Norwegian Artificial Intelligence Symposium, Gjøvik, 22 November 2010

Towards a Simple Augmented Reality Museum Guide

Jo Skjermo[1], Michael J. Stokes[1], Torbjørn Hallgren[1], Anders Kofod-Petersen[1,2]
[1]Department of Computer and Information Science
Norwegian University of Science and Technology
NO-7491 Trondheim, Norway
[2]SINTEF ICT
S. P. Andersens vei 15 b
NO-7465 Trondheim, Norway

***Abstract*—modelling knowledge for agents and virtual entities can be hard. Often the domain knowledge needed increases as the complexity of the agent increases. What is worse is that for increasingly complex agents the needed knowledge of the underlying reasoning methods may also increase for the domain experts. In light of this we present early work and thoughts on a virtual museum guide system that should be easy to configure and maintain by domain experts.**

1. Introduction

Augmented reality museum guides have recently attracted a lot of attention [2]. However, many problems still remain unsolved. The goal of the work presented here is to design a suitable architecture for augmented reality museum guides in general, and for the *Sveresborg museum* in particular. *Severesborg museum*Êis a museum located in Trondheim, Norway where among others the ruins of the old Zion fortress is located. The Zion fortress has been subject to several related projects at the visualisation group at NTNU. Most recently the work presented here.

The virtual guide system presented here introduces virtual characters that server as user guides at the Zion fortress (see Fig. 1) by interacting with the users to impart knowledge about this cultural site.

Several virtual museum guide systems already exist. An early augmented reality museum tour guide was ARCHEOGUIDE with LIFEPLUS developed by Information Society Technologies (IST) [15]. It fills the cultural site with a multitude of virtual human characters going about their daily business. Instead of simply hearing about ancient life, users are able to see it in action, superimposed over the modern day site. The system also introduces a virtual guide character who interacts with the user, providing navigation assistance and historical information. However, the guide itself seems to be superimposed only onto the lower right-hand corner of the user's head mounted display. As such, it does not seem to inhabit and navigate around the same world space that the user and the other virtual life characters do.

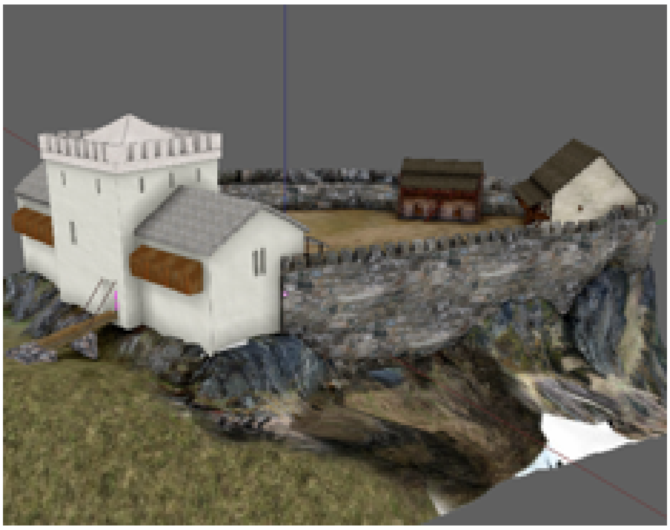

Fig. 1. The computer model of the Zion castle

This would limit the possible interactions between the guide and the user: for example leading and following actions would not be possible. The system did allow for planning of differentiating routes through the site based on a questioner a user filled out before starting. However, the system did not observe user behaviour for adaption of the plan.

Another early augmented reality museum guide is The Museum Wearable (MW), undertaken by Flavia Sparacino at the MIT Media Lab [14]. This system does not have an actual guide character; instead it superimposes contextual text labels, still images and video clips, combined with audio narration. It does however observe the user's behaviour and build up a model of their interests, which is then used to customise the audio-visual material presented to them. The system observes the exhibits the user visits, and measures the time spent at each exhibit. A Bayesian network is then used to classify the user into only one of three stereotypes, and different audio and video clips are presented based on the inferred user type.

Both ARCHEOGUIDE/LIFEPLUS and MW has several restrictions, but together give an idea for the required functionality such a system can contain. To move toward a more general solution for a virtual museum guide in an augmented or virtual world we start by identifying requirements and main problem areas.

