

Prototyping Expressive 3D Social Worlds

David John^{*1}, Christos Gatzidis¹, Anthony Boucouvalas², Fotis Liarokapis³ and Vesna Brujic-Okretic⁴

¹School of Design, Engineering & Computing, Bournemouth University, Poole, UK; ²Faculty of Science and Technology, University Of Peloponese, Tripolis, Greece; ³Faculty of Engineering and Computing, Coventry University, Coventry, UK; ⁴Department of Information Science, City University, London, UK

Abstract: 3D virtual worlds are increasingly popular arenas for social interaction. There are new opportunities and possibilities for the style of communication but important aspects present in face-to-face meetings are absent in the artificial environment, including the visual cues of emotion that are provided by facial expressions, and a realistic representation of one's geographical location. Our research attempts to enhance social interaction within virtual worlds by proposing a framework that would enable fully expressive Internet communication with the use of 3D expressive models. The framework contains three separate systems that support different aspects of social interaction within virtual worlds. Firstly, the Virtual City Maker creates a believable environment by automatically creating realistic identifiable geo-referenced 3D environments from a variety of aerial and GIS image data. Secondly, an automated 3D head modelling system provides a mechanism for generating and displaying expressions. Thirdly, the Emotion Analyser provides a mechanism for triggering the display of appropriate expressions by automatically identifying emotional words contained in text messages, the person to whom the emotional words refer to and the intensity. Each system in the framework is discussed with reference to their contribution to enhancing social interaction within 3D social worlds.

Keywords: Virtual Worlds, social interaction, urban modelling, facial animation, emotion extraction.

INTRODUCTION

The increase in popularity of social networking sites and 3D virtual worlds has enabled people to interact socially in new and diverse ways. With the development of virtual world technologies participants can explore more realistic 3D environments and interact with each other or artificial inhabitants of the environment using increasingly sophisticated avatars. Social interaction within the virtual world is a different experience for participants compared to face-to-face meetings. Virtual environments offer new opportunities and possibilities for user interaction but there are important aspects that are absent including the visual cues of emotion that are provided by facial expressions, and the shared experience of the participants geographical or urban location.

Our work focuses on enhancing social interaction within 3D virtual worlds, where the main method of communication is text messages that are passed between participants. Text has traditionally been the standard medium of communication between members of virtual worlds, and is still the dominant medium of communication. Written text sentences are rich in conveying emotional information, but people have to be focused to read and comprehend the text. Text is a familiar mode, but without any visual cues there is an increased need for remote participants in the conversation to make abstract judgements. Interpretation of the content is

aided by considering it in the appropriate context. This includes the interaction of the parties with the location as well as facial expressions, body language, voice intonation and language. Context within virtual worlds can be derived from the shared interactive location and enabling the avatars to present appropriate facial expressions.

One aim of 3D social world research has been to make the social interaction between the inhabitants of virtual worlds more comparable with face-to-face meetings, by making the avatar and their environment more believable [1, 2]. This does not necessarily mean virtual worlds should be made as close to the real world as possible, as making the avatar more photographically realistic does not increase its social presence [3]. Instead, other techniques could be used to make the interaction more realistic, such as enhancing the presentation of avatar, automating facial expressions, and placing the interaction within an authentic and recognizable location.

The important role that emotion plays in human-computer interaction and user interactions with synthetic agents has been recognised by many researchers of human-computer interfaces and synthetic agents including, Bartneck [4], Bianchi-Berthouze and Lisetti [5], Kiesler *et al.* [6], Koda [7], Paiva [8], Parkinson [9] and Picard [10]. In face-to-face interactions emotions are integral to the interpersonal processes that help develop mutual rapport between the communicators. However, when communicating using text through a computer network, the non-verbal cues are not present which can result in misinterpretation or extreme emotional reactions [9]. Without the visual cues, or the op-

*Address correspondence to this author at the School of Design, Engineering & Computing, Bournemouth University, Poole, UK; Tel: +44 (0)1202 9665078; Fax: +44 (0)1202 9665314 E-mail: djohn@bournemouth.ac.uk

eration of social normative influences that occur in the presence of other human beings, users are more likely to be uninhibited and aggressive compared to their behaviour in face-to-face meetings [6]. Various approaches can be followed to provide some expressive visual clues that are present in face-to-face interaction. For example, the explicit use of textual symbols, e.g. :-) and emoticons can provide a purely textual communication with a sense of emotion expression, however, these do not indicate intensity or the full range of emotions expressed in the whole message.

Another approach for communicating emotion can be through the use of avatars. An avatar is “an incarnation in human form or an embodiment (as of a concept or philosophy) often in a person” [11]. In the context of computer mediated communication, software avatars are often referred to as artificial intelligence agents that computer users are willing to perceive as believable or life-like. Prendinger and Mitsuru [12] highlighted the importance to avatars of demonstrating intelligent life-like behaviour to make them believable in social computing, while Hayes-Roth [13] advocated seven qualities that inferred life-likeness. In order to make an avatar more believable, characters should seem conversational, intelligent, individual, and should respond to individual users in a unique manner and show emotional responses [2, 8, 13].

Most avatar software systems have embedded functions to calculate the avatar’s emotions and present facial expressions visually, and various methods have been developed. Prendinger *et al.* [14, 15] have developed MPML and SCREAM as a Mark-up Language used to script emotionally and socially appropriate responses for animated characters. Andre *et al.* [1, 16] enhanced the presentation of information through the automated interaction of several life-like characters. A key aspect of making the dialogue believable is the design of distinctive emotional dispositions for each character. Johnson *et al.* [17] associate appropriate emotive behaviour with animated pedagogical agents to reinforce feedback during the learners’ problem-solving activities. Gratch *et al.* [18] identified the three key issues for creating virtual humans as face-to-face conversation, emotions and personality, and human figure animation. They make the distinction between communication-driven approaches that use deliberately conveyed emotion, as opposed to simulation-based approaches that aim to simulate “true” emotion based on appraisal theories such as the Ortony, Clore, and Collins’ cognitive appraisal theory (the OCC model) [19]. Thalmann *et al.* [20] presented an approach to the implementation of autonomous virtual actors in virtual worlds based on perception and virtual sensors.

Various synthetic facial models close to real human faces have been created. However, the use of realistic facial models for Internet communication requires significant computation time and large amounts of data transfer, and most currently existing avatars have a limited number of predefined emotion expressions. Cassell *et al.* [21, 22] have developed a framework for designing automated response for animated characters who participate in conversational interaction with human users, with REA as an example embodied conversational agent that attempts to have the same properties as humans in face-to-face conversation. Jack was developed as a 3D interactive environment for controlling articulated figures

at the Center for Human Modeling and Simulation [23]. It features a detailed human model and includes a range of realistic behavioural controls. Badler *et al.* [24] have a similar goal of making interaction with embodied characters the same as with live individuals with their Parameterized Action Representation system (PAR) that is designed to bridge the gap between natural language instructions and the virtual agents who are to carry them out in real-time. Conati [25] uses a Dynamic Decision Network (DDN) to model the probabilistic dependencies between causes, effects and emotional states to generate consistent believable responses for the virtual agents. De Rosis *et al.* [26] have created an XDM-Agent (Context-Sensitive Dialog Modeling) that simulates behaviour in different contexts and making semi-automatic pre-empirical evaluations of consistency and complexity. Hayes-Roth has used embodied characters in virtual theatres and interactive learning environments. In the Virtual Theatre Project [27] synthetic actors improvise their behaviour based on the directions they receive and the context. In the Animated Agent project [13, 28] Smart Interactive Characters maintain credibility by having distinctive personas, well-defined functional roles, multi-dimensional mood dynamics, natural language conversation, social and learning skills, and adaptation.

From the above there are three main questions for supporting social interaction in virtual worlds:

1. How to create believable environments
2. How to create believable avatars
3. How to trigger the display of appropriate expressions

In this paper we propose a framework [29] that could contribute to answering these three questions and enhance social interaction within virtual worlds; the Virtual City Maker to create realistic geo-referenced 3D environments, a 3D facial animation system to display facial expressions, and the Emotion Analyser to automatically identify emotions in text messages.

