events; furthermore, the context model should accommodate the semantic categories that the conceptual (or world knowledge), visual and linguistic information sources introduce into the discourse.

In order to be able to resolve an evoking reference an NLVR system must have access to some *a priori* conceptual knowledge. In particular, it should have access to taxonomic information defined for each type of entity in its world: entities, type, their geometry, the features the objects possess (e.g., colour etc.), the range of values these features can take (e.g., brown etc.). For example, in order to resolve the reference *a red house* in the input *add a red house* the system must be able to access and instantiate the geometric description and other defaults associated with the class of objects described by the noun *house* and the RGB value described by the adjective *red*.

In order to interpret an underspecified exophoric reference, an NLVR system must be able to define the set of objects in the local visual context that fulfil the linguistic description given in the referring expression and rank the elements of this set by their visual salience. In order to meet these requirements the context model must integrate a model of visual salience and implement a mechanism for establishing a set of objects with a particular property or feature value.

There are, however, exophoric underspecified references where the candidate referents need to be ordered based on a different criterion than visual salience. For example, in visually situated language locative expressions of the form  $(NP_{\text{subj}} + \text{Prep}_{\text{loc}} + NP_{\text{obj}}$ , e.g., *the tree in front of the house*) are one of the more common and complicated forms of exophoric reference. Resolving a locative expression involves selecting the object whose location is

best described by the locative.<sup>7</sup> Here the objects fulfilling the description given in the subject noun phrase should be rated based on their location within the area described by the locative and in the event of a tie by visual salience. Noun phrases containing comparative or superlative adjectival descriptions also require the candidate referents to be ordered based on criteria different to visual salience. For example, the reference *the shortest blue house* can be used to denote an (already encountered) object that is not currently visually salient. Resolving these references requires the context model to be able to overlay multiple ordering criteria on a set of elements.

To resolve anaphoric references the system must maintain a list of the discourse representations that previous expressions have introduced into the discourse. Similar to the visual context, the linguistic context should be integrated with the conceptual knowledge, i.e., the representation introduced into the linguistic context by a referring expression should be linked to the conceptual knowledge associated with the expression's referent. Furthermore, as with exophoric references, we take salience (in this instance linguistic salience) to be one of the primary drivers in resolving anaphoric references. The referent of a given anaphoric referring expression is the most linguistically salient entity in the linguistic context whose associated object fulfils the linguistic description of the expression. There is a substantial research literature on the topic of linguistic salience. Much of this work has formulated linguistic salience in terms of *hierarchical recency* [29,31–33].<sup>8</sup> These hierarchical models of discourse structure maintain a tree structure representation of the discourse. The representation of a previous utterance is hierarchically recent to the current utterance if it is adjacent to the current utterance within this tree structure. However, to a first approximation the basic idea underlying these methods is similar: among the available potential referents, the more recently a referent has been mentioned the more salient it is in the linguistic discourse. Consequently, in order to model linguistic salience, the context model should at least accommodate recency of mention.

Interpreting some forms of anaphoric reference may require the context model to store other categories of information. For example, in order to resolve *other-anaphora* combined with *one-anaphora* it may be necessary to store the linguistic access mechanism used to refer to a preceding referent. Other-anaphora occurs when a definite description contains the modifier *other*. In DRT “other must be represented by a discourse referent that is presented as distinct from some discourse referent already introduced in the DRS” [28, p. 463]. One-anaphora occurs when the token *one* is used as a substitute for the head of a noun phrase. In these uses, *one* picks up some property of an antecedent noun phrase's referent. In the context of an interaction dialogue between a user and an NLVR system, the type information of an expression's referent is normally given by the head noun of an antecedent referring expression; i.e., the token *one* picks up the type information from the preceding reference. However, *one* can also pick up a preceding adjectival description. For example, taking Fig. 7 as the initial visual context, Fig. 8 illustrates an interpretation of a user input

---

<sup>7</sup> The process of grading the locations of candidate referents can require access to several conceptual categories that would not otherwise be required. For example, if an object has an intrinsic frame of reference associated with it, this information should be stored as part of the system's conceptual knowledge. See [30] for details of the LIVE model of the semantics of projective prepositions.

<sup>8</sup> Of these only [29] provides an operational definition of hierarchical recency.

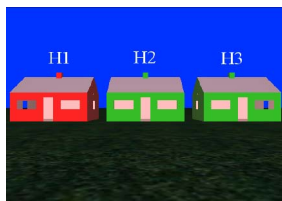


Fig. 7. The initial visual context (H1 = red house, H2 = green house, H3 = green house).

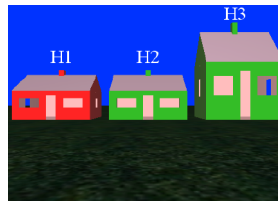


Fig. 8. The visual context after the input *make a green house taller* has been processed.

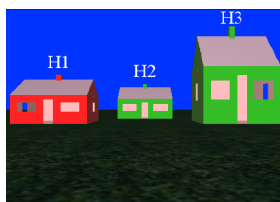


Fig. 9. The visual context after the input *move the other one back* has been processed.

*make a green house taller*, with house H3 being selected as the referent. If the user now inputs the command *move the other one back*, the system can exclude H3 as a possible referent because of the semantics of the modifier *other* requiring the referent to be distinct from H3. However, the system still needs to distinguish between H1 and H2. We argue that in these situations the token *one* picks up both the type and the adjectival description used in the referring expression in the preceding utterance. Consequently, the LIVE system will in this instance resolve *the other one* as *the other green house*. Fig. 9 illustrates the visual context after the input *move the other one back* has been processed.

In summary, a basic NLVR context model should minimally:

- (1) have access to a model of conceptual knowledge (such as object type, object properties and property values),
- (2) incorporate perceptual information such as what objects are in the local visual context and the relative visual salience of these objects,
- (3) incorporate linguistic information such as when an object was last referred to and the semantic content used in the referring expression,
- (4) mark the salience ascribed to an object in a given context as a result of linguistic reference,
- (5) define a mechanism for modelling the effect of recency on both visual and linguistic salience,
- (6) be dynamically updated as a result of both linguistic and visual discourse events (we define a visual discourse event as the rendering of a scene),
- (7) facilitate the interaction between the different modalities by using a generic data structure for storing visual and linguistic information,

- (8) provide local interpretive contexts for each discourse event,
- (9) define a mechanism for grouping and ordering the elements within a particular local context model based on one or more criteria.

### 5.1. The LIVE discourse context model architecture

The LIVE context model is divided into two stacks of local contexts. One stack represents the evolving visual context and is called the visual domains list (VDL). The other stack represents the evolving linguistic context and is called the linguistic domains list (LDL).

Following [34] and [35], we assume that the process of resolving a referential expression is achieved by accessing and restructuring a local context. In [34] these local contexts are referred to as *cognitive domains*, however, in this paper, following [35], we use the term *reference domain*.

The VDL and LDL both contain a stack of reference domains. New reference domains are added to these stacks in response to discourse events occurring within the context they are modelling. We define the rendering of a scene as a discourse event in the visual context, and the interpretation of a referring expression as a discourse event in the linguistic context.

The reference domains in the VDL are created using the output of the real-time visual saliency algorithm described in Section 4.2. To account for rapidly changing simulated environments the visual saliency algorithm runs each time a scene is rendered and returns a list of the objects in the scene along with a relative salience ranking for each object. Each reference domain in the VDL describes a particular set of objects in a given scene. New reference domains are added to the top of the VDL stack. Consequently, the more recently the scene was rendered, the higher in the stack the scene's reference domains are stored. Mimicking the degradation of human memory, the VDL and LDL *forget* objects that have been rendered or referred to recently. Currently, each stack contains up to 3000 reference domains and once full they discard the old reference domains at the bottom of the stack as new ones are added at the top. A reasonable extension to this approach would be to use the focus stack framework [29] to manage the LDL stack, as it also uses a stack based representation. The focus stack framework uses a model of speaker intention to define segments within the discourse and stack elements get pushed and popped at the transition from one discourse segment to the next. However, this approach could not be used to manage the VDL as there is no analogous process to speaker's intentionality in the global structuring of the visual context.<sup>9</sup>

The reference domains in the LDL are created by the LIVE linguistic interpretive module. A new reference domain is added to the top of LDL as the result of the interpretation of a referring expression that has been input by the user. The more recent the referring expression was used, the higher the reference domain representing the interpretation of the expression is in the LDL stack.

Because newly created reference domains are added at the top of each stack, reference domains are chronologically ordered. This allows the LIVE context model to capture the

---

<sup>9</sup> However, a model of an agents's intentions could be used as a factor in computing visual salience, as entities that are relevant to the agents intentions would be more salient.



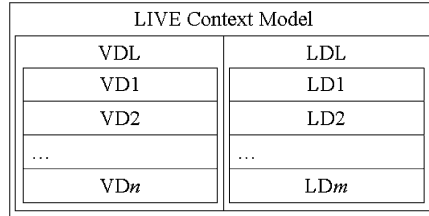


Fig. 10. Schematic of the LIVE context model architecture.

