

SESSION

HUMAN-COMPUTER INTERACTION, GUI AND USER INTERFACE ISSUES

Chair(s)

TBA

Enhancing Large Display Interaction with User Tracking Data

S. Thelen¹, M. Deller², J. Eichner¹, and A. Ebert¹

¹Computer Graphics and HCI Group, University of Kaiserslautern, Kaiserslautern, Germany

²Augmented Vision Group, German Research Center for Artificial Intelligence, Kaiserslautern, Germany

Abstract—*In this paper we present a method for enhancing interaction on large multitouch displays by exploiting tracking information of users working with the device. Conventional systems do not distinguish between individual users and process all touch gestures equally. The method we propose enables us to customize event handling and personalize the response of applications according to the user who triggered the event. In our setup, a conventional tabletop display is augmented with an overhead mounted depth sensor providing the information required to link touch events to persons interacting with the device. In the following, we introduce the approach and elaborate advanced interaction features that can be realized by applying our technique. Furthermore, we present a prototype that supports map-based collaboration on an interactive tabletop system and discuss details of the implementation in terms of hardware and software.*

Keywords: Multitouch, large display, user tracking, personalization, depth sensor

1. Introduction

Large multitouch displays are employed in collaborative scenarios to support users in a common task. Due to their extended screen real estate they provide sufficiently high numbers of pixels for the visualization of complex data and enable users to interact with an application simultaneously through touch gestures. While most systems are able to process gestures of multiple users, one limitation often remains: Conventional multitouch setups cannot distinguish events on a user basis, i.e., all events are processed equally, regardless of which user actually performed the gesture. However, many features that could improve collaboration rely on the ability to link an event to the person who triggered it.

Desktop applications, for instance, provide an edit mode that enables users to undo/redo their modifications by maintaining a global operation history. In collaborative scenarios the situation is more complex and a separate operation history has to be kept for each user to provide the opportunity to undo/redo modifications without affecting other collaborators. This requires that each operation can be linked to a group member which can be achieved by incorporating tracking information.

Another scenario that benefits from user tracking information is permission control. Members of a user group can play

different roles in a collaborative process and have different privileges. A session "chairman", for instance, is able to perform operations that normal users do not have the privileges for. Identifying the person behind a gesture adds further control to applications and lets them block actions that do not conform with the rights of a user.

Round-based interaction schemes that are known from classical board games offer a straight forward way to address these issues: In round-based scenarios the order in which users interact with applications is predefined, making it easy to link actions with certain users. However, round-based patterns limit concurrency and prevent people from freely interacting with the system.

In this paper we present an approach to deal with these shortcomings while imposing as few restrictions on user interaction as possible. Our technique induces minimal overhead and employs standard customer hardware components. It offers a flexible solution for linking events to users and employs tracking data from an infrared (IR) depth sensor which is mounted perpendicularly above the display. Tracking data is exploited for enhanced interaction metaphors which have been integrated into an interactive tabletop prototype. The prototype is employed in collaborative map-based decision making scenarios and features a 46" diffused illumination (DI) tabletop display and a Kinect depth sensor. The paper is structured as follows: First, we give an overview of related work that is relevant to our technique. We then discuss the general setup for systems implementing our approach and describe the peculiarities of a prototype installation. Furthermore, we outline details of the tracking algorithm applied to determine the user which performed a touch. Subsequently, we elaborate the potential of user tracking and discuss its integration into a prototype application. The paper concludes with a discussion and outlook on future work.

2. Related Work

Part of the techniques that have been proposed for personalizing touch interaction make use of additional more or less complex hardware components:

Dietz et al. [1] introduced DiamondTouch, a technique for creating touch sensitive display surfaces that allows multiple users to interact simultaneously. The method is based on location-dependent, modulated electric fields which are capacitively coupled through the user's body to receivers installed in the work environment. The receivers are integrated

3.1 Hardware Aspects

The prototype consists of a 46" Cell [13] tabletop system featuring multitouch interaction and a Kinect controller with built-in depth sensor that is mounted approx. 2.0m above the display.

The tabletop display is equipped with light-emitting diodes (LEDs) and emits IR light through the glass panel. Hands touching the surface reflect some of the emitted light back inside the display where it is captured by an IR camera to track hands and fingers.

Kinect [14] is a state-of-the-art game controller that enables gesture-based interaction with modern video games. Besides a multi-array microphone, the device features an RGB camera and an IR depth sensor. The depth sensor consists of an IR laser projector emitting a continuous pattern of structured light that is perceived by an IR camera inside the controller. The controller deciphers the received light coding and calculates depth values for each pixel of the sensor image.

In our setup, both devices are driven by separate threads running on dedicated machines. In order to communicate with each other, both exchange XML messages via Ethernet network.

3.2 Software Aspects

Applications running on the touch display are built upon the Cornerstone SDK [15] which processes touch inputs from an arbitrary quantity of fingers and hands. However, the framework has no information about objects above the display surface and is therefore unable to assign gestures to individual users. User tracking as described subsequently is implemented in the Kinect module.

The implementation of the Kinect module is based on OpenNI [16] and OpenCV [17]. OpenNI provides an interface to access Kinect's depth data and functions for further processing. We deliberately circumvent using the built-in hand tracker which has not been optimized for being employed in an overhead setup. However, the shape tracker extracting the shapes of users from a scene is rather robust and therefore used in the algorithm to determine u_{id} . OpenCV is used for all tasks that are not supported by OpenNI.

4. Linking Gestures and Users

The Kinect module T_c determines u_{id} for each touch detected at the tabletop. The coordinates of an event are given in the screen space of the multitouch application and first need to be converted into camera space. Then, the user whose hand is closest to the touch event is identified.

4.1 Calibration and Mapping

Since tabletop and overhead camera have different reference systems, the coordinates of a touch event given in

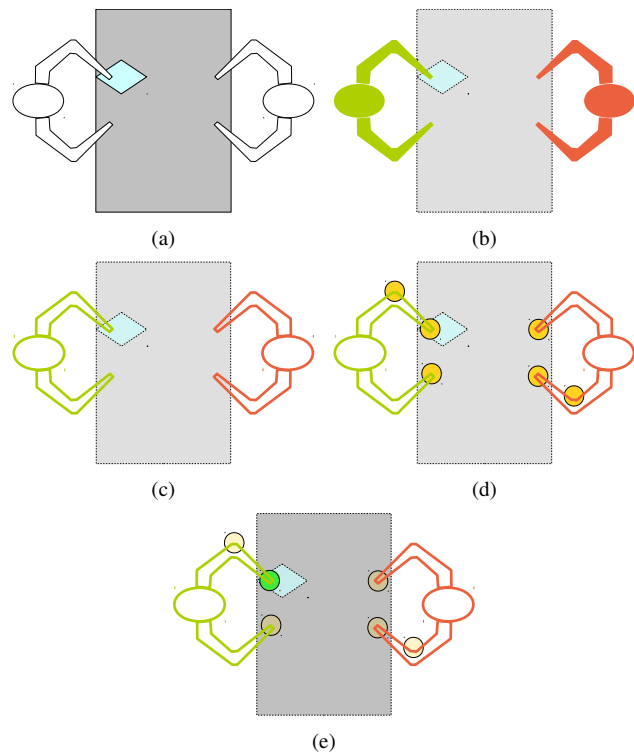


Fig. 2: Determining u_{id} a) touch detected, b) shape texture tex_s , c) contour texture tex_c , d) hand candidates including false positives, e) closest hand candidate and lookup of u_{id} from tex_s .

display screen space must first be transformed into camera space. The transformation is based on a homography, i.e., a linear mapping in projective space that preserves collinearities of points and concurrencies of lines. The homography maps points in tabletop coordinates to their corresponding points in the camera image. It is uniquely defined by four 2D point pair correspondences $\{c_i \leftrightarrow m_i\}_i$ and computed by determining the coefficients $\{h_{ij}\}_{i,j}$ of a 3×3 homography matrix H , so that $c_i = H \cdot m_i$. The point pair correspondences are established in a calibration stage by selecting the corners of the tabletop display in the camera image via mouse click.

The transformation between reference systems is straight forward:

$$(x_c, y_c, 1)^T = H \cdot (x_m, y_m, 1)^T.$$

4.2 Hand Tracking

The next step is to determine the ID of the user whose hand is closest to (x_c, y_c) . Therefore, all hands above the display surface must be identified.

Due to the DI display technique the volume above the surface is illuminated with IR light within approximately

1-2 cm. The illumination interferes with the IR pattern of the depth sensor, resulting in missing data points or incorrect depth values, and needs to be filtered out. Filtering is accomplished by a threshold criterion, setting regions with missing depth information (i.e., represented by zero depth values) or obviously wrong depth data (i.e., values greater than the distance between depth sensor and tabletop surface) to NaN. Recognizing fingers on the display surface is difficult due to the IR illumination and the limited resolution of the depth sensor. However, hand shapes clearly stand out against the NaN background. As a consequence, the system is designed to identify the shape of the associated hands. This approach works well in practice yielding correct user IDs for touch events. Determining u_{id} is accomplished as follows:

- 1) The shapes of all users are extracted and stored in a texture tex_s (see figure 2(b)). An entry $tex_s(x, y)$ is set to u_{id} if pixel (x, y) belongs to the shape of the user with ID u_{id} . The functionality is provided by the OpenNI [16] library (see `xn::UserGenerator`).
- 2) The contours of all user shapes are extracted from tex_s and stored in a texture tex_c (see figure 2(c)).
- 3) A k-curvature [18] algorithm is applied to tex_c in order to identify possible hand shapes based on the contour curvature (see figure 2(d)). Since the algorithm identifies false positives in regions with characteristic curvature, only those points (x_{hp}, y_{hp}) within a reasonable range d above the display surface will further be considered and stored in a list of candidates l_{hp} (see figure 1). The distance of a candidate to the surface can be determined from the depth texture tex_d , storing the distance of a point to the sensor in each pixel of the image.
- 4) For each element (x_{hp}, y_{hp}) in l_{hp} the system calculates the distance to (x_c, y_c) . If (x_{min}, y_{min}) is the point with minimal distance, the corresponding user ID can be retrieved from tex_s , i.e., $u_{id} = tex_s(x_{min}, y_{min})$ (see figure 2(e)).

Sensor images of the controller are given in figure 3. Figure 3(a) shows an on-top view of the current scene as perceived by the RGB camera of the controller. Figure 3(b) illustrates the result after performing steps 1-4 of the algorithm. For convenience, the results of all steps are displayed in one single texture: Each user is assigned a unique ID represented by a unique shape color. The extracted user shape contours are outlined in yellow color. White squares indicate regions with characteristic contour curvature for potential hand point candidates. Notice, the algorithm identifies all hands in the scene as candidates, including one false positive match near the shoulder of the left user. Based on their distance to the tabletop surface it is possible to eliminate ineligible candidates. Yellow circles indicate the hand points to be checked when a touch event is triggered.



(a)



(b)

Fig. 3: Sensor images a) On-top RGB view, b) hand tracking according to algorithm.

5. Contribution and Benefits

Information about the person behind a touch event offers a wide range of applications: Tracking information provides the basis for user-specific operation histories that enable users to selectively *undo or redo* their modifications without affecting other collaborators. Many large display applications neglect this feature and either provide no option for undoing operations or only "global" undo functionality. However, the more users work with an application the more important the opportunity to selectively reverse modifications of a single user becomes. Compared to other approaches based on mutually exclusive interaction regions for each user [8], our method provides

more flexibility and scales well with the total number of users interacting with an application.

Tabletop displays do not impose a particular physical orientation, i.e., users can interact with applications from any side of the device. However, a variety of GUI components must be oriented correctly so that users can work them, e.g., virtual keyboards, documents, inputs forms, and images. Our method can be employed to automatically *align* these elements relative to the person who is working with them. For instance, a document can be aligned so that the person who opened the text is able to read it without changing the position at the display or rotating the document.

The more users interact with a large display application in parallel the higher the risk of *gesture interference* becomes. Interferences arise, for instance, when one user tries to move a widget into one direction, while another user tries to move it into the opposite direction. Without user tracking, this gesture is interpreted as a pinch & zoom operation that unintentionally scales the element. The approach discussed in this paper offers a way to resolve such ambiguous situations. If multiple users intent to modify a widget, the element can be blocked in order to prevent undesired behavior. Alternatively, the application can process only the gestures of one of the collaborators.

The overhead depth sensor in our setup enables the implementation of *3D gestures* [3] that also consider the space above the display surface for interaction. Depth data can be combined with touch information from the tabletop in order realize new enhanced gestures, e.g., a one-finger touch event on the display defines the center of a zoom operation whose level is specified by the height of the secondary hand.

User tracking makes handling of *state information* straightforward. Input events may cause transformations between user states that can be exploited for interaction. A particular state may affect all actions of a single user and spare other collaborators, facilitating the implementation of user-specific edit modes, e.g., selection mode or delete mode.

Furthermore, users can be assigned different levels of access privileges, which can be exploited for *security control* and *group management*.

6. Prototype Application

We implemented a large display application that visualizes geo-referenced data on top of a three-dimensional terrain model (see figure 4). The application supports decision making processes in the agricultural industry and enables users to interactively explore geo-spatial information. Data is structured in layers whose items can be added, edited, or deleted from a scene. Layers store domain-specific information, such as field record data, GPS location information of agricultural machines, precipitation data, or humidity data. The terrain model can be explored in a 3D environment based on terrain data provided by the OpenStreetMap 3D project [19].

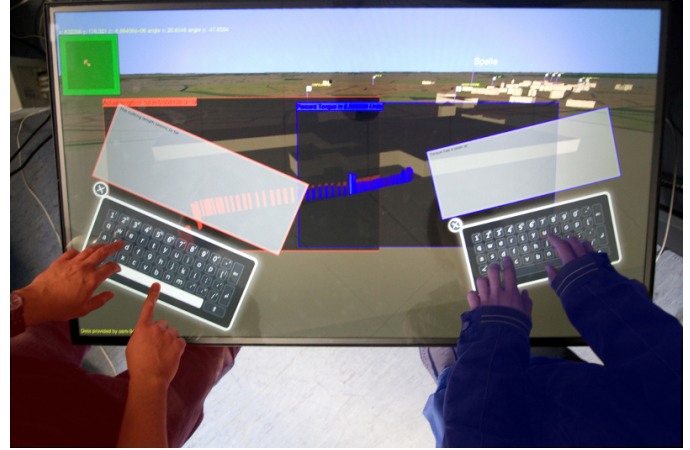


Fig. 4: Keyboards are aligned relative to users. Users are assigned unique colors to make elements distinguishable.

The system features user-specific undo functionality based on operation histories for a predefined set of operations. Users can revert geometric operations applied to graphical elements, i.e., rotations, scalings, and translations. In the future we plan to extend the module and to consider all types of user input.

When a 2D GUI element is opened, it is aligned relative to the user who is interacting with it (see figure 4). Since users may interact with the prototype from any side of the display, the feature significantly improves readability when working with text forms and virtual keyboards. The method mostly obviates the need for tedious manual alignments.

Further tracking information is exploited when editing data objects. Any modification of an object causes the application to change the color of the item to indicate which user is interacting with it. Furthermore, in order to demonstrate the concept of user-specific privileges, items can only be deleted by a designated user - the "chair" of the session. The chair is identified via a unique ID that is initially defined when starting the application.

In order to avoid gesture ambiguities and to prevent users from unintentionally transforming a scene, global view transformations are only processed if they do not interfere with other users, i.e., if exactly one user is interacting with the application.

7. Discussion

User tracking provides the opportunity to improve interaction on large displays by considering what is happening outside the display box. Our method facilitates the implementation of enhanced, personalized multitouch features based on additional information about users working with the application. A prototype could be realized with minimal hardware overhead using commodity components. In our example tracking information and depth data are obtained

from a Microsoft Kinect controller. Exploiting depth data for user tracking revealed a variety of advantages over alternative approaches: Tests confirmed that our approach is mostly immune to ambient light. Tracking works well under quickly changing light conditions which is a major advantage over RGB-based methods. Furthermore, we circumvent common problems with skin color segmentation that make tracking error-prone with respect to varying skin tones or require users to wear short-sleeved shirts.

Precision and robustness of the prototype are satisfactory. Tracking builds upon OpenNI's user generator which was optimized for usage in a horizontal setup where the controller is placed in front of the scene. As a consequence, the generator occasionally loses track of users or certain body segments when capturing scenes from its overhead position. Currently, we compensate for missing tracking information in a frame by maintaining a buffer storing the tracking results of the previous 16 frames.

Tracking data exploited in the prototype distinguishes between users but cannot determine their "real" identity. If a user leaves the scene and reenters after a certain period, the system assigns a new ID since it cannot distinguish between a new person joining the session and someone that has already been tracked. As a work-around, we offer an opportunity to manually edit and assign user IDs.

Regarding scalability, the prototype currently consists of one 46" tabletop display which is augmented by a single depth sensor. In case of larger display areas or multiple display surfaces, tracking would have to be distributed across multiple sensors. Sensor images would have to overlap in their border regions and would have to be stitched together in order to analyze the complete scene. Experiments revealed that IR interference between adjacent sensors is negligible. Thus, although not trivial, tracking could be distributed in order to achieve good scalability in terms of display size and user group size.

8. Conclusion and Future Work

We introduced an approach to incorporate user tracking data into large display multitouch applications. The method employs data from an overhead depth sensor to enhance interaction by exploiting information about the person behind a gesture. The "beyond the display" information enables us to customize the response of applications to different user inputs. We elaborated general applications of our technique and discussed the setup and implementation of a prototype system. The prototype is used by decision makers in an agricultural context and operates with minimal hardware overhead.

In the future we plan to extend our set of enhanced interaction metaphors and evaluate our method by comparing it to systems providing conventional multitouch features. Furthermore, we intend to replace OpenNI features that have been optimized for front view setups.

9. Acknowledgements

This work has been funded by the Federal Ministry of Education and Research (BMBF), Germany, as part of the project *iGreen*.

References

- [1] P. Dietz and D. Leigh, "DiamondTouch: A Multi-User Touch Technology," in *Proceedings of the 14th annual ACM symposium on User interface software and technology*, ser. UIST '01. New York, NY, USA: ACM, 2001, pp. 219–226.
- [2] N. Marquardt, J. Kiemer, and S. Greenberg, "What caused that touch?: expressive interaction with a surface through fiduciary-tagged gloves," in *ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '10. New York, NY, USA: ACM, 2010, pp. 139–142.
- [3] N. Marquardt, R. Jota, S. Greenberg, and J. A. Jorge, "The continuous interaction space: interaction techniques unifying touch and gesture on and above a digital surface," in *Proceedings of the 13th IFIP TC 13 international conference on Human-computer interaction - Volume Part III*, ser. INTERACT'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 461–476.
- [4] V. Roth, P. Schmidt, and B. Güldenring, "The IR ring: authenticating users' touches on a multi-touch display," in *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, ser. UIST '10. New York, NY, USA: ACM, 2010, pp. 259–262.
- [5] T. Meyer and D. Schmidt, "IdWristbands: IR-based user identification on multi-touch surfaces," in *ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '10. New York, NY, USA: ACM, 2010, pp. 277–278.
- [6] C. T. Dang, M. Straub, and E. André, "Hand Distinction for Multi-Touch Tabletop Interaction," in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '09. New York, NY, USA: ACM, 2009, pp. 101–108.
- [7] F. Wang, X. Cao, X. Ren, and P. Irani, "Detecting and Leveraging Finger Orientation for Interaction with Direct-Touch Surfaces," in *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, ser. UIST '09. New York, NY, USA: ACM, 2009, pp. 23–32.
- [8] K. Kyung Tae, T. Kulkarni, and N. Elmqvist, "Interaction Workspaces: Identity Tracking for Multi-user Collaboration on Camera-based Multi-touch Tabletops," in *Proceedings of the Workshop on Collaborative Visualization on Interactive Surfaces*, 2009.
- [9] K. C. Dohse, T. Dohse, J. D. Still, and D. J. Parkhurst, "Enhancing Multi-user Interaction with Multi-touch Tabletop Displays Using Hand Tracking," *International Conference on Advances in Computer-Human Interaction*, vol. 0, pp. 297–302, 2008.
- [10] D. Schmidt, "Know Thy Toucher," in *CHI 2009 Workshop: Multitouch and Surface Computing*, 2009.
- [11] A. D. Wilson, "Using a Depth Camera as a Touch Sensor," in *ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '10. New York, NY, USA: ACM, 2010, pp. 69–72.
- [12] A. D. Wilson and H. Benko, "Combining Multiple Depth Cameras and Projectors for Interactions On, Above and Between Surfaces," in *Proceedings of the 23rd annual ACM symposium on User interface software and technology*, ser. UIST '10. New York, NY, USA: ACM, 2010, pp. 273–282.
- [13] "Website of MultiTouch Ltd." <http://multitouch.fi/>.
- [14] "Website of the Xbox 360 Kinect Controller," <http://www.xbox.com/en-US/kinect>.
- [15] "Website of Cornerstone," <http://cornerstone.multitouch.fi/>.
- [16] "Website of OpenNI," <http://www.openni.org/>.
- [17] "Website of OpenCV," <http://opencv.willowgarage.com/wiki/>.
- [18] J. Segen and S. Kumar, "Human-Computer Interaction Using Gesture Recognition and 3D Hand Tracking," in *Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on*, oct 1998, pp. 188–192 vol.3.
- [19] "Website of the OpenStreet3D Project," <http://www.osm-3d.org/home.en.htm>.

Implementation of a Touch Based Graphical User Interface for Semantic Information System

Sergio L. Mendoza¹, Adrienne Lam¹, John Paul Adigwu¹, Aleksander Milshteyn¹, Jorge Estrada¹,
Evan Tsai¹, Gabriel Nunez¹, Gregory Carter¹, Neil Arellano¹, Airs Lin¹,
Dr. Charles Liu¹, Dr. Helen Boussalis¹

¹Structures, Propulsion and Control Engineering (SPACE) University Research Center (URC)
College of Engineering, Computer Science, and Technology
California State University, Los Angeles
5151 State University Drive
Los Angeles, CA 90032

Abstract - *This paper focuses on the development and integration of the next generation Graphical User Interface (GUI) into a Semantic Information System (SIS) platform [1]. The SIS platform is originally designed by the CSULA SPACE Center¹. It allows for collaboration between multiple users in a dynamic environment, facilitating users' ability to generate metadata and content in real-time. These are stored in a database while their semantics are represented in the XML-based tree-structure display [2]. While the traditional GUI provides navigation via a myriad of mouse clicks and cumbersome pull-down menus, we present an effective solution to modernize this GUI by implementing touch-screen capabilities to work in unison with an OpenGL 3D representation of the tree-structure [3]. The tree structure, which consists of objects being rendered, is then visually manipulated using finger gestures such as flick, scroll and pinch-zoom, providing seamless navigation through content.*

Keywords: Graphical User Interface (GUI), Semantic Information System (SIS), Touch Screen, XML, OpenGL.

1 Introduction

Today's applications are moving towards touch-screen based interfaces. Traditional GUIs have lacked touch-screens partly because these hardware devices were costly and hence did not have the programming support that is available today. As the costs of hardware have dropped and the technical support for touch-screen capabilities has become widely available, touch screen interfaces are now widespread on the market. Compared to traditional mouse and keyboard devices, touch-screen movements are now used for more quick and efficient navigation.

The Semantic Information System (SIS) is developed as a supplement for dynamic project environments, drawing upon characteristics from both a "Semantic Web" and a "Semantic Network" while building on previous work [4]. The SIS addresses issues associated with a traditional working environment; information is generally scattered and traditional technologies have less support for collaborative project environments. Moreover, representing semantic

relationships in a 2-D space is not practical when there is a large overlap between information.

The purpose of a Semantic Web is to provide a means for information to be processed by machines, allowing for better organization and retrieval of such information. On the other hand, a Semantic Network emphasizes how nodes of information relate to each other to form a network and focuses on how to visually represent this network of relationships. A previously explored method of referencing information in a machine-friendly manner can be seen in the implementation of the Aerospace Information Server, in which XML (Extensible Markup Language) is used to store tags and other metadata [5].

The SIS can be used as a collaborative work tool or as a central location for organizing relevant project knowledge. In such a scenario it is intended to increase synchronization of work, improve the integration of work on a larger scale project, provide a shared understanding of a project's progress among members and expand the accessibility of information.

Capabilities incorporated into the SIS include an integrated web browser, a scraping function for gathering content and a 3-D representation of semantic relationships. A key aspect of the SIS is the ability for users to generate new content. This can be done by gathering information from the web or incorporating local content into the network. In this context, new nodes in the network are referred to as objects.

2 SIS Content Generation

2.1 The Tree Structure

During earlier renditions of the Semantic Network, a 2-D tree structure was implemented to represent relationships between objects. New objects and their content are incorporated via the Scraping Tool, which will be described below. The contents of the tree are stored in XML format to take advantage of XML's inherent hierarchical structure. The Tree display is a convenient interface for users to browse and access information on their network.

When an object in the tree is selected, the metadata is immediately displayed to the user. In the current application, the metadata consists of basic parameters such as Title, Date

Created and Description. A recently added parameter is a list of related objects. This is necessary due to the fact that a 2-D display cannot show cross-category relations. However, the implementation of a 3-D tree display to better visualize semantic relations is described below. Rather than overwhelm the user with the main content contained in each object, this allows them to choose whether or not this object is relevant to their search criteria. The tree can be expanded and collapsed while navigating the available content.

2.2 The Scraping Tool

The mechanism for gathering content and populating the Tree Structure is the Scraping Tool. Through this process (referred to as publishing), users can gather relevant content from different sources and store or integrate them in a central location. The Scraping Tool is initiated upon user selection and provides a pop-up menu with several steps, as seen in Figure 1.

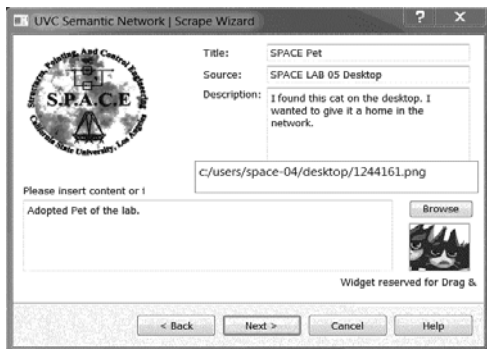


Figure 1: Scraping Tool Object Generation

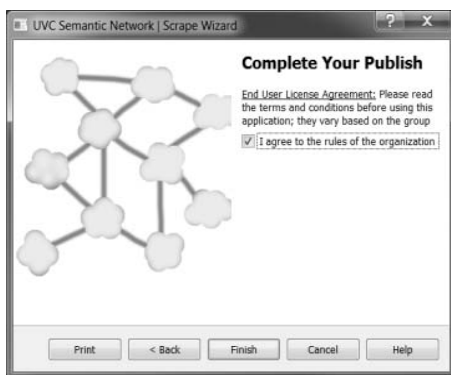


Figure 1 cont.: Scraping Tool Object Generation

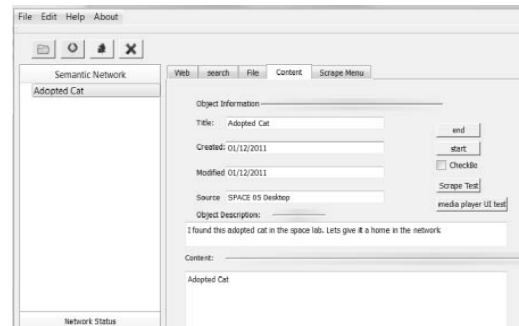


Figure 2: Traditional GUI Display after Object Generation

After an object is published, its metadata is stored in a new entry in a database while its content (Word document, PDF, image, audio, video, etc) is stored on a file server. The retrieval of metadata and content is conducted through these respective channels. Their implementation will be discussed in further detail below. Steps involving the database and file server are transparent to the user, occurring in the background of the application. The Tree Structure is immediately updated after objects are created, to allow access to new content.

It was previously mentioned that the content itself can consist of project work itself or merely information related to the project. In either case, the SIS is more effective when the scope of its usage is more focused. This is a characteristic inherited from the Semantic Network; as it becomes broader in focus its benefits are minimized. Seeking out information in an unfocused Semantic Network would bear little difference to a search on the Internet. In this area it is dependent on the user to add content that is relevant to project or knowledge base.

2.3 Content and Metadata Storage

The database serves as a means for storing metadata, facilitating searching functions and managing user accounts. The objects within the network can be searched by the metadata parameters such as date or file type. For larger and more heavily populated networks, the ability to search within specific branches can be compounded on top of the metadata parameters.

Each object within the database has a unique ID associated with it. These IDs are present in the XML tree file and are called when a user selects the object. The metadata for that particular object is pulled from the database by using the object's ID. As the Tree is populated with new objects, their respective database ID's are added to the XML tree file to provide a reference.

The connection between the XML tree and the database is necessary for retrieving content as well. In addition to the aforementioned metadata parameters, the database also keeps a listing of files for each object. These must be known in order to retrieve files from the FTP server. Whenever a user views or downloads an object's content, it is actually being retrieved from a file server. Again, these processes are transparent to the user. The database can be used to control

users' access to content via the objects that they can view. For redundancy, a user's access to content on the file server can also be adjusted according to their account.

3 Traditional GUI Operations

3.1 SIS Traditional GUI Operations

This section describes the operations for the traditional GUI used in displaying a tree structure. Previous GUI shows a hierarchical tree structure as the sole method to navigate through the Information system. Such methods of navigating through files have become uncommon and have little visual impact on the user. On the left hand side in Figure 3, the tree structure is displayed using XML giving the user the ability to expand and collapse parent/child content and see what branches it contains. Users click on the arrow next to a brief description which expands the tree.

When users select a node of information its respective metadata is displayed on the content page. The metadata includes the following node information: the title, date created, date modified, source information, object description, and file type. This tab allows modification and creation of nodes on the user loaded semantic tree structure. Users can also open a different tree by clicking on the folder button on the top right corner below the file drop down menu. Nodes can easily be deleted when selected by cursor followed by pressing the delete button.

The SIS also includes an embedded browser and a web-page parser. When the user visits a certain web-page s/he is able to utilize the current version of the parser for identifying related terms. Current version of the parser is limited in listing several related objects to the user-provided keyword within the web-page. After entering the keyword in the Search Tab and invoking "Search" function, the web-site in the embedded browser is parsed for the input terms specified by the user.

The Display Tab returns the output results with words preceding and succeeding the term; providing a context for the result. Instead of looking back through the web-page to determine the context of the keyword, the user can quickly identify the relevance of results on the results tab. When users find content relevant to their query, they can create a new object on the tree via the Scraping Tool or update content on an existing object.

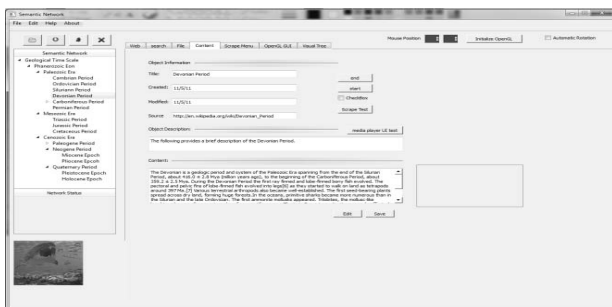


Figure 3: Traditional GUI of the displayed XML Tree and Object Information

3.2 Shortcomings of Traditional GUI

The traditional GUI represents an effort to display the content on the network but it falls short when it comes to graphical capabilities and user control. The traditional GUI does not offer an instant visual representation of the content. SIS participants will have an advantage in navigating through SIS database when offered a visual representation of the content, as it will aid them in faster decision-making by instantly identifying the necessary information.