The question about how to create believable environments is answered by the automatic creation of identifiable locations.

The question about the creation of believable avatars is answered by the development of an automated 3D head modelling system for the display of expressions.

The question of how to trigger the display of appropriate expressions is answered by a mechanism for automating the identification of emotion contained in text messages.

ARCHITECTURE

The proposed framework that would enable fully expressive Internet communication with the use of 3D expressive models for is shown in Fig. (1).

Individual users of the system will be represented by realistic 3D facial models placed within a 3D virtual world that is dynamically constructed to represent their actual geographic location. The 3D environment is constructed by the Virtual City Maker software [30]. Users enter their geographical location (e.g. postcode) and a 3D representation of the outside surroundings is automatically generated. The 3D model is then passed on to the virtual world.

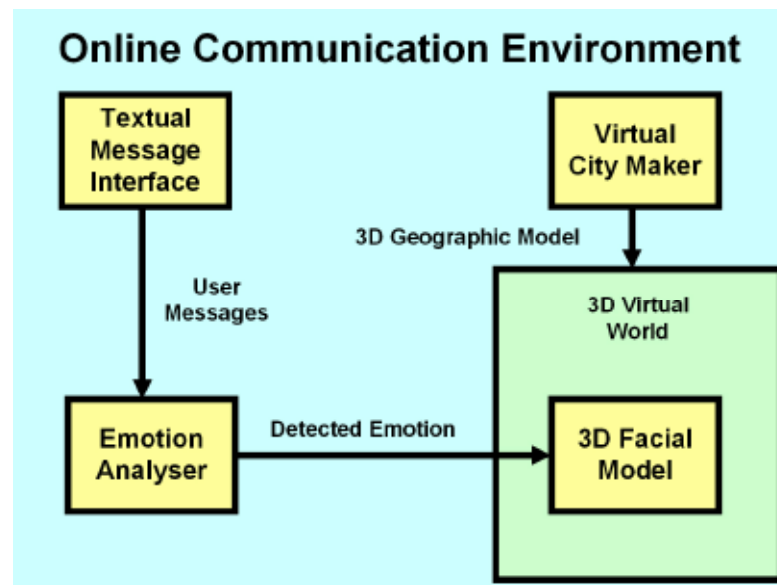


Fig. (1). Proposed architecture.

Users of the system will be able to pass textual messages to each other using a textual message interface. Individual user messages will be parsed by the Emotion Analyser [31, 32] to identify the emotional content. Any relevant detected emotions will be passed into the 3D facial model to automatically display the current emotion of the user.

The remainder of this paper will describe each of the three systems in turn; firstly the Virtual City Maker, secondly the automated 3D head modelling system and thirdly, the Emotion Analyser, concluding with a discussion of the research issues.

DEVELOPING AN URBAN-AWARE 3D VIRTUAL CITY ENVIRONMENT

The question of how to create believable environments is answered by the automatic creation of locations that are identifiable by the users. Although comprehensive procedural virtual city-generation software exists today such as the one presented by Mueller *et al.* [33], producing 3D urban content in an accurate and geo-referenced manner, which can also be integrated in a larger framework like the one presented in this paper, remains an issue. To address this, we propose the Virtual City Maker modelling tool, a system comprising of *two modes*; the *automatic* and *semi-automatic*.

Overview Of The Virtual City Maker System

First of all, regarding the input data used, the authors of this paper have selected a combination of aerial photographs and also 2D ground maps, at least for the outline of the buildings modelled. This more hybrid approach combines the strengths of the two different techniques for a more efficient result.

Aerial images ensure accurate results and the bypassing (to a certain extent at least) of a generalization process, while ground maps provide positional and geographic information. Delivering accurate location information in 3D is crucial to location-aware social interaction. Therefore, geo-referenced results and accuracy are key issues with errors in geometry

placement only afforded to be marginal making this hybrid data use approach even more justified.

Having placed building footprint outlines in a precise manner, these outlines have been extruded to real world heights using, if needed, GIS vector data in the form of the .shp file format (Fig. 2). This particular file format operates as a combination of CAD data with several added metadata attributes.

Using this particular data is a key part of our system, since it can not only compliment the aerial image information but also solve many inconsistencies and thus produce output of far greater quality compared to other systems (particularly regarding geo-references). This is because not only does it provide structure outlines for individual buildings but other 2.5D information, such as different heights, shape values, age values, type information and even land use.

The Automatic Mode

The key method in the automatic reconstruction mode of our proposed system is the detection of building outlines and roofs from aerial images and/or 2D ground maps. The second other important step is the extrusion to real-world heights and that is handled, when needed, by a parser developed to read the GIS .shp files. The main processes involved in the building and roof detection system are illustrated in Fig. (3), with the implementation based on the pioneering algorithmic work in this area conducted by Lin *et al.* [34, 35].

The philosophy behind this has been to make only those decisions in the 3D analysis stage of the detection process that can be made in as much assurance at each level illustrated as possible. Therefore, the hypothesis generation process creates hypotheses that may have somewhat weak evidence, which is where information from additional input data comes into use. That process is then followed by a subsequent selection process which picks these based on more global evidence, filtering out the unwanted ones. The hypothesis verification process uses the most of this global



Fig. (2). shp attributes for a single London building, containing key values for reconstruction such as height(s), age and land use.

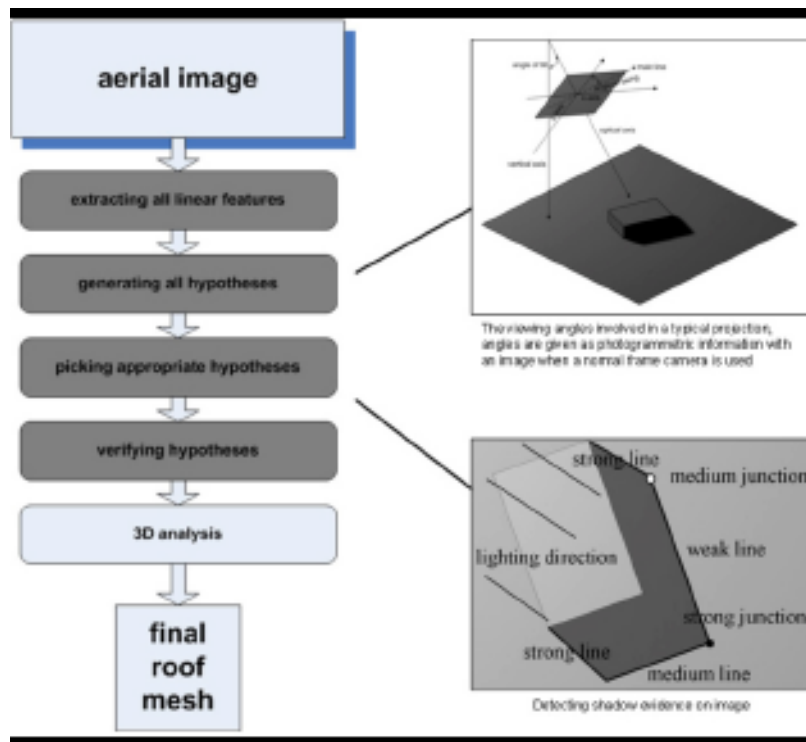


Fig. (3). Roof/building detection block diagram, based on the work of Lin et al. [34, 35].

information and can therefore make better decisions, resulting in a more accurate output.

To begin with, initial hypotheses are generated for 2D projections of building roofs. The reason behind this is that it is generally accepted that roofs are usually the most distinct features in aerial images from a nadir point of view (such as

the ones our system uses) and that focusing on the roofs helps reduce the number of hypotheses to be examined later on [34]. The generation of all roof hypotheses is gathered by the restriction of the building shapes to rectangular form or similar compositions and also of roofs to be flat. Roofs of buildings of this shape should project into a parallelogram-like shape.

As a more detailed example of the process we followed here, as Lin *et al.* [34] suggest, a 90° degree corner projected to an angle, O , given the angle π that one side of the projection makes with the horizontal and the viewing angles ψ and ξ is given by the following equation (1).