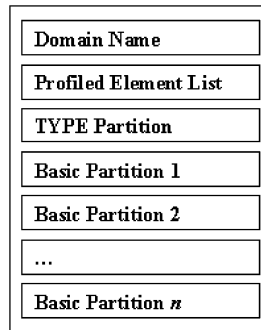


Fig. 11. The internal structure of a reference domain.

effect of recency in the evolving contexts. Fig. 10 gives a schematic of LIVE's context model architecture. In this figure, VD1...VD $n$  are reference domains representing the semantics of visual discourse events and LD1...LD $m$  are reference domains representing the semantics of linguistic discourse events. The lower the index of the reference domain [1... $[n/m]$ ], the more recently the discourse event occurred.

### 5.2. The LIVE discourse context model: Reference domains

The basic units of the LIVE context model are the reference domains. The reference domains in the VDL and LDL share a similar internal structure. The sharing of an internal structure across the different contexts is important because the process of resolving a referring expression can result in a reference domain being copied to a different stack; e.g., interpreting an exophoric reference will result in a reference domain from the VDL being restructured and copied to the LDL. The complexity of the resolution process would be greatly increased if a reference domain had to be ported to a different format before being inserted into a different context.

Each reference domain contains a domain name, a profiled element list, a type partition, and a set of zero or more basic partitions. Fig. 11 illustrates the internal structure of a reference domain.

The reference domains are named after the type of objects they contain. For example, if a domain describes a set of houses it is named *house*, if it describes a set of objects of different types it is set to a generic type *thing*.

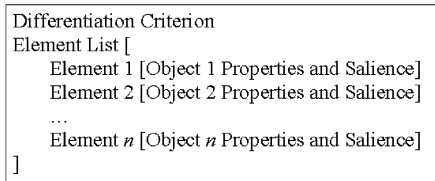


Fig. 12. The internal structure of a partition.

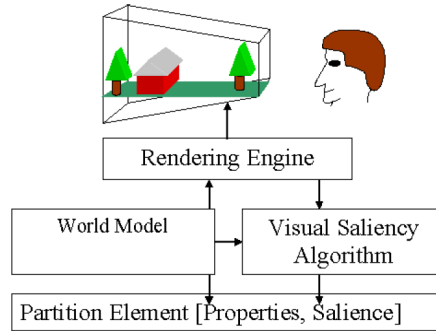


Fig. 13. The internal structure of a partition's element.

The profiled element list stores the representations of the objects in the local context, modelled by the reference domain, that are prominent in that local context. In the LIVE system, an object is marked as profiled if it was referred to in the linguistic discourse.

The function of the partitions, both type and basic, is to predict the different ways that a user may refer to an object in the domain. Each partition is comprised of a differentiation criterion and an element list. The differentiation criterion of the type partition is set to the domain's type (i.e., the domain name). This partition lists all the elements in the domain apart from the profiled element(s). The differentiation criterion of a basic partition is the attribute that distinguishes the elements of a partition from the elements of the reference domain that are excluded from that partition. The partition's element list is the data structure where the partition's elements are stored. The elements of a partition in a particular reference domain represent objects in the 3-D simulation that are of the correct type for the reference domain and that have the property specified by the partition's differentiation criterion. For example, the basic partition whose differentiation criterion is *red*, and whose reference domain is named *house* and which is at the top of the VDL will contain elements representing the red houses that are currently visible. Fig. 12 illustrates the internal structure of a partition.

Each element in a partition has two components: a pointer to an object in the world model and the visual saliency rating ascribed to the object in the world that the element's pointer denotes. Fig. 13 illustrates the internal structure of a partition's element and how it relates to other components in the LIVE framework.

Where a partition's differentiation criterion describes a type or a colour, the elements in the partition's element list are inserted into the list in ascending order based on their visual saliency scores. The partitions use a last-in-first-out access policy. The insertion process results in the element with the highest visual saliency score being inserted at the first access location within the partition's element list. This organisation reflects one of the fundamental assumptions underlying the LIVE approach: that is, all other factors being equal, objects which have a higher visual saliency are more likely to be the referents of a referring expression than objects which have a lower visual saliency. Fig. 14 illustrates the ordering of elements in a partition's element list based on their visual saliency scores.

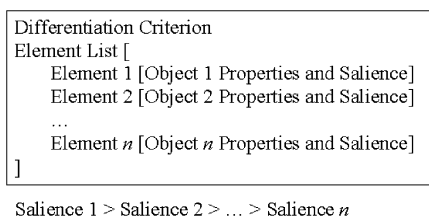


Fig. 14. The ordering of elements in a partition based on their visual saliency scores.

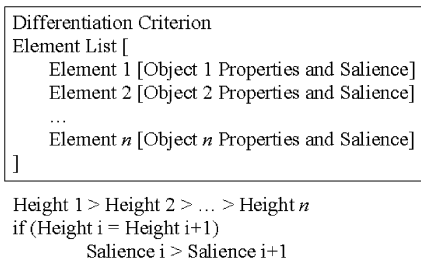


Fig. 15. The ordering of elements in a partition based on a gradeable property: in this instance height.

However, if a partition's differentiation criterion describes a gradeable property such as height or location within a preposition's spatial template<sup>10</sup> the elements are ordered based on their fitness with respect to the partition's criterion. In situations where two elements within a partition are equal with respect to the differentiation criterion, the element with the lower visual saliency rating is inserted first. Fig. 15 illustrates the ordering of elements in a partition modelling a gradeable property of the elements in a reference domain. In this example, the property modelled by the partition is height. Consequently, the taller the element the lower its index in the list.

## 6. Reference resolution in visually situated discourse

In this section we present the general interpretive algorithm used by the LIVE system. The LIVE system handles several types of referring expressions: definite descriptions (including other-anaphora and one-anaphora), indefinites, singular demonstratives accompanied by a deictic gesture (input using the mouse pointer), pronominal reference (*it*), and spatial locatives. The system's interpretive module uses a three step algorithm to handle these references, see Algorithm 2. Each of these steps is described in detail below.

**Algorithm 2** (*The LIVE reference resolution algorithm*).

- (1) Select the general context for the interpretation: VDL for exophoric references, LDL for anaphoric references.
- (2) Select the reference domain from the relevant context: this reference domain functions as the local context for the interpretation.
- (3) Select the referent(s) of the referring expression from the local context and mark it/them as being prominent within the reference domain.

<sup>10</sup> A spatial template is a representation of the regions of acceptability associated with a given preposition. See [30] for information on how the LIVE system models the semantics of prepositions.

### 6.1. *Selecting the general context: VDL or LDL*

In the LIVE context model the LDL is the appropriate context for anaphoric references and the VDL is appropriate for exophoric references. However, in a visually situated discourse deciding whether a reference is anaphoric or exophoric is extremely difficult because most forms of referring expression can be used in both an anaphoric and exophoric manner. Definite descriptions are one of the most difficult types of referring expression to categorise. They “apply freely to objects introduced in discourse, present in the physical environments, or available through the common background of the discourse participants” [36, p. 371]. Indeed, “the two most common cases of definite descriptions in the TRAINS<sup>11</sup> conversations are anaphoric definites and definites interpreted with respect to the visual situation” [37, p. 214]. In some instances, the categorisation of a definite can be based on syntactic information [38]. However, in the absence of a syntactic cue, one is forced to use heuristic rules.

Given that in an NLVR scenario the main information source is the visual simulation, it is expected that in the majority of cases definite descriptions will be exophoric. While we acknowledge that this assumption is a simplification of the issue, we take the exophoric interpretation to be the default and treat the anaphoric interpretation as an exception. We define two conditions that must be met in order to trigger an anaphoric interpretation of a definite description. These are:

- (1) the profiled element in the reference domain at the top of the LDL stack must fulfil the description provided in the definite description,
- (2) the profiled element in the reference domain at the top of the LDL stack must be currently visible.

The motivation of condition (2) is to catch situations where a user has referred to an object and subsequently navigated to a different part of the simulation. If the user then referred to an object of a type similar to their previous referential utterance and the system did not check whether the previous referent was still visible, the interpretation of the user’s new utterance would be applied to an object off screen. If both conditions are fulfilled, the LIVE system takes the definite description to be an anaphoric reference and selects the LDL as the general context; otherwise, the default exophoric interpretation, using the VDL as the general context, is triggered.

### 6.2. *Selecting the local context*

Having categorised the reference as either anaphoric or exophoric, the next stage in the interpretation algorithm is to select a reference domain to act as a local context. The reference domains are named based on the types of objects they contain and are chronologically ordered (see Section 5.1). The process for selecting the reference domain uses both

---

<sup>11</sup> The TRAINS corpus is a multimodal corpus created at the University of Rochester. See <http://www.cs.rochester.edu/research/trains/> for more information.

the temporal and lexical domain information. The general approach is to select the most recent domain within the relevant domain list that contains one or more elements which match the description of the object in the expression. Where no description is given (*it, this* etc. without accompanying gesture), the selection process returns the most recent domain in the selected general context.