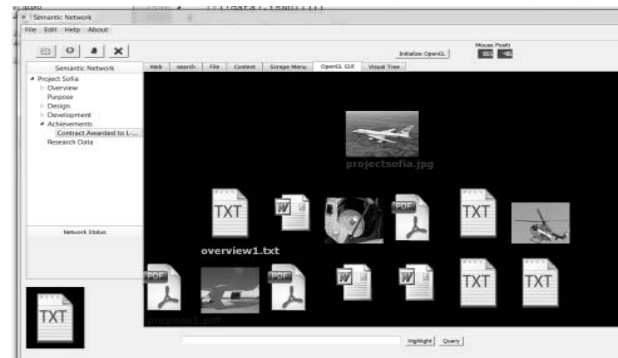


Figure 4: 2D OpenGL-based Simplistic GUI

4 Modernized GUI for SIS Operations

4.1 OpenGL within Qt

The Open Graphics Library (OpenGL) Application Programming Interface (API) was chosen due to its multi platform capabilities. Qt has OpenGL library available as part of its SDK, hence QGL Widget is used within the original SIS GUI application [6].

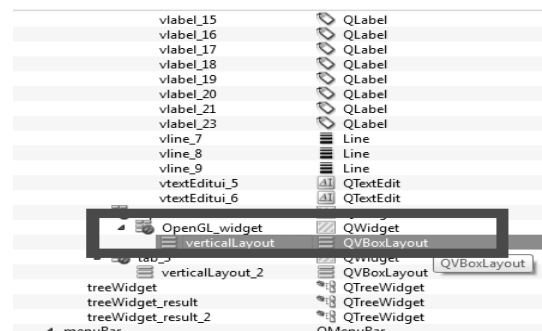


Figure 5: Vertical Layout is an embedded object of the QWidget. (QGL Widget is added onto the Vertical Layout)

OpenGL works through a low level programming, and hence allows the programmer to control the rendering of the environment. The programming language allows for what is called texture mapping. This scheme allows the programmer to map pixels around primitive polygons.

When doing so the pixels conform to the primitive shape instantiated (e.g. flat picture mapped to a quad). In order to visually represent the objects from SIS database, first the

primitive polygons are mapped on the QGL Widget on the stationary coordinates.

The image files representing information nodes are loaded into memory and converted to GL data. Then, the individual textures are generated for each specific GL image that represents its own information node, and finally these GL images are bounded to the specific quads (primitive polygons) and become textures. This process is called texture mapping.

4.2 Modernized GUI Features

Current touch-screen features of the Modernized GUI include regular zoom in/out and tree rotation functionalities. The later allows the user to rotate the tree by ten degrees either along the x or y axes using either mouse or touch-screen display.



Figure 6: Touch-based Zoom and Rotation Buttons

Besides default zoom operations, the right portion of the QGL Widget region is reserved for what is referred to as a "Local Zoom Menu". It is designed to conserve visual space on the QGL Widget. Upon user-selection of a necessary level of viewing, the local zoom capability will filter out (visually remove) the most distant levels of information from the QGL Widget.

This feature will help user to visually concentrate on the specific level of information, displaying only the selected level, as well as the nearest preceding and succeeding levels of information. The visual data of all other levels within the tree structure will be temporary concealed from the user.

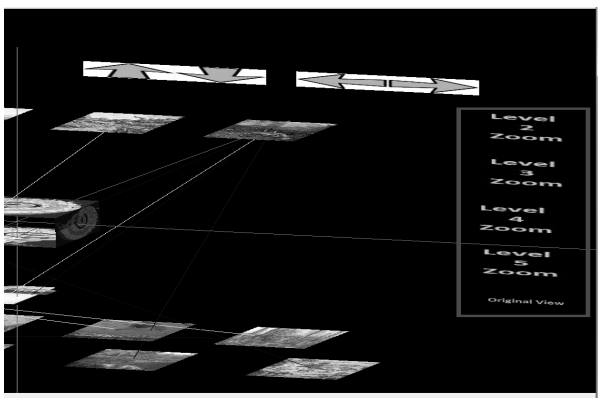


Figure 7: Local Level Zoom and Original View Features

4.3 Implementation and Results

Two sets of operations needed to be executed in order to project the visual information on the QGL Widget. The first set of operation is necessary to enable OpenGL environment and parsing of the existing XML tree. Then, the second set of operations results in visual data rendering. Both sets are shown in Figure 8.

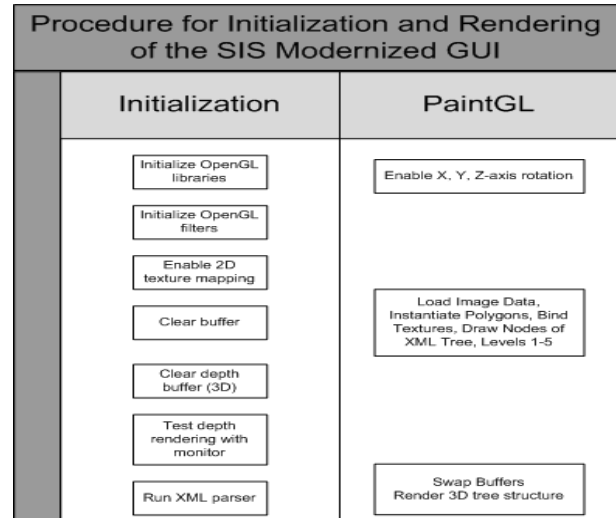


Figure 8: SIS Modernized GUI Flow Chart for OpenGL Initialization and Rendering

As it was previously mentioned throughout the paper, the Qt SDK comes with the OpenGL library support. Configuring Qt project file with the following switch command enables OpenGL support "Qt += opengl". Then the event filters have to be installed for all the widgets. This is done in order for the widget to be able to recognize the mouse and touch-based events. If the event filter will not be enabled, then the widget will not respond to the user commands.

The third step for initializing OpenGL is to enable 2D texture mapping. This will allow image data to be bound to the primitive polygons. In this project, quads are used for the image texturing.

After enabling the texture mapping, the QGL Widgets color and depth buffers have to be reset and cleared from any information. The depth buffer has to be tested in order to remove hidden surfaces from the QGL Widget. Finally and XML parser reads the XML file and populates information in the traditional GUI fields as well as loads the textual node information in the hierarchical structure. The parser also identifies the names of the files to be loaded for the OpenGL texturing.

The PaintGL process deals explicitly with mapping the textures as well as keeping track of their positioning. First important procedure is to enable three axes to respond to the user-controlled rotation. When user touches or clicks an arrow button the XML tree will rotate around x or y (up/down or left/right) axes with the 10 degrees interval. It will allow the user a 360 degrees view of the whole tree structure.

The second block under PaintGL column represents a set of graphics operations including the data loading from the image file, converting the image data to GL data type (for further manipulation), binding textures onto the primitive quads, and finally mapping the textures within the QGL Widget.

The final step is to swap the buffers in order for QGL Widget to update all the coordinate information. Thus, any changes made on the QGL Widget require the buffer swapping function call at the end of all the drawing procedures.

Figures 9 and 10 show three-level and five-level models of the Semantic Information System's XML tree in a three-dimensional environment.

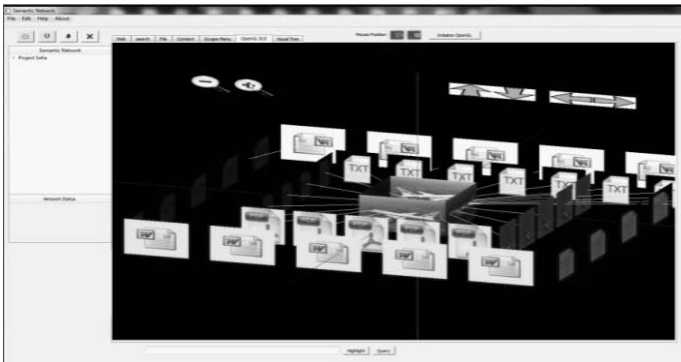


Figure 9: Example of a 3D Three-Level Modernized GUI in 3D

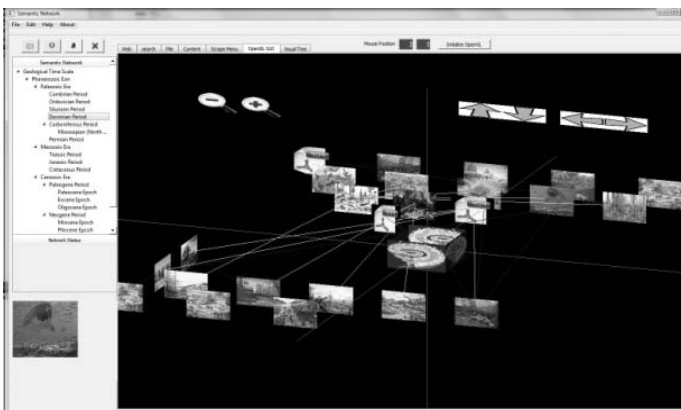


Figure 10: Example of a 3D Five-Level Modernized GUI in 3D

5 Conclusion

In this paper we have discussed how the Next Generation GUI allows SIS platform participants to effectively recognize, edit, and manipulate visual information. Accessing the content is accomplished via traditional user interface.

The traditional user interface is simple and straightforward at retrieving the content but lacks the visual aspects and touch-screen capabilities that play important roles in presenting information to the user.

The next-generation GUI complements this SIS application by adding the visual aids and touch-based control panel, which allow users to navigate the SIS database in a

seamless manner. By providing operations such as scroll, flick, and pinch zoom users can effectively search for content in a snap.

6 References

- [1] J. P. Adigwu, Dr. H. Boussalis. "Semantic Information System : Applicaitons in K-12 Education," *The Journal of Computing Sciences in Colleges (Vol . 26, Num. 4) , April 2011.*
- [2] "XML Marker" URL: <http://symbolclick.com/navigation.htm>
- [3] "OpenGL Coding Resources" URL: <http://www.opengl.org/resources>
- [4] D. Huynh, S. Mazzocchi, R. Lee. "SIMILE Project. Piggy Bank". Massachusetts Institute of Technology. 2003-2008 URL: http://simile.mit.edu/wiki/Piggy_Bank
- [5] A. Alegre, A. Beltran, J. Estrada, B. Coalson, A. Milshteyn, C. Liu, H. Boussalis, "Development and Implementation of an Information Server for Webbased Education in Astronomy." Proceedings of the International Joint Conferences on Computer, Information, Systems, Sciences, and Engineering (CISSE) 2007.
- [6] "QT Cross-Platform Application and UI Framework" URL: <http://qt.nokia.com/>

Dynamic Depth and Immersive 3D Interaction

Sugon Kim, Min-Joon Kim, and Chieteuk Ahn

School of Integrated Technology, Yonsei University, Incheon, South Korea

Abstract—*This paper focuses on a new immersive 3D interaction method where depth has more emphasis. We first define what we consider dynamic and static depth information. We propose a new user interface method with emphasis on depth-based 3D environments. The main idea is linking additional information on objects' superficial depth that we know as closeness. On top of that we propose a system where the user can easily interact with the 3D objects by controlling and defining the depth of the objects. We envision an array of applications where dynamic depth and depth interaction can take place, despite the fact that our current prototype demonstrates only basic interaction at the moment.*

Keywords: Immersive 3D interaction, 3D interaction, depth manipulation, active depth, depth information

1. Introduction

In recent years we have seen a massive increase of products and research in 3D contents and its related fields of study. Starting with the movie "Avatar" in 2009, more and more 3D movies have been released in succession. Wide success in the movie industry has propelled 3D technology to its current state [1]. With 3D movies and television shows becoming more and more widespread with the commercialization of 3D television sets, it is possible to conclude that we will see more 3D contents in the foreseeable future. With the rise of these 3D contents comes the question: how do we interact with these 3D contents?

Traditional mice have been such a success in navigating 2D environments that we are so accustomed to using each day. But these tend to be limited in degrees of freedom for easy 3D navigation [2]. On the other hand advanced devices developed in order to deal with specific tasks in 3D. Numerous products have been developed, starting from familiar devices such as the traditional mouse to the more advanced 3D mouse, and even gesture recognizing devices such as Nintendo's Wii™[3] and Microsoft's Kinect™[4]. Such devices have been developed for gaming purposes, but they also have contributed towards making 3D navigation and 3D interaction in general easier to the user. While these devices have greatly improved 3D user interactions it does not mean that such devices do not have their shortcomings.

3D mice, although have the degrees of freedom to successfully navigate 3D scenes, tend to lack the haptic feedback. The lack of feedback to the user tends to make a device more difficult to use [2]. While the suggested solution would be

adding some sort of feedback in the form of pseudo-haptics, it is not easy to provide appropriate responses to the users relying solely on software. Another issue with 3D mice is the high learning curve the users go through to get comfortable with the device. While more degrees of freedom are added for easier navigation, it may not be so in the beginning as the user is only able to achieve the ease of use through repeated trials and errors. For professionals this may not pose a problem, as they are required to use the device more often than not. However, to a regular, non-professional user 3D mice may not suit them particularly well. Even with some difficulties, it is more likely that the average user will be able to navigate and manipulate 3D objects with a 2D mouse. 3D navigation devices should be accessible to a wide range of users, and to achieve that intuitiveness is key.

Devices such as the Wii™ and Kinect™[4] have extended the ways a user interacts with a computer with their versatility. Microsoft's Kinect is widely used in many areas such as gesture recognition and depth mapping, and Nintendo's WiiMote has found usage as a device for 3D interaction [3][5]. Not only these devices are meant for games, but also they have been used for many research topics across the globe. While intuitive and easy for the user to "just use" the device, it has been observed that using these devices for long periods of time cause fatigue to the user.

A common topic that the devices mentioned above do not consider is the manipulation of depth, or rather the proactivity of depth. In our own opinion, depth has been a passive aspect in many 3D interfaces. Objects have been given depth values in order to show the closeness of an object to the users. Zooming in or out and differing the views of the environment has long been used and is still the same. It can be said that depth data has only been used in terms of the traditional meaning of depth: how close or how far? In this research we give additional values to depth, such as information or multimedia data, in order to make depth a more active part of a 3D interface. Users will be able to interact with an object's depth and its additional information associated with their depth values. This is an attempt to extend the user's control and allow 3D interaction more intuitive and easier to engage.

First, we provide background information and related work in section 2. In section 3 we discuss our approach and concept as well as some example applications. We then present our prototype in section 4 along with results of our experiment demonstrations and conclude our paper in section 5.

2. Related Works

While there are many papers and research regarding virtual reality interfaces, we have not been able to find a paper that incorporates interaction with depth as in this paper. Regardless, our research is relatable to several research topics in regards to virtual reality and virtual interfaces. This paper builds upon the following areas of previous research: 1) gesture recognition and virtual interfaces; and 2) varying the stereoscopic 3d display to the user based on head/eye location.

2.1 Gesture Recognition and Virtual Interfaces

Currently the way we interact with a computer is through a monitor and a set of input methods such as the keyboard and mouse combination or one's finger to name a few. We have been using this idea since the first graphical user interface came to desktop. Here we have all our files in a two-dimensional space where depth is nonexistent. We are familiar to this desktop concept and we have been using this for quite some time.

Nowadays we have gone beyond the desktop concept, adding in three-dimensional information and trying to simulate not only one's desk, but also one's entire workspace. New interaction methods have been proposed for out-of-screen objects and different input methods such as gesture in a virtual environment. Du et al. [6] use a 3D TV and Wii remote devices to simulate interaction with out-of-screen 3D objects displayed on the television. They envision menu interactions, volume control, and browsing functions with out-of-screen interactions and gestures.

Mine et al. [7] has provided a set of general rules to follow in virtual-environment interactions. The problem is that in virtual environments, the users lack haptic feedback in order to be able to properly orient themselves to the objects they are manipulating. They suggest the use of proprioception for the control of such objects.

Many studies do implement proprioception in some ways; as we can see in studies conducted use gestures, cameras, and other methods in order to correctly position the user to the relative material on the screen. Keefe et al. [8] have developed an artistic medium called CavePainting, which is a fully immersive 3d interface. The users are able to paint on a screen "directly" using a brush or various other methods such as throwing paint on the canvas using buckets.

Banerjee et al. present Pointable [9], which is an in-air pointing technique to manipulate out-of-reach targets on tabletops. They have demonstrated a gesture-based selection technique using one's index finger and thumb, and have shown accurate results in terms of selecting out-of-reach targets. While they are virtual environment input methods using cursors and other small hand-held devices such as CavePainting mentioned above and Swordplay [10], Pointable has been able to deliver accurate results using one's fingers.

In this research, we intend to combine elements of a 2D interface and that of a virtual reality interface in order to efficiently and accurately control depth and also objects of interests to the user.

2.2 Varying Stereoscopic 3D Displays to the User based on Head/Eye Tracking

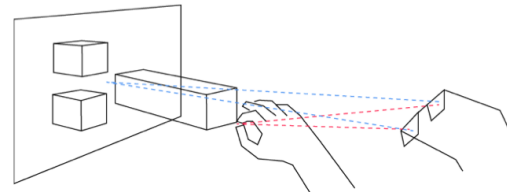


Fig. 1: Concept diagram

Our intention is to combine 2D interface and 3D interface elements together, and this includes displaying the display according to the user's gaze or location of the head and its features. Such research done by Kwon et al. [11], Newman, et al. [12], and Chen, et al. [13] have been tremendous help in terms of implementing our own prototype. While our initial solution does not include the use of eye and gaze tracking, we do use facial features in order to figure out the viewing angle of the user so proper adjustments can be made in the display.

3. Dynamic Depth and Depth Interaction

Before we further explain our ideas, we will illustrate what we define as "dynamic" depth information and "static" depth information. Static depth information is the depth value itself. Information such as how far an object is away from the base of the screen is an example. Zooming in and out and changing the display also is considered static depth.

On the other hand, we consider dynamic depth information to be adding information relative to the object's depth value. We have envisioned some examples that can be implemented using our system. One example is a 3D file navigation system. One can imagine this as a 3D cabinet. A user can "press" on a folder and pull his/her finger away from the monitor. The contents of that particular folder will reveal the contents, just like an actual file cabinet. One can then navigate through the files inside the selected folder.

Another application can be found in architecture and 3D modeling. Take a multi-story building for example. Each story's floor plan and other information can be linked to the depth (z) value of one's finger. Instead of the traditional method of viewing each story by going "into" the floor, the user can use his or her finger to navigate each floor. The first floor can be accessed by having one's finger closer to the monitor. In order to view higher stories the user will

only have to move his or her finger further away from the monitor.

The last example we present applies to e-book interfaces. This does not necessarily apply to dynamic depth, but in terms of bettering the user experience. Currently visualizing page turning is available whether it is on the web or on a tablet PC such as the iPad. However, we have envisioned an advancement of the current state by having the page actually move out of the screen while the user is turning the page. The 3D monitor will display the page turning based on the movement of the fingers of the users.



Fig. 2: Depth-based e-book interface

In order to demonstrate the 3D interaction we constructed a simple prototype, which consists of a monitor with webcams attached to the top and sides. There are multiple methods that we have designed to implement our system, but for the sake of brevity we have decided on using readily available devices and software. Our initial setup consists of a regular monitor with multiple webcams. While we have not discussed in detail the pros and cons of different prototype components, we selected this setup as the base that we could build upon.

We envision this prototype to be a connection between the traditional 2D desktop interfaces and 3D virtual interfaces. Because the objects itself are displayed in a 2D manner until an object is selected and is forced to come out of the screen by the user. With the user using his or her fingers and touch mechanisms for selecting the object, the interface is not entirely traditional nor is it entirely a virtual one. There are elements of both types of interfaces present in our prototype.

We have chosen three PlayStation™ Eye™ cameras for our webcam for easy color detection, head tracking, and its availability. One camera is mounted on the left side of the monitor, while the remaining two are placed on top of the monitor. We used CodeLaboratories™ CL-Eye SDK [14] and OpenCV for the base development platforms of our interface.

We categorize the webcams based on its function in our

model. In the first group, one camera will be used for head tracking. Head tracking is used in our prototype in order to differ the view based on how the user is looking at the screen. In the second group of cameras, two cameras are used to detect the position of the user's fingers in x, y, and z-axes. The camera mounted on the side will detect the y-position of the finger and the camera placed on top of the monitor will compute the x-position of the finger. Both cameras will be used in coordination to get the z-position of the finger.

4. Prototype Implementation

While the prototype description mentioned above is our main intent and concept, we have created an alpha version for the sake of fast prototyping and quick implementation. In our initial prototype of the product, we use a single camera that is pointed towards the computer screen. The camera replicates the user's point of view, thus being able to obtain the x, y, and z-coordinates of the user's fingers. The x and y-coordinates are obviously tracked by the vertical and horizontal movements of the finger. The calculations have been made easier by wearing different colored caps on each finger. The program will only detect and read data from the colored objects.



Fig. 3: Prototype design

The z-coordinates are estimated by the size of the hand the camera reads in. In order to get a more accurate estimation of depth we have attached a colored square on the backside of the hand. The camera will compute the size of the square in order to give the object a z-depth value.

We implemented few basic interactions in our prototype: moving an object, zooming in and out, and rotating an object. But before all of these the first function to be taken care of is simulating a mouse click. To simulate a mouse-click, the program simply calculates the distance between the colored caps. If the program determines that the two colored caps are touching each other, then it is considered a "click" or selection. Once the desired object is selected, the user is only able to interact with that object only. The user is able to "let go" of the object after performing a double click.

Once an object is selected, the user can move his or her fingers to a new location. If the user performs another click and moves his or her fingers then the selected object will

follow the fingers and relocate. In order to place the object on the desired location a double click will suffice.

In order to zoom-in or out, the program uses the depth (z) value that it obtains from the fingers' location as mentioned earlier in this section. Once an object is selected the user can either zoom in or out by distancing his or her fingers from the screen. We have designed the program so that a pulling gesture would bring the desired object closer to the user, thus zooming in. Zooming out the object can be achieved by "pushing" it away.

The last functionality to introduce is rotating the object. Once the user selects the object by performing the clicking (pinching) action and holding it, the user can rotate the object to his or her freedom. The user will lose control of the object by letting go of the hold of the clicking action.

5. Conclusion

In this paper we discussed giving 3D objects' depth more information than the traditional near/far-ness and also presented a way to interact with those objects. Traditionally, depth has been related to how far or how close an object is to the user. We believe that this has somewhat limited how users can interact in a 3D scene. By supplementing depth data with information such as files, floor-by-floor information in architecture and other examples mentioned above depth can assume a bigger role in 3D user interaction.

6. Future Works

We imagine implementing the system from a single-camera prototype to a multi-camera prototype will greatly increase its accuracy in detecting the three-dimensional coordinates of the finger. Greater accuracy in coordinate calculations means a better display and interaction of 3D objects. We would also like to try different combinations of cameras in order to test out which setup would be best fit for our idea. The use of depth cameras has been discussed as well as fish-eye lenses. With a new setup most definitely the algorithms to calculate the exact position of one's finger and to display the 3D objects must be reconsidered as well. Other than changing the sets of cameras the next task would be to use an actual 3D stereoscopic monitor in order to make 3D interaction more desirable and "tangible" to the users. While in our current prototype we have emulated 3D interactions through visualizing them through a 2D image, making this type of interaction available in 3D technology is the ultimate form we have in mind. Next up in line is the actual usability testing of our prototype, which we will conduct as soon as we deem the prototype ready to go. While currently the prototype is under construction further improvements should go a long way to solidify what we have envisioned in terms of dynamic depth and depth interaction.

7. Acknowledgements

This work was in part supported by the MKE (Ministry of Knowledge Economy) under the "IT Consilience Creative Program" support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2012-H0201-12-1001) and the Yonsei University Research Fund of 2012.

References

- [1] P. Surman, K. Hopf, I. Sexton, W. Lee, and R. Bates, "A roadmap for autostereoscopic multi-viewer domestic TV displays," *Multimedia and Expo, 2006 IEEE International Conference on*, pp. 1693–1696, 2006.
- [2] C. Hand, "A survey of 3D interaction techniques," *Computer graphics forum*, vol. 16, no. 5, pp. 269–281, 1997.
- [3] Wii - Official Website at Nintendo. [Online]. Available: <http://www.nintendo.com/wii>
- [4] Kinect - Xbox.com. [Online]. Available: <http://www.xbox.com/en-US/kinect>
- [5] D. Bowman, S. Coquillart, B. Froehlich, M. Hirose, Y. Kitamura, K. Kiyokawa, and W. Stuerzlinger, "3d user interfaces: New directions and perspectives," *Computer Graphics and Applications, IEEE*, vol. 28, no. 6, pp. 20–36, 2008.
- [6] L. Du, P. Qin, J. Song, W. Song, Y. Xu, and W. Zhou, "Immersive 3D user interface for 3D TV," in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON), 2011*, 2011, pp. 1–4.
- [7] M. Mine, F. Brooks Jr, and C. Sequin, "Moving objects in space: exploiting proprioception in virtual-environment interaction," *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 19–26, 1997.
- [8] D. Keefe, D. Feliz, T. Moscovich, D. Laidlaw, and J. LaViola Jr, "CavePainting: a fully immersive 3D artistic medium and interactive experience," *Proceedings of the 2001 symposium on Interactive 3D graphics*, pp. 85–93, 2001.
- [9] A. Banerjee, J. Burstyn, A. Girouard, and R. Vertegaal, "Pointable: an in-air pointing technique to manipulate out-of-reach targets on tabletops," *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pp. 11–20, 2011.
- [10] M. Katzourin, D. Ignatoff, L. Quirk, J. LaViola, and O. Jenkins, "Swordplay: Innovating game development through VR," *Computer Graphics and Applications, IEEE*, vol. 26, no. 6, pp. 15–19, 2006.
- [11] Y. Kwon, K. Jeon, J. Ki, Q. Shahab, S. Jo, and S. Kim, "3D gaze estimation and interaction to stereo display," *The International Journal of Virtual Reality*, vol. 5, no. 3, pp. 41–45, 2006.
- [12] R. Newman, Y. Matsumoto, S. Rougeaux, and A. Zelinsky, "Real-time stereo tracking for head pose and gaze estimation," *Automatic Face and Gesture Recognition, 2000. Proceedings. Fourth IEEE International Conference on*, pp. 122–128, 2000.
- [13] J. Chen and Q. Ji, "3D gaze estimation with a single camera without IR illumination," in *ICPR 2008. 19th International Conference on Pattern Recognition*, 2008, pp. 1–4.
- [14] Code Laboratories > About CL Eye Platform. [Online]. Available: <http://codelaboratories.com/products/eye/sdk/>

User Interface for Creating Traditional Chinese Architecture

Soon Tee Teoh

Department of Computer Science, San Jose State University, San Jose, California, USA

Abstract— We present an interface for users to conveniently and quickly create a settlement with various standard structures in traditional Chinese architecture. The main page of the interface is a top-down map view of the canvas which allows users to place different structures, such as pavilions, pagodas, halls, houses, gates, moon-gates and bridges onto the scene. Then, the user can custom-design each structure. Because traditional Chinese architecture has a standard form, we simplify the design of a building into a few parameters which users can edit, such as roof curvature and roof height. This allows the user to design the building according to their preference and still adhere to the traditional Chinese architecture form.

Keywords: computer graphics, procedural modeling, architecture, culture, user interface, 3D models

1. Introduction

3D models of buildings and other structures in traditional Chinese architecture are popular because of their beauty and because of the evocative environment they create in CGI movies and video games. In 2008, for example, DreamWorks Animation's feature film *Kung Fu Panda* won the Academy Award for Best Animated Feature, and was followed by a sequel in 2011.

In this paper, we describe our work in creating an interface for users to conveniently and quickly create 3D models of a variety of structures in traditional Chinese architecture. The structure of traditional Chinese architecture follows strict rules, therefore it is suitable for computer-aided design, however, some structures, such as the roof, are also very complex, and therefore require computer-aided design for users to create quickly.

Our program is able to create some main structures found in a traditional Chinese settlement, including pagodas, pavilions, halls, houses, bridges, platforms, fortified walls, garden walls, gates, moon-gates, and ceremonial arches. We provide a convenient interface for the user to place structures on the map, and therefore create a traditional Chinese settlement. We also provide the user with the interface to custom-design each structure created.

Because traditional Chinese architecture has a standard form, we simplify the design of a building into a few parameters which users can edit, such as roof curvature and roof height. This allows the user to design the building according to their preference and still adhere to the traditional Chinese architecture form.

2. Related Works

Our work uses procedural modeling to create and describe building structures. Birch et al. [1] introduced one of the earlier works on procedural modeling of architectural structures. They make use of a large library of prototypes to create 3D models of buildings. They have windowsplitting and curve-generation components, and their program can add the 3D models into a scene-graph.

Wonka et al. [9] described a method for modeling architecture using split grammars to generate non-trivial volumetric shapes. They created a large database of rules from which they modeled different designs. With a parameter-matching system, users can specify high-level design parameters to influence the output. They also introduce control grammars, which grow buildings according to architectural principles. Lipp et al. [5] later extended this work to create user-friendly interactive tools so that the user can interactively edit and control the output of the grammar-based procedural architecture system.

CGA Shape [7] is a shape grammar introduced by Muller et al. It has production rules that iteratively generates building models with increasing detail, such as window facades and roof surfaces. This powerful grammar can describe complex building models not restricted to axis-aligned shapes. They are able to create buildings and entire with buildings of varied designs and styles of significant complexity and realism, from medieval towns to futuristic skyscrapers. A user interface lets the user control parameters to determine the appearance of the automatically-generate buildings. CGA Shape is now available in a commercial package called CityEngine.

Several previous works specifically create buildings in traditional Chinese architecture. For example, Chan et al. [2] modeled the Temple of Heaven in Beijing, as part of the effort to preserve historical architectural knowledge, using photographs of perpendicular elevations of the buildings.

Liu et al. [6] built 3D models of vernacular houses in the Southeast China style. They model the vernacular house to significant detail, with elements like roofs, windows, and various types of walls.

Liu et al. [4] introduced a procedural modeling technique based on Chinese Ancient Architecture design theory. They formalized the design patterns into a set of Constructive Grammar rules.

Teoh [8] formulated a description for traditional East Asian architecture using procedural modeling. Noticing that

the structure of traditional Chinese, Korean and Japanese architecture all follow a similar design, but with different parameters, Teoh developed a procedural model for halls and pagoda towers, and by changing parameters such as the number of sides of a pagoda, or the curvature and height of the roof, the user can create structures of different styles.

Hou et al. [3] presented a modeling framework for building 3D models of traditional Chinese architecture from elevation drawings. They use automatic drawing recognition to extract production rules from elevation drawings. They cluster repetitive horizontal regions from the drawings to form architectural components, which serve as elements for 3D model generation. Then, these components are structured into a shape tree and organized hierarchically using Markov Random Fields. Finally, shape grammar rules are derived to construct the 3D semantic model and variations.

3. Interface for creating Chinese architecture

There are two steps in the creation of a traditional Chinese settlement. On the first page of the interface, we allow the user to drag and drop various types of traditional Chinese structures, such as halls and pagodas. The second step is to allow the user to click on one structure to edit. The interface will then focus on the selected structure, providing a user-friendly interface for the user to adjust various parameters associated with the structure type, to custom-design the structure.

3.1 Creating a Settlement

The main page contains a simple interface for the user to create a settlement. On the left is a canvas for the user to drag and drop various structures. On the right is a control panel for the user to choose which structure to create, by clicking on the representative icon. The structures available are: Hall (or House), Pagoda, Pavilion, Platform, Staircase, Corridor, Gate (Ceremonial Arch or Moon-Gate), Wall, Bridge (Wood or Stone Arch) and Tree. The user simply chooses the desired structure, then clicks on empty space on the canvas, and a new structure of the selected type will be created and placed on the canvas. An example is shown in Figure 1.