$o = \pi \tan(\lambda, \omega)$, where

$$\lambda = [\cos^2(\pi + \psi) \cos(\xi)] + \frac{\sin^2(\pi + \psi)}{\cos \xi}$$

and

$$\omega = \sin(\pi + \psi) \cos(\pi + \psi) [\cos(\xi) - \frac{1}{\cos(\xi)}] \quad (1)$$

Any parallelogram found must satisfy the relationships above (Formula 1). This is followed by a selection process which eliminates or keeps hypotheses based on what detectable evidence there is for them in the aerial image and on the global relationships amongst them all.

The basis for the verification of 3D outlines in our work are the variety of shadows buildings cast on the ground in aerial images. It is assumed that the direction of illumination in the image is known and also that the ground surface in the vicinity of the structure is flat. These two assurances allow for the computation of accurate height information.

To offer more insight in the algorithms involved in this area, the building height is related to several parameters that can be measured from the image given. With X (pixels/meter) the image resolution in the neighbourhood of the building location, H the projected wall height, can be computed (in pixels again) from the building height, B (in meters), and also the viewing angles by the following algorithm, $H = (B \cdot X \cdot \sin \xi)$. The projected shadow width, W , can be calculated from the building height, the viewing angles and the sun angles (the direction of illumination, x , the direction of shadow cast by a vertical line, y , and the sun incidence angle l using the following algorithm (2), as introduced again by Lin *et al.* [35]).

$$W = \begin{cases} B \cdot X \cdot \tan l & \text{when } \gamma = 0 \\ \frac{B \cdot X \cdot \sin(l + \xi)}{\cos i} & \text{when } \begin{cases} \xi \neq 0 \\ y = x = 270^\circ - \psi \end{cases} \\ \frac{B \cdot X \cdot \sin(l - \xi)}{\cos i} & \text{when } \begin{cases} \xi \neq 0 \\ y = x = 90^\circ - \psi \end{cases} \\ \frac{B \cdot X \cdot \sin(\xi - l)}{\cos i} & \text{when } \begin{cases} \xi \neq 0 \\ y = x + 180^\circ \end{cases} \\ \frac{B \cdot X \cdot \sin \xi \cdot |\cos(y + \psi)|}{|\cos(y - x)|} & \text{else} \end{cases} \quad (2)$$

However, when surfaces are not flat in the aerial image or are cast on surfaces of nearby buildings, the quality of the 3D information is low. While still usable for other purposes it was decided to have an alternative. Nevertheless then, in this eventuality, the average height value from the GIS .shp data is used. This illustrates in detail why our choice of using a number of different input data can be so useful. The concept behind shadow analysis used in this work derives from

the establishment of relationships between shadow casting elements and shadows cast [35].

Provided the direction of lighting is known, an attempt is made to establish these relationships between shadow casting lines and the actual shadow lines and subsequently between shadow casting vertices or junctions and shadow vertices (illustrated in Fig. 3). The ambiguous / non-visible vertical edges of buildings cast visible shadows in a direction parallel to the projection of the illumination on the ground. These shadow lines start at the edge of a building and can be detected fairly easily. There are occasions when these lines are cast in part on the ground and in part on nearby buildings or structures, but even then the projection constraint can be justified and they can still be detected.

The Semi-Automatic Mode

There are cases where end-users/software developers for social virtual worlds would require more customizable urban content than that generated by their postcode location and real-world maps, i.e. urban content for their placement which is not derived from GIS coordinates but can be fictitious. The semi-automatic mode of the application presented here can cater for that via individual plug-ins. Interfaces for Virtual City Maker can be either stand-alone as individual plug-ins or unified (Fig. 4).

Creating Buildings

Every virtual city consists of buildings, with these structures being the most important ones for the overall scene. One of the most important plug-ins for interactive or semi-automatic editing and creation of virtual cities, and thus buildings in particular, with the Virtual City Maker tool is the Building Creator script.

The creation of 3D buildings can either be imported from CAD files such as 2D ground plans or can be freeform ones that the user can draw. Multiple floor segmentation for the building is offered as well as roof modelling with controllable height and overhang based on the individual user's selection (out of a range of roof shapes such as gabled, cross-gabled, flat, mansard, hipped, cross-hipped, pyramidal and shed, see Fig. 4). Furthermore, the ability to create insets for additional geometric detail (for example segmentation between floors or creating a ground surface surrounding the building or even a terrace-like shape before the roof) is offered, as well as allowance for texturing.

This final process has been implemented by means of assigning different numerical material IDs to each floor and roof so that when the user selects an image to map onto a surface it will be saved according to this ID selection. This speeds up and simplifies the otherwise tedious process of texturing while at the same time giving the user full control on it (see Fig. 4, for step-by-step creation process of a building).

Different Types Of Shading

Recent research suggests [36, 37] that other ways of rendering (beside the photorealistic one) 3D urban content can also be beneficial for communicating elements of numerous other principles such as cognition, cartography and non-photorealism, which could potentially offer benefits to ex-

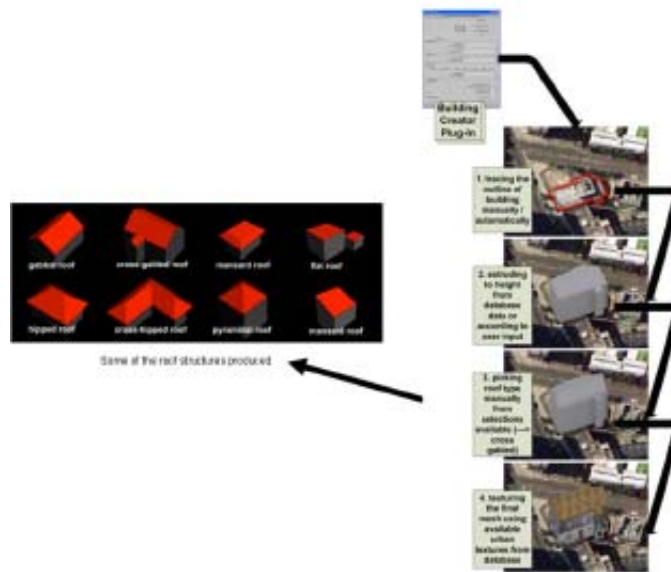


Fig. (4). Building Creator Plug-In.

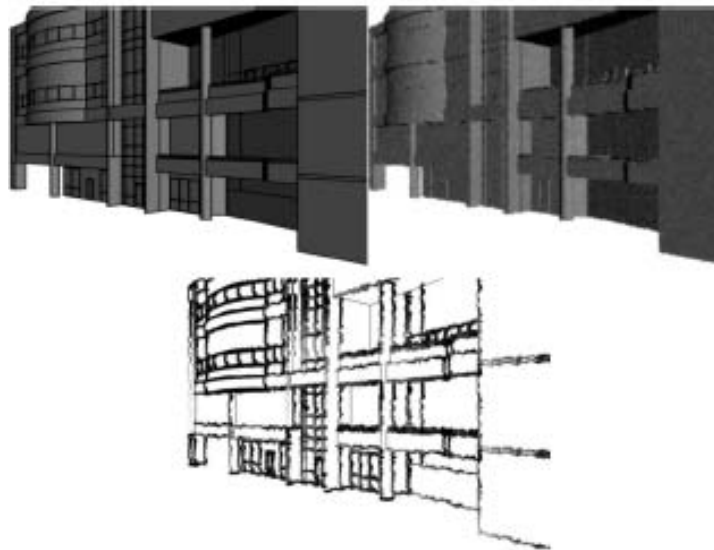


Fig. (5). Various views of urban models using the City Shader.

pressive communication. Thus an additional plug-in has been incorporated to the Virtual City Maker solution called City Shader in order to explore these possibilities. This facilitates the production of more complex non-photorealistic (NPR) styles (Fig. 5) resembling more abstract human 2D artwork such as cel-shading, hatching, line rendering and watercolour rendering. Most importantly, these different types of shading can be exported to suitable file formats for real-time use, which involves a number of issues considering real-time virtual worlds are not ideal to handle expressive urban-area visualizations (software that produces NPR renders automatically exists today [38, 39] but focuses on render-effect output rather than geometry-based one like ours, thus making it unusable in our chosen area).