For this stage of the interpretive process, all exophoric references are treated as equivalent. Selecting their reference domain involves searching for the most recent domain in the VDL that contains elements fitting the description given in the referring expression.

Selecting the reference domain for an anaphoric referring expression is more complicated because different forms of anaphoric expressions make different presuppositions about the structure of their local context. For example, (singular) other-anaphora designates an object that has been excluded from a specified or implied group. Following this, the reference domain selected as the local context for an other-anaphoric expression should contain a specified or implied grouping and at least one element that has been excluded from this group that fits the description given in the referring expression. In the terminology used in the LIVE framework, these considerations translate into the requirement that the reference domain should contain one or more profiled elements and at least one non-profiled element that fulfils the property and type descriptions given in the reference.

### 6.3. *Selecting the referent and updating the context model*

In the LIVE framework the referent of a referring expression is the most salient element in the local reference domain that fits the description given in the referring expression.

Some referring expressions, such as the pronoun *it*, carry very little semantic content. We treat these expressions as a continuation—in the spirit of [39]—of the current linguistic context. They are understood as referring to the most salient element in the linguistic context, i.e., the object most recently referred to. The representation for this object in the context model is always located in the profiled element list of the reference domain at the top of the LDL stack. The continuation of the context is modelled by inserting a copy of this reference domain at the top of the LDL stack.

However, for more complicated types of referring expression (*definite descriptions, other-anaphora*, etc.) the selection of the referent is a two stage process:

- (1) Select the element in the reference domain that represents the expression's referent.
- (2) Profile (mark as being prominent in the reference domain) the selected element.

A crucial factor in the selection of a referent is the internal structure of the reference domains. Recall, from Section 5.1, that each domain is divided into partitions and each partition defines the set of objects in the reference domain with a particular attribute or attribute value (specified by the partition's differentiation criterion). Moreover, the elements of each partition are ordered by their visual saliency within the reference domain or by their fitness with respect to the differentiation criterion and then by their visual saliency.

Selecting the reference domain element that represents the expression's referent is different for each type of referring expression. For an exophoric definite description the

referent is the most salient element that fulfils the description of the referent in the expression in the most recent VDL reference domain of the correct type. Consequently, if no adjectives are used in the referring expression, the most salient element in the domain is selected. For example, the exophoric interpretation of the expression *the house* is the most salient element in the type partition within the most recent VDL reference domain of type *house*. For exophoric references which contain an adjectival description, the selection procedure consists of searching the most recent VDL reference domain of the correct type for a basic partition whose differentiation criterion matches the adjectival description and selecting the most salient element in that partition. For example, under an exophoric interpretation, the referent ascribed to the expression *the red house* would be the most salient element in the *red* partition within the most recent VDL reference domain of type *house*.

Once the referent of an expression has been selected the reference domain is restructured to mark the prominence of the referent within the domain. Algorithm 3 lists the steps in this process.

**Algorithm 3** (*The LIVE profiling algorithm*).

- (1) If there are element(s) in the profiled element list, remove them and insert them back into the type and basic partitions.
- (2) Copy the element(s) that represent the referent(s) of the referring expression into the profiled element list and delete them from the type and basic partitions in the domain. Removing the newly profiled elements from the type and basic partitions in the reference domain means that other-anaphoric expressions can be interpreted by selecting the most salient element in the type and basic partitions whose differentiation criterion matches the description in the expression.
- (3) Mark the partition that modelled the decomposition of the reference domain that was used to denote the referent(s) as profiled. Marking the partition used to select the referent records the semantic information used to denote the referent; i.e., the differentiation criterion of a profiled partition stores the description used to denote the elements that are profiled in the domain. This information is useful for resolving anaphoric referring expressions such as *the other one*.

The final step in the interpretation process is to insert the restructured domain at the top of the LDL for subsequent anaphoric reference. This final step interrelates the VDL and LDL context models.

#### 6.4. Reference resolution: A worked example

To illustrate the dynamic updating and interaction between the visual and linguistic contexts in the LIVE context model, a brief worked example is provided here. The example focuses on the interpretation of an underspecified exophoric definite description. The initial visual context for the example is shown in Fig. 16.

Assuming that there has been no previous linguistic input, Fig. 17 provides a schematic illustration of the state of the context model when the scene in Fig. 16 is on screen. As there has been no previous linguistic input the LDL is empty. However, there is a visual context

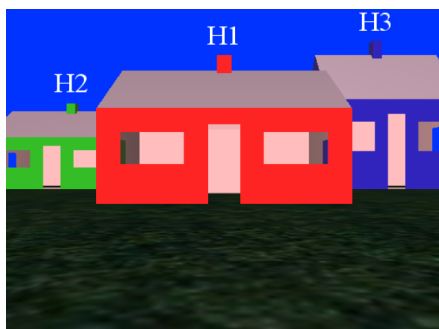


Fig. 16. The initial visual context.

LIVE Context Model	
VDL	LDL
VDL Domain Name: house Profiled Element List: null Type Partition { Diff Criterion: house, Element List: [ [H1, Saliency 1.0000], [H3, Saliency 0.2001], [H2, Saliency 0.1131] ] } Basic Partitions: 0 ... n ...	null

Fig. 17. Context model resulting from processing Fig. 16.

which is represented by the reference domain at the top of the VDL. This reference domain is of type house and the three houses in the scene have representations in this reference domain. In the reference domain's type partition these representations are ordered by their normalised visual saliency scores: H1 has a visual saliency rating of 1.000, H2 a rating of 0.1131 and H3 a rating of 0.2001.

Given this context, the command *make the house taller* triggers an exophoric interpretation of the referring expression *the house* and the VDL is selected as the general context for the interpretation.

For all exophoric referring expressions the local context used for their interpretation is modelled by the most recent reference domain in the VDL that contains elements fitting the description given in the referring expression (see Section 6.2). In this example, the reference domain at the top of the VDL stack, VD1, is a suitable context.

Next the referent for the expression must be selected. For an exophoric definite description the referent is the most salient element that fulfils the description of the referent in the selected reference domain. As there is no adjectival description in the referring expression, the reference domain's type partition contains the elements representing the objects that fit the description. However, the referring expression *the house* is underspecified as there is more than one object in the context fitting the description given in the reference; this is reflected in the context model by the fact that there is more than one element in the reference domain's type partition. In this instance, however, the system can resolve this ambiguity using the visual saliency values associated with the elements in the partition. The element in the first access location in the type partition is taken to be the referent: H1.

It is important to note that the LIVE framework tries to recognise cases of genuine ambiguity arising from underspecification. It does this by comparing the visual saliency ascribed to the primary candidate against the visual saliency of the other candidate referents. If the primary candidate's visual saliency does not exceed the visual saliency of the other candidates by more than a predefined *confidence interval*, the reference is deemed to be ambiguous and a message to that effect is output to the user. Since the LIVE visual saliency scores are normalised, we have set the system's confidence interval to the midpoint of the saliency value range 0.5. This value, however, can be adjusted to reflect a stricter or looser interpretation of a referring expression.

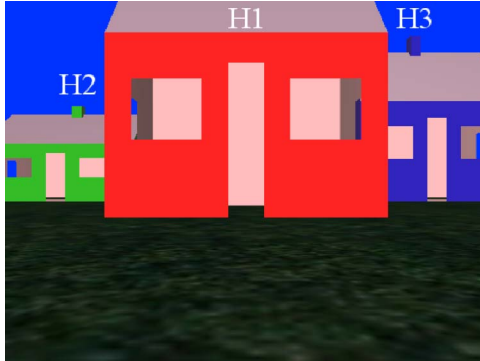


Fig. 18. The visual context after the interpretation of *make the house taller*.

LIVE Context Model	
VDL	LDL
<b>VDL</b> Domain Name: house Profiled Element List: null Type Partition [ Diff Criterion: house, Element List: [ [H1, Saliency 1.0000], [H3, Saliency 0.1343], [H2, Saliency 0.0858] ] ] Basic Partitions: 0 ... n ...	<b>LDL</b> Domain Name: house Profiled Element List: [ H1, Saliency 1.000 ] Type Partition [ Diff Criterion: house, Element List: [ [H3, Saliency 0.2001], [H2, Saliency 0.1131] ] ] Basic Partitions: 0 ... m ...

Fig. 19. The context model after interpretation of *make the house taller*.

Having selected a referent for the expression, the final stage of the interpretation process involves restructuring the reference domain to reflect the prominence of the referent in the context and updating the context model. Algorithm 3 in Section 6.3 defines the restructuring process. The result of this process is that the element representing H1 is stored in the domain’s profiled element list and is removed from the domain’s type and basic partitions. Once the reference domain has been restructured, the linguistic context is updated by inserting the restructured reference domain at the top of the LDL stack. Fig. 18 illustrates the visual context after the input *make the house taller* has been processed. Fig. 19 illustrates the updating of the LDL context by the insertion of the restructured reference domain. The difference in the visual saliences between the VDL and the LDL for the objects H3 and H2 in Fig. 19 is because the LDL reference domain models the context that the utterance was interpreted in and the VDL reference domain models the resulting visual context (i.e., the visual scene in Fig. 18).