The rest of the paper is organised as follows: In Section 2 the architecture and design of a augmented reality museum guide and and overview of the different components is given; Section 3 describes the initial implementation of the proof-of-concept application; Finally, Section 4 draws up the conclusion and points to future work.

2. Architecture and design

Any augmented reality system must related to both the physical and virtual environment and the entities present. In the case of this museum guide the physical environment consists of the end-user and museum as such including the different features of interests. The virtual environment consists of the representations of the physical environment as well as the virtual tour guide. The following sections describes the architectural and design choices made when developing this augmented reality museum guide.

2.1. The mobile unit

The mobile unit is the equipment worn by the end-user when visiting the museum. Its primary objectives are to measure position and orientation; capture visual input from the end-user's perspective; and combine the physical with the virtual world.

2.2. Representing the physical environment

As described above, the physical environment consists of the end-user, the museum grounds and the features of interest. Representing the physical world can be approached by several avenues. Perhaps the most popular way is a *triangle mesh*. A triangle mesh consists of a set of vertices and a set of triangles, where each triangle is defined in terms of three ordered vertices. Each vertex has a 3D position vector and may also have a normal vector and a texture coordinate.

Fig. 2. The guide next to an activation point

Triangle mesh is a very simple representation compared to other approaches such as *winged-edge* and *half-edge*. While a simple representation might appear as a disadvantage, it does mean that they can be regarded as a sort of lowest-common-denominator. Thus, they can generally be generated from any source representation.

Triangle mesh can be used to render the geometry of (ancient) buildings and other features that are relevant for the guide system. The secondary application of triangle mesh is to be used in the physics simulator (see Section 2.7). The simulator will need geometrical information about the environment to properly simulate dynamic objects (such as the tour guide). Triangle meshes are generally not the most efficient representation for this, however they are useable.

2.3. Features of interest

Features of interest are the entities in the physical world that are (assumed) to be of interest. These points combine the physical and the virtual world. Given that triangle meshes are used to represent the physical environment it would seem like the obvious choice also for features of interest. However, there are some significant disadvantages to using this method. The amount of 3D editing work required is not insignificant and the technical competence required of the editor is significant. Finally, it would often be the case that features of interest do not have a clear geometrical shape. They would rather be general areas.

One alternative method is to employ bounding boxes. While this would solve the technical skill problem, it does not address the situations where a feature of interest do not have a clear geometrical shape. Finally, there is no clear way to display them to the end-user. Thus, no intuitive way for the user to "activate" them can be presented.

Activation point (see Fig. 2) with an associated activation radius, as employed in LIFEPLUS [15], appears to be the most promising. The work presented here employs this method. Where a feature of interest is marked with the position and will appear as a flowing, glowing information icon. This approach clearly signals the user that there is something that does not belong to then environment. When the user approaches it and crosses the activation radius. The actions from the systems, which are associated with this point are activated.

In addition to these activation point, location points and guide points can also be placed. Both are invisible to the user and serves the purposes of marking the location of a feature of interest and marking the point where the guide should stand while engaging the user.

Fig. 3. The guide

2.4. *The guide character*

The guide character will be one of the most visual elements in the virtual world. Thus it is desirable that it is an aesthetically pleasing as possible. This means having a fairly detailed character model: fluid believable animations; and a clear, understandable voice with good lip-sync. Therefore there are three main areas to consider: the model, the animations and the speech system.

The character model is the 3D representation of the guide character. There is no one character who will be appropriate for all sites or users. Thus a library of characters are supported. There is further the dilemma of a character being too lifelike, yet to artificial. This problem is known as the "uncanny valley" [9]. The author suggested that to achieve the maximum level of familiarity with an object (in this case the guide), models should assume a medium level of human likeness, rather that striving for the highest possible level. Thus a character should be human, yet somewhat cartoon like. Figure 3 depicts the guide designed following this guideline.

The animation of the guide character must also use the chosen triangle mesh representation. However, triangle mesh only describes rigid bodies without any possibility for animation.