The algorithm used for one of the results above (and for various other cel-shading visuals in our system) is a variant

of the one presented by Lake [40], which rather than smoothly interpolating shading across a model as in Gouraud shading (another popular approach), finds a transition boundary and shades each side of the boundary with a solid colour.

Optimising Building Structures

Following that, another very important component script for the overall tool is the previously mentioned Polygon Reductor. The algorithm which forms the basis of the Polygon Reductor script is based on a concept called *edge decimation* [41, 42]. Some concepts known as *vertex decimation* [43] and *triangle decimation* [44] are popular alternative approaches to polygon reduction / mesh simplification processes. In this case, i.e. with edge decimation, polygons are removed from the mesh by collapsing or contracting edges.



Fig. (6). Polygon reduction algorithm used.

That effectively means removing two triangles from an area of the mesh’s surface thus simplifying it. The two vertices of the collapsed, *decimated*, edge are merged into one endpoint and the triangle adjacency lists of the two original vertices are concatenated for the newly formed vertex. It should be noted that this contraction process has been made both progressive (for performance reasons, more on that below) and also reversible.

The algorithm we have opted to use is the fundamental edge decimation one (as described by Ronfard [45], amongst others) both for pre-processing and visualizing the optimized mesh (Fig. 6).

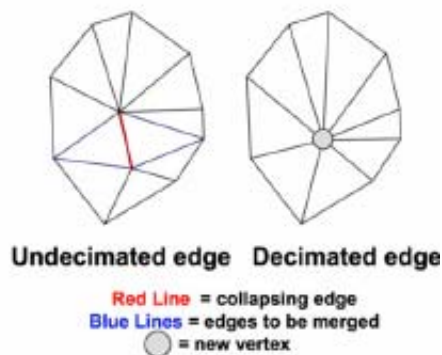
This approach has the following advantages: a) progressive representations of the original building model with the continuous family of meshes is very space-efficient plus has a smaller storage than the results of standard triangle/vertex approaches b) level-of-detail can be supported via transformation by just applying the *i*-th vertex split/edge collapse operation c) the model can offer view-dependent or selective refinement.

Finally, to select the target edges for the edge collapse we have opted to use an energy-derived error-based function introduced by Hoppe [42, 46], which has yielded the best results for our urban meshes compared to other error metrics. There are two ways to decrease the polygon count on a building mesh using this plug-in. The first is automatically, i.e. by selecting one of the four different levels of detail presented with each progressively decreasing polygon count by 25%. The second is semi-automatically, by decreasing either the vertex or the polygon count/percentage. The resulting output can be previewed within the script, have its settings reset and also exported to the file format for use with our visualization engine.

Enhancing the Urban Environment and Other Plug-ins

The impact of a virtual city both on a navigational and also on an aesthetical level can be enhanced by surrounding world objects. The main drawback of using these is that it can be very time consuming to create and place them on a fairly large scale urban model.

A solution to this, tackling foliage meshes, is the Tree Maker plug-in (user-interface illustrated in Fig. 7). The Tree



Maker script is in many ways representative of all the other work conducted on generation of models other than buildings for the Virtual City Maker solution. It targets performance and also automation.

All tree models generated consist of planes with adjustable height and width. These planes have opacity based bitmaps that create the illusion of a 3D mesh via transparency. They can have 2, 4, 6 or 8 sides, the more sides the better the illusion of a 3D mesh is, with a cost to real-time visualization performance.

An extensive texture library offers the user selection of plants in order to cover a number of different tree types. The user can also select any plane from the virtual city scene as a ground the models will rest upon and then generate rows (or columns of trees) in both the *x* and *y* axis (with controllable distances).

Fig. (7) illustrates some types of trees generated by the Tree Maker script and also how they can be placed into an urban scene, all developed by the rest of our solution.

Other plug-ins developed, include tools for the generation of terrain meshes (Terrain Generator) encompassing the urban areas and also tools for producing an environment for the scene (Environment Maker) with a minimum amount of effort from the side of the user and with fully-controllable parameters. The terrain can be created with a different number of segments to provide different LODs (levels of detail) while mountain and hill like shapes can be created and instantiated with appropriate user input (positioning, size and noise).

As for generating the environment for the urban mesh, some of the parameters supported include adjusting camera position and sunlight parameters such as lighting according to the time of the day, the shadows as well as the sun system orientation. Effects such as self-illumination, clouds (with user-adjusted size, quality and density) and finally a fog system can also be added.

3D FACIAL ANIMATION

In order to answer the question about how to create believable avatars, work is currently underway to create an automated 3D model that is able demonstrate human-like

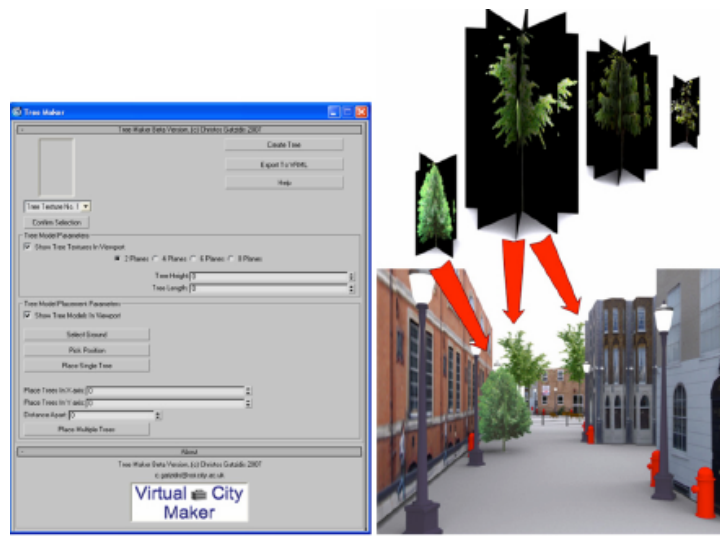


Fig. (7). Generating & enriching scene with trees.

features and expressions of emotion in real-time. This system is developed using Discreet’s 3D Studio Max scripting language called MaxScript which allows us direct access to the SDK of the modelling package. The purpose of the system is to both interface with these two applications in order to extend the emotional facial animation seen already from 2D [31] to 3D, and also offer us with a system that would severely cut down on the number of hours spent on the tedious manual task of lip-synching.

A skeleton-based hierarchy was created for the purposes of this with joints controlling human anatomy parts, multiple times for some of them. Using this hierarchy for each 3D head model, control can be exerted on areas such as the forehead and eyebrows, eye, eyelid, both sides of the nose, cheekbones, jaws, tongue and lips. This hierarchy can also be replicated (and scaled appropriately) to different 3D head models via a feature implemented on the system.

Following that, four different interfaces have been developed to control a number of important functions that all contribute to realistic 3D facial animation; a) expressions b) phonemes and visemes c) extra movements d) focus of the eyes.

These functions are controlled mostly via buttons and sliders on a range of 0-10 (Fig. 8) and can of course be blended together to create complex facial poses. Expressions cover all the emotions the character can depict such as happy, sad, anger, fear, disgust, and surprise. Phonemes (the sound we hear in speech) and visual phonemes (visemes, the mouth shapes and tongue positions that you create to make a phoneme sound during speech) are also controlled and blended together via buttons.

We have implemented 16 visemes for the system (Fig. 9) unlike the usual 9 often used for facial animation. Extra movements include additional, more natural sometimes, reflex motions such as blinking, eyebrow raising, random head motion (very important to create realistic facial animation) and breath. Finally, the focus of the eyes is facilitated by manipulating the joints that control the eyes by constraining them using aim constraints. This facilitates often overlooked motions, such as for example glancing away during a con-



Fig. (8). Facial Animation Control Panel.



Fig. (9). Visemes using the prototype facial animation system.

versation that can add considerable emotional impact and realism on any 3D face pose.