## 7. Visual salience and natural language generation

Section 6 illustrated the importance of visual salience to visually-situated natural language understanding, showing how visual salience underpinned reference resolution in visually situated expressions. We will follow a similar approach to illustrate the importance of visual salience to visually situated natural language generation.

### 7.1. Generating referring expressions and visual salience

The Generation of Referring Expressions (GRE) is one of the most fundamental and ubiquitous tasks in natural language generation. Reference generation focuses on the semantic questions involving the factual content of the description, and does not concern itself with the linguistic realisation of the description. Many GRE algorithms have been proposed [7,40–49]. Most of these algorithms deal with the same problem definition: given a single target object, for which a description is to be generated, and a set of distractor ob-



jects, from which the target object is to be distinguished, determine which set of properties is needed to single out the target object from the distractors. On the basis of these properties a *distinguishing description* of the target object can be generated. A distinguishing description is a description of the target object that excludes all the elements of the distractor set. In this section we present a GRE algorithm that utilises the visual contextual and salience information stored in the LIVE context model to generate references that are linguistically underspecified yet sufficiently detailed to be resolved.

The current state of the art for GRE is the Incremental Algorithm [7]. The Incremental Algorithm “sequentially iterates through a (task-dependent) list of attributes, adding an attribute to the description being constructed if it rules out any distractors that have not already been ruled out, and terminating when a distinguishing description has been constructed” [7, p. 247]. If the end of the list of attributes is reached before a distinguishing description has been generated, the algorithm fails. The target object’s type is always included in the description generated even if it has no distinguishing value.

Most of the later GRE algorithms extend the original Incremental Algorithm in some respect: [44,45] extend the algorithm to handle targets as sets; [44] also extends the algorithm to handle context-dependent and vague properties; [47,48] concentrate on extending the algorithm to handle boolean properties (containing negation, conjunction).

The algorithms described in [42,43,49] focus on multimodal GRE, generating references that include deictic gestures. Similar to the LIVE GRE algorithm, these multimodal algorithms can generate linguistically underspecified references, where the ambiguity is resolved through the accompanying deictic gesture. However, most multimodal GRE algorithms, for example [42,43], assume that the deictic gesture is precise and unambiguous. Consequently, the generated expressions tend to be relatively simple, containing only a head noun with a pointing gesture. Furthermore, they tend to use context-independent criteria for deciding whether or not to include a deictic gesture in the reference: [42] generates a reference that includes a pointing gesture when it is not possible to generate a fully-specified linguistic description; [43] generates a multimodal reference for all objects that cannot be unambiguously described using a pronoun.

Unlike [42,43], the algorithm presented in [49] provides for various gradations of preciseness in the pointing gesture, ranging from unambiguous to vague. This algorithm uses a graph-based representation, where objects are represented as vertices, and properties and relations of these objects are represented as edges. Each edge has a cost associated with it. The problem of finding a referring expression for an object is treated as finding the cheapest subgraph which is isomorphic to the intended referent but not to any other object. Some aspects of this algorithm are similar to the LIVE GRE algorithm. [49] uses the size and distance of the target object to compute the costs associated with the different types of deictic gesture, similar to the size and centrality factors captured by the LIVE visual salience algorithm. There are, however, several differences between the LIVE GRE algorithm and the algorithm presented in [49]. First, the inclusion of a deictic gesture within a generated reference provides a focus of attention that allows the algorithm in [49] to disregard distractor objects outside this focus. By contrast, the LIVE GRE algorithm uses visual salience. Second, the LIVE GRE algorithm provides a fully operational definition of how the salience ratings of objects in the context should be computed. By contrast, [49] does not provide a precise method for calculating the costs associated with the absolute

and relative properties of the objects in the context set. This is an important issue for [49] as the output of the algorithm is dependent on a trade-off between these costs and the costs ascribed to the different forms of pointing gesture. Third, [49] uses a relatively more complex domain model. Consequently, it incurs a higher cost when updating and maintaining this context. In order to update its domain model [49] it must update the costs associated with each pointing that it may wish to generate for each object in the scene. However, the LIVE GRE algorithm need only update the visual salience score for each object it can see. This allows the LIVE framework to keep its domain model relatively simple, an essential aspect for situated real-time systems processing rapidly changing environments.

The output of the Incremental Algorithm is, to a large extent, determined by the context set it uses. However, Dale and Reiter eschew a precise definition: “[w]e define the context set to be the set of entities that the hearer is currently assumed to be attending to” [7, p. 236]. One possibility is to use the domain of discourse  $D$ , the total set of entities that can be referred to. However, using  $D$  as the context set can result in the Incremental Algorithm generating longer descriptions than are necessary: depending on the linguistic and/or perceptual context a reduced description may suffice and may in fact be more natural to the discourse.

Theune [50] Chapter 4<sup>12</sup> discusses whether restricting the context set to a proper subset of  $D$  containing those entities of  $D$  that have been referred to before would enable the Incremental Algorithm to generate reduced anaphoric descriptions. She concludes that restricting the context set in this way has an unwanted consequence: “the descriptions of all domain entities will be made relative to this restricted set” [50, p. 106]; as a result, the descriptions generated for new entities entering the discourse by the algorithm may not be sufficiently detailed to distinguish the target object from the other entities in the domain that are not in the restricted context set. Theune’s solution is to structure  $D$  by marking certain entities as more linguistically prominent than others. This is achieved using a framework for modelling linguistic salience that is a synthesis of the hierarchical focusing constraints of Hajicová [51] and the constraints of Centering Theory [39]. Essentially, in Theune’s extension to the Incremental Algorithm, “an entity that is being newly introduced into the discourse must be distinguished from all other entities in the domain, whereas an entity that has been previously mentioned can have a reduced description” [50, p. 106]. The idea underlying Theune’s extension is to modify the definition of a distinguishing description as:

“A definite description *the N* is a suitable description of an object  $d$  in a state  $s$  iff  $d$  is the most salient object with the property expressed by  $N$  in state  $s$ ”. [50, p. 101]

Theune’s structuring of  $D$  focuses on linguistic salience and enables the Incremental Algorithm to generate reduced anaphoric references. Like [50] we use a saliency measure to restructure  $D$ . Unlike [50], however, our saliency model is based on visual rather than linguistic salience. Consequently, our modified algorithm can generate underspecified exophoric references.

---

<sup>12</sup> See also [46].

We have developed a version of the Incremental Algorithm that uses the LIVE visual saliency algorithm and exploits humans' abilities to resolve underspecified references given a visual context. The integration of visual saliency information enables the algorithm to generate underspecified yet sufficiently detailed descriptions. The integration is achieved in two steps.

Firstly, the output of the LIVE visual saliency algorithm is used to create the context set used by the LIVE GRE algorithm. This excludes all the objects in the world that are not currently visible and improves the ability of the algorithm to generate relevant references. A further advantage is that the objects in the context set can be ordered by their visual saliency scores.

Secondly, we modify the definition of a distinguishing description to exploit the visual saliency scores associated with each object in the context set. Our definition of a distinguishing description requires that the target object should not only be the most salient object fitting the generated description in the scene but its salience should exceed the salience of the elements in the distractor set that fulfil the description by a predefined confidence interval; i.e., *a description is distinguishing if it excludes all the distractors that have a visual salience greater than the target object's salience minus a predefined confidence interval*. The motivation for this definition is that a small difference in visual salience is not normally sufficient to resolve underspecified references. We have set the LIVE system's generation confidence interval to 0.5, as this is the midpoint of the visual salience range. Algorithm 4 lists the LIVE GRE algorithm.

**Algorithm 4** (*The LIVE algorithm for generating referring expressions*).

*Input:* A context set containing a target object and a distractor set; each element in the context set must have a visual salience score ascribed to it, and a confidence interval (Default 0.5).

*Output:* A description of the target object that excludes all the elements of the distractor set that have a visual salience score greater than the target object's visual salience minus a predefined confidence interval.

- (1) Sequentially iterate through the predefined list of preferred attributes (type, colour, size).
- (2) For each attribute create a set  $F$  containing the objects that have the attribute plus all the previously accepted attributes.
- (3) If the number of elements in  $F$  is less than the number of elements in the set created using the previously accepted attributes and the target object is an element of  $F$ , add the current attribute to the list of accepted attributes.
- (4) Terminate if (i) the target object is the only element of  $F$  or (ii) when the target object's visual saliency score exceeds the visual saliency scores of the other objects in  $F$  by the predefined confidence interval. Always include the target object's type in the set of accepted attributes.
- (5) If the end of the preferred attributes list is reached and the algorithm has not terminated, return fail.