One approach to solve this problem is to represent an animation sequence as a set of keyframes, where each keyframe contains a complete triangle mesh model, just in a slightly altered pose that the preceding one. This is basically the approach of traditional (hand drawn) animation films. This method is simple to implement. However, the amount of storage required to maintain a suitable set of animations is not feasible. Further, this type of animation is closely tied to the specific character. Thus, it is next to impossible to transfer onto other characters.

A better solution is to use *skeletal animations* or *skin deformations* [16]. This approach introduces a skeleton onto which the triangle mesh is attached. Thus, the triangle mesh will deform in accordance with changes in the skeleton.

There are several ways of deforming a triangle mesh based of the skeleton model. The most common method is *vertex blending* or *matrix palette skinning*. Here each vertex in the mesh is linked to every bone that affects its movement. Mesh transformation is achieved by by transforming each vertex be each linked bone and blending the result according to the weights [16].

The final part of the animated guide is its ability to speak. The two main options that stand out is to either employ a text-to-speech system or use pre-recorded audio files.

2.5. *Guide behaviour*

The virtual guide character is not intended to be realistically human, however it must behave in an intuitive way that allows the user to get the best out of their museum experience, and access the knowledge they want to access. This means that its behaviour must be consistent with the user's expectations.

There should be at least two affecting forces behind the guide's behaviour: the first might be a state machine that can drive the guide's lower level functions such as walking, leading, following and discussing features. The second might be a learning system that allows the guide to learn about the user, present features from the best perspective, and suggest features to visit along a custom user-specic tour.

This section will deal with the first force: the state machine. State machines have been successfully used to drive virtual guide behaviour in related systems [13]. Such a state machine should preferably be able to handle a minimum set of functionality.

When the user first enters the system the guide should appear to them in a non-threatening manner and intro-

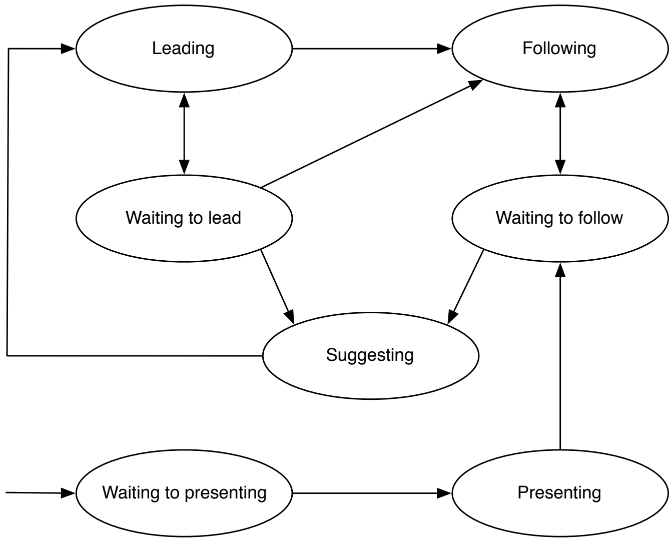

Fig. 4. The state diagram for the guide's behaviours

duce itself. This can be accomplished via an introduction script, consisting of a voice recording and potentially some custom animations. It should tell the user what to expect:

- It should explain that the user will see information icons and should touch them to hear about a feature.
- This should be accompanied by the guide pointing out an example information icon in the vicinity, and perhaps inviting the user to test it out. Once completed, the guide could say "very good" and move on.
- It should also explain that the user is free to walk away from an icon at any time, even if the guide is still talking. The user is the boss.
- Finally, the guide should invite the user to follow them to the rst feature, but also let them know that they are free to walk somewhere else if they want to. Again, they are the boss.
- The guide would then begin walking toward the first feature.

The state diagram for a minimum state machine as discussed here can be seen in Figure 4.

2.6. Personalisation

Without any information about the user, minimising the distance walked is probably the best effort a tour guide can make. If information is available, however, it can be used to customise and personalise the tour.

Information about the user can be used to form a "user model". Rich [12] proposes three dimensions of classication for user roles: whether they apply to a canonical user or an individual user, whether they are specified explicitly or implicitly inferred from behaviour, and whether they describe long-term or short-term characteristics.

The first dimension is easy to resolve: the virtual guide system will need to develop a user model that applies to the individual who is currently using the software.