The next step involves interfacing in real-time with the Emotion Analyser parser (described below). The facial animation system first lays down keyframes for the phonemes. It does this by taking the information from the Emotion Analyser output file and matching it, phoneme against length of time. After that the system also adds emotions with given values. These values correspond with those on the parser. For the extra movements the system takes the text file and creates emotions from the markup language as it matches phonemes and timings. It also begins the process of laying down a series of secondary animations and keying these (blinking, random head motion, nodding and shaking of the head, breathing etc.). For example, blinking is controlled by the emotion that is set in the text file. If the 3D head that needs to be animated has anger set using the markup language, then it will only set blinking keyframes once, say, every six seconds for example. According to popular research, the normal blinking rate is once every four seconds, and if the character is in an extreme state (such as for example lying or acting suspiciously) then this rate increases to once every two seconds. Moving on from that, random head motion is keyed only when keyframes are present for phonemes, meaning that the character always moves his head

when he is speaking. This results in a very subtle yet noticeable effect. Another example of building on emotional facial animation with extra movements is by breathing, an automatic process keyed in values when we reach the end of a sentence. This value can differ depending on the physical state of the character (again, an extreme value mimics gasping for breath).

TRANSLATING EMOTION FROM TEXT

To answer the question of how to trigger the display appropriate facial expressions we propose the use of the Emotion Analyser to automatically detect emotions within text communications [31, 32]. The Emotion Analyser is a word tagging system that identifies explicit emotional keyword and phrases using an emotion extraction engine. The system enhances text communication by detecting emotions embedded in user-typed text, displaying appropriate facial expression images on the screen in real time.

Fig. (10) depicts an overview of the components of the Emotion Analyser. The emotion extraction engine comprises three components: input analysis, tagging system and a parser.

A set of rules has been set up to specify the engine’s behaviour at each stage.

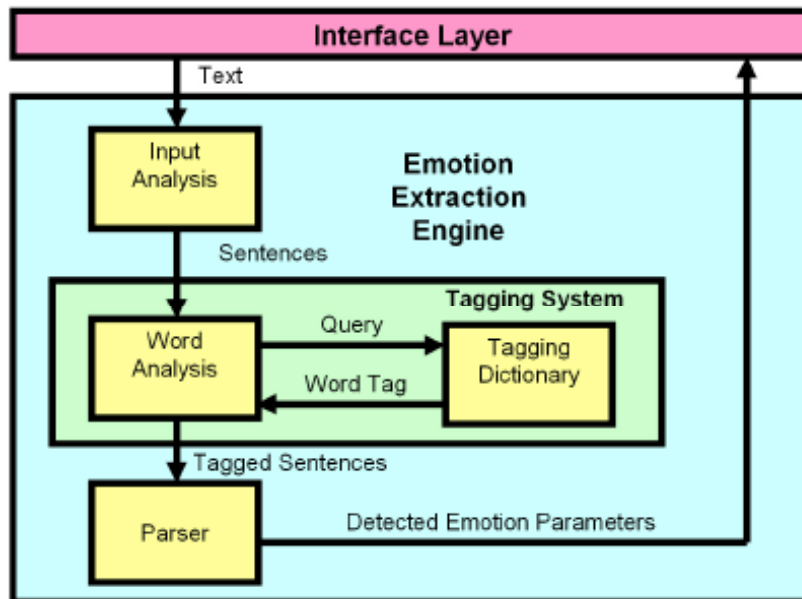


Fig. (10). The Emotion Analyser components.

Input Analysis

The engine receives the text that is passed between participants in the interface layer. The length of the text can vary from just a few words to a hundred-thousand word article. An article may contain hundreds of sentences and even in a chat environment a user may type more than one sentence at a time.

The text is divided into individual sentences and each sentence is analysed one at a time. Sentences are identified where termination characters such as exclamation marks, question marks and full stops are identified. The context of the full stop is tested to ensure the intention is to terminate the sentence and it is not being used for another purpose such as a decimal point. Repeated exclamation and question marks are noted as these raise the intensity of any detected emotion.

The input analysis component passes each sentence to the tagging system.

Tagging System

The tagging system replaces the individual words in the sentence with tags to indicate the word category and highlight the emotional words ready for further analysis by the parser. Although some tagging systems already exist, such as Brown Corpus [47] and BNC [48], these tagging systems do not fulfil the engine's requirement of simplicity for fast, real-time response. In order to determine how to define the tag set it was necessary to consider how sentences are constructed and how the emotional words should be classified.

Sentences are made up of two main groups of words: function words and content words [49, 50]. Function words include prepositions, conjunctions and auxiliary verbs, while content words include nouns, adjectives, verbs and adverbs. All these groups contain sub classes, which can be represented by tree diagrams [51] as shown in Fig. (11).

By classifying all the words in the sentence and analysing the relationship between them it is possible to make inferences about the construction of the sentence, including whether the sentence is a statement, a negative statement, or a question, and who is the subject of the sentence (whether the person who wrote the sentence is referring to themselves, others or it is neutral).

In addition to classifying the word type, the tagging system also classifies words that convey emotion. There are a number of different classifications of emotional words. It was decided not to attempt to classify words according to a complex hierarchy of emotions, such as the OCC model [19] that defines a hierarchy of 22 categories of emotion, as this would be too complex for the development of believable characters [52], and it would be extremely difficult to map all of the 22 emotion categories into recognizable expressions. Instead, we used a classification that directly related to facial expressions. Ekman [53-55] proposed six basic expressions which can be communicated efficiently through distinct facial expressions that are recognized across many cultures. The categories are happy, sad, anger, fear, disgust and surprise. Within each of the categories, a wide range of intensity of expression and variation of detail exists.

A dictionary of over 21,000 words has been compiled for use with the Analyser. Word entries contain the word (including any possible prefix and suffix), a tag which indicates which emotion category they belong, and an ambiguity tag that indicates whether they have different meanings in different contexts. In our tag-set the mark "EMO_W" is assigned to each possible emotional word. There are six expression categories and so the tag-set uses numbers from 1 to 6 to represent happy, sad, fear, surprise, anger, and disgust. Each emotional word tag also indicates the intensity level information: low, medium and high, e.g. the tag *IN2* stands for expression category "Happy" and intensity level 2. Examples of the tag set are shown in Table 1.

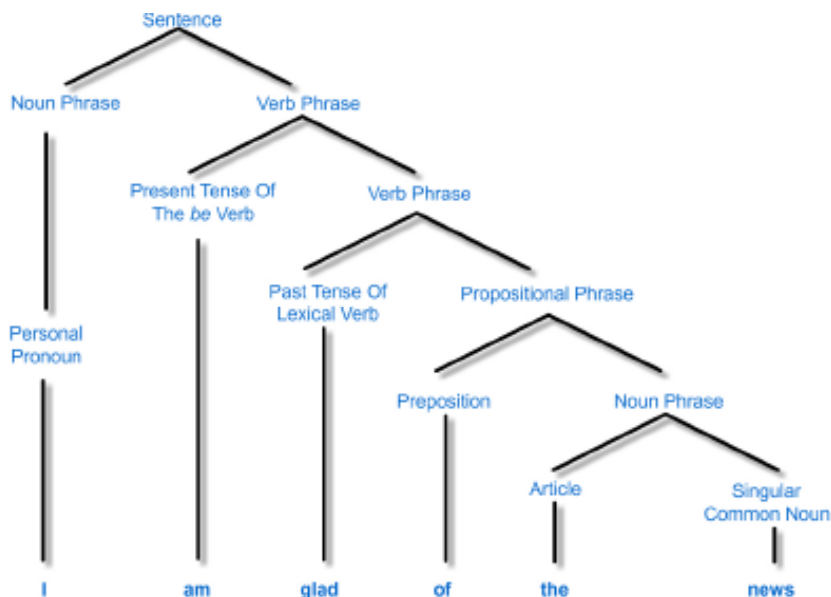


Fig. (11). Tree Representation of a Sentence.