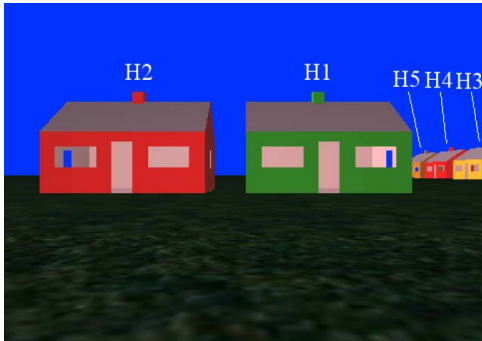


Fig. 20. The visual context.

H1	Green	1.0000
H2	Red	0.8646
H3	Yellow	0.0235
H4	Red	0.0111
H5	Brown	0.0149

Fig. 21. The LIVE system’s analysis of Fig. 20 listing the objects’ IDs, colour attributes and visual saliency scores.

Fig. 20 illustrates a scene from the LIVE system and Fig. 21 lists the LIVE visual saliency ranking of the objects in the scene. Note that in this scene all the objects have the same absolute height, width and depth values. It is the effect of perceiving the scene from a particular point of view as the user moves through the simulation that is captured by the visual saliency algorithm. Given this context, assuming that H2 is the target object, the system would begin trying to generate a description using the target type attribute: *house*. None of the objects in the scene are excluded by this description. Moreover, the target object’s visual saliency score does not exceed all the scores associated with the elements in this set by the predefined confidence interval. Indeed, H1’s visual saliency score exceeds that of the target object. As the first attribute in the preferred attribute list does not result in a distinguishing description the system tries to generate a description by using the first and second attributes in the list: type and colour. This results in the set of objects fulfilling the description *red house*. Only two objects in the scene fulfil this description: the target object H2 and one of the distractor objects H4. Furthermore, H2’s visual saliency rating exceeds H4’s by more than the predefined confidence interval (0.5). Consequently, the description *red house* is deemed to be a distinguishing description and the GRE algorithm terminates. It should be noted that in this instance Dale and Reiter’s [7] algorithm would fail to generate a description as it would not be able to distinguish between H2 and H4 using only adjectival and type attributes.

## 8. Evaluation

In the previous sections we have argued that computer systems processing visually situated discourse must make use of visual salience for both resolving and generating references. We have shown that the use of visual and linguistic salience allows the LIVE system to move beyond other current approaches in its ability to resolve linguistically underspecified references and to generate underspecified but sufficiently detailed referring expressions in contexts where visual salience is important.

Our claims about the role of visual salience in NLVR systems raise a number of questions. What evidence is there that visual salience is important in reference resolution? Does the LIVE system accurately mirror people's use of visual salience in reference resolution? To what extent does the system provide users with a natural and easy to use NL interface to its virtual environment? In this section we present 3 different evaluations of the LIVE system.

In Section 8.1 we describe an experiment examining the role of visual salience in reference resolution. In this experiment participants were shown a series of images, each paired with an underspecified linguistic description. Each description could refer to two different objects in the image. The competing reference objects differed in visual salience to a greater or lesser degree in different images. For each image-description pair, participants either marked the object in the image to which they felt the description referred, or else ticked a box indicating they felt the description was ambiguous. The results showed a reliable link between reference resolution and visual salience: the greater the difference in visual salience between the two competing reference objects in an image, the more likely participants were to resolve the description given with that scene as unambiguously referring to the most salient object. Conversely, the smaller the difference in visual salience between the competing reference objects, the more likely participants were to judge the description as ambiguous.

In Section 8.2 we describe a more specific evaluation of the LIVE system's account of visual salience and reference. In this evaluation the system was given the same images and underspecified object descriptions as in the experiment described in Section 8.1. For each image the system was asked to either resolve the given underspecified description, or to mark that description as ambiguous. The results showed a reliable correlation between the system's responses and participants' responses to the same images in the experiment reported in Section 8.1, both on average and at the individual participant level.

Finally, Section 8.3 describes an open ended interactive 'usability test' of the system, examining the system's ability to understand NL input used by participants when using the system.

### 8.1. *Evaluating the visual salience and ambiguity hypothesis*

A central hypothesis of the LIVE system is that visual salience is important in the resolution of underspecified linguistic references. To test this hypothesis, we developed a series of 20 images and paired each image with an object description. Each image was designed so that the corresponding object description matched at least two alternative *reference objects* in that image: each object description was underspecified. The set of 20 images were designed so that the competing reference objects differed in visual salience to a greater or lesser degree in different images. Each image also contained a number of other objects apart from the two competing reference objects. For example, Fig. 22 shows the image that was paired with the object description *the red tree*. In this image, there are two competing reference objects for the description *the red tree*: trees A and B. However, one of these (tree A) is more visually salient than the other (tree B). For simplicity, the LIVE system was used to create the images and to compute the differences in visual salience between the reference objects in each image.



Fig. 22. The experimental image paired with the object description *the red tree*.

### 8.1.1. Procedure

10 participants took part in the experiment. For each participant, a set of 10 image-description pairs were selected from the 20 generated, in such a way that each set of 10 covered the entire range of possible visual-salience differences. Each group of image-description pairs was presented to participants in a booklet, with an image and a description on each page. For each image-description pair, participants either marked the object in the image to which they felt the description referred, or else ticked a box indicating they felt the description was ambiguous. In addition, each booklet contained one image that was paired with an unambiguous, completely specified description (one for which there was only one possible referent in the image). This was provided as a baseline to ensure that participants had an example of an completely unambiguous description to which they could refer. Each booklet started with a page giving the participant the following instructions:

On each page in this document there is an image, a caption, and a box marked *ambiguous*. You are to mark the object in the image, by drawing an X on it, that you think is being described by the caption. If you are not sure which object is being described, tick the box labelled *ambiguous*.

### 8.1.2. Results

All participants correctly judged the fully-specified image-description pair as unambiguous, and selected the correct referent for that description. This indicated that they understood the task. When participants judged a description as unambiguous, they were asked to indicate, by marking a cross on the image, the object to which they thought that description referred. For all such unambiguous descriptions the participants selected the more visually salient of the competing reference objects in the image as the referent for the given description. The 10 image-description pairs seen by each participant were divided into two groups: the *high visual salience difference* group (the 5 images in which the competing reference objects had a large difference in visual salience) and the *low visual salience difference* group (the 5 images in which the competing reference objects had a small difference in visual salience). To compare the system's judgement with that of individual participants in the experiment, we examined each participant's responses for the

Table 1  
Average number of image-description pairs rated ambiguous and unambiguous by each participant

	Ambiguous	Unambiguous	SD
Low visual salience difference	4.4	0.6	1.26
High visual salience difference	1.3	3.7	1.06
Total	5.7	4.3	1.94

images the system judged ambiguous and for the images the system judged unambiguous. All 10 participants produced more *ambiguous* responses for the images the system judged to be ambiguous, and more *unambiguous* responses for the images the system judged to be unambiguous; 0 participants produced the opposite pattern ( $p < 0.01$ , sign test). Table 1 shows the average number of image-description pairs rated ambiguous and unambiguous by each participant in these two groups. Clearly, the difference in visual salience between competing reference objects has a strong influence on peoples' judgements of ambiguity for a linguistic description.

To analyse the relationship between degree of visual salience difference and degree of ambiguity, the 20 underspecified image-description pairs used in the experiment were ranked according to the percentage of participants who judged the given description as ambiguous: this was taken as a measure of the ambiguity of that description. This was compared with the degree of difference in visual salience between the two competing reference objects in each image. Values for visual salience for the competing objects in each image were computed using the LIVE system's visual salience algorithm. Table 2 shows the degree of ambiguity and the visual salience difference rank (from 20: highest difference in visual salience between competing objects, to 1, lowest difference in visual salience) for all 20 images. There was a significant correlation between degree of ambiguity and difference in visual salience ( $r = 0.80$ , %var = 0.65,  $p < 0.01$ ), indicating that the lower the difference in visual salience between the competing reference objects for a given description, the more likely participants were to judge that description as ambiguous. This supports the LIVE system's assumptions linking visual salience and linguistic ambiguity.

In 9 out of the 20 images used in the experiment, the less visually salient of the two competing reference objects was not fully contained in the image (the edge of the object was outside the image frame). Fig. 23 shows an example of one such image. These 9 images were primarily those with a high difference in visual salience between competing objects, and with low degrees of ambiguity (note (a) in Table 2). It could be that the relationship seen in Table 2 arises because, in images with a high difference in visual salience between competing objects, one of those objects is only partially in the image and so the other object is unambiguously the intended reference. To assess this possibility, we extracted and analysed the remaining 11 images in which both competing reference objects were fully within the image boundary. For these 11 images there was a significant correlation between degree of ambiguity and difference in visual salience ( $r = 0.70$ , %var = 0.48,  $p < 0.01$ ), showing that the observed relationship between ambiguity and difference in visual salience does not arise because in some images one of the competing reference objects is only partially shown. The correlation between ambiguity and difference in visual salience for these images is lower than that for the full set of 20 images because these 11

Table 2  
Relationship between degree of ambiguity and difference in visual salience

Image	Degree of ambiguity	Visual salience difference rank	Image	Degree of ambiguity	Visual salience difference rank
Image 2 <sup>a</sup>	0%	20	Image 14 <sup>a</sup>	80%	8
Image 3 <sup>a</sup>	0%	17	Image 13	80%	7
Image 7 <sup>a</sup>	0%	14	Image 15	80%	6
Image 1 <sup>a</sup>	20%	19	Image 16 <sup>a</sup>	80%	5
Image 6	20%	16	Image 17	80%	3
Image 5 <sup>a</sup>	20%	15	Image 10	100%	12
Image 9 <sup>a</sup>	20%	11	Image 12	100%	9
Image 4 <sup>a</sup>	40%	18	Image 18	100%	4
Image 8	40%	13	Image 19	100%	2
Image 11	80%	10	Image 20	100%	1

<sup>a</sup> Images where the less visually salient of the two competing reference objects were not fully within the image frame.