The control panel also contains icons for manipulating the structures, such as rotate, translate and scale. The user clicks on the desired action, then clicks on the structure to move, and holds down the mouse button while moving the mouse, and the selected structure will transform accordingly.

Structure such as platforms, halls and pagodas are simply placed on the map with a single mouse click. However, other structures such as Bridge and Gate are placed with two mouse clicks, one on each end-point of the bridge or gate. After placing the bridge or gate on the map, the user can move it by right-clicking on one endpoint and dragging it.

To the best of our knowledge, this is the first program that enables users to drag and drop structures to create a settlement in traditional Chinese architecture. This allows artists to conveniently design a settlement. The top-down map view gives the user an overview of the settlement.

3.2 Designing a building

The user can rotate, translate and scale a structure from the main canvas. However, to custom-design a building, the user has to click on the building, and select the "Edit building" mode. The program will then switch to the Edit Building interface. An example of the interface is shown in Figure 2, which shows how a user can edit a Hall structure.

The Figure shows the parameters that can be controlled by the user. For example, the user can click on the control point to adjust the Roof Height, or the Ceiling Height. The user can also adjust how far the pillars and the walls recede from the ceiling edge. The user can also adjust the curvature of the eaves and roof by moving the control points. Other parameters such as wall texture and number of storeys are entered in the adjacent control panel.

In the control panel, the user can also toggle between Hall type and House type. The difference is that the Hall type has eaves on all four sides, while the House type has solid wall on the sides all the way to the top. Examples of houses are shown in Figure 3. The tip of the roof can also be toggled between Rounded style and Pointed style. This Figure shows the difference between the two types of roofs.

The most elaborate part of traditional Chinese architecture is the roof. Roofs are found in different structures: Houses, Halls, Pavilions, Pagodas and Corridors. To structure the roof, our program separates a roof into three parts: (1) the Main Roof, (2) the upper eave, which is above and inside the ceiling line, and (3) the lower eave, which is below and outside the ceiling line.

For a Hall, which has eaves on all four sides, the Main Roof is a prism shape with triangles on the sides, and a single plane on the front and back of the Hall. The user-controlled parameters are: Width, Height and Curvature of the Main Roof. Upper and lower eaves exist for every storey of a House, Hall and Pagoda. The user-controlled parameters are: Width, Height and Curvature of the eaves. By controlling these parameters, the user is able to create structures with different styles. For example, the Tang style tends to have straighter roofs, while the more recent styles have curved roofs for Halls. Houses also tend to have straight roofs, and some also have rounded instead of pointed roof-tops.

The bracket (dougong) system used in the construction of traditional Chinese buildings is also very complex. They are difficult to model for the average user, but they are a distinctive part of traditional Chinese architecture, so they are essential. Our system automatically generates these bracket systems. The bracket structure holds beams which hold up the roof, which can be very heavy. In our program,

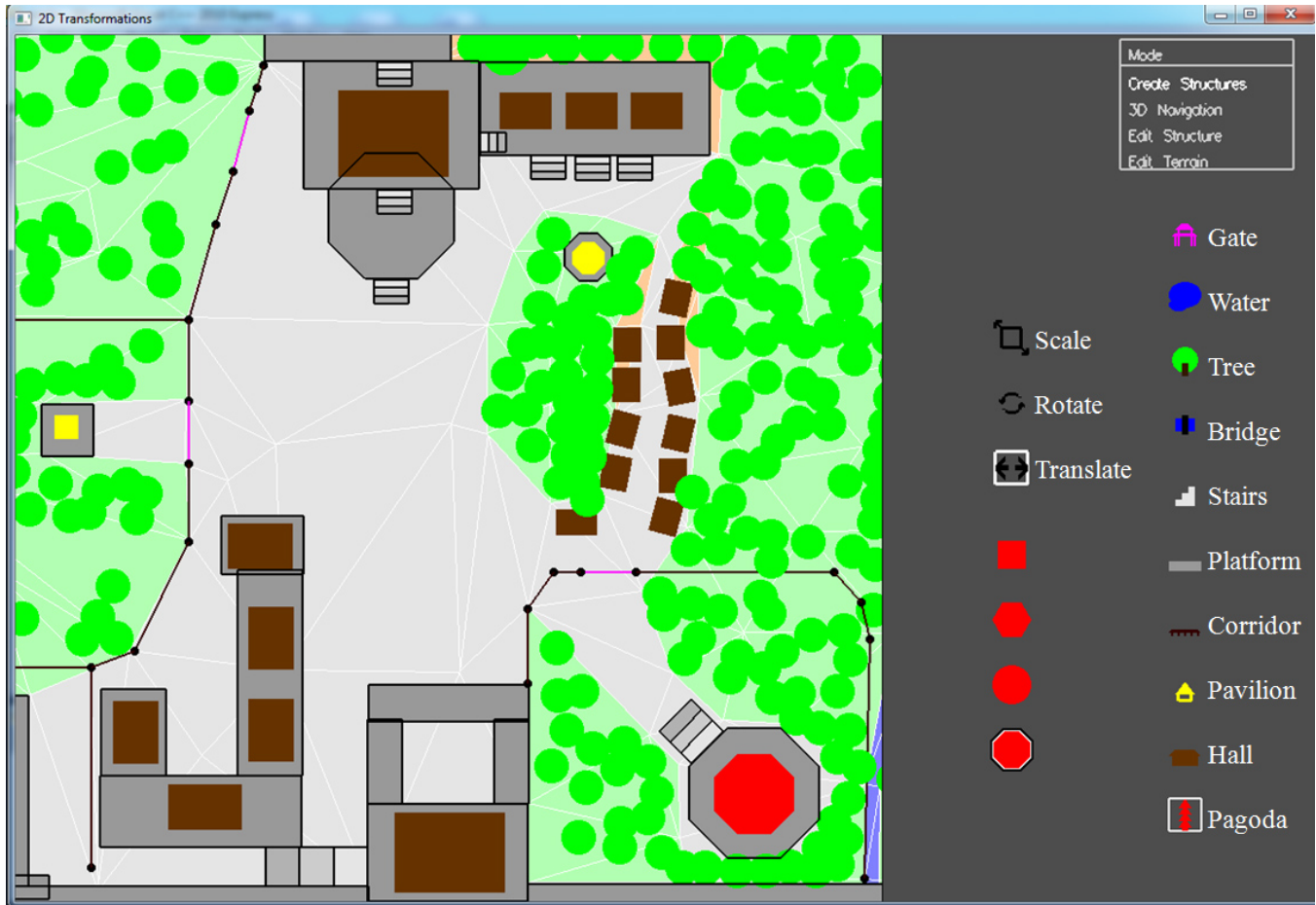


Fig. 1: Interface for creation of structures. Right: Control panel to select structure and interaction. Left: Canvas map to place the structures.

we allow the user to specify the number of pillars on each side of the building. Then, for each pillar, the program automatically creates brackets at the top of the pillar to support the roof. Taking into consideration outer-eave height and width, and pillar offset from the ceiling line, the program calculates the dimensions of the brackets and beams and automatically creates the brackets. An example is shown in Figure 4.

Another structure we allow the user to create is the Gate. There are two types of gates supported: (1) the Moon Gate, and (2) the Ornamental Gate ("Pailou" or "Paifang"). Both are characteristic of traditional Chinese architecture. An example of the Moon Gate is shown in Figure 5 and an example of an Ornamental Gate is shown in Figure 6.

The user creates a gate on the map by clicking on two endpoints of the gate on the map. When a gate is selected, we allow the user to toggle between Moon Gate and Ornamental Gate types. The Moon Gate is an essential part of a Chinese garden, and is connected to a garden wall, as shown in the

example. The Moon Gate contains a circular opening. Like the garden wall, the Moon Gate also has an ornamental tiled roof.

The Ornamental Gate ("Pailou" or "Paifang") is typically a free standing structure. There are two main styles, both of which are shown in Figure 6. The user is allowed to toggle a selected Ornamental Gate between the two different styles. The gate in the foreground has three sections, with smaller boards between each section. The gate in the background has three tiers with no boards between the tiers. The Ornamental Gate uses the same roof structure and bracket structure found in a Hall. The roofs and brackets are complicated structures and are automatically created.

The Bridge is another common characteristic feature of a traditional Chinese settlement. There are two types of bridges, which the user can choose. First is the Arch Bridge, which is typically made of stone. The Arch Bridge has arches below the bridge. The user-controlled parameters for an Arch Bridge are: width, height, height offset of arches, number of

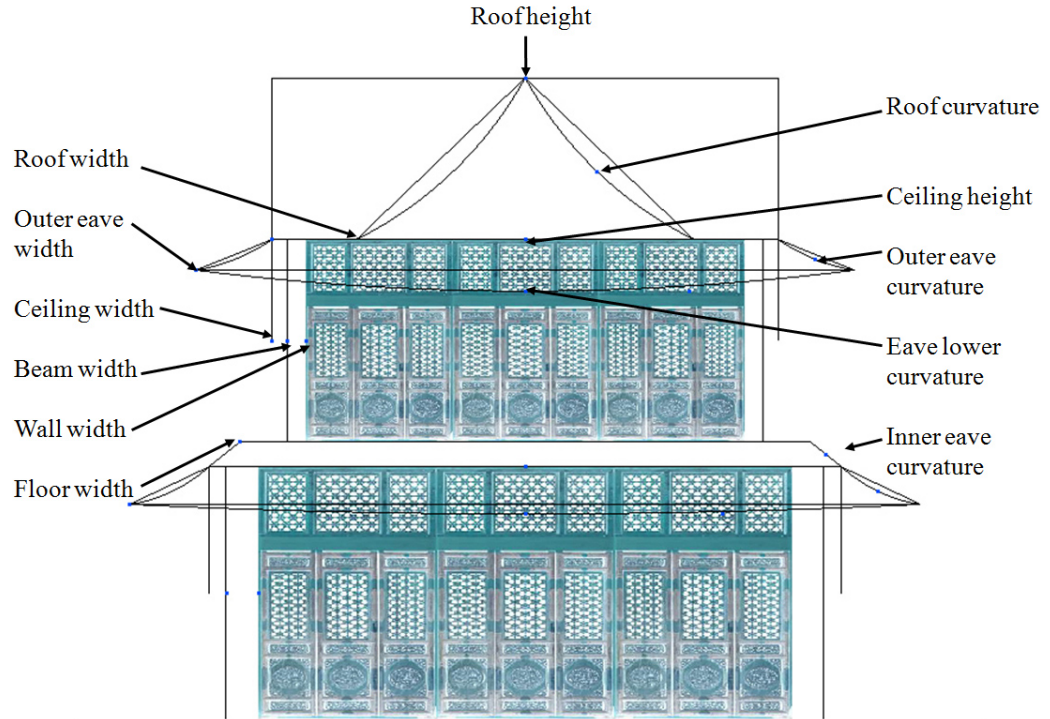


Fig. 2: Interface for the user to edit a Hall. The parameters controlled by the user are labelled. The user controls the parameters by clicking on the appropriate control points and moving the mouse. Other parameters such as wall texture and number of storeys are entered in the adjacent control panel.

arches, width of each arch, height of each arch, and curvature of the top of the bridge. Except for the number of arches, all the other parameters are controlled by the mouse in the edit mode for the Arch Bridge. The number of arches is controlled by the control panel. An example is shown in Figure 7.

The other bridge type is the Wood Bridge. This is a simple arc, supported by stilts. An example is shown in Figure 8. Wood bridges are typically small and found in gardens. By contrast, the Arch Bridge are much more massive and are found in palaces and are meant for transport of cavalry.

Figure 7 also shows an example of Platforms. Platforms are blocks of stone. They can be used to make fortified walls by making the shape thin. Any The Figure shows a Hall on top of a Platform. When the user places a building on top of a Platform, the program automatically sets the base height of the building to the height of the Platform. The user can put multiple layers of platforms on top of one another. In addition, via a check-box in the control panel, the user can also create an arched entrance at the bottom of a platform block. This creates a city gate, an example of which is shown in Figure 7.

Figure 8 also shows an example of a Pagoda, which is one of the largest, tallest and most visible structures in traditional Chinese architecture. The user can set the height of the

pagoda, and also the width of each storey, and the curvatures of the roof and the eaves at each storey, as well as the height of the roof.

Figure 8 also shows an example of the garden wall. A garden wall is constructed just like a Corridor, which is shown in Figure 9. The user selects the GardenWall/Corridor icon and clicks on the map canvas. The user can then toggle between these two types. Note that the user can connect multiple corridors to a single control point. In this way, branches can be made to create a corridor network. The user can click on a button to synchronize the height of all corridors connected by the network. The program automatically creates a corridor that is joined properly at the intersections.

4. Conclusions

We have described our work in creating an interface for users to conveniently and quickly create 3D models of a variety of structures in traditional Chinese architecture. Our program is able to create some main structures found in a traditional Chinese settlement, including pagodas, pavilions, halls, houses, bridges, platforms, fortified walls, garden walls, gates, moon-gates, and ceremonial arches. We provide a convenient interface for the user to place structures on the map, and therefore create a traditional Chinese settlement.

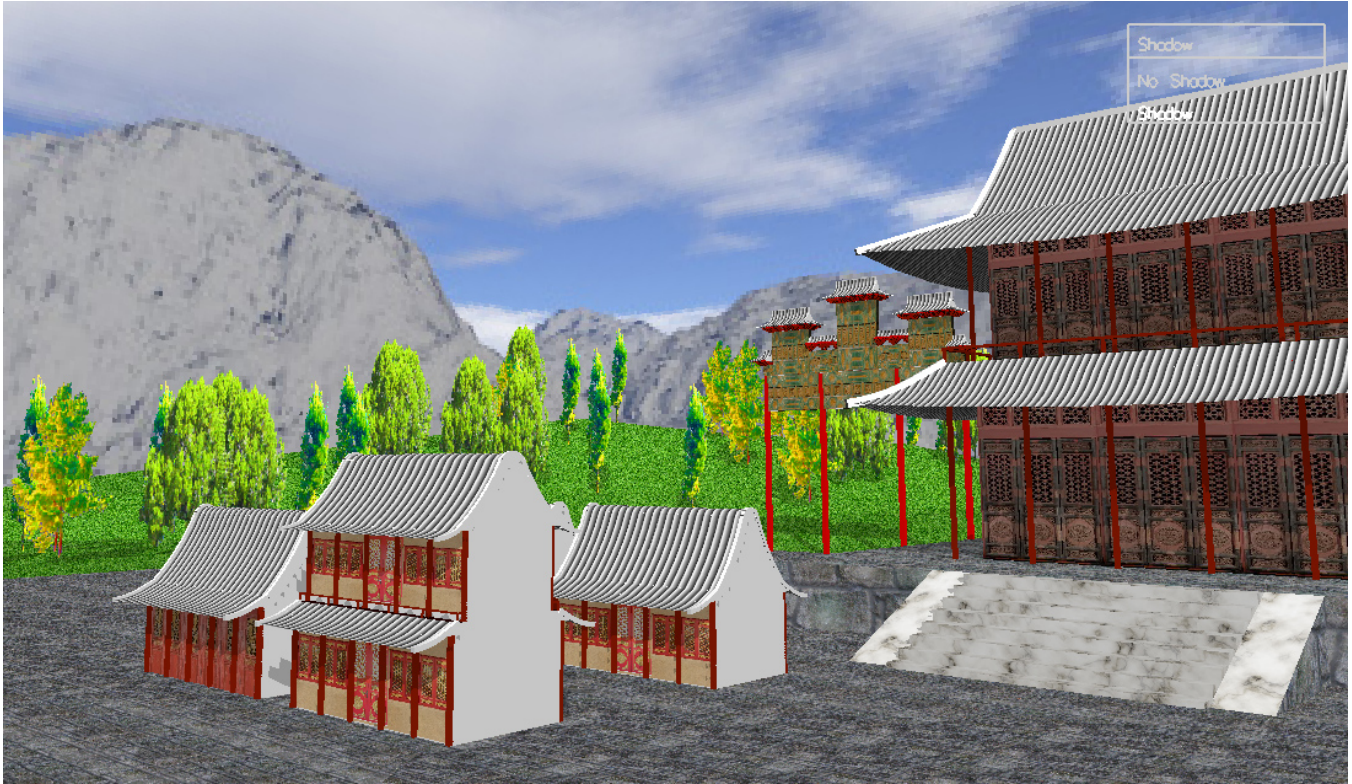


Fig. 3: Example with houses with different number of storeys and with pointed and rounded roof types. Also in the scene is a hall, rock platform and marble stairs.

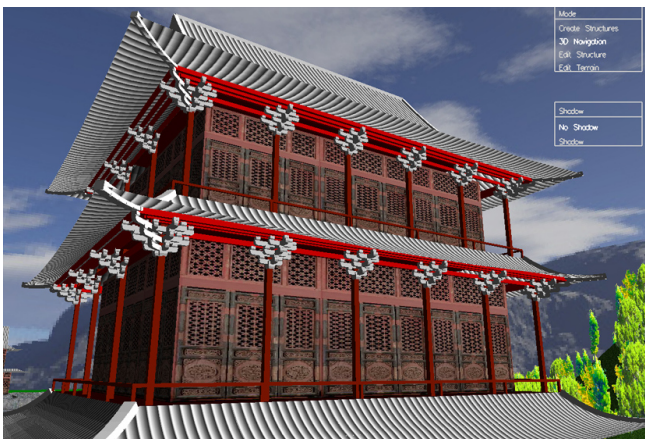


Fig. 4: View of a Hall showing automatically created brackets at the top of every pillar. The bracket structure holds beams which hold up the roof.

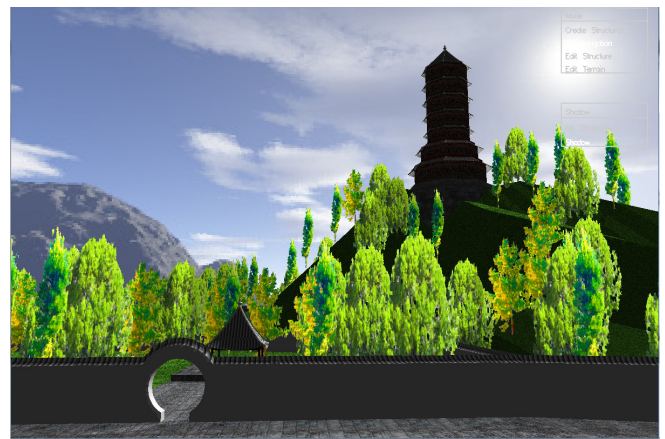


Fig. 5: Scene with Garden Wall, Moon Gate, Pagoda and Pavilion

We also provide the user with the interface to custom-design each structure created.

Because traditional Chinese architecture has a standard form, we simplify the design of a building into a few parameters which users can edit, such as roof curvature and roof height. This allows the user to design the building

according to their preference and still adhere to the traditional Chinese architecture form. Our program automatically creates complex structure such as the roof and the bracket and beam system.

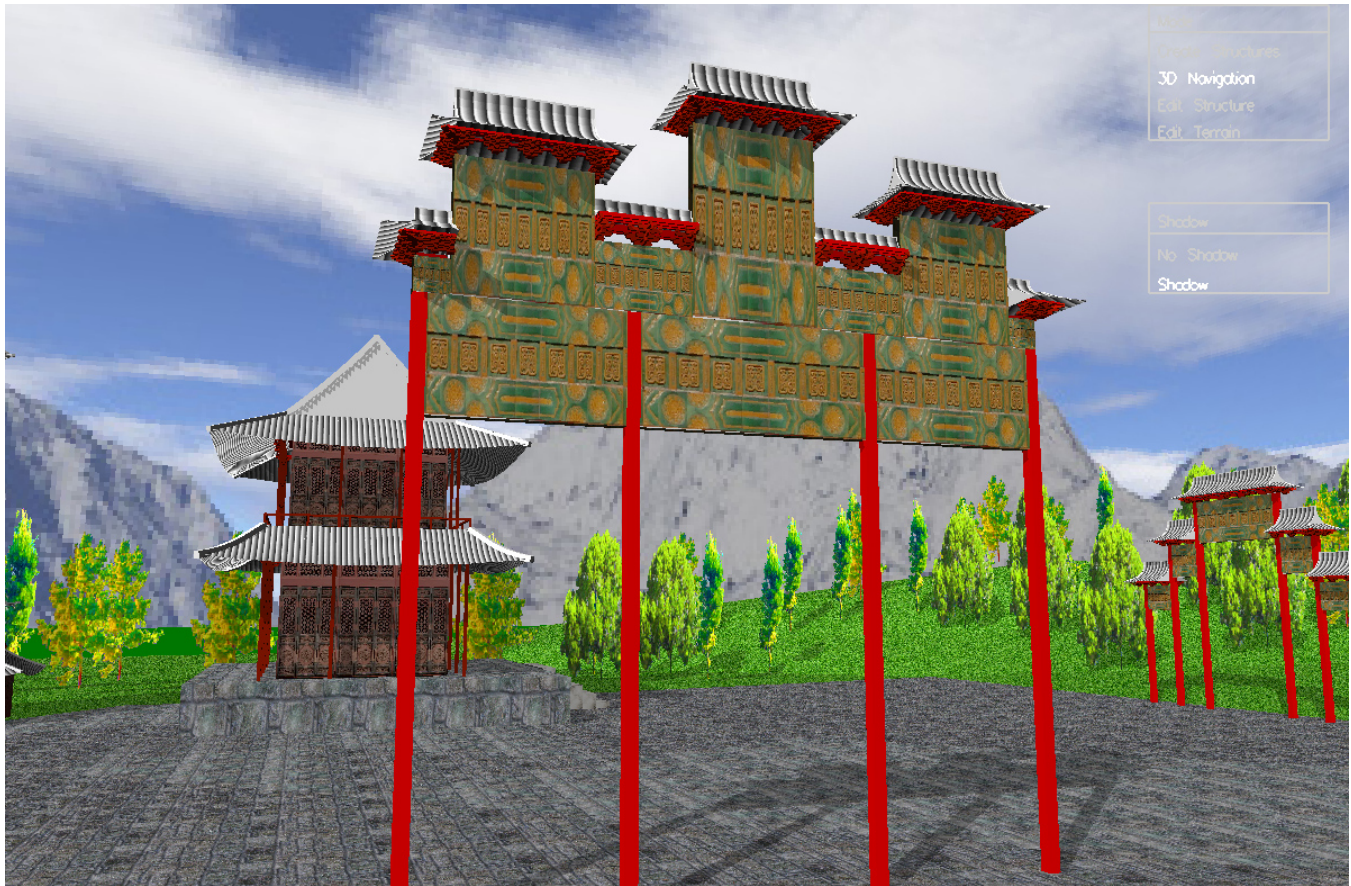


Fig. 6: Scene with two Ornamental Gates ("Paifang" or "Paifang")

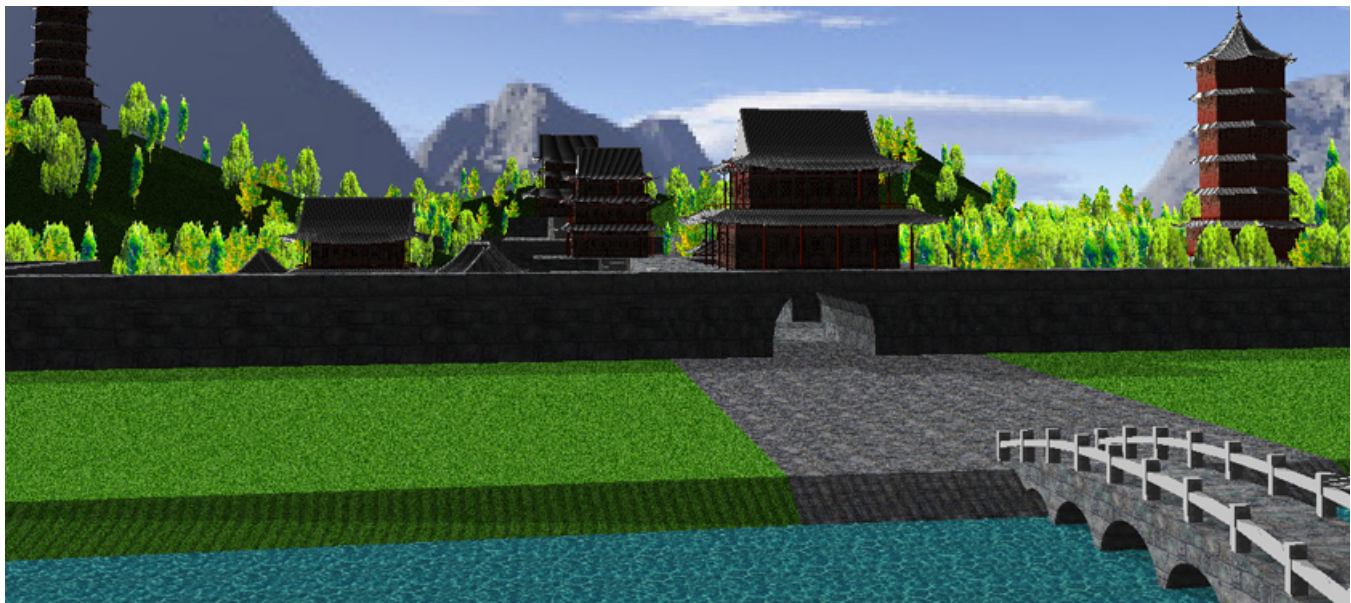


Fig. 7: Scene with Marble Arch Bridge, Stone Fortified Gate, and Stone Fortified Wall

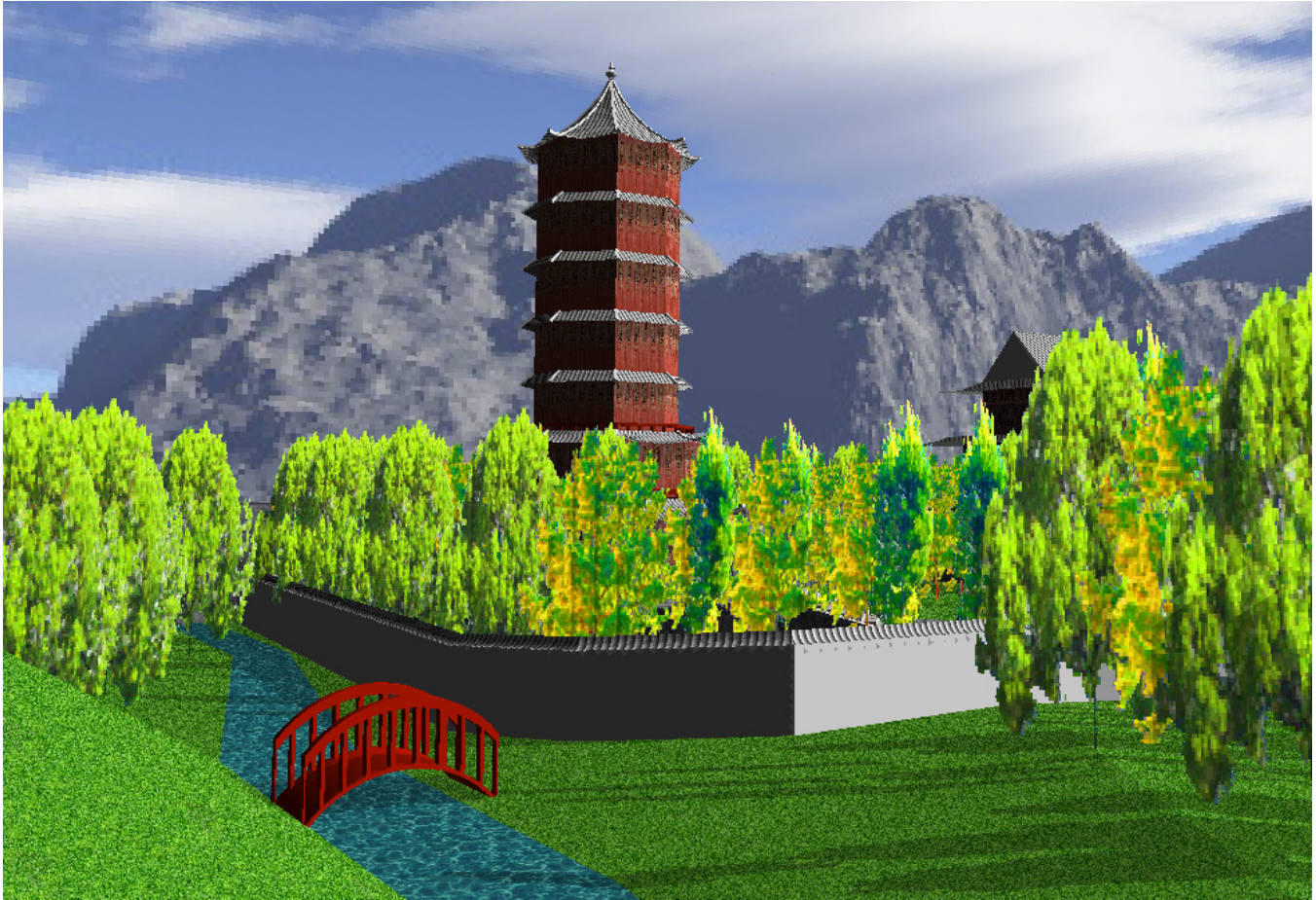


Fig. 8: Scene with Wood Bridge and Pagoda

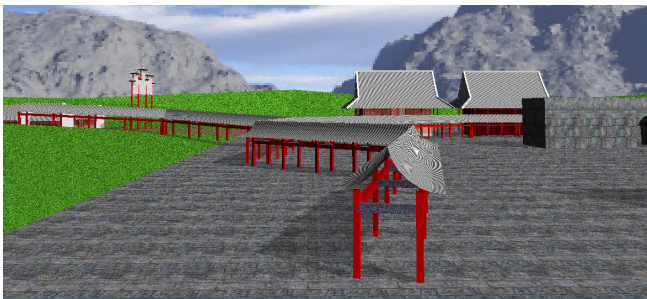


Fig. 9: Scene with a long Corridor

References

- [1] P. Birch, S. Browne, V. Jennings, A. Day, and D. Arnold, "Rapid procedural-modelling of architectural structures" *Proceedings of the 2001 Conference on Virtual Reality, Archeology, and Cultural Heritage*, pp. 187-196, 2001.
- [2] C.-S. Chan, Z. Tong, A. Dang, and J. Qian, "Virtual reality modeling of traditional Chinese architecture" *Proceedings of the 9th International Conference on Virtual Systems and Multimedia*, pp. 13-22, 2003.
- [3] F. Hou, Y. Qi, and H. Qin, "Drawing-based procedural modeling of Chinese architectures" *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 18, no. 1, pp. 30-42, 2011.
- [4] H. Liu, Q. Wang, W. Hua, D. Zhou, and H. Bao, "Building Chinese ancient architectures in seconds" *Proceedings of the Workshop on Computer Graphics and Geometric Modeling (CGGM 2005)*, pp. 248-255, 2005.
- [5] M. Lipp, P. Wonka, and M. Wimmer, "Interactive visual editing of grammars for procedural architecture" *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 102:1-10, 2008.
- [6] Y. Liu, C. Xu, Z. Pan, and Y. Pan, "Semantic modeling project: Building vernacular house of Southeast China" *Proceedings of the 2004 ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Applications in Industry (VRCAI '04)*, pp. 412-418, 2004.
- [7] P. Muller, P. Wonka, S. Haegler, A. Ulmer, and L. Goog, "Procedural modeling of buildings" *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 614-623, 2006.
- [8] S. Teoh, "Generalized descriptions for the procedural modeling of ancient East Asian buildings" *Proc. of the International Symposium on Computational Aesthetics in Graphics, Visualization, and Imaging (CAE'09)*, pp. 17-24, 2009.
- [9] P. Wonka, M. Wimmer, F. Sillion, and W. Ribarsky, "Instant architecture" *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 669-677, 2003.