Table 1. The Tag Subset

Tag	Description	Example
NOUN	Noun	Kindness
1N2 EMO_W	Happy, Noun, intensity 2	Happy
2N2 EMO_W	Sad, Noun, intensity 2	Sad
3N2 EMO_W	Fear, Noun, intensity 2	Fear
4N2 EMO_W	Surprise, Noun, intensity 2	Surprise
5N2 EMO_W	Anger, Noun, intensity 2	Angry
6N2 EMO_W	Disgust, Noun, intensity 2	Disgust
PCF	Present continuous	Is
PP	Present perfect	Have
Past	Past	Was
VI	Verb	Breathe

In addition a number of common phrases are categorized to infer emotions that may not be evident from the individual words viewed in isolation, e.g. “cold feet”.

When receiving the input sentence, the tagging system splits the sentence into individual words, and searches through the tagged dictionary to find each word and the corresponding tag category. If a word is not found in the dictionary, the engine will check for suffix and prefix combinations. The words in the sentence are replaced by the corresponding word-category tags and the output is sent to the parser for further analysis.

Parser

The parser examines the tagged sentence to establish the emotional state of the sentence. Only sentences that contain

an emotional word are considered for further analysis. A set of rules are followed to identify which emotional words are present, the person to whom the emotional words referred and the intensity of the emotional words. For chat applications, only sentences that express emotions in the present tense are considered by the engine.

It is important to establish to whom any expressed emotion refers to, as only emotions relating to the first person should be represented in the interface. If no subject is found, the engine will assume the emotion refers to the person who is in communication e.g., “apprehensively happy” will be treated as “I am apprehensively happy”.

If there is one more emotional word in a sentence and they are connected by a conjunction and they are both positive or both negative emotions, then the parser will combine

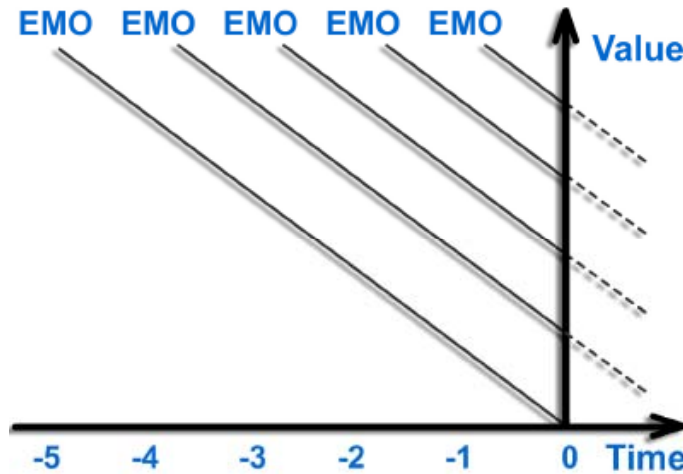


Fig. (12). The Decay Of Emotion Over Time.

these two emotional states e.g., “*I am surprised and happy about it*”.

The parser also checks whether there is a condition attached to the emotion e.g. “*I’m happy when she’s there!*” is interpreted as an emotional sentence, but with reduced intensity.

When exclamation marks are found, the intensity of the corresponding emotion is increased. The Parser will also increase the intensity where adjectives are repeated as in “*I am very very sorry*”.

If the sentence is a question the engine cannot identify emotions, for example, the sentence “are you happy?” does not represent a happy emotion.

If an emotional word is found in negative form, the parser will not treat the sentence as containing an emotional word, the negation of one emotion does not automatically indicate the presence of the opposite emotion.

The engine has limited success dealing with irony, except where the text contains explicit acronyms and emotion symbols. For example, the sentence “*I hate you :-)*” presents an ironic and happy feeling despite the existence of the emotional word “hate”. When emotion acronyms or symbols are found, the parser will discard other emotional words and set the sentence’s mood according to the category of the acronym.

The parser also resolves situations where a conflict between positive and negative emotions is found in the same sentence. Decisions are made based on the momentum of emotion. Emotions do not disappear once their cause has disappeared, but decay through time [56], therefore greater weight is given to the emotions contained in the most recent sentence with progressively less weight to earlier sentences. This is illustrated in Fig. (12), where the current mood of the user at position 0 on the time axis is calculated with the emotion state (EMO) of recent time periods (-1 to -3) having a greater current value than older time periods (-4 to -5).

The momentum of emotion is calculated by the Average Weighted Emotional Mood Indicator (AWEMI) shown in Formula 3:

$$AWEMI = \frac{1}{N} \sum_{i=1}^N \alpha_i E_i \quad (3)$$

In which i is an index increasing from 1 to N corresponding to the sentences taken into account, up to maximum N . E_i corresponding to the sum of the signed emotion intensities (momentum, with a positive or negative sign) of i -th sentence and α_i is the weight given to each emotion within a sentence. The emotion decay is represented by the weight parameter α which is calculated using Formula 4.

$$\begin{aligned} \alpha_i &= \alpha_{i+1} - 0.1 \text{ where } i \in \{-5, -4, -3, -2\} \\ \alpha_{-1} &= 0.5 \\ \alpha_0 &= 1 \end{aligned} \quad (4)$$

Using this formula the influence of the emotion in the current sentence will have greatest influence on current mood ($\alpha_0 = 1$) while the emotion in contained in the previous sentence will have half the influence ($\alpha_{-1} = 0.5$) and the emotion contained in the previous four sentences will have a gradually diminishing influence ($\alpha_{-2} = 0.4$, $\alpha_{-3} = 0.3$, $\alpha_{-4} = 0.2$, $\alpha_{-5} = 0.1$).

The emotional states of each sentence are stored for the analysis of average mood calculation. The current emotion is based solely on the information presented within a single sentence while the current mood is calculated from the current and previous weighted emotions.

When a sentence contains both positive and negative emotions, e.g., “*I am happy, but I am worried about the future*” a conflict occurs. When conflicting emotions are detected, the engine will consult the current mood calculated by the average mood calculator. If the current mood is a particular emotion, then the new sentence’s emotion will be changed to that category with the lowest intensity, otherwise the conflicted emotions will remain unchanged.

An array (E) is assigned to contain the accumulative intensity values of the six emotion categories. The array elements 0 to 5 represent the accumulative intensity of emotion happy, surprise, anger, disgust, sad and fear. The accumulative intensities are calculated using Formula 5:



Fig. (13). 2D and 3D Internet chat application interface.

$$E[x] = \sum_{i=-n}^0 \alpha_i I_i(x) \text{ where } x = \{0, 1, 2, 3, 4, 5\} \quad (5)$$

The values of array E depend on the relative intensities over the last n time periods. In this case n is chosen to be 5 as it is assumed that in a chat environment users only remember the most recent dialogs. $I_i(x)$ is the intensity of emotion category x at discrete time i and the value of $I_i(x)$ varies from 0 to 3, which represents the lowest intensity to highest intensity. When i is 0, $I_i(x)$ contains the intensity values of the current emotions.

The output from the parser is sent to the interface layer, including the current sentence emotion category with intensity and the average mood with intensity.

The Emotion Analyser text-to-emotion engine that has been applied in a number of different applications including an Internet chat application, for which both 2D and 3D (Fig. 13) interfaces have been developed. In both versions two or more users can exchange text messages that appear in speech bubbles located above facial images that represent individual users. The 2D version displays a static image that represents an office table that the participants gather around. The 3D version enhances the interface by providing a simple 3D environment for user to explore. Whenever an emotion is detected in a text message that refers to the sender their facial image displays an appropriate expressive image. The applications illustrate the operation of the Emotion Analyser, however the interfaces could be enhanced further. There is no animation between changes of expression, and the environment is not realistic.

The text chat interface has also been used as an accompaniment in an online chess game application. In addition to this, a prototype virtual theatre application has been created where 2D characters read through a predefined script and dynamically change their expressions to match the current text sentences. The Emotion Analyser can also be used to assess the emotion state of whole documents; text can be selected from an embedded web browser, and sent to the Analyser for classification. A specialized application has been customized to search for words relevant to the movement of the value of shares in the stock market.

DISCUSSION AND CONCLUSION

The combination of the above three systems forms a framework that could be used within 3D virtual worlds to enhance social interaction. The major three contributions this framework could make is a) provide a mechanism for automatic identification of emotions in text communications b) provide a more realistic 3D virtual environment and finally c) provide a mechanism for displaying more believable facial expressions.