Fig. 23. An experiment image where the distractor object is not fully in the image. The object description matched with this image was: *the tall tree*.

images cover a more limited range of visual salience difference values (all have a visual salience difference rank of 13 or less). Statistically speaking, the correlation coefficient is sensitive to the ‘range of talent’ (variability) in variables being measured: the smaller the range of talent, the lower the absolute value of the coefficient, other things being equal.

## 8.2. Applying the LIVE system to the experimental data

To evaluate the LIVE system’s account of linguistic reference resolution mediated by visual salience, the system was given the same 20 images and underspecified object descriptions as in the experiment involving human participants described in Section 8.1. The system was then asked to resolve the underspecified object descriptions. In attempting to resolve an underspecified linguistic reference, the system has two options: either to identify the most visually salient object in the current scene that matches that reference, or (if the difference between the most visually salient and the next most visually salient matching object is too small), to identify the description as ambiguous and ask the user for more in-



Table 3  
Participants' classification of descriptions as ambiguous and unambiguous, compared with LIVE system's classification

	50% or more of participants judged descriptions	
	Ambiguous	Unambiguous
LIVE judged descriptions ambiguous	10 <sup>a</sup>	0
LIVE judged descriptions unambiguous	1	9 <sup>b</sup>

<sup>a</sup> For all 10 images, 80% or more of participants judged descriptions as ambiguous.

<sup>b</sup> For 7 out of 9 images, 80% or more of participants judged descriptions as unambiguous.

formation. The system makes use of a *confidence interval* parameter in deciding whether a given linguistic reference is ambiguous or not: if the difference between the most visually salient and the next most visually salient matching object is greater than this confidence interval, the system takes the most visually salient matching object to be the referent of the current description; otherwise the system takes the description as ambiguous, cf. Section 6.3.

To apply the system to the experimental input data, we must select a value for the confidence interval parameter. Since the visual salience values computed by the system range from 0 (minimum salience: object not present) to 1 (maximum visual salience), differences in visual salience between objects also fall between 0 and 1. We therefore chose a value of 0.5 (the midpoint of this range) for the confidence interval when applying the system to the experimental data. This means that the system will treat a description with two competing reference objects whose difference in visual salience is less than 0.5 as ambiguous. With this confidence interval, the system judged 10 of the 20 image-description pairs to contain ambiguous descriptions; the remaining 10 pairs were judged to contain descriptions that unambiguously referred to the most visually-salient matching object. All 10 participants produced more *ambiguous* ratings for images which the LIVE system judged to be ambiguous than for images which the system judged to be unambiguous; 0 participants produced the opposite pattern ( $p < 0.01$ , sign test). In all cases in which participants judged a description to be unambiguous, they (like the system) selected the most visually-salient matching object as referent. Table 3 compares the LIVE system's classification of descriptions as ambiguous or unambiguous with the human participants' classification of these descriptions. This table shows that in all but 1 of the 20 image-description pairs used in the experiment, both the system's judgements of ambiguity for descriptions, and its selection of a referent for unambiguous descriptions, matched those of the human experiment participants.

The above analysis applies the system to the overall results of the experiment described in Section 8.1. A second worthwhile analysis is to compare the system's pattern of responses to image-description pairs in the experiment with the pattern of responses produced by *individual* participants in the experiment. In any experiment, different participants will have different biases that influence their responses. In the current experiment, for example, there were two participants who judged 8 out of 10 image-description pairs to be ambiguous: compared to other participants, those two were biased towards identifying descriptions as ambiguous. There was one participant who only judged 1 of the 10 image-description pairs to be ambiguous: compared to other participants, that participant was

Table 4  
Match between individual participant's responses and LIVE system's responses (with different confidence interval values)

	Participant judged		Confidence interval	LIVE system matched judgement		
	Ambiguous	Unambiguous		Ambiguous	Unambiguous	Errors
Participant 1	4	6	0.6	4	6	0
Participant 2	5	5	0.5	5	5	0
Participant 3	5	5	0.6	4	5	1
Participant 4	4	6	0.5	4	5	1
Participant 5	2	8	0.65	2	6	2
Participant 6	2	8	0.6	2	6	2
Participant 7	4	6	0.6	4	6	0
Participant 8	1	9	0.1	1	9	0
Participant 9	4	6	0.5	4	5	1
Participant 10	4	6	0.7	3	6	1

The 'ambiguous' and 'unambiguous' columns under the 'LIVE system' heading represent the number of descriptions judged (un)ambiguous by a given participant that were also judged (un)ambiguous by the system. In all cases where both the system and a participant selected an unambiguous referent, both selected the more visually salient matching object.

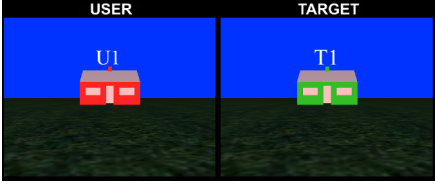
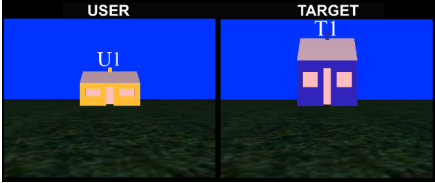
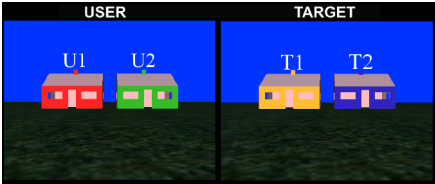
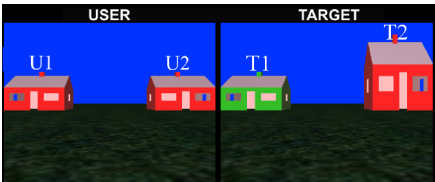

biased against identifying descriptions as ambiguous. To assess the LIVE system's ability to account for the experimental results at the individual participant level, the system's responses to image-description pairs were compared with each individual participant's responses to those pairs, and a different value for the system's confidence interval parameter was selected for each. Table 4 shows the number of ambiguous and unambiguous responses from each participant, the confidence interval at which the LIVE system best matched that participant's responses, the number of descriptions judged ambiguous(unambiguous) by that participant that were also judged ambiguous (unambiguous) by the system, and the number of times the system's response did not match that participant's response (the number of errors). As can be seen, the number of errors produced by the system was low (one error or no errors for most participants).

These results show that the LIVE system's use of visual salience to resolve under-specified references closely matches how participants resolve such references, both at an aggregate and at an individual participant level. In the next section we describe a further evaluation of the system, this time focusing on the system's ability to handle different types of reference.

### 8.3. Interactive evaluation of the system's ability to interpret different forms of reference

In order to test the system's ability to interpret different forms of reference we designed 5 interactive tasks. Each task consisted of a *user* scene and a *target* scene. 5 human participants were asked to interact with the system online and in real-time to change the user scene into the target scene. Each task was designed to elicit the use of particular forms of reference from the participants, by presenting situations and tasks that provided an opportunity to use those forms. Table 5 lists the visual context provided for each task and the target interactions that we hoped the task would elicit from the participants.

Table 5  
The visual contexts and target interactions for each of the interactive tasks

	<p>Task 1. Target interaction type: full definite noun phrases. Example: <i>make the house green</i>. U1 = red, T1 = green.</p>
	<p>Task 2. Target interaction type: full definite noun phrases and anaphoric pronominal references. Example: <i>make the house taller; make it blue</i>. U1 = yellow, T1 = blue + tall.</p>
	<p>Task 3. Target interaction types: full definite noun phrase, spatial locatives and other anaphora. Example: <i>make the [red house   house on the left] yellow; make the [other one   house on the right] blue</i>. U1 = red, U2 = green, T1 = yellow, T2 = blue.</p>
	<p>Task 4. Target interaction types: spatial locatives and other anaphora. Example: <i>make the house on the left green; make the other one taller</i>. U1, U2 = red, T1 = green, T2 = red + tall.</p>
	<p>Task 5. Target interaction types: underspecified full definite noun phrases and spatial locatives. Example: <i>make the [house in the middle   blue house] taller</i>. U1, U3, T1 = blue, U2, T2 = green, U4, T4 = red, U5, T5 = yellow, T3 = blue + tall.</p>

The participants were given no prior training with the system and, not surprisingly, the analysis of the logs revealed that the system was not able to parse all the participant inputs. There were two main causes for this: (1) unknown words, participants sometimes misspelt words or used words outside the system's vocabulary (for example, *facade*, *please*<sup>13</sup> or unknown verbs); (2) participants used sentence structures the parser could not process.