SESSION

NOVEL ALGORITHMS AND APPLICATIONS

Chair(s)

TBA

Initial Design of a Multimodal Collaborative Mobile Application for Real Time Decision Making

Gregory Burnett¹, Thomas Wischgoll², Victor Finomore¹, and Candace Washington¹

1. US Air Force Research Laboratory, 711th Human Performance Wing, Wright-Patterson AFB

2. Department of Computer Engineering and Computer Science, Wright State University

Abstract – *Mobile devices, smartphones and tablets, are continually expanding their computational performance capabilities through improved processing, interconnectivity network abilities, resource management and ease of use user interfaces. As such they are gaining interest as a means to support on-the-move remote collaboration for military personnel executing real time decision making tasks. This paper focuses on a software-based implementation of a prototype multimodal Android application that was designed to capture and disseminate real time battlefield perspectives to distributed entities. Moreover the mobile application enables remote experts to interactively collaborate through multimodal functionality to provide directives to the mobile user that should be applied to the local scene. The design of the mobile application interaction is discussed as well as the results from an initial demonstration where remote guidance was present to a mobile user attempting to defuse an improvised explosive device. Additionally, we report future implementation capabilities and projected military usage.*

Keywords – Multimodal, Mobile Computing, Remote Collaboration

1. Introduction

With changing environments and emerging unrecognizable threats, dismounted Battlefield Airmen (BA) need to carry advanced technologies to survive and effectively prosecute their missions. Requiring a high degree of situation awareness and the ability to multitask between various mission essential responsibilities, BA rely on mobile computing devices to stay informed of battlefield conditions. Within recent years, ground military research and development efforts have shifted from funding the minimization of rugged laptops and ultra mobile personal computers to leveraging mobile devices such as smartphones and tablets [1]. These newer mobile devices continue to evolve at an unprecedented rate offering vast and ubiquitous capabilities. They allow users to perform tasks while on-the-move; as well as to be interoperable with heterogeneous

distributed systems through a variety of communication channels. Mobile devices are increasingly becoming more user friendly, offering intuitive user interface controls and advanced features. Mobile device operating systems, such as Android, are improving power management and consumption of frequently used embedded features. Accordingly, mobile devices and their capabilities are being examined as potential decision making tools for military personnel.

A desired capability that dismounted warfighters seek to have is the ability to collaborate with non-located individuals on mission objectives and tasks. Through the use of mobile devices and their intrinsic communication and interface capabilities, real-time, on-the-move remote collaboration can be performed in dynamic battlespaces.

Remote collaboration on physical tasks is defined by Kraut et al. [4] to be: “A general class of ‘mentoring’ collaborative physical tasks, in which one person directly manipulates objects with the guidance of one or more other people, who frequently have greater expertise about the task.” (p. 16) Task knowledge is important to enable remote collaboration; however to effectively collaborate, Clark et al [6] report there needs to be a mutual understanding between helper and worker to ensure *common ground*. This common ground can be achieved through various modalities. For example, when sharing a video feed of the active workspace, visual modal understanding can be achieved as the helper monitors the worker actions engaging in the task following an instruction. Alternatively, evidence of non-grounding could be observed if the worker hesitates to act out the remote instruction showing signs of confusion. Auditory, grounding can occur with verbal acknowledgments such as “okay”, “got it”, “un-huh”. Conversely, audio modal non-ground can be heard through utterance such as “what”, “huh”, “I don’t understand”.

In this paper, we report the design and implementation of an Android mobile application that supports multimodal remote collaboration. We first highlight related work in

which we draw features that will be incorporated in our application. These capabilities are implemented to achieve a higher common ground between non-located parties cooperatively working together on a task. Next, we discuss the methods and result from an initial demonstration of the mobile application, where participants worked jointly to defuse a simulated improvised explosive device (IED). Finally, future work to improve the prototype for military operations and potential military career fields that could benefit from real-time mobile collaboration in support of decision making is provided.

2. Related Work

Cooperative interactive systems between distributed parties working together to complete a physical task have been researched and implemented using several approaches and apparatuses.

Kraut et al. [4] developed a wearable system consisting of head borne CCD camera, a VGA (640x480) display and microphone headset. The configuration had each worker don the system which shared video and duplex audio between a helper and worker. The helper was able to monitor the video perspective of the worker's active workspace and use verbal instructions to guide maintenance procedure toward the task of repairing a bicycle. The authors' investigation concluded that field workers completed the task "more quickly and accurately when they have a remote expert helping them". Moreover, having a shared perspective positively influenced the verbal directives and feedback given between worker and helper.

Kirk, Fraser, & Rodden [3] designed a collaborative video/audio environment that sought out to address mixed reality ecology by conjoining two separate but similar workspaces into one hybrid workspace. The interactive system overlaid video captured gestures and workspace elements of the helper onto the active workspace of the worker through the use of projectors. Creating a linked collaborative workspace, the helper could direct the workers actions through the use of simple hand gestures, audio commands, and/or both. Their results showed that task completion time and accuracy mistakes were less than voice alone when the helper used the combination of voice plus gestures.

Authors Ou, Fussell, Chen, Setlock and Yang's [2] DOVE (Drawing Over Video Environment) remote collaborative system facilitated a "remote helper to draw on a video feed of a workspace as he/she provides task instructions". The DOVE system supported both freeform annotation as well as gesture fitting recognition to generate a markup pers-

pective shared to the worker. Results from their research suggest that markup capability "significantly reduces performance time compared to camera alone".

Interactive collaborative systems designed to support remote guidance evaluate their performance through various metrics. However, common metrics are task completion time, accuracy of task, response time, and mistakes made. Our paper focuses on these factors-- assessing the usefulness of our prototype Android application as a tool to support real-time decision making. Moreover, cooperative systems utilize several modalities to achieve communication grounding between helper and worker.

Wickens and McClarley's [5] report that systems and interfaces utilizing multiple modalities are more advantageous to the user than those that do not. Their paper suggests that multimodal interfaces allow their users to process different modality information concurrently for better cognitive understanding of the task at hand. For military use cases, cooperative interfaces leveraging various perception channels to communicate orders, instructions, battlefield information, etc. are essential. Warfighters often conduct operations in cognitively demanding environments that often require them to share their cognitive focus and attention across several events and stimulus happening concurrently. Additionally, auditory and visual distracters and/or masking are a battlefield constant. Special attention to designing independent or redundant multimodal capabilities is needed to provide information management to the warfighters.

3. Collaborative Prototype Design

Advancing user-centric cognitive interfaces for Battlefield Airmen (BA), researchers working in the US Air Force Research Laboratory, Human Effectiveness Directorate, Warfighter Interface Division began designing and implementing an Android mobile application for remote collaboration. Leveraging multimodal perception functionality, the mobile application sought to improve the warfighters capability by providing context rich information while supporting interactive collaboration of non-located parties. As depicted below, a cooperative application, running on a mobile device, could be utilized in the field to capture, disseminate, and interact with remote experts. These interactions can be support preplanned, dynamic, or time sensitive operations.



Figure 1: Mobile Device used to capture and share in field deployed perspective with remote experts

Drawing collaborative features from the highlighted related research approaches and other mobile application resources, our collaborative prototype identified the following set of capabilities for inclusion that military operators could benefit from:

- 1) Sharing live video of active worker's workspace
- 2) Sharing full duplex audio between linked users
- 3) Support free form and predefined markup annotation on captured still from live perspective.
- 4) Adjustable transparency of overlaid markup image on-top of live workspace.
- 5) User configurable preview/live adjacent windows or merged preview and live perspective

3.1 Live Video Sharing

Streaming video of the active workspace has been shown to improve communication grounding and thus warranted its integration into our prototype application. Our software design utilizes the integrated camera of the mobile device platform or can connect to an off board camera through 802.11, Bluetooth, or USB. Once connection is made to the video capture device, an image buffer is used to store the camera's acquisition at an upper limit of 30 frames per second and adjustable down to a lower limit of 5 frames per second. The configurable sampling and pending transmission of the image is scalable to conserve power consumption. Prior to network transmission, the image buffer is compressed to an 800x600 jpeg to improve network utilization as well as maximize the receiving military parties' ability to process the image natively without preprocessing the data received.

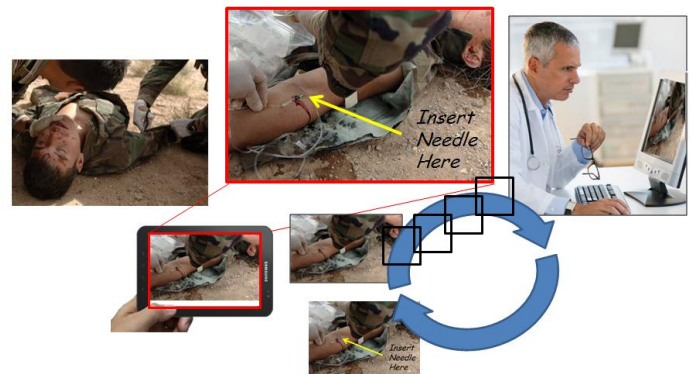


Figure 2: View Sharing between Helper and Worker

3.2 Audio

Audio communication was implemented using Voice over Internet Protocol (VoIP) functionality. The software supports live "hot" microphone as well as push to talk execution. Both audio input options were implemented to address constraints of military use cases that could limit operators' available hands-on with the mobile device and for battery consumption considerations. The application supports stereo or mono input and can be configured to use 8 or 16 bits per second at 11KHz, 22KHz, 44KHz, etc samples per second. The transmission and reception of network audio information are threaded to perform concurrently without delays.

3.3 Markup Stills

Extending the visual modality collaboration capability, annotation of still images was implemented into our application. As Ou et al showed, cognitive performance and understanding improved through the use of markups. We designed functionality that allowed free-form and predefined elements to be captured and transmitted between helper and worker. The graphical inputs were collected

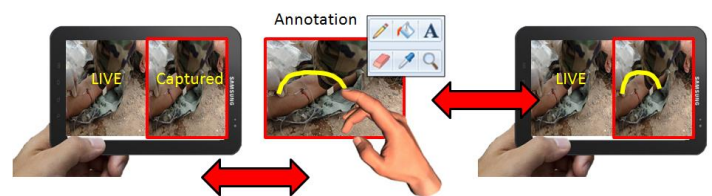


Figure 3: Markup Capability

through touch screen inputs on the mobile device display. When edits are completed the markup are merged with the still capture image for transmission. Image compression is applied and the resultant image is transmitted across the network.

3.4 Configurable Interface

Our mobile application supports two display configurations. 1) Full screen mode that renders the live perspective merged with a markup image on the same preview surface and 2) Adjacent mode that displays the live perspective and a markup image in separate preview surfaces.



Figure 4: Full Screen Mode and Adjacent Mode

The rationale behind having two display configurations can be drawn from the dependence on the cooperative markup received for a collaborative task. In the full screen mode, the markup can act as an exact guide or placement for the worker to perform task objectives. For example, the markup could identify a vein in a soldier's arm where an IV needle should be inserted. In the adjacent mode, the markup can serve more as a reference than a precise guide for a task. For example, place the tool on this shelf.

3.5 Transparency Adjustment to Markup

Markup images received from remote helpers will be displayed to the user and fused with the live perspective in the full screen mode. When attempting to perform a task, illustrated through markup context, the mobile user may choose to adjust the degree of transparency of the markup to better observe the active workspace. This capability was implemented in our application through intuitive touch screen actions. As depicted in figure 5, the user can adjust the transparency level applied to the markup simply by running a figure up or down the vertical dimension of the mobile device's screen. This dynamic ability to make the markup fade in and out of the live perspective serves to enable the user with an on demand overlay assisting collaboration.



Figure 5: Markup Transparency

4. Demonstration Experiment

To assess if cooperative interaction between distributed users can affectively be achieved on mobile devices, a relevant military task was evaluated. Improvised explosive devices plague military operations worldwide. With no constant design, IEDs have numerous wire configurations and trigger features. Defusing IEDs involve systemic sequential wire identification and disarming (cutting or re-routing wires) making the IED inert.

4.1 Participants

Twelve participants volunteered for this study, 8 men and 4 women, ranging in age from 23-30 ($M = 25$) years. All participants had normal hearing and normal, or corrected-to-normal vision.

4.2 Design

A within-subject design was employed with four levels of Modality Interface (Audio, Video with Markup, Video with Audio & Video with Markup and Audio). The order for which each participant utilized a modality was controlled by counterbalancing the order so not to bias the experimental conditions. All participants took part in a training session to familiarize themselves with the task and devices. The participants trained defusing four IEDs per experimental condition. Participants were given the option for more practice trials; however, none of them felt the need for more. The four experimental conditions and IED configuration were randomized per participant.

4.3 Apparatus

Four simulated IEDs were used in the experiment. Each IED consisted of a clock, power source, control chip, and explosive charge containers as seen in Figure 6. There were nine wires on each IED, seven were active and two were distracters. The participants worked cooperatively with a remote confederate, who had detailed instructions for disarming each IED and experience communicating through the various multimodal communication capabilities. Participants used a Samsung Galaxy Tablet running our developmental Android application to interact with the remote confederate through a Wi-Fi connection. The Galaxy Tablet was mounted on a stand to allow the participant to freely use their hands, as seen in Figure 7. The remote confederate was situated in front of a workstation, which was isolated from the experimental area. The confederate's workstation allowed them to communicate via VoIP, capture, and annotate images from the participant's tablet to assist them in their task.

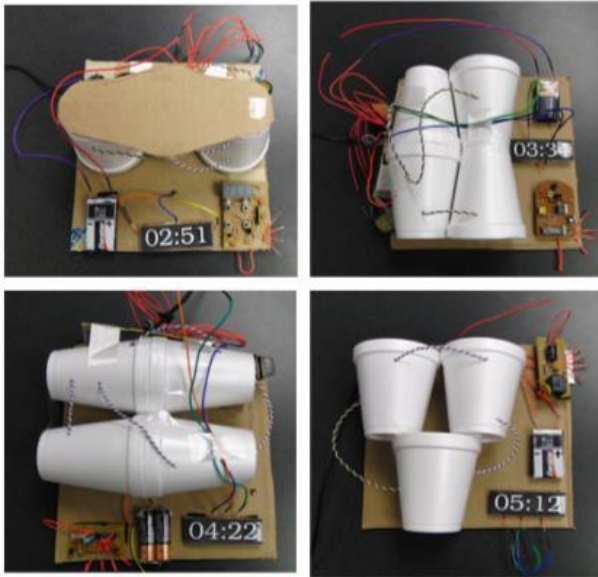


Figure 6: Simulated Improvised Explosive Devices

4.4 Procedure

Four conditions were evaluated. 1) Audio only where the confederate could not see the worker's workspace. 2) Video with Markup where the confederate monitored the workers workspace and provided markup directives. 3) Video with Audio where the confederate monitored the workers workspace and provided verbal directives. 4) Video with Markup and Audio where the confederate monitored the workers workspace and could provide directives through both markup and verbal interactions.

In the Audio condition, participants spoke to the confederate via VoIP where they had to describe the IED in order for the confederate to relay the sequence for which to disconnect active wires. The Video with Markup condition consisted of the confederate capturing a picture of the IED from the tablet's perspective then annotating the picture in real-time on their workstation. The annotated image, which showed the order of wires to disconnect, was sent to the participant to defuse the IED. The Video with Audio condition consisted of the confederate monitoring the participant's perspective while supplying verbal instructions to defuse the IED. The Video with Markup and Audio condition combined the Audio and Video conditions so that the confederate and participant were able to talk to each other as well as send annotated images.

For each condition subjects defused a unique IED. They were asked to complete the task as fast as possible without making any errors. A countdown clock was used to impose time pressure initially starting at one minute and decrementing each second.



Figure 7: Participant diffusing IED with Tablet

5. Results

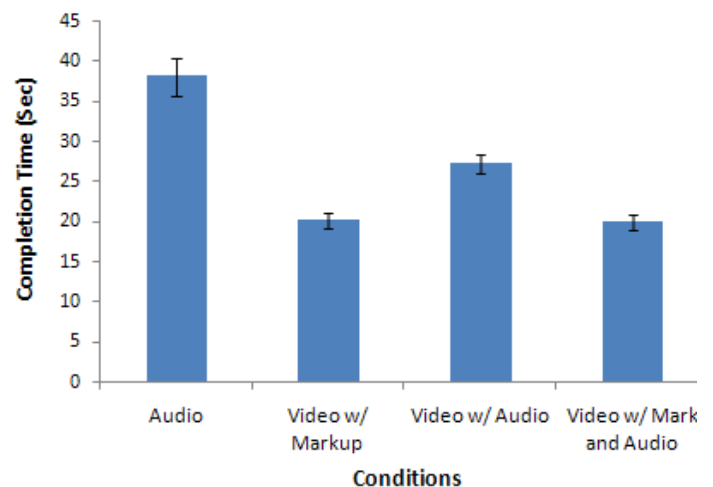
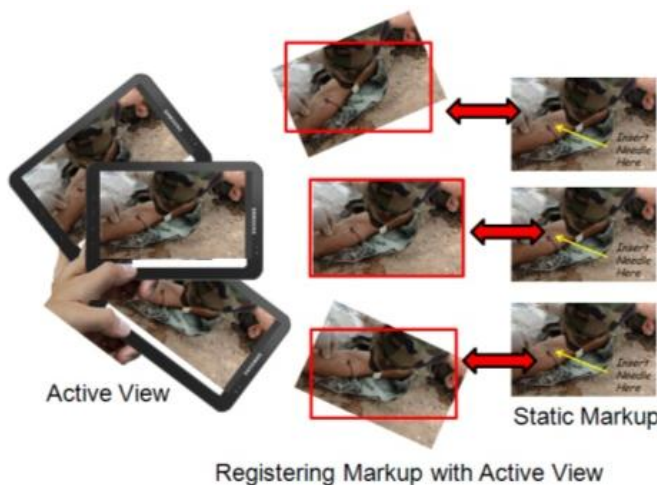


Table 8. Mean completion times for each of the four experimental conditions. Error bars are standard errors.

Mean task completion time and their respective standard errors for the four experimental conditions are displayed in Table 8. A 4 condition repeated measures Analysis of Variance of this data revealed a statistically significant main effect for conditions, $F(3,33) = 70.88, p < .05$. Subsequent post hoc Tukey-tests with alpha set at .05 revealed that participants using Video with Markup and Video with Markup and Audio completed the task statistically faster than the other two conditions but were not different from each other. The Tukey-test also found that participants using Video with Audio were faster than Audio alone.

6. Future Work

Mobile devices' form factor make them convenient for portability and on-the-move processing; however their compact size lends them to not remain static as users manipulate and interact with the device's context. When collaboration is performed through markup images the offset between the current live perspective and the captured markup perspective may slightly differ. In our current design when merging the live and markup in the full screen mode a ghost effect could be rendered if the two orientations do not align. To prevent this visually distracting effect, the ability to auto register the still markups with the active live perspective is desirable. Computer vision techniques can be incorporated into the cooperative Android application that performs feature extraction and image transformation on the annotated markups that orients them to the dynamic live image captured on the mobile device, as seen in Figure 9.



Registering Markup with Active View
Figure 9: Image registration concept

Battery capacity is a major limitation of mobile devices. With regards to remote collaboration, power consumption and battery depletion prior to the completion of a cooperative task could be detrimental to the success of the task. Currently, it is left up to the mobile user to monitor their device's state of charge and adjust the mobile device's properties to prolong the runtime (e.g. reduce the screen brightness, disable updates, etc.). An additional improvement that we plan on incorporating into our Android application is a battery power status message. The message will transmit as additional information used in the collaboration between the helper and worker. With the mutual understanding of the current battery state of charge on the mobile device, both helper and work can monitor and negotiate features that could be reduced (frequency of active workspace transmission, sample rate of audio, on-board vs. off-board image registration, etc.) extending the runtime of the mobile device.

7. Conclusion

Previous work in computer supported cooperative work has shown that sharing multimodal information (video and audio) improves joint task completion. This paper reported the implementation of an Android cooperative collaboration application and its initial evaluation and demonstration use for defusing IEDs. Results from the experimentation showed that mobile device can support the communication capability to successfully complete a task jointly executed by a remote helper. The ability to share and annotate images was found to be the effective means to communicate directives. This initial investigation provided justification for further development of a mobile, multi-modal collaborative application for distributed operators. This application seeks to equip BA with a direct on demand link to SME to maximize mission effectiveness.

8. References

- [1] Ackerman, S. (2011). Army Taps Android Phones for 'Wearable Computers'. *Danger Room* (6 Sept 2011), <http://www.wired.com/dangerroom/2011/09/nett-warrior-smartphone/>
- [2] Ou, J., Fussell, S.R., Chen, X., Setlock, L.D., & Yang, J. (2003) Gestural Communication over Video Stream: Supporting Multimodal Interaction for Remote Collaborative Physical Tasks. In *Human-Computer Interaction Institute*. Paper 59.
- [3] Kirk, D. S., Rodden, T. & Stanton Fraser, D. (2007) Turn It This Way: Grounding Collaborative Action with Remote Gestures. In *Proceedings of CHI Conference on Human Factors in Computing Systems*, 28th April-3rd May, ACM: San Jose, CA, pp1039-1048
- [4] Kraut, R. E., Fussell, S. R., & Siegel, J. Visual information as a conversational resource in collaborative physical tasks. *Human Computer Interaction*, 18, 1 (2003).13-49.
- [5] Wickens, C. D., & McCarley, J. Applied Attention Theory, In Taylor & Francis, Boca-Raton, FL. (2008)
- [6] Clark, H. & Wilkes-Gibbs, D. (1986) Referring as a collaborative process. *Cognition* 22, 1-39.

GRAPH DRAWING BY STOCHASTIC CONTINUATION

Marc C. Robini

CREATIS (CNRS UMR 5220, INSERM U1044), INSA Lyon, 69621 Villeurbanne, France

ABSTRACT

The problem of producing aesthetically pleasing drawings of graphs is often formulated in terms of optimization. One of the most popular energy model was introduced by Kamada and Kawai [1], who proposed to measure the quality of a layout as a weighted sum of squared differences between Euclidean and graph-theoretic inter-vertex distances. However, minimizing this energy is challenging even for simulated annealing (SA). In this paper, we consider a recently introduced class of generalized SA algorithms—namely, stochastic continuation (SC)—in which the temperature controls the energy function and the candidate-solution generation mechanism in addition to the acceptance rate of uphill moves. Compared to SA, our SC approach to minimizing the Kamada-Kawai energy reveals the complexity of the optimization problem gradually and increases the mixing rate at low temperatures. Experiments on examples from graph drawing contests show that the proposed SC algorithm outperforms SA with optimal communication setting both in terms of minimizing the energy function and in terms of standard aesthetic criteria.

1. INTRODUCTION

The problem of producing aesthetically pleasing drawings of undirected, connected graphs is the subject of considerable attention in computer graphics research. We refer to [2, 3] for a comprehensive introduction to this topic. The most popular approaches to graph drawing are based on physically-inspired models that are expressed in terms of forces acting on the objects (the vertices) of the system (the graph) or in terms of a potential energy reflecting the internal stress of the system; the layout is then the result of the simulation of the relaxation of the system, which amounts to minimize a cost function defined either implicitly [4, 5] or explicitly [1, 6]. Among these approaches, the energy model introduced by Kamada and Kawai [1]—which measures the quality of a layout as a weighted sum of squared differences between Euclidean and graph-theoretic inter-vertex distances—has the following advantages: (i) it is simple compared to cost functions that involve multiple constraints and multiple free parameters (such as in [6]), and (ii) it favors smooth drawings with small maximum-to-minimum edge-length ratio, small edge-length dispersion, and high amount of symmetry [7], which are three consensus aesthetic criteria (we refer to [8, 9] for discussions

on the criteria for judging the aesthetic quality of a graph layout). However, minimizing the Kamada-Kawai energy is a challenging task.

An attractive solution to the Kamada-Kawai optimization problem is to use simulated annealing (SA), whose popularity stems from its ease of implementation and its global convergence properties. The key feature of SA is to allow uphill moves (that is, moves that increase the value of the energy function) in order to escape local minima. By analogy with the physical process of annealing in solids, uphill moves are accepted with some probability controlled by a temperature parameter that decreases monotonically to zero. As the temperature goes to zero, the invariant measure of the underlying Markov chain model concentrates on the global minima of the energy, and thus we can expect that the process converges to a global minimum if the cooling is sufficiently slow. Early results [10, 11, 12] show that this is indeed the case if the temperature is inversely proportional to the logarithm of the iteration index, but this theoretical advantage is outweighed by well-known practical disadvantages: logarithmic cooling is very slow and the design of SA algorithms has very few degrees of freedom. In fact, it is common practice to allow extra design freedom by relaxing some of the convergence assumptions, but variations on the theme of SA usually come without optimal convergence guarantees. A promising alternative is the theoretically-grounded variant of SA called stochastic continuation (SC), which has been proposed in [13] and recently refined in [14, 15]. In a nutshell, SC is a generalization of SA that allows the objective function and the candidate-solution generation mechanism to depend on temperature. The idea is to ease the annealing process by revealing the complexity of the optimization problem gradually and by adapting the exploration of the state space to the temperature regime.

We propose an original application of SC to minimizing the Kamada-Kawai energy. In particular, we make the energy and the candidate-solution generation mechanism vary with temperature in an effort to speed up the annealing process while satisfying the convergence assumptions of SC. Our experimental results show that the resulting SC algorithm substantially outperforms the enhanced version of the Kamada-Kawai algorithm described in [16, §4.2] as well as SA in terms of energy, maximum-to-minimum edge-length ratio, and number of edge-crossings.

This paper is organized as follows. The graph drawing problem is stated formally in Section 2, and the SA and SC optimization approaches are discussed in Section 3. Section 4 is devoted to the application of SC to graph drawing and experimental results are presented in Section 5.

2. FORMAL STATEMENT OF THE PROBLEM

We confine ourselves to straight-line representations, in which each edge is mapped to a line segment in the plane. Let $G = (\mathcal{V}, \mathcal{E})$ be an undirected, connected graph with vertex set \mathcal{V} and edge set $\mathcal{E} \subseteq C_2^{\mathcal{V}}$, where $C_2^{\mathcal{V}}$ denotes the set of 2-subsets of \mathcal{V} , and let $\mathcal{L} = \llbracket 1, L \rrbracket^2$ be a 2-D square lattice representing the set of possible vertex positions ($\llbracket 1, L \rrbracket$ is a shorthand notation for $\{1, \dots, L\}$). Let $\Omega = \mathcal{L}^{\mathcal{V}}$ be the set of maps from \mathcal{V} to \mathcal{L} , that is, the set of straight-line drawings of G on \mathcal{L} . The objective is to find a global minimum of the energy $U : \Omega \rightarrow \mathbb{R}$ defined by

$$U(\varphi) = \sum_{\{u,v\} \in C_2^{\mathcal{V}}} \left(\frac{\|\varphi(u) - \varphi(v)\|}{d_G(u,v)} - \lambda \right)^2, \quad (1)$$

where $\|\cdot\|$ is the standard Euclidean norm, $d_G(u,v)$ is the graph-theoretic distance between vertices u and v (that is, the length of the shortest paths in G between u and v), and the constant λ is the ideal length of an edge in the drawing. A typical choice for λ is

$$\lambda = \frac{L+1}{\text{diam}(G)}, \quad (2)$$

where $\text{diam}(G)$ stands for the diameter of G .

The energy (1) is a particular case of the raw stress criterion associated with least-squares multidimensional scaling (see, for instance, [17]). Indeed, given an arbitrary permutation $(v_1, \dots, v_{|\mathcal{V}|})$ of \mathcal{V} , we have

$$U(\varphi) = \sum_{1 \leq i < j \leq |\mathcal{V}|} w_{ij} (\lambda_{ij} - \delta_{ij}(\varphi))^2, \quad (3)$$

where $w_{ij} = 1/d_G^2(v_i, v_j)$, $\lambda_{ij} = \lambda d_G(v_i, v_j)$, and $\delta_{ij}(\varphi) = \|\varphi(v_i) - \varphi(v_j)\|$. From [18, 19], we know that the extension of (3) to $(\mathbb{R}^m)^{\mathcal{V}}$ generally has non-global minima and that the associated local minima issue is severe when the dimensionality m is small, as it is in our case.

3. FROM SIMULATED ANNEALING TO STOCHASTIC CONTINUATION

3.1. Simulated annealing

SA uses a so-called *communication mechanism* to propose a new candidate solution from the current solution, which is represented by a symmetric and irreducible Markov matrix q on the state space Ω . (Simply speaking, the irreducibility

means that any state can be reached from any other state in a finite number of moves, and the symmetry guarantees that any such path can be traveled in the opposite direction.) In practice, the construction of a communication mechanism uses a neighborhood system \mathcal{N} on Ω that defines the set of allowed moves and a function Γ that assigns a positive weight to each move:

$$q(\varphi, \psi) = \begin{cases} c\Gamma(\{\varphi, \psi\}) & \text{if } \psi \in \mathcal{N}(\varphi), \\ 1 - c \sum_{\omega \in \mathcal{N}(\varphi)} \Gamma(\{\varphi, \omega\}) & \text{if } \psi = \varphi, \\ 0 & \text{otherwise,} \end{cases} \quad (4)$$

where $0 \leq c \leq 1 / (\sup_{\varphi \in \Omega} \sum_{\omega \in \mathcal{N}(\varphi)} \Gamma(\{\varphi, \omega\}))$.

Given a communication matrix q on Ω , an SA algorithm on the *energy landscape* (Ω, q, U) is a discrete-time, nonhomogeneous Markov chain $(\Phi_n)_{n \in \mathbb{N}}$ with transitions

$$\begin{aligned} & \mathbb{P}(\Phi_{n+1} = \psi \mid \Phi_n = \varphi) \\ &= \begin{cases} q(\varphi, \psi) \exp(-\beta_n(U(\psi) - U(\varphi))^+) & \text{if } \psi \neq \varphi, \\ 1 - \sum_{\omega \neq \varphi} \mathbb{P}(\Phi_{n+1} = \omega \mid \Phi_n = \varphi) & \text{if } \psi = \varphi, \end{cases} \end{aligned} \quad (5)$$

where $t^+ := \max\{t, 0\}$ and where $(\beta_n)_{n \in \mathbb{N}}$ is a nondecreasing, divergent sequence of inverse temperatures called a *cooling schedule*. This transition mechanism is implemented as follows. Let φ_n be the current state at step n . A new candidate solution ψ is first drawn from the distribution $q(\varphi_n, \cdot)$ on Ω . If the move from φ_n to ψ decreases the energy, that is, if $U(\psi) \leq U(\varphi_n)$, then the new current state φ_{n+1} is set to ψ without further condition. Otherwise, if $U(\psi) > U(\varphi_n)$, then φ_{n+1} is set to ψ with probability $p_n = \exp(-\beta_n(U(\psi) - U(\varphi_n)))$ and to φ_n with probability $1 - p_n$.