Applications where this framework could potentially be applied on include most of the main open-ended virtual worlds such as Home [57], Habbo Hotel [58], ActiveWorlds [59] and Second Life [60]. All of them use avatars (but without any real emotional impact) and furthermore allow the creation and display of urban buildings (but without any real attempt at geo-referencing or specifying real-world location).

Moreover, internet MMORPGs (Massively Multiplayer Online Role-Playing Games) such as for example Guild Wars [61] or World of Warcraft [62] rely today even more heavily on social interaction/live chat and chatting by using a narrative and having quests (where users can join guilds and fight monsters or collect objects) so any potential technology that could enhance emotional 3D participation in any communication context would be very beneficial.

However, virtual worlds such as Second Life and RPG games (Role Playing Games) are not the sole application area of the system/framework presented here. Emotional 3D live chat can find a place in virtual learning with the option for learner groups studying out of normal hours, or even beyond the timeframe of the course, thus opening up real potential for learning outside the confines of any standard or traditional institutional framework.

Virtual IT support is another possible application area where institutions seeking to provide ways of supporting new emerging technologies and applications can often be obstructed by existing architecture. Our framework could be applied towards a move that enables service-orientated architecture and interoperability, making remote geographical IT support easier and less complex.

The framework presented here also provides a number of areas for further research, particularly in the area of assessing emotional 3D internet interaction. Some of the questions that are generated by the research presented here so far, which we plan to explore in the near future, include:

- a) What is the suitable range of expressions to display in 3D internet emotional communication, should there be more or fewer than the ones we already have on our system?
- b) How can the intensity of emotion be fine-tuned to suit expression and perhaps even gestures (an aspect which our framework so far does not cover), i.e. should the display of emotions be exaggerated or more subtle?
- c) Does the Emotional Momentum mechanism included in the Emotion Analyser resolve conflicts of emotion correctly?
- d) Does the inclusion of expressive characteristics make the interaction more coherent or does it distract?
- e) How do characters interact using text? For example, in traditional chat environments only those within the meeting can exchange text / send private messages to individuals. Within virtual worlds it is more complex, should it be possible to send emotional interaction content only to those within view or broadcast to all users?
- f) Should the 3D urban environment be navigatable world that users can explore or a mere display of the user's geographical location in 3D?

We plan to explore all of the questions above plus other additional issues by conducting evaluation studies with a range of subjects.

We have discussed three separate systems that can be combined to form a framework that could be adopted to enhance the social interaction of the inhabitants of virtual worlds. The Virtual City Maker system automatically generates a realistic and recognizable environment to locate the interaction, the 3D facial animation system provides a mechanism for displaying more believable facial expressions, and finally the Emotion Analyser provides a mechanism for the automatic identification of emotions in text communications. Combined within a framework each system provides aspects that make the user experience within virtual worlds richer. We have described some of the anticipated benefits for participants of virtual worlds and discussed some of the areas of interest of each system individually and in combination that would require further research.

REFERENCES

- [1] E. Andre, M. Klesen, P. Gebhard, S. Allen and T. Rist, "Integrating models of personality and emotions into lifelike characters", *In Proceedings of the workshop on Affect in Interactions - Towards a new Generation of Interfaces*, A. Paiva and C. Martinho, Eds. Siena, Italy, October 1999, pp. 136-149.
- [2] J. Bates, "The Role of Emotion in Believable Agents", *Commun. ACM*, vol. 37, no. 7, 1994, pp. 122-125.
- [3] J. N. Bailenson and J. Blascovich, "Avatars", *In Encyclopedia of Human-Computer Interaction*, W. S. Bainbridge, Ed. MA: Berkshire Publishing Group, 2004, pp. 64-68.
- [4] C. Bartneck, "How convincing is Mr. Data's smile: Affective expressions of machines", *User Model User-adapt. Interact.*, vol. 11, 2001, pp. 279-295.
- [5] N. Bianchi-Berthouze, C. L. Lisetti, "Modelling Multimodal Expression of User's Affective Subjective Experience", *User Model User-adapt. Interact.*, vol. 12, no. 1, 2002, pp. 49-84.
- [6] S. Kiesler, J. Siegel and T.W. McGuire, "Social psychological aspects of computer-mediated communication", *Am. Psychol.*, vol. 39, 1984, pp. 1123-1134.
- [7] T. Koda, and P. Maes, "Agents with Faces: The Effects of Personification of Agents", *Proceedings of HCI'96, London*, August 20-23, 1996, pp. 98-103.
- [8] A. Paiva, Ed, "Affective Interactions: Towards a new generation of computer interfaces", Berlin/New York: Springer, 2000.
- [9] B. Parkinson, "Emotions in direct and remote social interaction: Getting through the spaces between us", *Comput. Hum. Behav.*, vol. 24, no. 4, 2008, pp. 1510-1529.
- [11] Merriam-Webster online dictionary, <http://www.m-w.com/home.htm>. [Accessed September 27, 2003].
- [12] H. Prendinger and M. Ishizuka, "Introducing the cast for social computing: life-like characters", *In Life-like Characters. Tools, Affective Functions and Applications*, H. Prendinger and M. Ishizuka, Eds. Berlin: Springer 2003, pp. 3-16.
- [13] B. Hayes-Roth, "What makes characters seem life-like", *In Life-like Characters. Tools, Affective Functions and Applications*, H. Prendinger and M. Ishizuka, Eds. Berlin: Springer 2003, pp. 447-462.
- [14] H. Prendinger, S. Descamps and M. Ishizuka "Scripting affective communication with life-like characters in web-based interaction systems", *Appl. Artif. Intell.*, vol. 16, nos. 7-8, 2002, pp. 519-553.
- [15] H. Prendinger, S. Saeyor and M. Ishizuka, "MPML and SCREAM: scripting the bodies and minds of life-like Characters", *In Life-like Characters. Tools, Affective Functions and Applications*, H. Prendinger and M. Ishizuka, Eds. Berlin: Springer 2003, pp. 213-242.
- [16] E. Andre, T. Rist, S. van Mulken, M. Klesen and S. Baldes, "The automated design of believable dialogue for animated presentation teams" *In Embodied Conversational Agents*, J. Cassell, J. Sullivan, S. Prevost and E. Churchill, Eds. Cambridge, MA: The MIT Press, 2000, pp. 220-255.
- [17] W. L. Johnson, J. Rickel and J. C. Lester, "Animated pedagogical agents: face-to-face interaction in interactive learning environments", *Int. J. Artif. Intell. Educ.*, vol. 11, 2000, pp. 47-78.
- [18] J. Gratch, J. Rickel, E. Andre, J. Cassell, E. Petajan and N. Badler, "Creating interactive virtual humans: some assembly required", *IEEE Intell. Syst.*, vol. 17, no. 4, pp. 54-63, July/August 2002.
- [19] A. Ortony, G. L. Clore, A. Collins, "Cognitive Structure of Emotions", Cambridge, UK: Cambridge University Press, 1988.
- [20] D. Thalmann, H. Noser and Z. Huang, "Autonomous virtual actors based on virtual sensors", *In Creating Personalities for Synthetic Actors*, R. Trappale and P. Petta, Eds., Berlin/New York: Springer 1997, pp. 25-42.
- [21] J. Cassell, T. Bickmore, L. Campbell, H. Vilhjmsson and H. Yan, "Human conversation as a system framework designing embodied conversational agents", *In Embodied Conversational Agents*, J. Cassell, J. Sullivan, S. Prevost and E. Churchill, Eds. Cambridge, MA: The MIT Press, 2000, pp. 29-63.
- [22] J. Cassell, "More than just another pretty face: embodied conversational interface agents", *Commun. ACM*, vol. 43, no. 4, 2000, pp. 70-78.
- [23] N. I. Badler, D. Metaxas, B. Webber and M. Steedman, "The center for human modeling and simulation", *Presence (Camb)*, vol. 4, no. 1, 1995, pp. 81-96.
- [24] N. I. Badler, R. Bindiganavale, J. Allbeck, W. Schuler, L. Zhao and M. Palmer, "Parameterized action representation for virtual human agents", *In Embodied Conversational Agents*, J. Cassell, J. Sullivan, S. Prevost and E. Churchill, Eds. Cambridge, MA: The MIT Press, 2000, pp. 256-284.
- [25] C. Conati, "Probabilistic assessment of user's emotions in educational games", *Appl. Artif. Intell.*, vol. 16, 2002, pp. 555-575.
- [26] F. de Rosis, B. Carolis and S. de Pizzulito, "Software documentation with animated agents", *Proceedings of the Fifth ERCIM Workshop on User Interfaces For All*, Dagstuhl, Germany, November, 28 - December 1, 1999.
- [27] D. Rousseau and B. Hayes-Roth, "A Social-Psychological Model For Synthetic Actors", *Proceedings of the Second International Conference on Autonomous Agents: Agents-98*, New York: ACM Press, 1998, pp. 165-172.