<sup>13</sup> One participant politely entered *please make the house greener* to which the system responded: *Sorry I don't understand the word "please", please reenter the sentence.*

Table 6  
The analysis of the interactive task logs

Participant		1	2	3	4	5
Task 1	Number of inputs	2	2	2	8	8
	Unknown words	0	1	1	4	5
	Unknown structure	1	0	0	3	2
	Ambiguous	0	0	0	0	0
	Reference type	A	A	A	B	A
Task 2	Number of inputs	2	2	3	6	3
	Unknown words	0	0	0	2	0
	Unknown structure	0	0	1	2	1
	Ambiguous	0	0	0	0	0
	Reference type	A,A	A,A	A,A	B,B	A,A
Task 3	Number of inputs	3	6	4	2	5
	Unknown words	0	2	0	0	0
	Unknown structure	1	1	2	0	3
	Ambiguous	0	1	0	0	0
	Reference type	C,C	B,B	C,C	B,B	B,C
Task 4	Number of inputs	2	2	2	5	3
	Unknown words	0	0	0	0	1
	Unknown structure	0	0	0	2	0
	Ambiguous	0	0	0	1	0
	Reference type	C,C	C,E	C,C	C,C	C,B
Task 5	Number of inputs	1	3	1	1	3
	Unknown words	0	0	0	0	1
	Unknown structure	0	2	0	0	1
	Ambiguous	0	0	0	0	0
	Reference type	D	C	C	B	B

Key for Reference Types: A = the + noun, e.g., *the house*; B = the + adjective + noun, e.g., *the red house*; C = the + noun + spatial locative, e.g., *the house on the left*; D = the + adjective + noun + spatial locative, e.g., *the red house on the left*; E = *the other house*.

When the system was unable to parse an input it output a message stating why it could not parse the input and requested the participant to enter another command.

Table 6 shows an analysis of the logs for each subject in each task. For each participant and task it lists: (1) the number of inputs used to complete each task, (2) the number of inputs that were not processed because of an unknown word, (3) the number of inputs that were not processed because the system could not parse the structure of the input, (4) the number of parsed inputs that the system deemed to be ambiguous, and (5) the types of reference used in the inputs that were parsed and not deemed to be ambiguous by the system. The reference types are listed from left to right in the order they were processed.

The analysis of the logs revealed that none of the participants used pronominal references at all. Instead, in the anaphoric task (Task 2) participants used anaphoric definite references, for example the input sequence: *make the house blue; make the house taller*. The analysis of Task 3 revealed that participants were equally likely to use colour based or spatial location based references. Both Tasks 3 and 4 were designed to elicit other-anaphoric references. However, there was only one use of this form of reference, participant

2 Task 4: *make the house on the right taller; make the other house green*. There were two underspecified definite descriptions used in Task 5.

On two occasions the system was able to parse an input but deemed the reference to be ambiguous: participant 2 in Task 2 input *make the house yellow* and participant 4 in Task 4 input *make the red house taller*. In both cases there had been no prior inputs successfully parsed and consequently an anaphoric interpretation of the references was not possible. On these occasions the system output a message that it could not resolve the reference because of ambiguity and asked the user to enter another input. All the other inputs that were parsed were correctly resolved.

Table 6 shows that the system's performance was disrupted when users gave the system words it did not know, or used grammatical structures the system did not recognise. In future work we will address this issue by extending the system's lexical and grammatical coverage. However, the results also illustrate the system's ability to resolve both anaphoric and exophoric references. Across the 5 tasks in experiment 3, the system successfully resolved: (1) exophoric definite descriptions, including linguistically underspecified definite descriptions; (2) anaphoric definite descriptions; (3) exophoric spatial locatives; and (4) other anaphoric references. The system's ability to resolve both anaphoric and exophoric references arises from its dynamic updating and interrelating of visual and linguistic discourse context representations within its context model.

## 9. Conclusions

The fundamental claim of this paper is that salience—both visual and linguistic—is an important overarching semantic category structuring visually situated discourse context representation. Computer systems attempting to process a visually situated discourse should integrate a model of visual salience within their NLP framework. Furthermore, the paper highlights the need for such systems to dynamically update and interrelate the visual and linguistic contexts if the system is going to handle exophoric and anaphoric references. The benefits of this approach are that the visual saliency scores associated with objects entering the context model through the visual context can, in many instances, be used to interpret underspecified exophoric references and to generate underspecified but sufficiently detailed references in contexts where previous GRE algorithms that do not accommodate visual saliency information would fail.

To support our claim we have designed, implemented and evaluated an NLP framework for a real-time NLVR system. One component of this framework is a DC model that integrates visual salience information with a model of the evolving linguistic context. There are several design features of the DC model that we feel are key to its ability to dynamically model and integrate the visual and linguistic contexts:

- (1) The division of the DC model into separate stacks of dynamically created local contexts means that the model can be updated as a result of both linguistic and visual discourse events. Using a stack architecture means that these local contexts are chronologically ordered. Consequently, the context model accommodates the effect of recency on salience.

- (2) The internal structure of the reference domains facilitates the integration of visual and linguistic semantic information. The reference domain partitions allow the objects in the domain to be grouped based on their type, attributes or attribute values. This form of domain decomposition accommodates some of the linguistic semantic categories that are of particular importance for reference resolution. The ability to order the elements within these partitions based on their visual salience accommodates some of the visual semantic categories impacting on reference resolution.
- (3) The use of a generic data structure to model discourse events from both modalities facilitates the interaction between these contexts. Moreover, the restructuring and inserting of reference domains from one modality's context model to another allows the LIVE system to update and interrelate events in the different modalities. For example, the perceptual event of seeing objects A and B on the screen introduces elements representing these objects into a reference domain in the VDL. The fact that this reference domain may be restructured and inserted into the LDL as a result of the interpretation of an exophoric reference to object A in the reference domain means that the element representing object B may also be introduced into the LDL even though there has been no prior linguistic reference to the object. This approach allows the system to resolve other-anaphoric references to objects, such as B, even though they have not been referred to in the linguistic discourse.

Using this DC model we showed how visual saliency data can be used to underpin two of the fundamental sub-tasks of NLP: reference resolution and GRE. Finally, system evaluation results were presented. The evaluation confirmed the proposed relationship between underspecified reference and visual salience and the LIVE system's interpretation of different forms of reference.

In conclusion, the focus of the LIVE framework has been to integrate visual context within a situated natural language processing framework. The main contributions of this work are illustrating the fundamental role of visual salience in situated discourse, the definition of a real-time model of visual salience, and the integration of this visual saliency information within discourse modelling, and reference resolution and generation. The main direction of future work will be to extend the components of the framework. One area of this work will involve the development of a more sophisticated management policy for the LDL stack. Currently, the LDL models *linear recency* of mention. However, its stack based representation lends itself to the integration of a hierarchical discourse structure such as [29]. Another area of future work is the LIVE reference resolution algorithm. Currently, it uses heuristic rules to classify an utterance as anaphoric or exophoric. As with all heuristics they do not cover all cases. One way of addressing this issue would be to adopt an approach to reference resolution that does not require this distinction to be explicitly made within the resolution process. For example, given a referring expression, one could compute for each candidate referent an overall discourse saliency using a weighted sum of the saliency of the candidates in the VDL and LDL. The candidate with the highest overall rating could then be selected as the referent. The weightings used in the integration should be dependent on the type of reference being resolved, for example, the weights associated with a pronominal reference should strongly preference an LDL interpretation. The LIVE

reference generation algorithm should also be extended to incorporate the insights from the other generation algorithms that extend the Incremental Algorithm [7].

Although our approach has been developed with simulated 3D environments, our main findings carry over to the interaction of artificial systems with sensor data derived from physical environments. To ground natural language interpretation and generation in the real world, systems need to model, update and interrelate both linguistic and visual context.