The second dimension is more complex. Information can be acquired from the user by posing them explicit questions, such as "are you male or female?", or "are you interested in medieval architecture?". This method has the advantage that it is easy to implement, and provides the user with fully conscious control over their experience. If they indicate that they are not interested in medieval architecture, they will expect not to hear much about it. There are signicant disadvantages, however. Asking the user to fill out a long questionnaire before they begin using the system is likely to be frustrating. The alternative to explicit questions is implicit inference. Information about a user can be inferred from their behaviour over time, and used to create and update a user model. In the context of the virtual guide system, a user's interest in a given feature can be inferred from the amount of time they spend looking at it or hearing about it.

The third dimension quanties whether the user model describes long-term or short-term characteristics. Long-term characteristics include facts like the user's gender and age, their interests, and other information that will be stable over time. Short-term characteristics may include things like the user's current focus, how busy they are, etc. The virtual guide system is primarily concerned with long-term characteristics, as these will be most relevant for selecting and suggesting features of interes

As this type of virtual reality guide can be seen as a kind of recommender system, we suggest to adopt the use of stereotypes in the tradition of Rich [12]. We adopt a similar approach as the one presented in [7]. Here stereotype modelling is combined with Bayesian networks [11], [6] to mitigate the knowledge bottleneck problem and the cold start problem.

Stereotypes contain characteristics that are commonly ascribed to a set of people, such as AI students enjoy solving NP hard problems. These characteristics are known as facets. Typically they represent a certain quality and is coded with values ranging from -5 (strong dislike) to 5 (like a lot). Each facet also has a certainty value assigned, between 0 and 1000. So a facet will describe how much a person fitting a stereotype enjoys a certain quality and how sure the system is on this assumption.

However, to achieve the continuously improvement and refining of a user's user model a sound knowledge maintenance technique is required. As described in [7], Bayesian networks allows for a continuously update of belief in the facets' values and rating. Thus, adapting a, initially rough user model to fit the idiosyncrasies of specific users.

2.7. Rendering and Physics Simulator

A number of pre-existing frameworks exist to support the rapid development of 3D applications. Basing the implementation on one or more of these frameworks would clearly save significant time. For the implementation Delta3D [3], OGRE[1] and OpenSceneGraph (OSG)[2] was considered.

Delta3D library is actually a collection of a multitude of other open-source frameworks, integrated together and exposed under a common programming interface. The complete framework handles functionality such as basic application support, physics simulation, rendering via OpenGL, characters and animation, networking, terrain, audio, scripting and 3D editing. OSG is an object-oriented scene-graph framework and provides a hierarchical scene-graph representation on top of OpenGL's state-based immediate-mode interface for rendering. OGRE is, like OSG, is also only an object-oriented scene-graph framework. Like OSG, OGRE focuses entirely on 3D rendering, with no support for other functionality like audio or physics withouth adding other frameworks. But, unlike OSG, OGRE can target both OpenGL and Direct3D as it's underlying graphics API.

3. Initial implementation

An initial guide system that support a set of minimum functionallity was made. As the system should support stereo rendering and head-mounted displays for both virtual reality and augmented reality usage, it would be preferable if the framework should support these options. None of the suggested frameworks does so by default. The VRJuggler[3] is a virtual reality hardware abstraction framework designed to insulate application code from the specifics of different virtual reality hardware configurations. VRJuggler is extremely easy to integrate with OSG (more so then with OGRE), and although OSG is actually one of the frameworks integrated in Delta3D, OSG using VRJuggler was chosen instead of trying to integrate VRJuggler into Delta3D. Unlike Delta3D, using OSG and VRJuggler makes it easier to only add in functionality that is actually needed, to keep the framework as simple as possible.

Although parts in the Delta3D framework could have been used to add 3D character animation in a relatively easy way to the project, it was decided that a custom solution should be developed. This would provide the maximum degree of flexibility and extensibility in the animation system. A basic skeleton based animation system was implemented, supporting bone interpolation for blending key-frames. The final pose is then converted to a set of matrices, and uploaded to the video acceleration hardware where matrix palette skinning [16] is used. A small library of animations was defined, including standing idle, walking, talking and pointing with the left and the right hand. Mixing these animations is sufficient to produce reasonably convincing sequences. In addition to the key-frame based animation, some bones in the skeleton can be driven directly by algorithms. The neck bone is driven manually so that the guide turns its head to face the user when appropriate. This is much more convincing than always rotating the entire body to face the user.