Let Ω_{inf} be the set of global minima of U . It is well-known that for logarithmic cooling schedules of the form $\beta_n = \beta_0 \ln(n+1)$, the convergence measure

$$\mathcal{M}(n) = \sup_{\varphi \in \Omega} \mathbb{P}(\Phi_n \notin \Omega_{\text{inf}} \mid \Phi_0 = \varphi) \quad (6)$$

goes to zero as $n \rightarrow +\infty$ provided β_0 is sufficiently small [11]. However, it is shown in [20] that $\mathcal{M}(n)$ cannot decrease faster than some optimal power α_{opt} of $1/n$ and that exponential cooling sequences must be preferred when the available computing time is finite; these sequences are of the form

$$\beta_n^N = \beta_{\text{inf}} \left(\frac{\beta_{\text{sup}}}{\beta_{\text{inf}}} \right)^{\frac{1}{\nu-1} (\lceil \frac{\nu}{N} n \rceil - 1)}, \quad (7)$$

where N is horizon of the algorithm, β_{inf} and β_{sup} respectively denote the initial and final inverse temperatures, $\lceil \cdot \rceil$ is the ceiling function, and ν is the number of constant-temperature stages.

3.2. Stochastic continuation

SC is a generalization of SA in which both the communication mechanism and the energy function can vary with temperature [14, 15]. Let $(q_\beta)_{\beta \in \mathbb{R}_+}$ be a family of Markov matrices on Ω that converges pointwise to a Markov matrix q_∞ on Ω (that is, $\forall (\varphi, \psi) \in \Omega^2$, $q_\beta(\varphi, \psi) \rightarrow q_\infty(\varphi, \psi)$ as $\beta \rightarrow +\infty$), and let $(U_\beta)_{\beta \in \mathbb{R}_+}$ be a family of real-valued functions on Ω that converges pointwise to the target energy U . An SC algorithm with *communication scheme* $(q_\beta)_\beta$ and *continuation scheme* $(U_\beta)_\beta$ is an inhomogeneous Markov chain $(\Phi_n)_{n \in \mathbb{N}}$ with transitions

$$\begin{aligned} P(\Phi_{n+1} = \psi \mid \Phi_n = \varphi) \\ = \sum_{\varphi \neq \psi} q_{\beta_n}(\varphi, \psi) \exp(-\beta_n(U_{\beta_n}(\psi) - U_{\beta_n}(\varphi))^+). \end{aligned} \quad (8)$$

(The implementation is the same as for SA except that the matrix q and the energy U are respectively replaced by q_{β_n} and U_{β_n} at step n .)

It is shown in [15] that SC is optimal under the following weak assumptions:

- (i) q_∞ has symmetric support, that is, $q_\infty(\varphi, \psi) > 0 \iff q_\infty(\psi, \varphi) > 0$ for all $(\varphi, \psi) \in \Omega^2$;
- (ii) q_∞ is irreducible;
- (iii) q_∞ can rest anywhere, that is, $q_\infty(\varphi, \varphi) > 0$ for all $\varphi \in \Omega$;
- (iv) $\Delta(q_\beta) = \Delta(q_\infty)$ for β sufficiently large, where $\Delta(q) := \{(\varphi, \psi) \in \Omega^2 \mid \varphi \neq \psi \text{ and } q(\varphi, \psi) > 0\}$.

More precisely, if the above conditions are satisfied, exponential cooling makes it possible for SC to have a convergence speed exponent arbitrarily close to the optimal exponent α_{opt} of SA; that is, for any $\alpha < \alpha_{\text{opt}}$, there exists a family $\{(\beta_n^N)_{n \in [0, N]}; N \in \mathbb{N}^*\}$ of exponential cooling sequences of type (7) such that $\mathcal{M}(N) \leq N^{-\alpha}$ for N large enough.

4. APPLICATION TO GRAPH DRAWING

Here, we describe our choices for the application of SA and SC to the graph drawing problem stated in Section 2. We start with the communication matrix of the annealing algorithm and we continue with the description of the continuation scheme $(U_\beta)_\beta$. Then, we discuss the design of the communication scheme $(q_\beta)_\beta$ whose definition involves q and whose choice is implicitly guided by that of $(U_\beta)_\beta$. We end this section with the specification of the functions that control the variations of U_β and q_β with β .

4.1. A communication mechanism for SA

The design of a communication mechanism for SA must balance two conflicting objectives. On the one hand, the annealing chain should mix rapidly at high temperatures to ensure efficient exploration of the state space, which calls for

large communication neighborhoods. On the other hand, it is desirable that the acceptance rate remains significant in the low temperature regime, which requires small communication neighborhoods.

We propose a vertex-by-vertex communication dynamics in which the amplitude of the vertex moves obeys a Rayleigh distribution: the neighbors of a given layout φ are the layouts that differ from φ by the location of a single vertex, and the difference vector between the candidate and the current positions of a vertex is of the form

$$\delta(r, \theta) = ([r \cos \theta], [r \sin \theta]), \quad (9)$$

where $[\cdot]$ denotes the nearest integer function, the angle θ is a sample from the uniform density on $[0, 2\pi]$, and the radius r is a sample from the density

$$f_R : r \in \mathbb{R}_+ \mapsto \frac{\pi r}{2R^2} \exp\left(-\frac{\pi r^2}{4R^2}\right), \quad (10)$$

where the mean value R is fixed. (The balance between rapid mixing at high temperatures and reasonable mobility at low temperatures is achieved by adjusting R , but the optimal value R^\dagger of R cannot be predicted because of its intricate dependence on the graph to be drawn.) The communication matrix associated with this dynamics is clearly symmetric and irreducible; we denote it by q_R . Formally, q_R is of the form of (4) with $c = |\mathcal{V}|$ and with neighborhood system and weight function respectively given by

$$\mathcal{N}(\varphi) = \{\psi \in \Omega \mid \exists! v \in \mathcal{V}, \psi(v) \neq \varphi(v)\} \quad (11)$$

$$\text{and } \Gamma(\{\varphi, \psi\}) = \int_{\mathcal{S}((\psi - \varphi)(v))} \frac{f_R(\|s\|)}{\|s\|} ds, \quad (12)$$

where $\mathcal{S}(\mathbf{a})$ denotes the unit square in \mathbb{R}^2 centered at point \mathbf{a} and v is the (unique) vertex at which φ and ψ differ.

4.2. The continuation scheme

We define $(U_\beta)_\beta$ from the target energy U by making the ideal edge-length vary with β , that is,

$$U_\beta(\varphi) = \sum_{\{u, v\} \in \mathcal{C}_2^{\mathcal{V}}} \left(\frac{\|\varphi(u) - \varphi(v)\|}{d_G(u, v)} - \Lambda(\beta) \right)^2, \quad (13)$$

where the control function $\Lambda : \mathbb{R}_+ \rightarrow (0, \lambda]$ increases monotonically to λ . The main feature of this scheme is the implicit control of the amplitude of candidate vertex moves. Indeed,

$$U_\beta(\varphi) \propto U^* \left(\frac{\lambda}{\Lambda(\beta)} \varphi \right), \quad (14)$$

where U^* is the extension of U to $(\mathbb{R}^2)^{\mathcal{V}}$. Therefore, moving a vertex location by a distance r when considering U_β is equivalent to a move of length $\lambda r / \Lambda(\beta)$ from the standpoint

of the target energy. In particular, using (13) together with the SA communication matrix q_R amounts to considering the target energy and replacing R by the “effective mean”

$$R_\beta = \frac{\lambda R}{\Lambda(\beta)}. \quad (15)$$

4.3. The communication scheme

Intuitively, the communication scheme $(q_\beta)_\beta$ should allow balanced exploration of the state space at the beginning of the SC process, and it should favor moves towards nearby minima by the end of the SC process. An efficient way to get this behavior is to design two distinct communication mechanisms for the high- and low-temperature regimes and to control the probability of choosing one over the other as a function of β ; that is,

$$q_\beta = (1 - \Xi(\beta)) \bar{q} + \Xi(\beta) \underline{q}, \quad (16)$$

where \bar{q} and \underline{q} respectively denote the high- and low-temperature communication matrices and $\Xi(\beta)$ is the probability of choosing \underline{q} rather than \bar{q} to generate a new candidate solution. We impose that the control function $\Xi : \mathbb{R}_+ \rightarrow [0, 1]$ is monotonically increasing and that $\Xi_{\text{sup}} := \lim_{\beta \rightarrow +\infty} \Xi(\beta) < 1$. It is then easily seen that the assumptions of SC hold if the following conditions are satisfied:

- (i) \bar{q} has symmetric support;
- (ii) \bar{q} is irreducible;
- (iii) \bar{q} can rest anywhere;
- (iv) $\Delta(\underline{q}) \subseteq \Delta(\bar{q})$.

A natural choice for \bar{q} is the SA communication matrix q_R with R large enough to ensure rapid mixing at high temperatures. In this case, the only constraint on the choice of \underline{q} is that $\Delta(\underline{q}) \subseteq \Delta(q_R)$, and thus any vertex-by-vertex communication dynamics is appropriate. We propose to use a mechanism similar to q_R , except that the angle θ is computed deterministically so as to move towards a minimum of the 2-D function obtained by viewing the target energy as a function of $\varphi(v)$ only. More precisely, given the current solution φ together with the vertex v subject to potential move, we consider the function

$$W_{\varphi,v} : \mathbf{a} \in \mathbb{R}^2 \mapsto U^*(\varphi_{v,\mathbf{a}}), \quad (17)$$

where $\varphi_{v,\mathbf{a}} : \mathcal{V} \rightarrow \mathbb{R}^2$ is given by

$$\varphi_{v,\mathbf{a}}(u) = \begin{cases} \mathbf{a} & \text{if } u = v, \\ \varphi(u) & \text{if } u \neq v. \end{cases} \quad (18)$$

Drawing inspiration from the vertex repositioning method of Kamada and Kawai [1], we approach a minimum of $W_{\varphi,v}$ by following the Newton search direction $\delta_N(\varphi, v)$ defined by

$$\nabla^2 W_{\varphi,v}(\varphi(v)) \delta_N(\varphi, v) = -\nabla W_{\varphi,v}(\varphi(v)), \quad (19)$$

where $\nabla^2 W_{\varphi,v}$ and $\nabla W_{\varphi,v}$ respectively stand for the Hessian matrix and the column gradient-vector of $W_{\varphi,v}$.

4.4. Control functions

We take the control function Λ of the continuation scheme to increase linearly with β : given $\lambda_{\text{inf}} \in (0, \lambda)$, we set

$$\Lambda(\beta) = \lambda_{\text{inf}} + (\lambda - \lambda_{\text{inf}}) \frac{\beta - \beta_{\text{inf}}}{\beta_{\text{sup}} - \beta_{\text{inf}}}, \quad (20)$$

where β_{inf} and β_{sup} are the initial and final inverse temperatures of the cooling sequence. Therefore, according to (15), the mean amplitude R_β of the proposed vertex moves decreases inversely linearly with β . We choose the parameter R of the high-temperature communication matrix $\bar{q} = q_R$ so that $R_{\beta_{\text{inf}}}$ is equal to $L/10$, that is,

$$R = \frac{L}{10} \frac{\lambda_{\text{inf}}}{\lambda}. \quad (21)$$

Besides, since $R_{\beta_{\text{sup}}} = R$, the mean amplitude of the proposed vertex moves at the end of the SC process can be freely chosen by adjusting λ_{inf} . We set $\lambda_{\text{inf}} = \lambda/10$ so that $R_{\beta_{\text{sup}}} = L/100$. (It should be stressed that our SC algorithm is not sensitive to the choice of Λ as long as R_{β_n} decreases smoothly with n . For example, the performance is the same when R_{β_n} decreases linearly with n or when R_β decreases linearly with increasing temperature.)

The choice of the control function Ξ of the communication scheme is slightly more delicate, as encouraging deterministic moves towards local minima too early in the optimization process is a waste of computation time. In order to delay the impact of the semi-deterministic communication mechanism \underline{q} , we propose to take

$$\Xi(\beta) = \Xi_{\text{sup}} \frac{\beta^2 - (\beta_{\text{inf}})^2}{(\beta_{\text{sup}})^2 - (\beta_{\text{inf}})^2} \quad (22)$$

with $\Xi_{\text{sup}} \in (0, 1)$. The parameter Ξ_{sup} is set to 0.8 in our experiments, but any value greater than 0.5 is good.

5. EXPERIMENTS

We selected five test graphs that challenge both the Kamada-Kawai algorithm and SA. These connected, undirected graphs are shown in Figs. 1 and 2 (the layouts were produced by our SC algorithm), they are from the 12th and 13th graph drawing contests held in conjunction with the 2005 and 2006 International Symposia on Graph Drawing. These test graphs have varying number of vertices, varying average vertex-degree and varying diameter; their basic characteristics are summarized in Table 1.

We consider the following algorithms:

1. the enhanced version of the Kamada-Kawai algorithm — denoted by KK — described in [16, §4.2];
2. the annealing algorithm SA_R defined by the communication matrix q_R (see Section 4.1) with $R = L/10$ and with $R = L/100$;

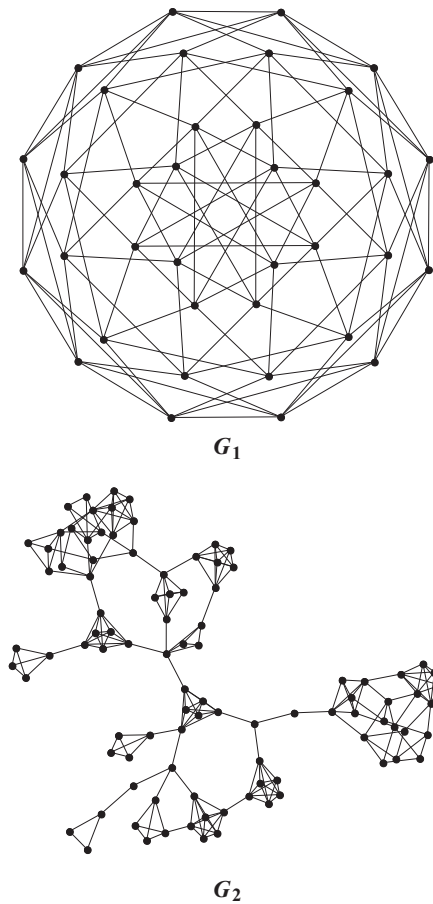


Fig. 1. Graphs G_1 and G_2 : graphs 4 and 1 of the 12th graph drawing contest held in conjunction with the 2005 graph drawing symposium in Limerick, Ireland.

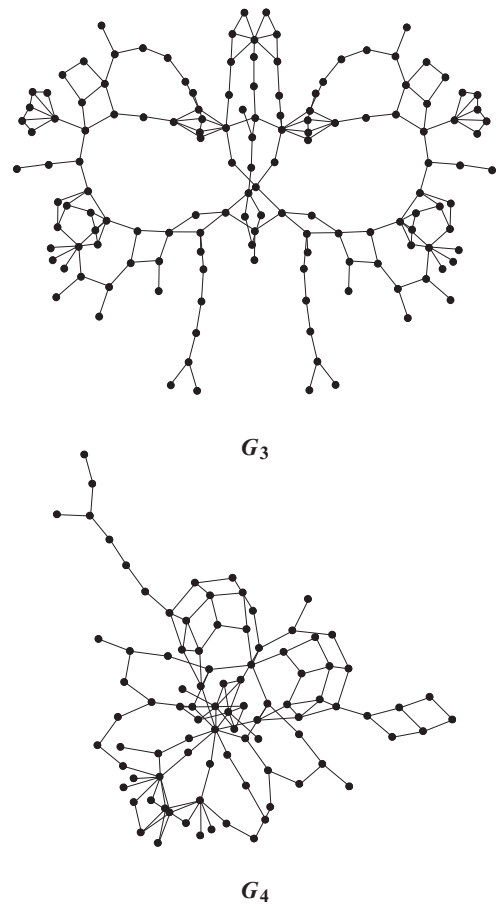


Fig. 2. Graphs G_3 and G_4 : graphs 4 and 2 of the 13th graph drawing contest held in conjunction with the 2006 graph drawing symposium in Karlsruhe, Germany.

3. the SC algorithm with continuation and communication schemes defined in Sections 4.2, 4.3 and 4.4.

The values $L/10$ and $L/100$ of the parameter R coincide with those of the mean amplitude of the vertex moves proposed at the beginning and at the end of the SC algorithm (from our experience, the performance of SA decreases when R lies outside the range $[L/100, L/10]$). The lattice size L is set to 10^3 and the ideal edge-length λ is defined as in (2). The cooling sequence of the SA and SC algorithms is of the form of (7) with horizon $N = 5 \cdot 10^3 |\mathcal{V}|$ divided into $\nu = 250$ constant-temperature stages of length $20|\mathcal{V}|$. The initial and final inverse temperatures β_{inf} and β_{sup} are estimated by means of the procedures proposed in [21].

For each graph G_i , each algorithm is run 100 times starting from the elements of a same set of randomly generated layouts. The performance is measured by the average \bar{U} of the Kamada-Kawai energy (1) and the average $\bar{\rho}$ of the maximum-to-minimum edge-length ratio over the drawings produced by the different runs. We also consider the number of edge-crossings, which is an important aesthetic criterion

for graph layout, although the Kamada-Kawai energy sacrifices it for symmetry. Edge-length dispersion, on the other hand, is left aside because it was not found to be discriminating. Obviously, the smaller \bar{U} , $\bar{\rho}$ and the number of edge-crossings, the better.

For proper comparison, we also include the best results obtainable by SA, that is, those produced by the SA algorithm with the optimal value R^\dagger of R ($R^\dagger = 39, 71, 84$ and 69 for G_1, G_2, G_3 and G_4 , respectively). However, it is important to be clear that the estimation of R^\dagger is unrealistic in practice. Indeed, the function that maps R to the final energy level achieved by SA_R has local valleys and hence, since the computational cost of a single evaluation of this function is that of annealing, the task of estimating R^\dagger is prohibitive in terms of computation time.

Our results are summarized in Table 2. We make the following observations.

Final energy level. Predictably enough, the optimally tuned annealing algorithm SA_{R^\dagger} outperforms KK for all test graphs. However, the performance of SA can vary signifi-

Table 1. Number of vertices, number of edges, average vertex-degree, and diameter of the test graphs displayed in Figs. 1 and 2

$G_i = (\mathcal{V}_i, \mathcal{E}_i)$	G_1	G_2	G_3	G_4
$ \mathcal{V}_i $	36	97	143	92
$ \mathcal{E}_i $	108	223	194	133
$2 \mathcal{E}_i / \mathcal{V}_i $	6	4.60	2.71	2.89
$\text{diam}(G_i)$	4	13	18	16

cantly with R (as is the case for G_2 and G_3) and SA can even perform worse than the KK algorithm for plausible values of R (as happens for G_3 with $R = L/100$). The average energy level achieved by the SC algorithm is lower than that associated with SA_{R^\dagger} in all cases.

Maximum-to-minimum edge-length ratio. The average maximum-to-minimum edge-length ratio of the layouts produced by SA depends strongly on R for all test graphs; it tends to increase with increasing R because large amplitude moves preclude fine edge-length adjustments. Surprisingly, SA_{R^\dagger} does not always perform better than KK according to this aesthetic criterion, as observed for G_2 , G_3 and G_4 . Since SA_{R^\dagger} outperforms KK in terms of minimizing the Kamada-Kawai energy, this shows that there exists low-energy drawings with unbalanced edge-length, and it is tempting to conclude that the Kamada-Kawai energy does not control edge-length dispersion. But the results produced by SC do not support this claim: SC systematically outperforms KK and SA_{R^\dagger} in terms of maximum-to-minimum edge-length ratio.

Number of edge-crossings. Depending on the graph structure, and just as for the two other performance measures, the average number of edge-crossing of the layouts produced by SA can vary significantly with R , as observed for G_1 and G_3 . The drawings produced by SA_{R^\dagger} have a smaller number of edge-crossings than those generated by KK, but SA_{R^\dagger} performs slightly worse than $SA_{L/10}$ in the case of G_1 . The SC algorithm outperforms all three annealing algorithms in any case.

In summary, SC outperforms the optimally tuned annealing algorithm SA_{R^\dagger} in terms of the Kamada-Kawai energy, the maximum-to-minimum edge-length ratio and the number of edge-crossings. It may happen that SA_R produces better results than SC according to either edge-length dispersion or edge-crossings for some particular value of R (for instance, the layout of G_3 produced by $SA_{L/100}$ has smaller edge-length dispersion than the one generated by SC). However, the performance difference in question is usually not significant and is always balanced by the the two other aesthetic criteria.

Table 2. Average results over 100 runs of the competing algorithms: Kamada-Kawai (KK), SA with mean vertex-move amplitude equal to $L/10$ and $L/100$ ($SA_{L/10}$ and $SA_{L/100}$), SA with optimal mean vertex-move amplitude (SA_{R^\dagger}), and SC. Given are the averages of the final energy level, the max-to-min edge-length ratio and the number of edge-crossings for each test graph.

Average final energy level ($\times 10^{-5}$)				
Algorithm	G_1	G_2	G_3	G_4
KK	61.429	9.654	11.388	10.008
$SA_{L/10}$	57.107	8.942	9.573	9.636
$SA_{L/100}$	57.100	9.510	11.447	9.771
SA_{R^\dagger}	56.771	8.736	9.403	9.547
SC	56.716	8.606	9.086	9.427
Average max-to-min edge-length ratio				
Algorithm	G_1	G_2	G_3	G_4
KK	4.30	4.30	3.85	2.93
$SA_{L/10}$	3.18	6.37	4.81	4.10
$SA_{L/100}$	2.83	3.96	3.51	2.85
SA_{R^\dagger}	2.66	4.99	4.77	3.41
SC	2.55	3.67	3.73	2.76
Average number of edge-crossings				
Algorithm	G_1	G_2	G_3	G_4
KK	252	136	18	46
$SA_{L/10}$	209	128	21	40
$SA_{L/100}$	237	132	15	41
SA_{R^\dagger}	212	127	15	38
SC	209	127	14	38

6. CONCLUSION

We proposed an efficient SC approach to graph drawing, which is based on clever temperature-parameterization of the communication mechanism and of the Kamada-Kawai energy. In the proposed algorithm, the evaluation of the parameterized energy difference that takes place at each iteration does not require more computational effort than evaluating the target energy difference, and thus our approach has the same time-complexity as that of SA. With this in mind, our experimental results show that well-designed SC algorithms can outperform SA with optimal communication setting at no additional cost. More generally, the flexibility and the global convergence properties of SC makes it potentially attractive for a wide range of difficult optimization problems.

7. REFERENCES

- [1] T. Kamada and S. Kawai, "An algorithm for drawing general undirected graphs," *Inform. Process. Lett.*, vol. 31, no. 1, pp. 7–15, 1989.
- [2] I. Tollis, G. Di Battista, P. Eades, and R. Tamassia, *Graph drawing: algorithms for the visualization of graphs*, Prentice Hall, 1999.
- [3] M. Kaufmann and D. Wagner, Eds., *Drawing graphs: methods and models*, vol. 2025 of *Lecture Notes in Comput. Sci.*, Springer, 2001.
- [4] T. M. J. Fruchterman and E. M. Reingold, "Graph drawing by force-directed placement," *Softw. Pract. Exper.*, vol. 21, no. 11, pp. 1129–1164, 1991.
- [5] A. Frick, A. Ludwig, and H. Mehldau, "A fast adaptive layout algorithm for undirected graphs," in *Proc. Graph Drawing'94*, vol. 894 of *Lecture Notes in Comput. Sci.*, pp. 388–403. Springer, 1995.
- [6] R. Davidson and D. Harel, "Drawing graphs nicely using simulated annealing," *ACM. Trans. Graphics*, vol. 15, no. 4, pp. 301–331, 1996.
- [7] F. Brandenburg, M. Himsolt, and C. Rohrer, "An experimental comparison of force-directed and randomized graph drawing algorithms," in *Proc. Graph Drawing'95*, vol. 1027 of *Lecture Notes in Comput. Sci.*, pp. 76–87. Springer, 1996.
- [8] H. Purchase, "Metrics for graph drawing aesthetics," *J. Visual Lang. Comput.*, vol. 13, no. 5, pp. 501–516, 2002.
- [9] C. Ware, H. Purchase, L. Colpoys, and M. McGill, "Cognitive measurements of graph aesthetics," *Inform. Vis.*, vol. 1, no. 2, pp. 103–110, 2002.
- [10] B. Gidas, "Nonstationary Markov chains and convergence of the annealing algorithm," *J. Statist. Phys.*, vol. 39, no. 1/2, pp. 73–131, 1985.
- [11] B. Hajek, "Cooling schedules for optimal annealing," *Math. Oper. Res.*, vol. 13, no. 2, pp. 311–329, 1988.
- [12] T.-S. Chiang and Y. Chow, "On the convergence rate of annealing processes," *SIAM J. Control Optim.*, vol. 26, no. 6, pp. 1455–1470, 1988.
- [13] M. C. Robini, A. Lachal, and I. E. Magnin, "A stochastic continuation approach to piecewise constant reconstruction," *IEEE Trans. Image Process.*, vol. 16, no. 10, pp. 2576–2589, 2007.
- [14] M. C. Robini and I. E. Magnin, "Optimization by stochastic continuation," *SIAM J. Imaging Sci.*, vol. 3, no. 4, pp. 1096–1121, 2010.
- [15] M. C. Robini and P.-J. Reissman, "On simulated annealing with temperature-dependent energy and temperature-dependent communication," *Statist. Probab. Lett.*, vol. 81, no. 8, pp. 915–920, 2011.
- [16] D. Harel and Y. Koren, "A fast multi-scale method for drawing large graphs," *J. Graph Algorithms Appl.*, vol. 6, no. 3, pp. 179–202, 2002.
- [17] I. Borg and P. Groenen, *Modern multidimensional scaling*, Springer, 2009.
- [18] P. Groenen and W. Heiser, "The tunneling method for global optimization in multidimensional scaling," *Psychometrika*, vol. 61, no. 3, pp. 529–550, 1996.
- [19] M. Trosset, "On the existence of nonglobal minimizers of the stress criterion for metric multidimensional scaling," in *Proc. Stat. Comput. Section, ASA*, 1997, pp. 158–162.
- [20] O. Catoni, "Rough large deviation estimates for simulated annealing: application to exponential schedules," *Ann. Probab.*, vol. 20, no. 3, pp. 1109–1146, 1992.
- [21] M. C. Robini, T. Rastello, and I. E. Magnin, "Simulated annealing, acceleration techniques and image restoration," *IEEE Trans. Image Process.*, vol. 8, no. 10, pp. 1374–1387, 1999.

FACIAL EXPRESSION RECOGNITION INTELLIGENT SECURITY SYSTEM FOR REAL TIME SURVEILLANCE

Ashraf Abbas M. Al-modwahi, Onkemetse Sebetela, Lefoko Nehemiah Batleng, Behrang Parhizkar, Arash Habibi Lashkari

Faculty of Information and Communication Technology
Limkokwing University of Creative Technology, Malaysia, Cyberjaya

Abstract: *Facial Expression Recognition is a part of biometric authentication that focuses on uniquely recognizing human facial appearance based upon one or more intrinsic physical or behavioral traits and inside emotions portrayed on one's face. This paper explains on research based on improving current surveillance systems by adding facial expression recognition to make a system that will detect a person's expression (who has intentions of causing harm) and report to the securities before the person can commit any prohibited work. The system captures ones feeling/expressions through a camera and send the footage for preprocessing before it can be extracted to see if the emotion/expression captured has the intentions of causing harm.*

Keywords: Facial Expression Recognition, Motion Detection, Computer Vision and Surveillance.

1 INTRODUCTION

Facial expression is regarded as way of communicating the emotional state of the individual to observers and humans can adopt a facial expression to help read as a voluntary action. The six basic facial expressions are: Joy, Surprise, Fear, Anger, Disgust and Sadness. The technology of facial expression is increasingly gaining attention of most researchers. A couple of systems have been developed based on facial expression, this includes among others: advanced human-computer interfaces, improved relations with the e-commerce consumers, and intelligent system for autistic children.

On the other hand Biometrics is seen as a way of recognizing an individual's distinctive physical or behavioral characteristics to distinguish or verify their identity. The common physical biometrics are fingerprints, hand or palm geometry, retina, iris, and facial characteristics. Many years ago biometric technologies have been used for entry access in secure environments and nowadays the primary application of biometrics is in physical security: to control access to secure locations (rooms or buildings). Biometric devices, typically hand geometry readers, are in office buildings, hospitals, casinos, health clubs and lodges. Biometrics is

useful for high-volume access control.

Facial expression recognition and biometrics can play a major role in preventing many criminal activities from happening. This project is aimed at combining both facial expression and biometrics technologies to designing an intelligent surveillance system that will be able to recognize and identify any person if they have any intentions of doing any criminal activities.

2 ANALYSIS

Facial expressions from ages have been linked with people's feelings. Lie to me, a television series that goes by the motto of "Words lie but your face doesn't", also emphasis that facial expression correlates with peoples feeling. Reading human facial expressions can help us in our day-to-day activities such as in security issues, gaming technology, marketing and advertising and expert system. From the observations and study we made, we have realized that most surveillance facial recognition systems only automatically check every person against a database of known suspects. This makes it hard for securities to catch someone who has the intentions of stealing or about to do any forbidden act. Available surveillance face recognition systems are only designed for blacklist searches in criminal investigation, for the control and enforcement of bans in areas such as airports, stations, as well as casinos and sport arenas. The systems do not detect the expression of the people in the area thus giving them an allowance to commit crimes.

Figure 1 shows an example of an existing surveillance face recognition system that checks every person against a database of known suspects.

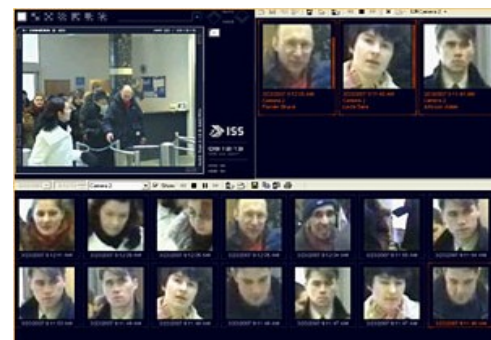


Figure 1: Example of facial recognition System

With these observations and research we see the need to research and develop a real time facial expression recognition system that will help read human facial expressions of people who their expressions show that they can cause harm or they are about to commit prohibited work and instantly alerting the guards through an alarm if the any threat has been detected.