## References

- [1] M. Tanenhaus, M. Spivey-Knowlton, K. Eberhard, J. Sedivy, Integration of visual and linguistic information during language comprehension, *Science* 268 (1995) 1632–1634.
- [2] I. Duwe, H. Strohner, Towards a cognitive model of linguistic reference, Report: 97/1, *Situierte Künstliche Kommunikatoren 97/1*, Universität Bielefeld, 1997.
- [3] M. Spivey-Knowlton, M. Tanenhaus, K. Eberhard, J. Sedivy, Integration of visuospatial and linguistic information: Language comprehension in real time and real space, in: P. Olivier, K. Gapp (Eds.), *Representation and Processing of Spatial Expressions*, Lawrence Erlbaum Associates, 1998, pp. 201–214.
- [4] D. Byron, 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, 2003.
- [5] A. Kehler, Discourse, in: D. Jurafsky, J. Martin (Eds.), *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, international ed., Prentice Hall, Englewood Cliffs, NJ, 2000, pp. 669–718.
- [6] K. Linden, Natural language generation, in: D. Jurafsky, J. Martin (Eds.), *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, international ed., Prentice Hall, Englewood Cliffs, NJ, 2000, pp. 763–796.
- [7] R. Dale, E. Reiter, Computational interpretations of the Gricean maxims in the generation of referring expressions, *Cognitive Science* 19 (2) (1995) 233–263.
- [8] J. McCawley, *Everything That Linguists Have Always Wanted To Know About Logic\** (but were ashamed to ask), second ed., University of Chicago Press, 1993.
- [9] H. Grice, *Studies in the Way of Words*, Harvard University Press, 1989.
- [10] T. Pechmann, Incremental speech production and referential overspecification, *Linguistics* 27 (1989) 98–110.
- [11] T. Winograd, A procedural model of language understanding, in: R. Schank, K. Colby (Eds.), *Computer Models of Thought and Language*, W.H. Freeman and Company, New York, 1973, pp. 152–186.
- [12] P. McKeivitt (Ed.), *Integration of Natural Language and Vision Processing (vols. I–IV)*, Kluwer Academic, Dordrecht, 1995/1996.
- [13] M. Maybury, W. Wahlster (Eds.), *Readings in Intelligent User Interfaces*, Morgan Kaufman, San Francisco, CA, 1998.
- [14] S. Oviatt, P. Cohen, L. Wu, J. Vergo, L. Duncan, B. Suhm, J. Bers, T. Holzman, T. Winograd, J. Landay, J. Larson, D. Ferro, Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions, *Human Computer Interaction* 15 (4) (2000) 263–322.
- [15] L. Kievit, P. Piwek, R. Beun, H. Bunt, Multimodal cooperative resolution of referential expressions in the DENK system, in: H. Bunt, R. Beun (Eds.), *Cooperative Multimodal Communication*, in: *Lecture Notes in Artificial Intelligence*, vol. 2155, Springer, Berlin, 2001, pp. 197–214.
- [16] T. Jording, I. Wachsmuth, An anthropomorphic agent for the use of spatial language, in: K. Coventry, P. Olivier (Eds.), *Spatial Language: Cognitive and Computational Aspects*, Kluwer Academic, Dordrecht, 2002, pp. 69–86.
- [17] P. Gorniak, D. Roy, Grounded semantic composition for visual scenes, *J. Artificial Intelligence Res.* 21 (2004) 429–470.
- [18] F. Landragin, N. Bellalelem, 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.

- [19] R. Forgas, L. Melamed, *Perception: A Cognitive Stage Approach*, McGraw-Hill, New York, 1976.
- [20] M. Chum, J. Wolfe, Visual attention, in: E.B. Goldstein (Ed.), *Blackwell Handbook of Perception*, in: *Handbooks of Experimental Psychology*, Blackwell, 2001, pp. 272–310, Chapter 9.
- [21] M.I. Posner, C.R. Snyder, B.J. Davidson, Attention and the detection of signals, *J. Experimental Psychol. Gen.* 109 (2) (1980) 160–174.
- [22] C. Koch, L. Itti, Computational modelling of visual attention, *Nature Rev. Neurosci.* 2 (3) (2001) 194–203.
- [23] D. Heinke, G. Humphreys, Computational models of visual selective attention: A review, in: G. Houghton (Ed.), *Connectionist Models in Psychology*, Psychology Press, 2004.
- [24] X. Tu, D. Terzopoulos, Artificial fishes: Physics, locomotion, perception, behaviour, in: *Proceedings of ACM SIGGRAPH*, Orlando, FL, 1994, pp. 43–50.
- [25] H. Noser, O. Renault, D. Thalmann, N. Magnenat-Thalmann, Navigation for digital actors based on synthetic vision, memory and learning, *Computer Graphics* 19 (1) (1995) 7–9.
- [26] J. Kuffner, J. Latombe, Fast synthetic vision, memory, and learning models for virtual humans, in: *Proceedings of Computer Animation Conference (CA-99)*, IEEE Computer Society, Geneva, 1999, pp. 118–127.
- [27] C. Peter, C. O’Sullivan, A memory model for autonomous virtual humans, in: *Proceedings of Eurographics Irish Chapter Workshop (EGIreland-02)*, Dublin, 2002, pp. 21–26.
- [28] H. Kamp, U. Reyle, *From Discourse to Logic*, Kluwer Academic, Dordrecht, 1993.
- [29] B. Grosz, C. Sidner, Attention, intentions, and the structure of discourse, *Computational Linguistics* 12 (3) (1986) 175–204.
- [30] J. Kelleher, J. van Genabith, A computational model of the referential semantics of projective prepositions, in: P. Saint-Dizier (Ed.), *Computational Linguistics Dimensions of the Syntax and Semantics of Prepositions*, Kluwer Academic, Dordrecht, 2005, pp. 200–215.
- [31] J. Hobbs, On the coherence and structure of discourse, Technical Report CSLI-85-37, Center for the Study of Language and Information, 1985.
- [32] W. Mann, S. Thompson, Rhetorical structure theory: Description and construction of text structures, in: G. Kempen (Ed.), *Natural Language Generation: New Results in Artificial Intelligence, Psychology and Linguistics*, Nijhoff, Dordrecht, 1987, pp. 83–96.
- [33] M. Walker, Limited attention and discourse structure, *Computational Linguistics* 22 (2) (1996).
- [34] R. Langacker, *Foundations of Cognitive Grammar: Theoretical Prerequisites*, vol. 1, Stanford University Press, Stanford, CA, 1987.
- [35] S. Salmon-Alt, L. Romary, Reference resolution within the framework of cognitive grammar, in: *Proceedings of the Seventh International Colloquium on Cognitive Science (ICCS-01)*, Donostia, Spain, 2001, pp. 284–299.
- [36] M. Pinkal, Definite noun phrases and the semantics of discourse, in: *Proceedings of the 11th International Conference on Computational Linguistics (COLING 86)*, Bonn, 1986, pp. pp. 368–373.
- [37] M. Poesio, *Discourse interpretation and the scope of operators*, Ph.D. dissertation, University of Rochester, 1994.
- [38] D.L. Bean, E. Riloff, Corpus-based identification of non-anaphoric noun phrases, in: *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics (ACL-99)*, University of Maryland, 1999, pp. 373–380.
- [39] B. Grosz, A. Joshi, W. Weinstein, Centering: A framework for modelling local coherence of discourse, *Computational Linguistics* 21 (2) (1995) 203–255.
- [40] D. Appelt, Planning English referring expressions, *Artificial Intelligence* 26 (1) (1985) 1–33.
- [41] R. Dale, *Generating Referring Expressions: Building Descriptions in a Domain of Objects and Processes*, MIT Press, Cambridge, MA, 1992.
- [42] W. Claassen, Generating referring expressions in a multimodal environment, in: R. Dale, E. Hovy, D. Rosner, O. Stock (Eds.), *Aspects of Automated Natural Language Generation*, Springer, Berlin, 1992.
- [43] J. Lester, J. Voerman, S. Towns, C. Callaway, Deictic believability: Coordinating gesture, locomotion, and speech in lifelike pedagogical agents, *Appl. Artificial Intelligence* 13 (4–5) (1999) 383–414.
- [44] K. van Deemter, Generating vague descriptions, in: *Proc. First International Conference on Natural Language Generation (INLG-00)*, Mitzpe Ramon, Israel, 2000.
- [45] M. Stone, On identifying sets, in: *Proc. International Conference on Natural Language Generation (INLG-00)*, Mitzpe Ramon, Israel, 2000.



- [46] E. Krahmer, M. Theune, Efficient context-sensitive generation of referring expressions, in: K. van Deemter, R. Kibble (Eds.), *Information Sharing: Reference and Presupposition in Language Generation and Interpretation*, CLSI Publications, Stanford, CA, 2002.
- [47] C. Gardent, Generating minimal definite descriptions, in: *Proceedings of the 40th International Conference of the Association of Computational Linguistics (ACL-02)*, Philadelphia, 2002, pp. 96–103.
- [48] K. van Deemter, Generating referring expressions: Boolean extensions of the incremental algorithm, *Computational Linguistics* 28 (1) (2002) 37–52.
- [49] E. Krahmer, I. van der Sluis, A new model for generating multimodal referring expressions, in: *9th European Workshop on Natural Language Generation (ENLG-03)*, Budapest, Hungary, 2003, pp. 47–57.
- [50] M. Theune, *From data to speech: Language generation in context*, Ph.D., Eindhoven University of Technology, 2000.
- [51] E. Hajicová, *Issues of sentence structure and discourse patterns*, in: *Theoretical and Computational Linguistics*, vol. 2, Charles University, Prague, 1993.