Although VRJuggler offers an abstracted view of the underlying audio hardware, the interface it provides requires all sound clips to be loaded into memory before being used. This approach is unfortunately incompatible with the goal of streaming lengthy compressed voice recordings. A custom 3D audio system was therefore implemented, with support for both cached and streamed sounds. For each active 3D audio source, the system calculates the distance between the source and the listener, and uses this to attenuate the sound. The system uses the OGG/Vorbis audio codec, doing real-time decoding of Vorbis data using public-domain code written by Sean Barrett [1]

To handle physics, Nvidia PhysX [10] was used. Unlike the other frameworks discussed in this section, PhysX is not free or open-source software. Its license does however permit use in commercial and non-commercial products at no cost. Although other physics engines could be used, PhysX is especially targeted toward usage in computer games. This means that although it trades some precision for performance, it is still well suited for the intended usage. Also, the PhysX interface is extremely clear and concise as the user is well insulated from more complex uses of physics engines.

The virtual guide's behaviour was modelled using the state machine shown in Figure 4. The state machine in turn drives the guide's movement system, animation system and audio system. The movement system allows the guide to walk toward nominated locations, as well as turn to face nominated directions. This functionality is implemented in a naive fashion without regard to path planning algorithms. Only PhysX is used to clip movement to avoid walking through solid geometry or the user. This approach was surprisingly successful. Even if the guide did not always choose an optional path to get from point A to point B, the direct path plus movement clipping usually resulted in a successful walk. If the guide encountered an obstacle, it would walk along the edge of the obstacle until it reached the corner, then continue toward its goal. It was of course possible to create situations where the guide would get stuck. A real path planning solution such as A*[4] would produce a more optimal results (see e.g. [5] for an example).

[1] http://www.ogre3d.org
[2] http://www.openscenegraph.org
[3] http://www.vrjuggler.org

Only a bare minimum of functionallity of a planning system based on individualized user models and implicit inference has been implemented for intial testing.

4. Summary and future work

The initial implementation for a virtual guide system supporting both virtual and augmented reality congurations has been developed and presented. A broad cross-section of this architecture has then been implemented as a working prototype, using the Zion fortress as a test site. The architecture presented in this report addresses many of the limitations of earlier virtual museum guide systems. The virtual guide character is a fully animated human gure, able to look at the user and talk to them, as a real museum guide would.

The guide's behavioural and learning systems allow it to interact with the user in two of the most fundamental and intuitive ways: following and offer help or cultural information whenever the they stop to enjoy an exhibit. Or, it can lead the user toward exhibits that they may be interested in, implicitly providing navigation assistance in a very human and personal way. The guide takes hints from the user's own behaviour and switches seamlessly between these two modes. Although not yet fully implemented in the prototype, the architecture proposes a mechanism by which the guide can build up a model of the user's interests and preferences, and use this to personalise not only the tour route, but also the way individual exhibits are presented. This personalisation process is invisible to the user, removing the need for run time questionnaires or other explicit inputs.

Currently the virtual guide presents every feature the same way, by pointing at it, facing the user and speaking. This is adequate but not nearly as good as it could be. A scripting architecture that would allow fexible scripts to control the guide's behaviour while presenting individual features could be used to improve the behaviour.

The Behaviour Markup Language (BML) [8] is an XML-based language for expressing the behaviour of a virtual human. It is intended as an interface between a behavioural planner and a behavioural renderer, however it can also be used to create a kind of behavioural script, and fed directly to the character renderer. When used in this manner, it enables a very high-level expression of what a virtual character should do, with commands such as "look at", "smile", etc.

Through the use of BML, a content creator, or maybe an intelligent part of the system itself, could create a simple script for the guide to walk to certain places, look at certain things, etc, all with a view to demonstrating some artefact or activity of cultural relevance. For more complex interaction with the environment, one could even consider recording motions using motion capure hardware, and translate this to BML, for playback by the guide and/or other characters in the environment. Such a translator could be of interest in other areas as well.