3 RELATED WORK

3.1. Facial Recognition

Most researches stated that in order to detect a face there are some factors that must be considered to get more accurate result. These factors may include the clothes a suspect is wearing, background color and skin illumination. (Peng, Zhu, Zhou, 2010) in their research stated that they used an adaptive Gamma Corrective method in order to get rid of influence of illumination on the skin color therefore making it easy to detect the face according to the structure of eye, nose and mouth. [1]. (Patil, Ginpunje, Bajaj, 2010) in their researched on facial expression recognition and head tracking in video using Gabor filter they pointed out that facial expression turn to become the key technology of advanced human computer as much as there is a rapid development of computer vision and artificial intelligence. In this study they proposed to use a method of detecting 28 facial key-points in images as a way of increasing the accuracy and robustness. They used Gabor filters because of its superior capability of multi-scale representation. In their process of facial expression recognition they used Gaussian filter in order to reduce some common types of noises in an image. They also applied the use of Gabor filters in the extraction process because it is used to define edges and it is the fastest method for training moderate-sized feed forward neural networks, the Levenberg-Marquardt algorithm was applied in training. [2]. In other researches (Zhao, Huang, Dellandré'a, Che, 2010) proposed automatic 3d facial expression recognition based on a Bayesian belief net and a statistical facial feature model. Bayesian Belief Network is used for 3D facial expression recognition in 3D environment because it has a novel structure that allows adding new feature and new expression flexibly. Also proposed was a novel method to compute parameters in BBN for inference belief of expression states based on the SFAM. The two (BBN with SFAM) were combined for the system to be capable of recognizing facial expression in 3D automatically and efficiently (Figure 2). [3]

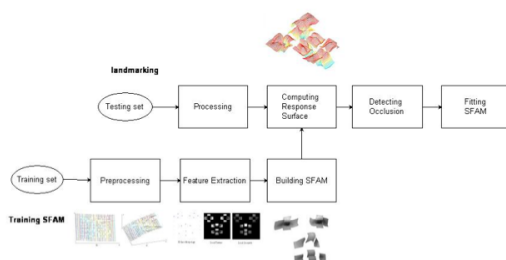


Figure 2: Flowchart for automatic land marking using SFAM

(Matai, Irturk, Kastner, 2011) presented the design and implementation of an FPGA-based real-time face recognition system (Figure 3). The system consists of 3 subsystems, which are face detection, face recognition that uses the Eigen face algorithm and down sampling. The complete system uses a camera to send the video data to the face detection subsystem, which in turn sends detected faces to the face recognition subsystem through the down sampling module that prepare images, it resize the image using the coordinates for input to face recognition. The design and implement of a face recognition subsystem on an FPGA uses both pipelined and non-pipelined architectures. FPGA Implementation of the face recognition subsystem is performed in two steps: generates the training data and the face recognition. The training data is generated using the OpenCV library and evaluate the feasibility of the face recognition subsystem using the ORL database. Face Recognition: stores the average image, the weight vectors for each image and the Eigenvectors in a block RAM. [4]

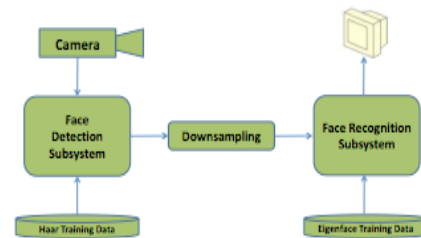


Figure 3: Flow of the FPGA-based real time face recognition system.

(Li, Phung, Bouzerdom, Tivive, 2010), proposed automatic recognition of smiling and neutral facial expression. The system has two major functionalities which are; face detection and alignment, and classification of smiling and neutral. In face detection and alignment method, the candidate eye regions are detected, based on the elongated shape of the eye and the circular shape the pupil. We will use a Gabor filter, the product of a harmonic function and a Gaussian function and a circular filter. In classification of smiling and neutral, the smiling and neutral facial expressions are differentiated using a new neural architecture, which consists of three processing stages. The three processing stage are Directional Filters; is designed to extract features at different orientations, Trainable Filters; aims to detect more complex features for classification and Classification; for classification, may use any type of classifiers. [5]

3.2. Computer Vision

(Miller G, Fels S, Oldridge S, 2011) they researched a conceptual structure for computer vision. They discuss scope of computer vision analysis and discuss a new categorization of the computer vision problem. The central theme of their contributions is to see computer vision becoming more accessible to researchers and developers alike. They use the axioms of vision as an

abstraction that builds on top of algorithmic approaches such as OpenCV to leverage the specialized knowledge contained in their impressive array of algorithms. In their axioms of vision, there are mathematical axioms, source axioms, model axioms and construct axioms. For their algorithms and problems, they discuss on how to use the axioms above using algorithm composition, problem decomposition and formal description model. They have presented the idea of an interpreter, which would take the formal description and translate it such that the correct algorithm for the problem may be chosen. [6]

(Ohsaki H, Nozaki K, Baba K, Sakane E, Sakamoto N, Koyamada K, Shimojo S, 2011) they discussed Peta-Flow Computing: Vision and Challenges. In the ICT, Peta-flow computing integrates the principal computing, networking, and interfacing technologies. It has got short, mid and long-term goals. To enable large-scale computing for geographically dispersed large volume data, Peta-scale computing is used. To enable transparent access to geographically disperse large-volume data Peta-scale networking is used and Peta-scale interfacing is a set of technologies for enabling input/output of geographically dispersed large-volume data from/to users. In all these technologies, it is noted that integrated high-speed networking with computing and interfacing technologies is necessary. At last it is important to note that high-performance computing such as large-scale computations/simulations have been leading applications of information and communication technologies, thus it is expected to continue in the peta-scale era. [8]

(Miller G, Fels S, Oldridge S, 2011) they researched a conceptual structure for computer vision. They discuss scope of computer vision analysis and discuss a new categorization of the computer vision problem. The central theme of their contributions is to see computer vision becoming more accessible to researchers and developers alike. They use the axioms of vision as an abstraction that builds on top of algorithmic approaches such as OpenCV to leverage the specialized knowledge contained in their impressive array of algorithms. In their axioms of vision, there are mathematical axioms, source axioms, model axioms and construct axioms. For their algorithms and problems, they discuss on how to use the axioms above using algorithm composition, problem decomposition and formal description model. They have presented the idea of an interpreter, which would take the formal description and translate it such that the correct algorithm for the problem may be chosen. [7]

3.3. Motion Detection

(Rao I, Murphy P, Nandy S, Rao V, 2009) they analyzed a real world system for detection and tracking. Their analysis comes as a necessity of inventing an intelligent surveillance system, which will make lives easier as well as enable competition with tomorrow's technology. Background subtraction is the method applied. They used pixel based approach so as to compare the corresponding pixel values of the foreground with reference to standard background. Their algorithm is implemented using image processing in MatLAB Environment of which they can achieve motion detection. They up an alarm or alert system, which is controlled by a Microcontroller

(AT89C51) which is interfaced with the personal computer system through a serial communication path initiated by the MatLAB code written and executed in the personal computer. By successfully interfacing the hardware and Personal Computer system with each other by serial communication, using RS232 protocol, the microcontroller-based hardware detects the data from the system and gives the desired output. [9]. (Xiaoping LI, Tao LI, Feng LING, Jianqiang XU, 2010) Studied the technology of motion detection and target contour extraction in the intelligent analysis of video. They gets gradient contour information according to operator Canny using double threshold values method based on statistic to divide moving target. Algorithms of Motion Detection discussed in this study are; 1) Frame Difference Method: it has nothing to do with the time varying. The movement of objects cannot be described with a still picture. But this algorithm cannot extract all the moving features of targets. 2) Background Subtraction Method: a method that realizes motion detection through the subtraction of image set and referenced background model, providing holistic data of features and extract relatively holistic moving target. 3) Moving Target Detection: uses the difference between varying image and relevant background image to detect moving objects for the current image. They extracted the contour lines of Snake Model Base on Cubic B-spline. The contour line of traditional Snake model is a polygon composing of some control points. B-spline curve is located in the polygon. [10]. (Chen Y K, Cheng T Y, Chiu S T, 2009) evaluated the Entropy algorithm of moving average of objects, an alarm system with dynamic threshold. Multi-periods background estimation algorithm is applied in the proposed algorithm to derive the subtraction video. Entropy: An $M \times N$ resolution digital image of frame f may be defined as a two-dimensional function where x and y are spatial coordinates, and the amplitude of I_f at any pair of coordinates (x, y) is the gray level of the image at that point. The proposed algorithm is using the Manzanera's multi-periods $\Sigma-A$ background estimation algorithm to derive the background video $M_f(x, y)$ from the input video. A single threshold is used to determine if the alarm is not suitable for the Entropy- A curve of the proposed algorithm. [11]

3.4. Surveillance

(Bhaumik G, Mallick T, Chowdhury K S, Dr. Sanyal G 2010) has done a research on analysis and detection of human faces by using minimum distance classifier for surveillance. This visual surveillance system in dynamic scenes that make an effort to detect, recognize and track certain object from image sequences and more generally to understand the human or object behaviour by identifying distinctive face features like the eyes, nose, mouth, and head outline, and defining a face model by the position, size, and relationships among these features. The analysis of the face is done using mathematical analysis which is has two phases, namely:

- 1) Calculating Eigen faces using Principle Component Analysis (PCA)
- 2) Face Image Classification. The type of algorithm used

is the computer algorithm [12] (Lianyang Ma, Zhijing Liu, 2008) researched about human motion recognition in video for surveillance systems (Figure 4). This system automatically detects the moving human (walking) using a camera and following these steps motion detection, objects tracking, and behavior understanding and human identification. (See the framework of detecting human behavior in figure) This is a spatio-temporal motion-based model and offers these two contributions:

1. Inferring and generalizing from just walker silhouette (about validity of much larger context of image patterns and behavior)
2. Using Hidden Markov Models to build human posture detection model and recognize the human motion.

The method used to determine the motion of a moving body is the contour model that goes through the Centroid, Expression of Motive Human beings' Image Sequence, Similarity Compare and standard deviation stages. Hidden Markov Models (HMM) defines a limited state set, and each state associates with a probability distribution (multi-dimension in general). The result of the test has shown a very low recognition rate in the matching method. [13]

(Zhen Lei, Chao Wang, Qinghai Wang, Yanyan Huang 2009) researched on real-time face detection and recognition for video surveillance applications with the aim of integrating and optimization of an automatic face detection and recognition system for video surveillance applications. The work is divided into two phases, the face detection stage and face recognition stage. Under the face detection stage they looked at the extended haar-like features to train the strong classifier that form the cascaded multilayer ear detector and the modest AdaBoost algorithm that generates a sequentially weighted set of weak base classifiers that are combined to form an overall strong classifier. In the face recognition stage, PC-ICA and hamming distance methods were reviewed. The PC-ICA is the method that first focuses on reducing the dimension of the data using principal component analysis (PCA), then adopt a ratio-factor-based ICA for face recognition. Hamming distance calculates the minimum distance between the test image and image to be recognized in the database. Tests were run in CMU+MIT frontal and the method has achieved a better detection rate in CMU+MIT as compared to other methods. [14]

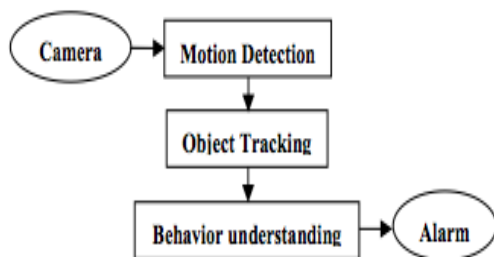


Figure 4: The framework of detecting human behavior

(Ting Shan, Shaokang Chen, Conrad Sanderson, Brian C. Lovell, 2007) examined automatic person recognition for intelligent-cctv based surveillance using one gallery

image. The system uses the pose compensation technique to synthesize realistic frontal face images from non-frontal views. It is based on Active Shape Models (ASMs) and Active Appearance Models (AAMs) which are deformable models popularized by Cootes et al.. it also uses pose estimation correlation models. Adaptive Principal Component Analysis (APCA) is another technique that is used in the system to get features/characteristics from both PCA and Fisher Linear Discriminant by warping the face subspace according to the within and between class covariance. [15]

(Lin F, Fookes C, Chandran V, Sridharan S, 2006) they investigated the role of motion models in super-resolving surveillance video for face recognition. The main aim of their paper was mainly to find out the way in which motion models of different super-resolution reconstruction algorithms affect reconstruction error and face recognition rates in a surveillance environment. They applied Super-resolution image reconstruction model whereby they combine multiple low-resolution (LR) images into one image with higher resolution (Figure 5). A XM2VTS database was used facilitate testing of multi-modal speech recognition systems. After carrying out an experiment, they discovered that super resolution would only be successful if peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) were indicative of recognition performance and if there is accurate registration. They aimed at investigating quantitative measures of image quality that will give an indication of face recognition performance and do experiments on real surveillance footage as their future work. [16]

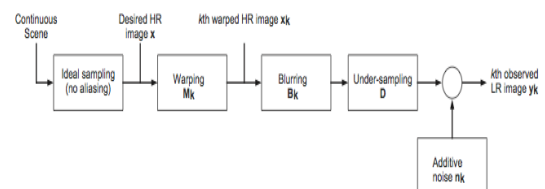


Figure 5: Super-resolution observation model

4 RESEARCH METHODOLOGY

Different projects require different methodologies to be developed. Methodology refers to the framework that is used to structure, plan, and control the process of developing a system [18]. This research follows steps that will help at the end of the research to have a clear understanding of the work and problems. These steps are:

- Understanding the nature of problem to be studied and identifying the related area of knowledge.
- Reviewing literature to understand how others have approached or dealt with the problem.
- Collecting data in an organized and controlled manner so as to arrive at valid decisions.
- Analyzing data appropriate to the problem.
- Drawing conclusions and making generalizations [17]

4.1. Objectives of Research

The research methodology is discussed in two parts: objectives of research and development methodology. The aim of the research is to:

- i. To study surveillance systems and Face expression recognition, and how to combine both technologies to improve face expression recognition (in terms of motion).
- ii. To develop an application for Facial Expression Recognition for surveillance systems.
- iii. To evaluate real time facial expression recognition for intelligent surveillance systems.

4.2. Development Methodology

Development methodology to be used in this research is waterfall Software Development Life Cycle (WF-SDLC). Since this is a security system that combines surveillance system and facial expression technologies to come up with a real-time intelligent surveillance security system we have seen the need to add collaborative methodology to the WF-SDLC because it isn't enough for the WF-SDLC to work alone in a collaborative system. Therefore the system will be designed following the Collaborative Waterfall Software Development Life Cycle (C-WF-SDLC) (Figure 6). This system is developed in a way that it will be able to detect the person's facial expression helping the guards to foresee if the person can cause harm to other people or he has any bad intentions of committing prohibited work. The C-WF-SDLC methodology follows six main phases, that is: Planning, Analysis, Design, Implementation and Maintenance.

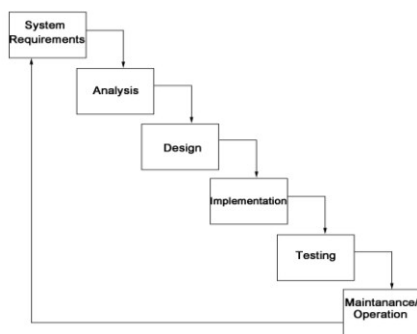


Figure 6: C-WF-SDLC

4.3. System Requirements Phase

The purpose of this phase is to determine project's main goal and how the system will function. To gather system requirements information, these are common questions that have to be answered:

- Why the system needs to be developed?
- Who are the users?
- How will they use the system?
- What are they using the system for?
- What are the input and output of the system?

This question needs to be answered thoroughly to come up with clear functionality of the system describing the

functions that the system should perform. All possible requirements of the system to be developed are captured in this phase. After the requirements are understood Software Requirement Specification (SRS) document is prepared to serve as a guideline for the next phase of the model.

4.3.1. Analysis Phase

In this phase analysis of the user's requirement is carried out. This is to determine the scope of the users based on the SRS prepared in the requirement phase and the observations made on the current systems. Thing to be cogitated are

- Scope of users
- Purpose of the system
- Information on surveillance systems
- Suitable equipment's (camera, laptop etc.)

The overall purpose of the analysis phase is to define the project goals that have been determined in the requirements phase into defined functions and operation of the intended system. It analyzes end-user information needs.

4.3.2. Design Phase

This is the plan of how the system will look like and how it works. It describes the desired features and operations in detail and may include screen layouts, process diagrams, pseudocode and other documentation. A sample of the project is developed in this phase. Design focuses on high level design like, what programs are needed and how are they going to interact, low-level design (how the individual programs are going to work), interface design (what are the interfaces going to look like) and data design (what data will be required). During these phases, the software's overall structure is defined and the logical system of the product is developed in this phase. It also helps in specifying hardware and system requirements and also helps in defining overall system architecture (Figure 7).

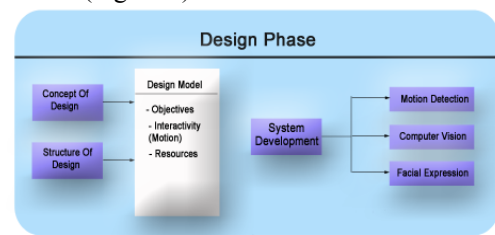


Figure 7: C-WF-SDLC Design Phase

4.3.3. Implementation and Unit Testing Phase

This phase is considered to be the longest phase of the software development life cycle. This is so because this is where the code is created and work is divided into small programs that are referred to as units. This phase include unit testing whereby units will be tested individually for their functionality before the whole system. Unit testing mainly verifies if the modules also known as units meet project specifications.

4.3.4. Testing Phase

This is the main testing phase in the SDLC, as the project is divided in small modules in the previous phase then the modules will be integrated together to test the system as whole. This is to make sure that the modules work together as intended by the developer (as in the specifications) and required by users. It also checks for bugs, errors and ensure the system is able to work in the intended platform. After ensuring that the product had solved the problem the system is then delivered to the customers (Figure 8).

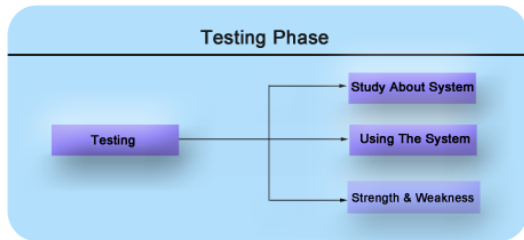


Figure 8: C-WF-SDLC Testing Phase

4.3.5. Maintenance/Operation Phase

Not all problems can be seen directly, but they occur with time and as other problems they needs to be solved. Usually these kinds of problems come in picture after the practical use of the system they are never found throughout the development life cycle. This phase of the Waterfall Model is considered to be very long, it never ends. The changes that occur after the product is handed to the users must not affect the main operation of the

system, so a system must be developed in a way that it will adapt to change.

5 Architectural Diagramme

The architectural diagram (Figure 10) shows the flow of events for the facial expression recognition intelligent security system for real time surveillance system. The camera captures the movements concentrating on the head and in the motion detection module the face will be detected together with the head pose estimation. After the face is detected comes behavior understanding of the facial expression. Behavior understanding has two phases: Verification and Identification. In verification phase the system tries to identify the expression detected by reading action units (the dots in eyes, nose, eyebrows and mouth (figure 9). In identification phase, the system attempts to match if the detected and verified expression portrays any feelings that have intentions of doing any prohibited work. [23].

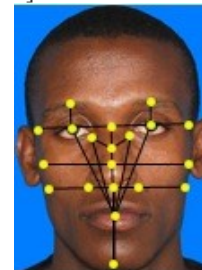


Figure 9: Facial Feature Points/Actin Units

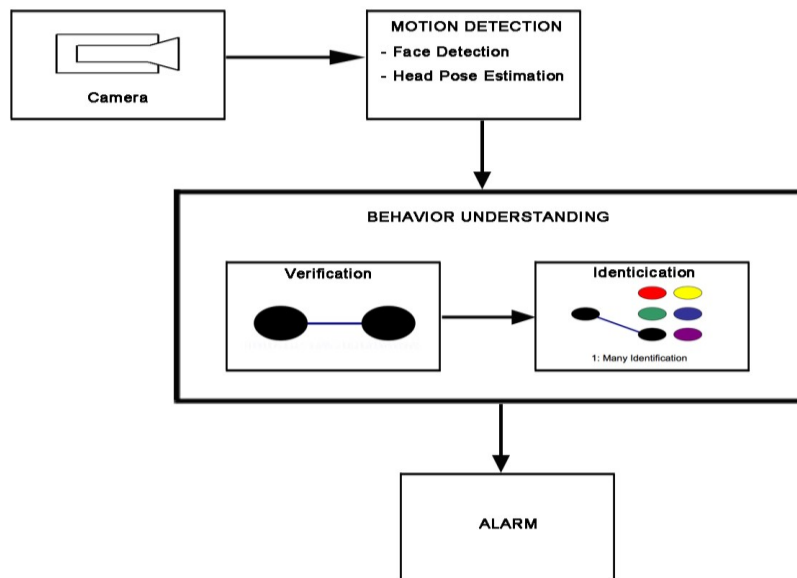


Figure 10: Architectural Diagram of the proposed system

6 Discussions and Conclusion

Facial expression recognition and biometrics can play a major role in preventing many criminal

activities from happening. This research covers the design and development of facial expression surveillance system based on the application domain of security surveillance systems and facial

expression recognition. The good thing about the system is that it will be real time, which will make it more effective and reliable for security guards to catch criminals at that particular juncture before they can even get away with the crime.

7 Acknowledgement

The special thank goes to our helpful advisor Dr. Arash Habibi Lashkari for his supervising and advising in the progression of our dissertation and project.

8 References

- [1] PENG Zhao-yi, ZHU Yan-hui, ZHOU Yu “Real-time Facial Expression Recognition Based on Adaptive Canny Operator Edge Detection”, 2010 Second International Conference on MultiMedia and Information Technology, pp 154-157, 2010.
- [2] PatilK, GiripunjeSD, BajajPR “Facial Expression Recognition and Head Tracking in Video Using Gabor Filter”, Third International Conference on Emerging Trends in Engineering and Technology, pp 152-157, 2010.
- [3] Zhao, Huang, Dellandréa, Che, “Automatic 3D facial expression recognition based on a Bayesian Belief Net and a Statistical Facial Feature Model”, 2010 International Conference on Pattern Recognition, France, pp3724-3727, 2010.
- [4] Matai, Irturk, Kastner, “Design and Implementation of an FPGA-based Real-Time Face Recognition System”, IEEE International Symposium on Field-Programmable Custom Computing Machines, United States, pp97-100, 2011.
- [5] Li, Phung, Bouzerdom, Tivive, “Automatic Recognition of Smiling and Neutral Facial Expressions”, 2010 Digital Image Computing: Techniques and Applications, Australia, pp581-586, 2010.
- [6] Miller G, Fels S, Oldridge S, “A Conceptual Structure for Computer Vision”, 2011 Canadian Conference on Computer and Robot Vision, Canada, pp168-174, 2011
- [7] Miller G, Fels S, Oldridge S, “A Conceptual Structure for Computer Vision”, 2011 Canadian Conference on Computer and Robot Vision, Canada, pp168-174, 2011
- [8] Ohsaki H, Nozaki K, Baba K, Sakane E, Sakamoto N, Koyamada K, Shimojo S, “Peta-Flow Computing: Vision and Challenges”, 2011 IEEE/IPSJ International Symposium on Applications and the Internet, Japan, pp256-259, 2011.
- [9] Rao I, Murphy P, Nandy S, Rao V, “A Real World System For Detection And Tracking”, 2009 International Conference on Advances in Recent Technologies in Communication and Computing, India, pp939-943, 2009.
- [10] Xiaoping LI, Tao LI, Feng LING, Jianqiang XU, “Motion Detection And Target Contour Extraction Based On Monitor System”, 2010 Second International Workshop on Education Technology and Computer Science, China, pp113-116, 2010.
- [11] Xiaoping LI, Tao LI, Feng LING, Jianqiang XU, “Motion Detection And Target Contour Extraction Based On Monitor System”, 2010 Second International Workshop on Education Technology and Computer Science, China, pp113-116, 2010.
- [12] Bhaumik G, Mallick T, Chowdhury K S, Dr. Sanyal G, “Analysis and Detection of Human Faces by using Minimum Distance Classifier for Surveillance”, 2010 International Conference on Recent Trends in Information, Telecommunication and Computing, pp265-267, 2010.
- [13] Ma L, Liu Z, “Human Motion Recognition In Video”, Fifth International Conference on Fuzzy Systems and Knowledge Discovery, pp49-53, 2008.
- [14] Lei Z, Wang C, Wang Q, Huang Y, “Real-Time Face Detection And Recognition For Video Surveillance Applications”, 2009 World Congress on Computer Science and Information Engineering, pp168-172, 2009.
- [15] Shan T, Chen S, Sanderson C, Lovell B C, “Towards Robust Face Recognition For Intelligent-Cctv Based Surveillance Using One Gallery Image”, pp470-475, 2007.
- [16] F. Lin, C. Fookes, V. Chandran and S. Sridharan, “The Role of Motion Models in Super-Resolving Surveillance Video for Face Recognition”, Proceedings of the IEEE International Conference on Video and Signal Based Surveillance (AVSS'06), 2006.
- [17] [ONLINE] Available at: <http://www.cairo.utm.my/webcairo/download/rese>

arch_method/Research%20Methodology%201.pdf
[Accessed 28 October 2011].

[18] [ONLINE] Available at:
<http://www.profit-labs.com/development-process/>
[Accessed 28 October 2011].

[19] Lecuona, M. (2007). A Collaborative System: Case Study in Solution-Architecture Development-Methodology Application. Available: <http://msdn.microsoft.com/en-us/library/cc168627.aspx>. Last accessed 02/12/11.

[20] Unknown. (2011). All About the Waterfall Model. Available: All About the Waterfall Model. Last accessed 2/12/11.

[21] Software Development Methodology. 2011. Software Development Methodology. [ONLINE] Available at:
http://www.hyperhot.com/pm_sdm.htm.
[Accessed 28 October 2011]

[22] Which SDLC Model to choose waterfall, Slam dunk, Spiral, Evolutionary, Stage Gate, Rapid Prototype, Agile, and Sync-Stable « ServiceXen Thoughts about Software Strategy, Marketing and Management. 2011. Which SDLC Model to choose waterfall, Slam dunk, Spiral, Evolutionary, Stage Gate, Rapid Prototype, Agile, and Sync-Stable « ServiceXen Thoughts about Software Strategy, Marketing and Management. [ONLINE] Available at:
<http://servicexen.wordpress.com/2008/09/06/which-sdlc-model-to-choose-waterfall-slamdunk-spiral-evolutionary-stage-gate-rapid-prototype-agile-and-sync-stable/>. [Accessed 31 October 2011].

[23] [ONLINE] Available at:
<http://www.biometrics.gov/documents/biointro.pdf>
[Accessed 8 December 2011]

[24] Michael V, Maja P, Ioannis P (2004). Motion History for Facial Action Detection in Video. Available: <http://pubs.doc.ic.ac.uk/Pantic-SMC04/Pantic-SMC04.pdf>. Last accessed 03 September 2011.

Speech-Driven Animation of an Expressive Talking Avatar

Kevin Gross, Jie Yan

Department of Computer Science, Bowie State University, Bowie, Maryland, USA

Abstract - *This paper explores a method of detecting the positive and negative emotions from speech signal. We use the frequency feature of the speech signal to detect the emotions and use the results to drive the animations of an expressive talking avatar.*

Keywords: Speech Driven, Animation, Facial Expressions, Avatar

1 Introduction

Human emotional characteristics can be identified by the features in the audio and speech signal such as frequency, speaking rate, and placement of stressed and unstressed syllables. The identifying factors enable computers to recognize basic human emotions, thus providing methods for computers to interact more effectively with humans.

In this research, we explored the frequency feature to detect emotions in the speech signal. And then we used the detected emotion to drive the animations of a expressive talking avatar.

It has been proven that different emotions are attributed to certain frequencies. For example, we would expect sadness to have a lower frequency than happiness. In addition, male and female vocal data provide, at times, differing emotional output. Next, we discussed speaking rate and determined that humans vary their rate of speech according to their emotions. Again, male and female data differed for certain emotions. For example, when displaying anger males decrease their speaking rate, while females increase their rate of speech. Lastly, we discussed the placement of stressed and unstressed syllables in speech. It was noted that when expressing certain emotions, such as anger, our vowels are expressed more prominently. Thus, after applying a Gaussian filter on the vocal file we are able to find stressed and unstressed syllables.

2 Related Work

One of the most important factors to consider when attempting to detect human emotions from speech signal is to determine the features used to provide information from which the emotion can be obtained. These features include: speaking rate, speaking frequency, and placement of accented syllables. Among these features, Fundamental frequency is expressed as

the inverse of the period of a waveform expressed in hertz. In Fourier theory, any signal can be expressed as sinusoids and the fundamental frequency is the lowest tone in a harmonic series. When we hear humans talk in daily activities we are hearing the fundamental frequency of their voice [8]. Generally, the frequency of a vocal sample is able to give an accurate indication of a subject's emotional characteristics. In addition, gender must be taken into consideration because male and female express certain emotions differently. Traunmüller and Eriksson state that when comparing male and female frequency and standard deviation, there is a notable difference in the gender data. In excited emotional states they found that women have a higher standard deviation than men and in non-excited states they have lower standard deviation than men. This trend seemed to hold mostly for actor portrayal and to fail in real life emotion analysis. However, it was noted in a Pegoraro-Krook study that the trend was perhaps dependent on the age of the subjects. For example in a cross gender study, those in the age ranges of, 20-29, 30-39, 40-49, and 70-79, the standard deviation was indeed higher for females when compared to that of men. Thus, the trend faulted only at those in the 50-59 and 60-69 age range [2].

Speaking rate is an additional tool used to verify human emotion. Speaking rate is the time for a specific utterance to complete. Mozziconacci and Hermes realized that, when studying speech rate, normally when a subject speaks neutrally their speech rate is similar to their speech rate when happy. When displaying certain emotions speaking rate differs according to the subject [3]. And also, a male expression of anger is generally characterized by slower speaking at higher pitches. In contrast, a female expression of anger is generally characterized as speech at a higher rate and at higher pitches. However, disgust is characterized as low pitch level and slower speech in both male and females [4].

Placement of accented and unaccented syllables give indication of emotion because humans usually stress differing portions of a sentence according to their emotional state. Wang and Narayanan performed syllable stress experiments based on Paul Taylor's work and determined that stressed syllables could be identified after smoothing the vocal data. They stated that when vocal data was smoothed the resultant peak frequency values could be used to give accented syllables. In order to achieve accurate smoothing of the

frequency a Gaussian filter must be applied to normalize the waveform [5][6].

Additional reasoning behind the validity of stressed and unstressed syllables was found in Kienast and Sendlmeier's research of emotional speech. They found that the pronunciation of words differ according to emotional states [7]. This observation is important because vowel placement has a direct relationship to how words are stressed. For example, when displaying anger vowels are pronounced more distinctly. Thus, in a prolonged angry vocal sample we would expect the peak accented syllables to be significantly more pronounced than with sadness.

3 Emotions Detection from Speech

Our research was based on the theory that emotions can be partially detected by the frequency data produced in human's speech. The speech can be obtained in two ways: a live recording of actual human interaction or by an actor portraying a specific emotion. Dan Jurafsky states that scripted speech is beneficial for analyzing frequency, energy, and speaking rate [1]. The use of an actor is beneficial because it allows more control over the emotional data and it allows for repetition, which is useful in developing a skeleton of the subject's emotional range. In addition, he states that actual human speech is beneficial for analyzing frustration, annoyance, and certainty and uncertainty [1]. Our initial research includes data samples taken from scripted speech. After samples are gathered, a method must be established to determine the characteristics of the vocal sample. If a subject's voice is separated as a sequence of frequencies then a pattern should develop according to the subject's vocal tendencies. This separation is performed by the use of Fourier Transformation. As stated before, Fourier theory allows a signal to be expressed as sinusoids. In other words, by use of a Fourier Transformation we can change the original data from the time domain to the frequency domain [9]. In other words, the Fourier Transformation allows the subject's raw vocal data to be converted into a series of frequencies. The frequency output is still organized by time, with the first time slice pertaining to the beginning of the vocal sample and the last time slice pertaining to the ending of the vocal sample.

In the experiment, we used the frequency data to determine positive and negative emotions. In theory, doing so would enable us to eliminate three of the six basic emotions. For example, if the frequency data indicated a positive emotion then emotions such as sadness and disgust could be eliminated. In "Prosody and Speaker State: Paralinguistics, Pragmatics, and Proficiency" Dr. Jackson Liscombe performed tests with speaker frequency and its relation to the emotion portrayed [10]. It was determined that frequency analysis is useful in finding differences in human emotion. I took multiple positive and negative emotional vocal samples from my subject and began the process of determining the subject's total maximum and minimum frequencies. Next, I

began to analyze samples from the same scripted emotion. This process was important because it allowed us to find, for example, where the subject's maximum and minimum frequencies were for a specific emotion. This process was repeated for both positive and negative emotions and provided a frequency range for the subject. After calculating the ranges separately, an overall range was calculated and a midpoint was established as the cutoff point. The frequency cutoff point served as the separation between positive and negative emotions. Thus, when the subject's voice was analyzed by the Fourier Transformation the resulting frequency fell into either a positive category or a negative category. Figure 1 shows the basic methodology for categorizing the emotion of the speaker subject.