- [28] B. Hayes-Roth, "Adaptive learning guides", *Proceedings of the IASTED, Conference on Computers and Advanced Technology in Education*, Banff, Canada, June 2001.
- [29] D. John, C. Gatzidis, F. Liarokapis, A. C. Boucouvalas and V. Brujic-Okretic, "A Framework for the Development of Online, Location-Specific, Expressive 3D Social Worlds", *Games and Virtual Worlds for Serious Applications*, 2009, VS-GAMES '09, Conference, Coventry, UK, March 23-24, 2009, pp. 219-220.
- [30] C. Gatzidis, V. Brujic-Okretic, F. Liarokapis and S. Baker, "Developing a Framework for the Automatic Generation and Visualisation Of 3D Urban Areas on Mobile Devices", *10th Symposium For Virtual And Augmented Reality*, Joao Pessoa, Brazil, May 13-16, 2008, pp. 151-162, 2008.
- [31] A. C. Boucouvalas, Z. Xu and D. John, "Expressive image Generation: Towards expressive Internet Communications", Special Issue on Context and Emotion Aware Visual Interaction, *J. Visual Languages Comput.*, vol. 17, no. 5, 2006, pp. 445-465.
- [32] D. John, A. C. Boucouvalas and Z. Xu, "Text-to-Emotion Analysis Engines -Theory and Practice", *Euromedia*, Athens, Greece April 17-19, 2006, pp.24-27.
- [33] P. Mueller, P. Wonka, S. Haegler, S. Ulmer and L. Van Gool, "Procedural Modelling Of Buildings", *ACM Trans. Graph.*, vol. 25, no. 3, 2006, pp. 614-623.
- [34] C. Lin and R. Nevatia, "Building Detection and Description from a Single Intensity Image", *Comput. Vis. Image Underst.*, vol. 72, no. 2, 1998, pp. 101-121.
- [35] C. Lin, A. Huertas and R. Nevatia, "Detection of Buildings using Perceptual Grouping and Shadows", *Proceedings of IEEE Computer Vision and Pattern Recognition Conference*, Seattle, WA, USA, June 21-23, 1994, pp. 62-69.
- [36] J. Dollner and M. Walthner, "Real-Time Expressive Rendering of City Models", *Proceedings of the Seventh International Conference on Information Visualization (IV'03)*, London UK, July 16-18, 2003, pp. 245-250.
- [37] J. Schumann, T. Strothotte, S. Laser and A. Raab, "Assessing the effect of non-photorealistic rendered images in CAD", *Proceedings of the SIGCHI conference on Human factors in computing systems: common ground*, Vancouver, British Columbia, Canada, April 13-18, 1996, pp. 35-41.
- [38] CEBAS' FinalToon. <http://www.finaltoon.com>. [Accessed Oct 29, 2008].
- [39] PSOFT's Liquid+. <http://www.psoft.co.jp/en/product/liquid/index.html>. [Accessed Oct 29, 2008].
- [40] A. Lake, C. Marshall, M. Harris and M. Blackstein, "Stylized rendering techniques for scalable real-time 3d animation", *Proceedings of NPAR 2000: First International Symposium on Non Photorealistic Animation and Rendering*, Annecy, France, June 5-7, 2000, pp. 13-20.
- [41] M. Garland and P. Heckbert, "Surface Simplification Using Quadratic Error Metrics", *Proceedings of SIGGRAPH '97: annual conference on Computer graphics and interactive techniques*, Los Angeles, USA, August 3-8, 1997, pp. 209-216.
- [42] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle, "Mesh optimization", *Proceedings of SIGGRAPH '93: annual conference on Computer graphics and interactive techniques*, New York, NY, USA, August 1-6, 1993, pp. 19-26.
- [43] B. Koh and T. Chen, "Progressive VRML Browser", *Proceedings of IEEE International Workshop on Multimedia Signal Processing*, Copenhagen, Denmark, September 13-15, 1999, pp. 71-76.
- [44] W. J. Schroeder, J. A. Zarge and W. E. Lorensen, "Decimation of triangle meshes", *Proceedings of SIGGRAPH '92: annual conference on Computer graphics and interactive techniques*, New York, NY, USA, pp. 65-70, 1992.
- [45] R. Ronfard and J. Rossignac, "Full-range approximation of triangulated polyhedra", *Comput. Graph. Forum*, vol. 15, no. 3, 1996, pp. 67-76.
- [46] H. Hoppe, "Progressive Meshes", *Proceedings of SIGGRAPH '96: annual conference on Computer graphics and interactive techniques*, New Orleans, Louisiana, USA, August 4-9, 1996, pp. 99-108.
- [47] W. N. Francis and H. Kucera. Brown Corpus Manual, Department of Linguistics, Brown University. <http://www.hit.uib.no/icame/brown/bcm.html>. [Accessed January 27, 2003].
- [48] G. Leech, "A Brief Users' Guide to the Grammatical Tagging of the British National Corpus". <http://info.ox.ac.uk/bnc/what/gramtag.html>. [Accessed January 27, 2003].
- [49] I. Robinson, *The New Grammarians Funeral*, Cambridge, UK: Cambridge University Press, 1978.
- [50] W. T. Gordon, *Saussure for Beginners*, New York, NY: Writers & Readers Publishing Inc, 1996.
- [51] S. Russell and P. Novig, *Artificial Intelligence: A Modern Approach*, Boston MA: Prentice Hall, 2002.
- [52] A. Ortony, "On Making Believable Emotional Agents Believable", *In Emotions in Humans and Artefacts*, R. P. Trappale, Ed. Cambridge, MA: MIT Press 2003.
- [53] P. Ekman, W. V. Friesen and P. Ellsworth. *Emotion In The Human Face: Guidelines For Research And An Integration Of Findings*, New York, NY: Pergamon Press, 1972.
- [54] P. Ekman, *Emotion in the Human Face*, Cambridge, UK: Cambridge University Press, 1982.
- [55] P. Ekman, W. V. Friesen and J. C. Hager, *The Facial Action Coding System*, *Research Nexus eBook*. Salt Lake City, UT, 2002: Available from: <http://www.face-and-emotion.com/dataface/est-ore/main.jsp>.
- [56] J. Velasquez, "Modeling Emotions and Other Motivations in Synthetic Agents", *Proceedings of the AAAI Conference*, Providence, RI, July 27-31, 1997, pp. 10-15.
- [57] PlayStation Home. <http://uk.playstation.com/psn/pshome/>. [Accessed June 29, 2009].
- [58] Habbo Home. <http://www.habbo.co.uk/community>. [Accessed June 29, 2009].
- [59] Active Worlds. <http://www.activeworlds.com/>. [Accessed June 29, 2009].
- [60] Second Life. <http://secondlife.com/>. [Accessed June 29, 2009].
- [61] Guild Wars. <http://www.guildwars.com/>. [Accessed June 29, 2009].
- [62] World Of Warcraft. <http://www.wow-europe.com/en/index.xml>. [Accessed June 29, 2009].

Received: September 10, 2010

Revised: October 20, 2010

Accepted: October 25, 2010

© John et al.; Licensee Bentham Open.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.