Augmented reality environments with virtual geometry are perhaps the most sophisticated application of the virtual guide system. These environments allow a user to walk around a physically real space, and see a composite image showing virtual geometry overlaid or otherwise combined with a real-world view. Installations of this kind would typically be established at outdoor museums, like the Sverresborg Folk Museum, with the augmented reality equipment offered for rent to museum visitors. Unfortunately the expensive and fragile nature of the equipment means that it will be a while before this kind of installation becomes viable.

Although designed to support an augmented reality configuration, the present implementation work in a regular virtual reality context using a tracked HMD and pointing device, or on a normal computer using mouse and keyboard input. The user is now free to explore the virtual environment from the comfort of a fixed installation, without physically walking around the real-world site. The use of virtual reality instead of augmented reality is a viable way to reduce the cost of using the virtual guide system in a museum environment. The virtual guide can be set up as a fixed exhibit, to be used by one or two visitors at a time. This drastically reduces the cost of equipment, as well as the potential for damage and general wear and tear. A fixed exhibit could still employ leading-edge technology, such as a head mounted display and orientation tracking, to give visitors the best possible feeling of immersion.

As the implementation can use a normal computer with mouse and keyboard input, users could in theory also enjoy a virtual tour from the comfort of their own home.

We note that in the future, recent development in the availability of consumer stereoscopic television sets and low-cost input devices like Wii controllers or the upcoming xBox kinect system, might be used to allow a much higher degree of immersion for people enjoying a particular site or museum from the comfort of their own home.

References

[1] S. Barrett. Public domain ogg vorbis decoder. 2008.

[2] T. Bickmore, L. Pfeifer, D. Schulman, S. Perera, C. Senanayake, and I. Nazmi. Public displays of affect: Deploying relational agents in public spaces. In *CHI '08*, 2008.

[3] R. Darken and P. McDowell and Erik Johnson. The Delta3D open source game engine. *IEEE Computer Graphics and Applications*, 25(3):10–12, 2005.

[4] P. Hart, N. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. In *IEEE Transactions on Systems Science and Cybernetics*, 1968.

[5] D. Jan, A. Roque, A. Leuski, J. Morie, and D. Traum. A virtual tour guide for virtual worlds. In *Intelligent Virtual Agents, Proceedings of the 9th International Conference (IVA 2009)*, number 5773 in LNAI,

pages 372–378, Amsterdam, The Netherlands, September 2009. Springer.
[6] F. V. Jensen and T. D. Nielsen. *Bayesian Networks and Decision Graphs*. Springer, Berlin, Germany, 2007.
[7] A. Kofod-Petersen and H. Langseth. Tourist without a cause. In Şule Yildirim and A. Kofod-Petersen, editors, *Proceedings of the second Norwegian Artificial Intelligence Symposium*, Gøvik, Norway, November 2010. Tapir Akademiske Forlag.
[8] S. Kopp1, B. Krenn, S. Marsella, A. N. Marshall, C. Pelachaud, H. Pirker, K. R. Thorisson, and H. Vilhjalmsson. Towards a common framework for multimodal generation: The behavior markup language. In *Proceedings of the 6th International Conference in Intelligent Virtual Agents 2006*, pages 205–217, 2006.
[9] M. Mori. The uncanny valley. *Energy*, 7(4):33–35, 1970.
[10] NVIDIA. Physx. http://www.nvidia.com/object/physx_new.html Sept. 2010, 2010.
[11] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Kaufmann, San Mateo, CA., 1988.
[12] E. Rich. Users are individuals:- individualizing user models. In *International Journal of Man-Machine Studies*, pages 199–214, 1983.
[13] J. Serrano. Virtual guide for a virtual heritage environment. Master's thesis, Norwegian University of Science and Technology, 2004.
[14] F. Sparacino. The museum wearable: real-time sensor-driven undestanding of visitors' interests for personalized virtually-agumented museum experiences. In *Proceedings of museums and the web*, 2002.
[15] V. Vlahakis, A. Demiris, T. Pliakas, and N. Ioannidis. Desgin and application of an augmented reality system for continuous, context-sensitive guided tours of indoor and outdoor cultural sites and meseums. In *Proceedings of the 2003 conference on virtual reality, archeology and cultural heritage*, 2003.
[16] J. Weber. Run-time skin deformation. In *Proceedings of Game Developers Conference*, 2009.