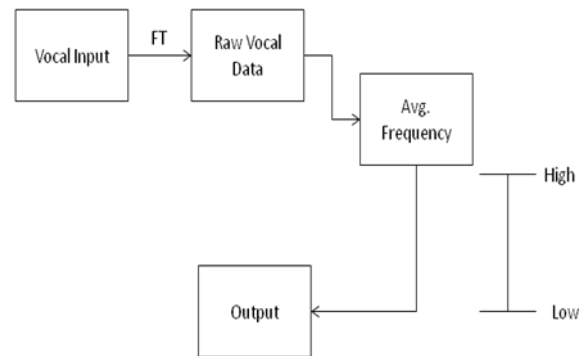


Figure 1 Basic methodology for categorizing the emotion of the speaker

Accurate processing of the subject's vocal frequency is paramount in determining their emotion. If the vocal input is clipped then the average frequency value will not be accurate, thus resulting in a false frequency value. In addition, if frequency processing begins before or after the subject speaks then the null frequency values will be averaged in with actual frequency values, also producing inaccurate data. In the Fourier Transformation, we had to be aware when the speaker began to speak and when the speaker stopped. In addition, the null values within the speaker's sentence had to be preserved to provide a non-skewed average frequency.

In order to achieve an accurate frequency output I perform two searches: one from the beginning of the frequency list and another from the end of the frequency list. The frequency calculation begins when a frequency value greater than zero is added to the list. Next, the list is searched from the end and continues to backtrack until it reaches a non-zero value. Thus, the entire frequency list is preserved and the preceding and ending zeros are not part of the resultant list.

4 Avatar Animation

In this research, we build an avatar model with six basic emotional targets from which all other emotions can be derived from. The six basic emotions known as surprise, happiness, anger, fear, disgust, and sadness [11], see figure 2 for the six basic facial expressions targets of the avatar.

Based on the emotion status detected from the speech signal, the avatar was animated by adding the animation tags in the phoneme transcription sequence. See Figure 3 for an example of the animation sequence. When the human speaker is saying "Jessica was so sad that she started to cry" with a very sad voice, the emotion detection result suggested that the speaker is having a negative emotion as vocalizing this sentence. Therefore, a sad tag will be added in the phoneme transcription sequence so the avatar can show the "sadness" target in the animation sequence.



Figure 2 Basic emotions of the animated avatar: surprise, happiness, anger, fear, disgust, and sadness

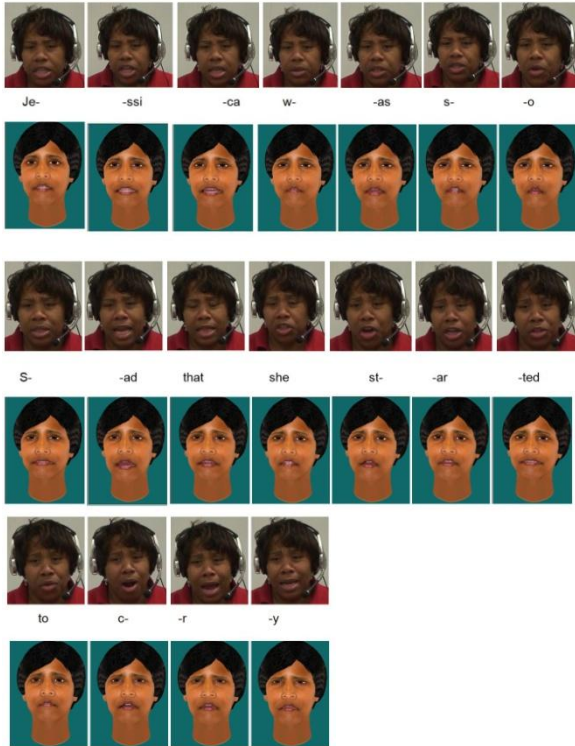


Figure 3 The original speaker and the avatar are saying the same utterance "Jessica was so sad that she started to cry"

5 Conclusions

In this paper, we presented a method of detecting the positive and negative emotions from the recorded speech signal by using the frequency feature of the speech signal, the emotion information will further be used to drive the facial expressions animation of an animated avatar. Experimental result shows that the animation sequences produced by the animated avatar are very similar to the original videos of the real human speaker. Our Future research work will focus on detecting all the six basic emotions and mixed emotions from the continuous speech signal and generate more expressive animated avatars.

6 References

- [1] JURAFSKY, D., MARTIN, J.H., *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* (2nd Edition.) Prentice-Hall, 2009
- [2] TRAUNMULLER, H., and ERIKSSON, A.. 1995. The frequency range of the voice fundamental in the speech of male and female adults, manuscript. http://www.ling.su.se/staff/hartmut/f0_m&f.pdf
- [3] MOZZICONACCI, S. J. L., and HERMES, D. J. (2000). "Expression of emotion and attitude through temporal speech variations," *Proceedings ICSLP 2000*, Beijing, China, Vol. 2, 373-378.
- [4] VERVERIDIS D. and KOTROPOULOS C., *Emotional speech recognition: Resources, features, methods, and applications*, Elsevier Speech Communication, vol. 48, issue 9, pp. 1162-1181, Sep. 2006
- [5] WANG, D. NARAYANAN, S., *An Acoustic Measure For Word Prominence In Spontaneous Speech.* *IEEE Transactions on Audio, Speech, and Language Processing*, (15)2: 690-701, 2007
- [6] TAYLOR P., *Analysis and synthesis of intonation using the tilt model.* *Journal of the Acoustical Society of America*, 107(3):1697-1714, 2000.
- [7] SENDLMEIER, K .M., WALTER F., *Acoustical Analysis of Spectral and Temporal Changes In Emotional Speech.*

Proceedings of the ISCA ITRW on Speech and Emotion, Newcastle, 5-7 September, 2000.

[8] A *fundamental frequency* analysis of harsh vocal quality. Ph.D. dissertation, *Stanford University*, 1964.

[9] HOMAK J.P., The Basics of NMR, on line textbooks; <http://www.cis.rit.edu/htbooks/mri/>

[10] LISCOMBE, J. J., Prosody and Speaker State: Paralinguistics, Pragmatics, and Proficiency. Ph.D. Thesis, Columbia University.2007.

[11] Ekman, P., & Rosenberg, E. L. , What the face reveals: Basic and applied studies of spontaneous expression using the Facial Action Coding System (FACS), 2nd Edition, New York: Oxford University Press., 2005.

A Novel Heuristic Algorithm for Solving Euclidean Steiner Tree Problem

Ali Nourollah^{1,2}, Elham Pashaei¹

¹ Department of Electrical, Computer and IT Engineering, Qazvin Islamic Azad University, Qazvin, Iran

² Department of Electrical and Computer Engineering of Shahid Rajaei Teacher Training University, Tehran, Iran

Abstract - *The Steiner problem leads to solutions in several scientific and business applications. Computer networks routing and electronic integrated circuits are few examples of it. Assuming some points in the Euclidean plane, we can construct a minimum spanning tree connecting these (terminal) nodes. It is possible to add some extra points (Steiner Points) in order to decrease the length of this tree, which would in turn lead to Euclidean Steiner Minimal Tree (ESMT). The problem is NP-hard, so polynomial-time heuristics are desired. We present a novel heuristic for the Euclidean Steiner tree problem. The algorithm utilizes the straight skeleton of simple polygon to generate candidate Steiner points, and a path heuristic to constructing Steiner minimum tree by using some of the candidate Steiner points in Euclidean plane. We present computational results on the Soukup test problems.*

Keywords: Euclidean Steiner Minimal Tree; straight skeleton of simple polygon; path heuristic

1 Introduction

The Steiner problem is applied in several scientific and business applications, such as computer networks routing, electronic integrated circuits, petroleum shaft and post networks. The computational features of this problem make it an important research subject in computational geometry. Having some points in the Euclidean plane, the shortest path for connecting these points leads to a tree which is called Euclidean Steiner Minimal Tree (ESMT). The Euclidean Steiner minimal tree problem is considered as a NP-hard problem [1]. ESMT may contain some nodes that are not in the set of the given nodes that are known as Steiner nodes and we call the given nodes as terminals. ESMT in a plane without any obstacles consists of unions of ESMTs with few terminals. It is unusual to encounter ESMTs with 6 or more terminals in literature (which is a serious constraint and our approach tends to be free of it) [2] and ESMTs connecting subsets of up to 4 terminals have proved to yield good quality solutions for the obstacle-free cases [3, 4]. Optimal solution algorithms for the Euclidean Steiner problem have been presented by Boyce and Seery [5], Cockayne and Schiller [6], Winter [7] and Cockayne and Hewgill [8]. These algorithms work by examining topologies (a topology being a set of

vertices and their associated edges) corresponding to full Steiner trees. Chang [9] presented an early heuristic algorithm based upon inserting vertices into the Minimum Spanning Tree (MST) in order to reduce the cost of the tree. This is a natural approach and has been used in many algorithms, for example Korhonen [10] and Smith and Leibman [11]. Smith, Lee and Leibman [4] presented an algorithm based upon Voronoi diagrams and Delaunay triangulations. Lundy [12] presented an algorithm based upon simulated annealing. Beasley in [3] have proposed a heuristic based upon finding optimal Steiner solutions for connected subgraphs of the minimum spanning tree of the entire vertex set. In this paper, we propose a new algorithm based upon straight skeleton of simple polygon and a path heuristic such as the Kruskal-based heuristic to solve the problem in a Euclidean plan for any number of terminals. Finally, we compare our results in Euclidean plane with data and results presented in [13]. This paper is organized as follows: Section 2 is dedicated to some basic definitions. We explain our algorithm in section 3. Computational results are presented in section 4. Section 5 draws a conclusion and makes suggestion for future works.

2 Basic definitions

2.1 Euclidean minimum steiner tree

A Steiner tree (ST) is a resulting tree from the addition of some special vertices strategically located and recognized as Steiner points, intended to shorten the total length of a minimum spanning tree (MST) even more. Clearly a Steiner minimum tree (SMT) has a general minimum length, corresponding to the shortest Steiner tree for a given set of Euclidean points. It is obvious that we can construct many different trees with this specification but the Steiner tree must meet the requirements of Theorem 1.

Theorem 1: (1) None of two edges of a Steiner tree can meet at an angle less than 120 degree (we call this as angle condition). (2) A Steiner tree has no crossing edges. (3) Each Steiner point of a Steiner tree has exactly degree of three.

Proof: see [14, 15].

For different set of points (terminals), the exact number of Steiner points required and their location is not known in advance. Theorem 2 establishes an upper limit for the number

of Steiner points required which leads to a Full Steiner Tree (FST) [16]. Assuming set of nodes Z , FST F spanning Z_k (which is a subset of k terminals in Z) has exactly $k - 2$ Steiner points, each one connected to three edges making angles of 120 degree.

Theorem 2: A Steiner tree for n points contains maximum of $n - 2$ Steiner points.

Proof: see [14, 15].

2.2 Straight skeleton

In geometry, a straight skeleton is a method of representing a polygon by a topological skeleton. It is similar in some ways to the medial axis but differs in that the skeleton is composed of straight line segments, while the medial axis of a polygon may involve parabolic curves. Straight skeletons were first defined for simple polygons by Aichholzer et al. and generalized to planar straight line graphs by Aichholzer and Aurenhammer [17, 18]. The straight skeleton of a polygon is defined by a continuous shrinking process in which the edges of the polygon are moved inwards parallel to themselves at a constant speed. As the edges move in this way, the vertices where pairs of edges meet also move, at speeds that depend on the angle of the vertex. If one of these moving vertices collides with a nonadjacent edge, the polygon is split in two by the collision, and the process continues in each part. The straight skeleton is the set of curves traced out by the moving vertices in this process [17, 18]. Fig. 1 of the illustration shows the straight skeleton of a polygon.

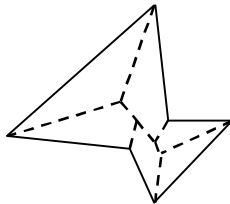


Figure 1: Straight skeleton of simple polygon

2.3 Generating simple polygons

In geometry, a simple polygon is a closed polygonal chain of line segments in the plane which do not have points in common other than the common vertices of pairs of consecutive segments [19]. There exist a vast number of algorithms to generate the simple polygons. Some of them is presented here:

2.3.1 Convexbottom polygon

It's a quite simple algorithm that constructs polygon which it evince a half-convex shape with order $O(n \log n)$. First, the two extreme points regarding x coordinate are determined, that is the two points with lowest respective highest x value. These points are connected by an imaginary line, so the point set is divided into an upper and a lower half. Now the convex hull of the lower half is computed, and the

imaginary line is removed again. The polygon's convex bottom has already been computed now. All remaining points that do not lie on the hull are now sorted from left to right by x coordinate and connected in this order. Finally, the two extreme points are connected to the convex bottom [20]. Fig. 2a and Fig. 2b illustrate process of algorithm.

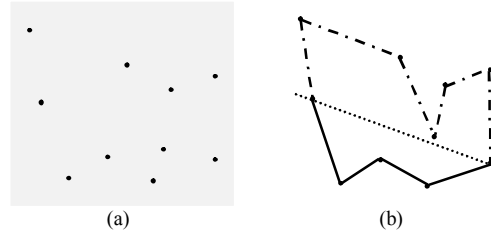


Figure 2: Convexbottom polygon

2.3.2 Star-Shape polygon

Algorithm starts with identifying a single point with lowest x value. A collection of n pairs of the form $P_i = (i, \alpha_i)$ is generated in which $0 \leq \alpha_i < 2\pi$ represents an angle between lowest x value point and other points in plane. The angles α_i are then sorted (in clockwise or counterclockwise order) and a polygon is constructed by connecting the points in a clockwise or counterclockwise fashion. Fig. 3a and Fig. 3b show an instance of star-shape polygon.

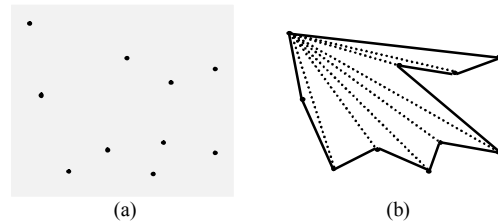


Figure 3: Star-Shape polygon

2.3.3 Mohammadi polygon

At the first step of algorithm, the convex hull of input points, are calculated. Then, an inner polygon with remain points inside the convex hull according on one of the mentioned algorithms are constructed. At the next step, we candidate two points of nearest edge of inner polygon from convex hull's edge and remove those two edges. Finally, edges between points of two removed edge of inner polygon and convex hull's edge are drawn [21]. Fig. 3a and Fig. 3b illustrate process of mohammadi algorithm.

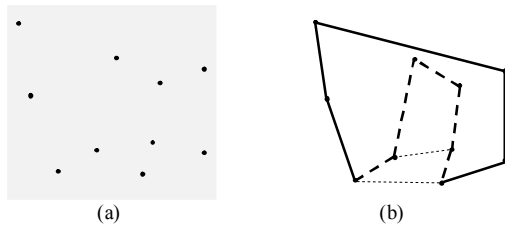


Figure 4: Mohammadi polygon

2.4 Path heuristics

Several heuristics for the Steiner tree problem on graphs can be characterized as path heuristics such as Shortest Path Heuristic (SPH) [22], Kruskal-Based Heuristic (KBH) [23] and Y-Heuristic (YH) [24]. Starting from an arbitrarily chosen terminal (or some other subnetwork of graph), the tree is gradually grown until it spans all terminals. The expansion is typically based on the addition of (shortest) paths between vertices already in the tree and terminals not yet in the tree.

Kruskal-Based Heuristic: The shortest paths heuristic is closely related to Prim's algorithm for minimum spanning trees [22]. Wang [23] suggested a heuristic which is closely related to Kruskal's algorithm for minimum spanning trees. The Kruskal-based heuristic (KBH) is as follows.

- Step1: Begin with a forest T_{KBH} consisting of all isolated terminals.
- Step2: If T_{KBH} is connected, then Stop.
- Step3: Find two trees in T_{KBH} closed to each other (ties are broken arbitrarily). Add to T_{KBH} a shortest path connecting those two trees. Go to Stop 2.

The worst case time complexity of this heuristic is $O(nv^2)$ that n denotes number of terminals and v denotes number of all nodes.

3 Skeleton based euclidean steiner tree

The proposed heuristic algorithm in this paper consists of three major steps. In first step, after assuming some terminals in the plane (Fig. 5a), we construct a simple polygon around the terminals. In our paper, we generate a Star-Shape polygon (Fig. 5b), a Convexbottom polygon (Fig. 5c) and a Mohammadi polygon (Fig. 5d). In the second step of algorithm, straight skeleton of polygons (Fig. 5b-5d) are constructed. With considering points of straight skeleton's intersecting edges solely, we have some candidate of Steiner points for given terminals in the plane. For going to next step, we overlap three produced simple polygons, then remove straight skeletons and simple polygons boundary. As result of preview steps, candidate Steiner points and terminals are remained (Fig. 5e). At the end, we generate a complete Graph that each node in it has an edge to all other nodes (composed all the terminals and candidate Steiner points), certainly. It is obvious that costs of the complete graph's edges are Euclidean distance between two nodes (Fig. 5f). We give this complete graph as an input to KBH algorithm that its output

is a Euclidean Steiner tree in plane with given terminals (Fig. 5g). Fig. 5 illustrates steps of our algorithm, respectively and Fig. 6 shows our suggested algorithm.

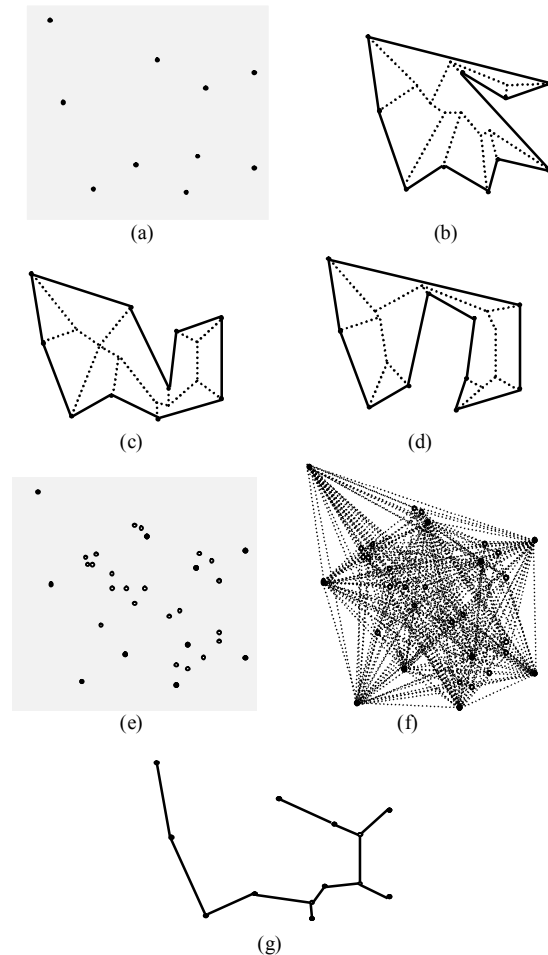


Figure 5: Skeleton based Euclidean Steiner tree in the plane

Algorithm 1: Skeleton Based algorithm for ESTP

- | | |
|----------------|---|
| Step 1: | Read n . //number of terminals. |
| Step 2: | Read coordinates of points in the main polygon. |
| Step 3: | Create three mentioned simple polygon around n points. |
| Step 4: | Draw straight skeleton of the polygons. //find candidates of Steiner points |
| Step 5: | Overlap these three produced simple polygons on the terminals coordinators. |
| Step6: | Construct complete graph from each points to others (terminals and candidate Steiner points). |
| Step 7: | Apply KBH algorithm to this complete graph. |
-

Figure 6: Suggested algorithm for ESTP in plane

4 Computational results

We have implemented our algorithms in C++ programming language. Our tests were built in Euclidean space on some of Soukup examples [13]. In Table 1 we compare some of our results with optimal results. As illustrated in Table I, our algorithms have achieved good results in all cases. Figure 7 shows the resulting tree of our straight skeleton based algorithm for the Ex.18 of Soukup exercise series. Figure 8 illustrate the comparison of our method with results of J. Beasley's heuristic on Soukup Examples [3], as one of the best others works were done in this field lucidly.

Table 1: Our algorithm in compare of Soukup Examples

Number of Soukup Example	Number of points	Optimal result	Our Skeleton based ESMT
EX. 1	5	166.44	170.90
EX. 2	6	150.05	155.60
EX. 2A	7	207.77	210.33
EX. 2B	8	213.88	215.40
EX. 2C	6	204.41	206.32
EX. 2D	12	218.42	225.78
EX. 2E	12	220.53	225.63
EX. 2F	12	217.78	220.30
EX. 2G	7	155.94	161.02
EX. 3	6	159.88	168.05
EX. 4	6	127.41	130.27
EX. 5	9	164.83	169.00
EX. 6	9	127.34	130.85
EX. 7	12	220.49	228.59
EX. 8	14	123.04	129.70
EX. 9	3	116.68	116.68
EX. 10	10	164.28	170.63
EX. 11	62	381.76	390.80
EX. 12	14	170.19	185.16
EX. 13	3	103.96	103.96
EX. 14	5	181.82	183.00
EX. 15	4	50.33	50.80
EX. 15A	4	51.30	53.19
EX. 16	4	25.28	27.70
EX. 16A	4	89.19	90.19
EX. 17	3	146.60	146.60
EX. 18	12	101.98	107.20
EX. 19A	14	233.22	237.34
EX. 19B	19	281.42	290.89
EX. 20	18	222.58	225.56
EX. 21	19	213.81	215.93

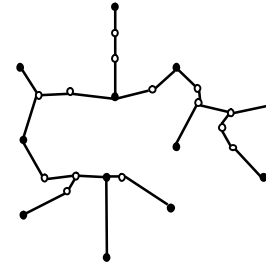


Figure 7: EX.18 of Soukup with Skeleton based algorithm

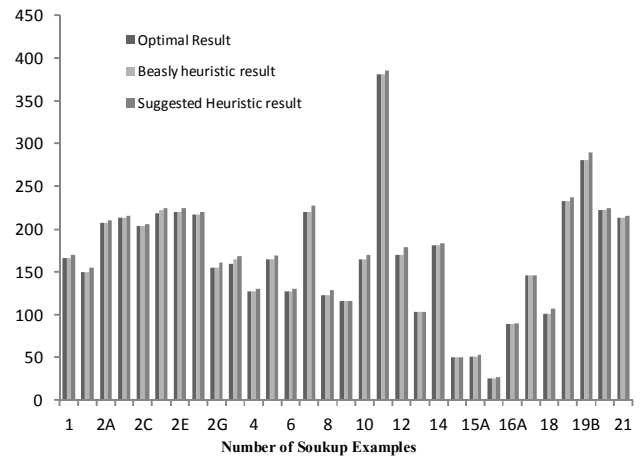


Figure 8: Soukup Examples and Beasley heuristic result in Compare of Skeleton based algorithm

Step 3 has $O(n \log n)$ time complexity. Order of Step 4 is $O(nm + n \log n)$, where n denotes the total number of polygon vertices and m denote the number of reflex vertices. Time complexity of Step 5 and Step 6 are constant, on the other hand, step 6 has $O(nv^2)$ time complexity. Totally, time complexity of our suggested heuristic is $O(nv^2)$, where n denote the number of terminals and v denote number of all points (terminals and Steiner point). The advantages of our heuristic are that it not only solves the Euclidean Steiner problem rapidly but also it has acceptable results. Additionally, it has absolutely novel idea in behind of its own.

5 Conclusions

In this paper we presented a novel algorithm which is capable of solving the Steiner problem in Euclidean plane for any number of given terminals. Our algorithm was based on straight skeleton approach in which we construct three simple polygons and calculate straight skeleton of them in order to find candidate Steiner points. Then apply KBH heuristic to find final Steiner tree. Computational results and time complexity for proposed algorithm are presented. A number of interesting issues remain open: Preprocessing required for optimization of results and run time, finding different algorithm of generating simple polygon to achieve good candidate Steiner points are a few example.

6 References

- [1] M.R. Garey, R.L. Graham, D.S. Johnson, "The complexity of computing Steiner minimal trees", *SIAM J. Appl. Math.*, 1977.
- [2] D.M. Warme, P. Winter, M. Zachariassen, "Exact algorithms for plane Steiner tree problems: a computational study", in: D.-Z. Du, J.M. Smith, J.H. Rubinstein (Eds.), *Advances in Steiner Trees*, Kluwer, Dordrecht, 2000, pp. 81–116.
- [3] J.E. Beasley, "A heuristic for the Euclidean and rectilinear Steiner tree problem", *European J. Oper. Res.* 58, 1992, 284–292.
- [4] J. MacGregor Smith, D.T. Lee, J.S. Liebman, "An $O(n \log n)$ heuristic for the Steiner minimal tree problem on the Euclidean metric", *Networks* 11, 1981, 23–29.
- [5] Boyce, W.M., and Seery, J.B., "STEINER72, an improved version of Cockayne and Schiller's program", *Computer Science Technical Report Number 35*, Bell Laboratories, Murry Hill, NJ, 1972.
- [6] Cockayne, E.J., and Schiller, D.G., "Computation of Steiner minimal trees", in: D.J.A Welsh and D. R. Wood (eds.), *Combinatorics, Institute of Mathematics and its Application*, Southend, 1972, pp. 53–71.
- [7] Winter, P., "An algorithm for the Steiner problem in the Euclidean plane", *Networks* 15, 1985, pp. 323–345.
- [8] Cockayne, E.J., and Hewgill, D.E., "Exact computation of Steiner minimal trees in the plane", *Information Processing Letters* 22, 1986, pp. 151–156.
- [9] Chang, S.K., "The generation of minimal trees with a Steiner topology", *Journal of the ACM* 19, 1972, pp.699–711.
- [10] Korhonen, P., "An algorithm for transforming a spanning tree into a Steiner tree", *Survey of Mathematical Programming (Proceedings of the 9th International Mathematical Programming Symposium)* 2, 1980, pp.349–357.
- [11] Smith, J.M., and Leibman, J.S., "Steiner trees, Steiner circuits and the interference problem in building design", *Engineering Optimization* 4, 1980, pp. 15–36.
- [12] Lundy, M., "Applications of the annealing algorithm to combinatorial problems in statistics", *Biometrika* 72, 1985, pp.191–198.
- [13] J. Soukup, W.F. Chow, "set of test problems for the minimum length connection networks", *ACM SIGMAP Bulletin*, 1973.
- [14] D. Cieslik, "Steiner Minimal Trees". Kluwer Academic Publishers, 1998.
- [15] F. Hwang, D. Richards, P. Winter, "The Steiner Tree Problem", Elsevier Science Publishers B.V, 1992.
- [16] M.Zachariassen, "Algorithms for Plan Steiner Tree Problem", PhD.Thesis, Department of Computer Science, University of Copenhagen, 1998.
- [17] P.Felker and S.Obrzalek, "Straight Skeleton Implementation", *Proceedings of Spring Conference on Computer Graphics*, 1998, pp. 210–218.
- [18] O. Aichholzer and D. Alberts and F. Aurenhammer and B. Gaertner, "A novel type of skeleton for polygons", *Journal of Universal Computer Science*, 1995, pp.752-761.
- [19] C. Zhu, G. Sundaram, J. Soneyink, and J. S. B. Mitchell, "Generating Random Polygon with Given Vertices", *Computer Geometry Theory and Application*, Vol 6, Issues 5, 1996, pp. 277–290.
- [20] Auer, T., Held, M. "Heuristics for the Generation of Random Polygons", *The 8th Canadian on Computational Geometry (CCCG'96)*, 1996, pp.38–41.
- [21] A. Nourollah, L. Mohammadi, "Three Heuristic Algorithms for Generation of Random Polygons by a Given of vertices", *The 6th Iranian Machine Vision and Image Processing Conference*, October, 2010, pp.588–592.
- [22] H. Takahashi and A.Matsuyama, "An approximate solution for the Steiner problem in Graphs", *Mach. Jap.* 24, 1980, pp. 573–577.
- [23] S.M. Wang, "A multiple source algorithm for suboptimum Steiner trees in graphs", in: H. Noltemeier (ed.) *Proc. Int. Workshop on Graph-Theoretic Concepts in Computer Science*, Wurzburg, 1985, pp.387–396.
- [24] N. P. Chen, "New algorithm for Steiner tree on graphs", *IEEE Symp. on Circuits and Systems*, 1983, pp.1217–1219.

MOBILE AR GPS-BASED FOR NAVIGATION PURPOSE

Behrang Parhizkar, Ceaser Bakka, Abdul Akbar Pandor, Ng Ah Ngan Mike Christian, Md. Abdullah - Al - Jubair, Arash Habibi Lashkari
 Faculty of Information and Communication Technology
 Limkokwing University of Creative Technology, Malaysia, Cyberjaya

ABSTRACT: *The ability of a mobile phone to support navigation systems and powerful operating systems creates an opportunity to explore the newly introduced technology – Augmented Reality – and its application with navigation systems. Augmented reality allows 3D Virtual Images to appear in a 3D real environment in real time and allow more interactivity for the users. In some ways Augmented reality navigation may be taken in with comparison to GPS navigation, but it concentrates more on enhancing reality than simply generating a map. Difference being seen in the way that the maps are created or generated, augmented Reality navigation displays a map through the use of the device's camera and GPS displays a 2D map. Instructions or commands for navigation are also of immense importance between the two, augmented navigation has easier commands to follow and GPS commands are just instructions, that is, augmented navigation will show the user where exactly to make a turn adding a safety feature of not letting the user get their eyes off the road as everything is in real time, so the user sees real driving conditions through their device.*

Keyword: Mobile Technology, GPS, Augmented Reality, Navigation System

1. Introduction

Throughout the years technology has advanced drastically with the sole purpose of bettering the lives of people that are highly involved in the use of sophisticated technology. In this regard, augmented reality has taken the levels of technology to another whole new level. Augmented reality has transformed the real world and virtual world creating a real life experience (Omar Choudary, Vincent Charvillat, Romulus Grigoras, Pierre Gurdjons) which is both in real-time, 3Dimensional and highly interactive. The Figure 1 shows the concept behind Augmented Reality.

But in order to fully understand the concept of augmented reality we have to explore its origins, the way it was initially designed, the graphics, and how it blended into the industry. Augmented Reality has a wide range of

applications that have been implemented. The implementation of these applications has been in various working environments adding simplicity within that field. These environments range from aviation, medicine, education to advertising, navigation and manufacturing.

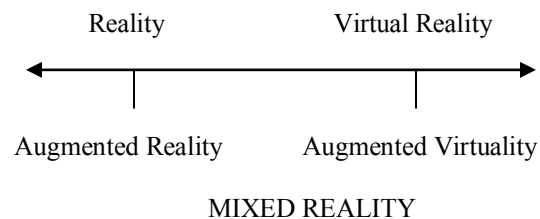


Figure 1: Concept of Augmented Reality

Augmented reality's main objective is to insert information and connotation to an actual object or situate (Educause Learning, 2005). Fortunately, the contrasting thing between virtual reality and augmented reality is that, augmented reality, suppresses the creation of a simulation of reality (Daniel Wagner and Dieter Schmalstieg, 2009). However, it uses a real object or space as the base and implements the use of technologies that add additional information that will increase the users understanding of the object (Educause Learning, 2005).

An augmented reality system is a system which creates a view of a real scene by incorporating digital virtual objects with full three-dimensional properties, into a scene (James Vallino, 1998). As the user of the system moves changes positions in the real scene the virtual objects somewhat appear just as they would as something that exists in a real world. These virtual objects should be able to interact with the user and real objects present in the scene in a normal, natural style. This way, we can see that augmented reality increases the user's performance and perception both in the real world (James Vallino, 1998).

In this paper, the authors discuss how a mobile Augmented

Reality navigation system is modeled to compensate for the needs of students within widely expanded universities. The users of the application will be able to Search for various places within a university, provided they have to idea of how to reach that place. The application will also allow its users to make transitions between theory related navigation, map navigation or the augmented navigation.

2. Problem Statement

Universities around the world are expanding structurally at a rapid rate especially when the number of students gets greater and greater with every enrollment period. As a result of being tremendous in size there is a great chance that some students would not be able to find their way around the campus with parents and newly enrolled students being the ones greatly affected. Data gathered from a research conducted in table 1 below madewith reference to Limkokwing University located in Malaysia, shows that from a pole of 100 students (50 being newly enrolled students and 50 old students of which 25 are males and 25 are females);

Gender	Find it hard to find a place around the University	
	New Students	Old Students
Males	22	15
Females	20	15

Table 1: Summary of Newly enrolled and old students that have issues related to getting around the University

The table above shows that forty two (42) newly enrolled students of the fifty (50) found it very hard to find a place and thirty (30) current students of the fifty (50) found it hard as well to navigate around the campus. This shows that student's do really have an issue of getting around the University. So it concludes that every student within the University if affected.

It is in this regard that the campus navigation system will be designed using a new technology referred to as Augmented Reality (AR). Augmented Reality will basically create a scenario whereby it will bring the virtual world into the real world. This will be an innovative way to make GPS navigation stand out –

implementing Augmented Reality into it. It will work by picking up video from the camera of an Android Smartphone or tablet and enhancing it using some Augmented Reality. This navigation system will be usable within campus with pre-determined destinations embedded into it.

Students will select a specific target location such as the clinic, library, and coffee shop and also restaurants and other recreational facilities situated within a particular area of the university. The system then computes the shortest path in a known network of possible routes. The information will be displayed as a series of waypoints that are visualized icons standing in the environment. These icons will be connected by arrows to show the direction the students should move between the waypoints. Simple directional information will also be displayed if the student is not able to distinguish the next waypoint if they are looking into the wrong direction.

3. Related Work

3.1. Mobile Augmented Reality

A vast area of technologies can be used together with augmented reality. In this year, plentiful augmented reality projects use handheld devices, especially mobile phones, in which they view data into the user's field of vision in conjunction with the object that the user is observing. These portable devices can use for example GPS data to provide users with context—including visual, audio, or text-based data—about real objects or places. Augmented reality is not an escort of text or multimedia files but a technology premeditated to view real objects or places and offer user's with appropriate information at the right time (Wagner, 2003).

One of the first ever created AR application embedded into a handheld device (Mobile phone) and dispatched to the public was the “invincible train” game (MiroslavAndel, Alexander Petrovski, 2006). The user would control the virtual scene, in this case the trains and be able to manipulate them to what they saw fit, unfortunately at this time the only form of interaction was via the use of a stylus pen.



Figure 2: Invincible train game

Augmented reality therefore stretches beyond personal computer platforms, there are mobile applications that have been created and are being used in various fields (Omar Choudary, Vincent Charvillat, Romulus Grigoras, Pierre Gurdjons, 2009). Augmented Reality delivers a highly visual interactive area for its users within a mobile computing application (Gerhard Reitmayr, Dieter Schmastieg, 2009).

Together with mobile devices, augmented reality has transformed the real world and virtual world creating a real life experience (Omar Choudary, Vincent Charvillat, Romulus Grigoras, Pierre Gurdjons, 2009).

3.2. Applications Of Mobile Augmented Reality

Augmented reality has been put to use in a number of areas, including medical imaging, where doctors can access data about patients; aviation, where tools show pilots important data about the landscape they are viewing; training, in which technology provides students or technicians with necessary data about particular objects they are working with; and in museums, where artifacts can be tagged with information such as the artifact's historical context or where it was discovered.

Within the academic section, AR is used for learning, mobile learning or web based learning. Educators are beginning to provide students with deeper, more meaningful experiences by linking educational content with specific places and objects (Oblinger, D, 2004). In many disciplines, field trips are part of the course; by supplementing these explorations with mobile technologies and data-collection devices (including digital cameras), the lessons can be extended beyond the trip (Manfred Bogen, Jürgen Wind and Angele Giuliano, 2006).

In some cases, augmented reality technologies have been integrated into educational games. In MIT's Environmental Detectives, for example, students learn about environmental sciences and ecosystems by finding clues and solving a mystery on the MIT campus using PDAs fitted with GPS devices.

3.3. Augmented Reality And Navigation Systems

One of the best applications designed for a mobile system to date is by far the most effective for those individuals that travel frequently. The way we navigate around has evolved from written directions to atlases to GPS systems and now available on mobile phones. The introduction of navigation systems provides users with useful tools for navigation, communication and interaction.



Figure 3: Mobile Augmented Reality Navigation

Augmented Reality in navigation gives users the ability to move about in an area and be able to interact with that area in order to attain the guidance they require (Gerhard Reitmayr, Dieter Schmastieg, 2009). The users are required to carry their handheld device as they move through the augmented scene. In order to aid the user during movement there is a pin that is placed around and identifies the current location that the user is standing at and the final destination that they have selected. The plus side of this is that Augmented Reality Navigation can be used with GPRS systems depending on their location, indoors, GPS cannot be used as compared to when you are outside (Nicola Lenihan, 2004).

3.4. Platforms For Navigation

A mobile platform, also known as a mobile operating system is what operates an item, basically it makes it work, in this case a mobile phone. Within these mobile platforms users can perform various tasks which include communication, internet connection and reading documents. Due to the high demand of mobile phones new features are being made or created in order to add more functions and features thus increasing user productivity. (Allan Hammershøj, Antonio Sapuppo and Reza Tadayoni, 2009). Currently, there are three widely used mobile platforms namely; Symbian, UNIX and Windows. Between the three of these platforms UNIX happens to be the only open source platform and easy to manipulate, whereas Windows and Symbian are under Nokia and Microsoft companies.

Under the UNIX operating system there two widely used operating systems; these are the iPhone OS from Apple, Blackberry OS by Blackberry and Android OS from Google. Within each platform there are software development tools (SDKs), these are provided by the producers of that platform to help third party users create their own creative applications which they can later use for their own benefit together with other users on their mobile platforms. The source of the structure of each OS, developers manipulates

different parts of function of the OS through the use of Application Programming Interfaces (APIs).

3.5. Apple OS

Apple OS is developed by the Apple Company. This OS is available on two types of Apple devices; these are the iPhone and the iPod. Currently, the iPhone OS is considered the leading operating system that is widely used on mobile platforms within the market. The iPhone OS is a UNIX based open source platform referred to as Darwin, (EmilianoMiluzzo, James M. H. Oakley, Hong Lu, Nicholas D. Lane, Ronald A. Peterson, and Andrew T. Campbell, 2008).



Figure 4: iPhone OS framework

As shown above the platform framework has four different layers. When application development starts the developers program the application using the SDK that is released the Apple Company. All the programming is done using the Objective-C programming language. Upon completion of the application it is distributed via the App Store which is an open market for application managed by Apple itself.

Advantages of using the iPhone OS platform is that its libraries have open source applications and well organized APIs which help as a guidance for people that are planning on developing certain applications.

The only disadvantage of the platform is that the framework structure of the OS does not allow more than one application to share resources with the others.

3.6. Google Android OS

Emerging greatly from behind the scenes is the Google Android OS, developed and managed by Google, just like the Apple iPhone OS, Android is an open source mobile operating system that is now operational on a wide variety of mobile devices such as HTC Magic, Sony Ericsson Play and Matorola Quench. The Android operating system runs on a Java virtual environment based on a Linux system. The OS being a Linux based OS and combining itself with a Java environment allows its application developers to program using Java (Benjamin Speckmann, 2008). Upon

completion of these applications they are published or posted in the Android Market then made available to other Android platform users.

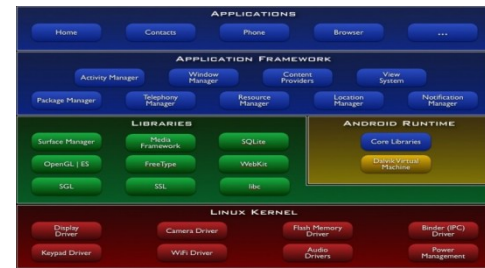


Figure 5: Google Android Framework

The advantage of using the Android OS is the fact that it is an open source and allows its users to connect and communicate using various hardware's. The architecture framework of Android also allows its developers to access the main components of the platform and transfer information from one application to another which unfortunately is not available with the Apple iPhone OS.

The disadvantage of using the Android OS is that it has a few developers that are knowledgeable with its programming language but statistics show that in the future Android will become the widely used mobile OS.

4. Research Objectives

This research aims at providing a simplified solution for students to know their way around the University and its areas through the use of Augmented Reality combined with a navigation system. Students should be able to walk around freely around the campus instead of walking around in circles and not getting any help. Some of the research objectives are;

- i. To develop a suitable development methodology model for the development of the Augmented Reality for the navigation system known as the Rapid Application Development Life Cycle (RAD).
- ii. To design and develop a suitable model of the Augmented Reality navigation System using the augmented reality techniques.
- iii. To implement the Augmented Reality Navigation System giving special attention to its user interface and usability.
- iv. To develop the prototype of the Augmented Reality Navigation System based on the Tangible User Interface.

- v. To test the strengths and weaknesses of the Augmented Reality Navigation System through Usability testing.

5. Methodology

In regards to the research, this paper focuses on the Rapid Application Development (RAD) concept. RAD is used due to the strong facts that it is a dynamic system which will allow review and changes to the system requirements whenever deemed necessary along the system development. However the RAD methodology will be used alongside the spiral model as this will allow any referral to any previous phase of the system development. Moreover, by using this methodology, less time is spent on system analysis and system design, and a prototype can be developed to test and evaluate the changes that need to be amended.

Below is an image that shows the concept of RAD.

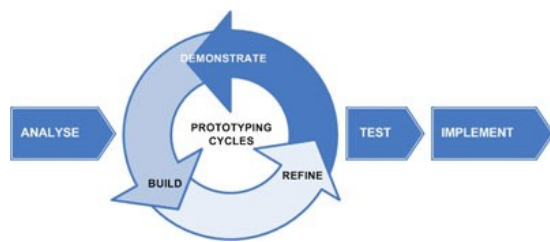


Figure 6: Rapid Application Development Model

The basis of RAD is that the chance of getting things the right time is very slim, but in order to at least give it a short the keys to RAD would be iteration, reasonable time schedules and teamwork.

The time schedules help in assigning the approximate time required to deliver a particular part of the system. You draw up a prioritized list of things that you will attempt to deliver. This list must contain some items that can be removed if the allocated time does not permit to create them. You are not allowed to exceed the time limit. The time schedule is to be measured in days, at most weeks, and never months. Focus is the key.

Given the general statement that you cannot get things right first time, you can have several attempts at using this method. The first iteration uses the first time schedule to get a prototype. The second and third iterations improve (not re-write) the results of the first iteration until the users and developers are happy that that part of the system is complete. You are not allowed to revisit any iteration.

A well-run RAD project will see a steady system grow around you. Everyone can see that the system is becoming more and more complete and can promote confidence in developers. RAD does tend to lead to well-designed systems that have low problem rates and meet the user's

true requirements. A badly-run RAD project will produce a large mess faster than other methodologies. You must choose the team well. As communication is an essential part of the success of this process, you should also ensure the users know the business and are not afraid to say when they don't know.

The life cycle model that has been chosen to help in developing the end product of the proposed system is the Spiral model. Spiral is actually a refine model of waterfall model. It recognizes the fact that most software is developed using an iterative or spiral approach rather than a linear approach. Also, since the proposed idea is dynamic in terms that the requirements tend to change along with time, spiral model is more appropriate as we can go back to previous phases. It allows us to repeat the different activities until we are satisfy with the end-result.

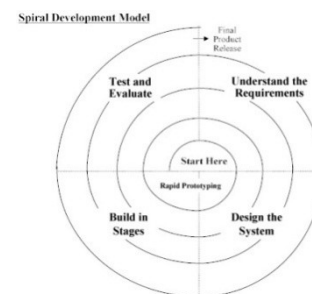


Figure 7: Spiral Model

6. Prototype

The prototype of the system has to accomplish its main goals and provide the user with his/her desired information. The system will use the camera for a mobile phone (provided it has an android platform) and augment various places of interest in and around the area you may be in.

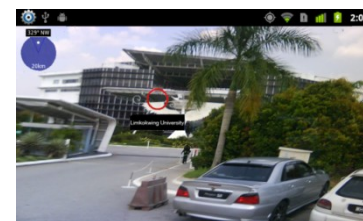


Figure 8: Screen shot of prototype

In the image shown above, the point of interest is marked with the red circular image. When the system is fully operational the object will have the name of the place that is being located together with the distance to be travelled in order to reach that place.



Figure 9: Prototype of GPS system

As also shown above, the system consists of a compass at the top left of the screen which shows the places located within a specific radius. Anything outside the radius will not be displayed.

7. Conclusion

Augmented reality has changed the way we may perceive life especially after its penetration into hand held mobile phones. Its changes over the years have led to the development of various applications on augmented reality that are now being used worldwide. In navigation systems it has proven to be of great importance especially if considered for tourism. Users now have a more interactive and fun way to navigate to their locations.

Since mobile phones are changing day by day we expect the augmented reality era to continue and evolve over the years. We expect user interfaces to be more graphical than they are today, less complex and more powerful so as to compensate for everyone's user-interaction needs and create a pure immersive environment.

When looking at navigation in perspective, augmented reality could be looked at for the next generation of augmented navigation systems. Voice implementation and image overlay for a travel path could be considered.

Acknowledgement

The special thank goes to our helpful advisor Dr. Arash Habibi Lashkari for his supervision and guidance in the progression of our dissertation and project.

8. References

[1] Allan Hammershøj, Antonio Sapuppo and Reza Tadayoni, "Mobile Platforms - An analysis of Mobile Operating Systems and Software development platforms", Aalborg University, Denmark, November 2009

[2] Benjamin Speckmann, "The Android mobile platform", Michigan University, April 2008

[3] Daniel Wagner and Dieter Schmalstieg, "History and future of Tracking for mobile phone Augmented Reality", Graz University of Technology. 2009

[4] Daniel Wagner and Dieter Schmalstieg "First steps towards Handheld Augmented Reality", White plains, NY, USA. 2003

[5] Educause Learning Initiative Advancing learning through IT innovation. "Things you should know about Augmented Reality". 2005

[6] Emiliano Miluzzo, James M. H. Oakley, Hong Lu, Nicholas D. Lane, Ronald A. Peterson, and Andrew T. Campbell, "Evaluating The Iphone as a mobile platform for people – Centric sensing applications", Dartmouth College, USA, November 2008

[7] Gerhard Reitmayr, Dieter Schmalstieg, "Global pose estimation using multi-sensor fusion for outdoor Augmented Reality", 2009

[8] James R. Vallino, "Interactive Augmented Reality", University of Rochester, New York, 1998.

[9] Manfred Bogen, Jürgen Wind and Angele Giuliano, "ARiSE – Augmented Reality in School Environments", Augustin, Germany, 2006

[10] Miroslav Andel and Alexander Petrovski "Interactive Collaborative Augmented Reality Using a distributed Scene Graph on Mobile Phones", Linkoping University, Sweden, June 2006.

[11] Nicola Lenihan. "A local Optimal User Position System for Indoor Wireless Devices". May 2004.

[12] Oblinger, D, "The Next Generation of Educational Engagement", Special Issue on the Educational Semantic Web, May 2004

[13] Omar Choudary, Vincent Charvillat, Romulus Grigoras, Pierre Gurdjans, "MARCH: Mobile Augmented Reality for Cultural Heritage", University of Toulouse, France. October 2009.

SESSION
VIRTUAL REALITY

Chair(s)

TBA

Designing a Low Cost Immersive Environment System Twenty Years after the First CAVE

Richard H. Fowler, Moises D. Carrillo, Raul A. Huerta, Wendy A. L. Fowler,

Department of Computer Science, University of Texas – Pan American, Edinburg, TX, USA

Abstract – *The earliest CAVE systems were a breakthrough technology made possible by the availability of graphics hardware capable of moderately high resolution and update speed. In the following twenty years the cost of computing and display equipment has decreased sufficiently that it is possible to configure a system with display capabilities equivalent to or better than early systems at orders of magnitude less cost, thus extending use to a much wider range of applications and users. In this paper we report our experience in design and implementation of an immersive environments laboratory used for research and education. The principle design considerations were that the system be relatively low cost, easy to configure and maintain, and fit in a small space, considerations common to the design of many similar systems. The paper also summarizes advances in cost/performance and ease of use since the first CAVE systems for each of the principle elements of an immersive environment system.*

Keywords: virtual reality, virtual environment, visualization, virtual reality tools, case study, design, low cost

1 Introduction

Immersive displays for virtual environments have been a hallmark of advanced graphics displays for twenty years. The first CAVE system [6, 7] was a breakthrough technology made possible by the availability of graphics hardware capable of moderately high resolution and update speed. These early CAVE systems typically employed a three sided space with walls and floor illuminated by stereoscopic projectors displaying graphic representations of physical environments. Essentially the same display paradigm is employed today in many systems, yet the quality of graphics and rate of display have dramatically changed in state of the art systems. The cost of computing and display equipment has decreased sufficiently that it is possible to configure a system with display capabilities equivalent to or better than early systems at orders of magnitude less cost, thus extending the range of immersive virtual environment displays to a much wider range of applications and users.

In addition to the reductions in computing and display cost, there has been significant decrease in effort required to program immersive virtual environments. The earliest CAVE

systems, as well as visualization systems generally, required significant programming efforts, typically unique to the system and application at hand. Much of the programming challenge was due to the need to use relatively low level graphics commands and individual computers for each display surface. In many current applications, programming approaches can rely on relatively high level tools for the programming needs of virtual environments. For display, a single computer can now readily control four or more graphics outputs, each with independent graphics processing unit. When individual computers are required, reliable software tools for configuration are available. Importantly for cost considerations, satisfactory tools for these programming tasks are publicly available.

In designing the system described in this paper our goals were to create an immersive display and virtual environment system with capabilities for both research and education. In its research capacity the system would support prototyping and development for work to be carried out in a CAVE located away from the university's central campus area, as well as stand-alone efforts. For educational purposes the system would be used in courses in graphics and visualization. The space available was quite constrained, yet the system needed to be of sufficient size to achieve high field of view with some measure of viewer mobility. Both hardware and software cost and maintenance were important factors. The system was to be designed to be easily reconfigured and extended.

The next section describes Related Work relevant to our design considerations, especially that for which cost was a significant factor. The section examines display techniques, computing and graphics systems, and software. The following section describes the System we have developed with a focus on its application in its targeted domains of education and research. Component costs are provided. Conclusions and References follow.

2 Related work

Predictably, the price/performance ratio for costs associated with immersive virtual environment systems has decreased by orders of magnitudes since the development of the first CAVE systems twenty years ago. However, for the several system elements decreases in costs have varied

widely. The most dramatic decrease in costs has been for computing and graphics equipment. Indeed, development of graphics processing units in the past decade, driven by a development focus on consumer demand for gaming and other visual elements, has provided some of the largest advances in computing hardware. For display elements, such as projectors and screens, decrease in cost has also been substantial, but these partially analog elements do not follow the cost reductions achieved for computing hardware, nor do the software elements required to integrate the system's components and create virtual environments.

The need to design and construct an immersive virtual environment under constraints of cost and space is not new, and many solutions have been offered. The remainder of this section considers this past work in the context of our system design and current product cost and availability, with sections focusing on displays, computing systems, and software.

2.1 Display

Room sized systems with images displayed on walls, floor, and ceiling are the most widely used technique for highly immersive virtual environment systems. Such systems can be characterized by the number of surfaces illuminated, from six, in the case of a fully immersive system with four walls, floor and ceiling used for display, to one, for a single wall display. Surely there are variants, such as the STARCAVE [10] with five sides, each with three panels, and spherical displays [11].

The essential design consideration for visual immersion is the portion of the viewer's complete field of view that is taken up at any time by displayed images [4]. As the user moves his or her view direction and/or position, the displayed image needs to be visible in the new area. Even a large single surface display will cover all of the field of view with static view direction and the user close to it. However, only with all six surfaces illuminated will full mobility with complete visual immersion be possible. Nonetheless, "successful" immersive displays have been constructed with less than all surfaces illuminated. "Success" depends on task at hand, nature of display content, and a host of other factors [23].

2.1.1 Single Surface Display

Single surface displays have long been proposed as alternatives to larger multi-surface displays, due to their lower cost, while still retaining elements of immersion suitable to the task at hand [36, 8]. Desktop size (~4' x 6') systems, such as the ImmersaDesk [8] and Responsive Workbench [17], providing stereoscopic viewing of rear projected images, were developed shortly after the first CAVE systems and have seen continuing application. Recent desktop systems have been developed that utilize commodity

stereoscopic television panel displays to significantly decrease cost [34].

Larger scale single surface stereoscopic displays were also developed during the early 1990's, such as the Infinity Wall [8] and PowerWall [36], again using rear projected images. Current approaches range from auditorium sized front projected stereoscopic displays to tiled displays made up of dozens of commodity monitors [19]. Research has shown that for appropriate applications single surface displays can provide a useful, low cost alternative to multiple surface displays [20, 23].

2.1.2 Two Surface Display

Two surface displays are capable of increased visual immersion for the viewer, but at increased cost and complexity. Typically, the image is as if projected in the corner of a room. With two surfaces the viewer has many of the advantages of display with three or more surfaces, but without complete mobility and scanning freedom. Nonetheless, with rear projection the user can stand and move about between illuminated surfaces, thus providing relatively high lateral field of view and mobility found to be useful in enhancing the viewer's experience of immersion.

Early examples of low cost two surface projection are the WEDGE [12] and VR2Go [27] systems, both using stereoscopic rear projection. Contemporary with these efforts is Jacobson et al.'s development of a low cost front projection system [14]. The system uses a display with vertically oriented projectors illuminating two screens using front projection. Vertical orientation of projectors results in the longer side of the projector's display being vertical, rather than horizontal. Arrangement of the two projectors allows the viewer to approach the corner to a point slightly within the screens on both sides without obstructing the illumination.

2.1.3 Three or More Surface Displays

A wide range of solutions has been offered to deal with the challenge of creating systems that provide three or four surface displays and achieve acceptable results with very limited cost. As noted, viewer mobility results in a more immersive experience for the user, and this is achieved most completely by rear projection. The increased sense of immersion is achieved with additional costs, including the translucent surface material for projection, together with its framing. Also, higher projector output is required, compared to front projection, due to loss through the translucent projection material. For space conservation, mirrors are used to fold the projected image.

A very low cost immersive system with images projected on three walls and the floor is described by Peternier and Cardin [22]. Rear projection onto canvas that allows some measure of light transmission is used for the walls. The wall

framework uses nylon line for corner tension, with image distortion due to line stretch corrected by software. Projection illumination is not folded by mirrors, saving cost, but resulting in an overall size some three to four times the length of a wall. To achieve active stereoscopic display, two projectors are used for each wall. Shutters for projector pairs control sequential left and right eye image display, and projector shutter timing is synchronized with viewer lcd glasses. Anaglyph display is also provided without shutters.

Recently, Juarez et al. [15] describe a similar approach to creating a room sized system using rear projection onto canvas, but without stereoscopic display. The use of short throw projector lenses allows overall size to be only about twice the size of the enclosed area. Pognadoresse et al. [24] describe an extremely low cost room sized system using tracing paper for projection. Cliburn [5] presents a user study using a very low cost three wall passive stereo rear projection system with on site painting of commodity acrylic display surfaces. Commodity mirrors are used to create a single fold of projections to reduce system size. Stereoscopic rear projection systems have been developed with space for only a single person [31, 39] in efforts to minimize space, as well as cost.

Front projection systems significantly constrain viewer mobility compared to rear projection due to possible occlusion of projected image, but offer significant reduction in cost and complexity. Additionally, front projection has the important advantage that space requirements are significantly reduced. The LAIR [9] system demonstrates a clever solution to four wall display using front projection. Two projectors are used for each wall, so that the projected image for each covers half of the wall. The viewer can then stand between the two projectors so that the projection is not occluded. With viewer position half the distance between the walls in front and behind, this arrangement of two projectors per wall can be repeated for the other walls to create a four wall display with an area in the center of the room in which a viewer does not block any of the front projections.

2.2 Computing Systems

In the decade following CAVE introduction, the most dramatic advances in cost were in the computing equipment required to achieve the frame rates and scene complexity necessary for interactive display of virtual environments. Relatively low cost computers and graphics systems replaced the early special purpose equipment with several orders of magnitude decrease in cost. In the decade following that, relatively low cost graphics processing units (gpus) with hundreds of processors have become available to support double buffered stereoscopic viewing of complex scenes at appropriately high refresh rates. Advances in computer architecture for moderately priced computers allow multiple gpus in a single system, thus for many applications

eliminating the need to network multiple systems and the associated cost of display synchronization.

The ability to achieve acceptable performance through the use of commodity computing and graphics hardware marked early efforts at low cost solutions [e.g., 1, 13]. A decade after the first immersive virtual environment systems were introduced in the CAVE, Pape et al. [20] and Lin et al. [18] provide 2002 graphics benchmark results comparing specialized graphics hardware, SGI Onyx, and low cost hardware of the day, detailing the orders of magnitude in improvement in price/performance ratio during that first decade. Today's commodity computers and moderately priced graphics hardware provide far greater performance than even specialized equipment of that era with yet more orders of magnitude improvement in price/performance.

To achieve acceptable graphics performance in low cost systems during the first decade of immersive virtual environment systems, it was necessary to utilize multiple computer systems with the associated complexities and cost of network communication and display synchronization. Early work focused on networking solutions as one component of low cost solutions [13, 18]. Much of the challenge of networked computer communication and configuration for low cost solutions was reduced by development of public domain software, e.g., ClusterJuggler [3].

Currently, it is possible to achieve acceptable computing and graphics performance far superior to early state-of-the-art systems at very reasonable cost. Indeed, for many applications satisfactory performance can be achieved using a single computer housing multiple graphics cards capable of stereoscopic display. A single computer system has the advantages of eliminating hardware costs associated with multiple computers and the complexity of networking the systems. Additionally, use of a single computer eliminates the need to synchronize display signals to projection devices, which often requires separate hardware components.

2.3 Software

In the first decade following CAVE introduction, software development changed from the major efforts of the first systems' completely unique applications, specific to hardware and display, to the widespread use of tools for graphics display and system configuration. The use of these early software libraries reduced much of the development effort, but nonetheless required application development in relatively low level languages, still requiring advanced programming skills and significant time tied to the integration of computing system and display. In the following decade and to date, such libraries are still widely used and necessary for many applications. However, the game industry has provided a set of tools that has crossed over to the development of immersive environments and can support

rapid low cost development of rich virtual environments with public domain tools, though integration of multiple displays still presents challenges with these tools.

CAVELib [21] was developed for use with Cruz-Neira et al.'s original CAVE system. Initially designed to run on specific hardware, it evolved to support general display facilities for multiple display and computer systems, as well as interaction devices. It was once among the most widely used systems by researchers and offered a reasonably robust and freely available solution for many immersive system software needs. Unfortunately for the wider research community, the system was commercialized, and now license costs present a barrier for low cost system development.

A number of other multiple display and computer integration software systems are freely available. VR Juggler [2] provides relatively high level multiple display management and device integration facilities. Its architecture allows the close coupling of additional native code in its use. It was extended to provide support for networked systems with Cluster Juggler [3]. Both are widely used and updated for new operating systems and languages [37]. OpenSG [26] provides another relatively low level software solution for multiple display immersive systems using scene graphs. The VRPN [38], Vrui [17] and FreeVR [33] libraries provide suites of interaction handling routines. Raffin and Soares [25] provide a review of multi computer and display systems for low cost immersive systems.

Within the past decade, the widespread availability of computer games and freely available programming tools to support them has led to their use in the development of immersive systems. These tools provide efficient graphics software and typically provide authoring tools for interaction handling, as well as display and modeling. An early example is Jacobson and Lewis' [14] use of the Unreal Tournament game software to develop a system supporting multiple displays. Their approach utilizes a multiplayer version of the game with separate computers executing different instances of the game. All instances use a single viewpoint, but with view direction set to provide views appropriate for the each of the displays. Views and positions on all computers are updated based on the viewpoint of the single user. Game engines for Half-Life [30] and CryEngine2 [15, 32] have also been used for virtual environment creation and display.

3 The System

The principle design considerations for the system we have developed were that the system be relatively low cost, easy to configure and maintain, and fit in a small space, considerations common to the design of many similar systems. The system's design required that it serve both research and educational purposes. In its research role the system would serve as a satellite development system for a CAVE located some distance away and maintain as much

software and hardware compatibility as possible. It would also be used to conduct experiments suited to its capabilities. The system's educational role would be to support courses in visualization and computer graphics.

Low cost for both hardware and software was a primary consideration. Additionally, the available space was only 12' wide. Nonetheless, the system needed to provide a high field of view for users. We also determined that some measure of viewer mobility, however limited by space, was required. To accommodate multiple users the system had to be easily reconfigured. In anticipation of future expansion the system design had to be readily extendable.

In considering the type of display system the principle decision was whether to use rear or front projection. Though tiled display walls can be used for large viewing surfaces, projects we are to undertake require clear legibility of text, which is difficult to achieve in tiled displays due to the screens' bezels. Rear projection has clear viewer mobility advantages over front projection, with viewers able to move anywhere in the viewing space. This advantage comes at the cost of more complex and typically more expensive screen material and mounting fabrication. Importantly for our design considerations, there is a much larger space requirement for rear projection than front projection, even with folded projection using mirrors. Given the available space, we could construct a viewing space 12' square with front projection, but only about half that size using rear projection.

We decided on front projection. This would allow a larger space for viewers than possible with a very small rear projection system, as well as more readily permit multiple configurations to support the multiple uses of the laboratory. Active stereo display was chosen, rather than passive stereo display, to minimize number of components. Figure 1 below shows two of the configurations routinely used with four stereoscopic projectors. Figure 2 shows an image projected using the three wall display.

The stereoscopic projectors used in the system are relatively low cost Lightspeed DepthQ projectors. For the three wall configuration shown in the left of Figure 1 the projectors are mounted vertically. Each projector's image is 8' tall and 6' wide. For the two wall configuration shown in the right of Figure 1 projectors are mounted horizontally and each projector's image covers an area 6' wide and 4.5' tall, providing a smaller, higher resolution display. This second arrangement of projectors leaves an area in the center of the projected area in which a single user can view the display from within the projected area. Though mobility is restricted, the user is still able to turn left and right, and the viewing location within this projected image space provides much of the advantages of immersion provided by a rear projection system. The configuration is easily extended to four walls by adding an additional four projectors to supply images for the other two walls, as in the LAIR system [9]. Though not shown in the figure, projectors can be stacked to project a

Figure 1. Two configurations of the 12' wide laboratory. On the left is a three wall, four projector display allowing multiple viewers to "look into" an 8' tall area. On the right a two wall four projector configuration with higher resolution allows a single user to sit or stand within the projected volume without obstructing the front projected image. The projectors can be also be stacked to provide a single 12'x7' 3.6 mp display.

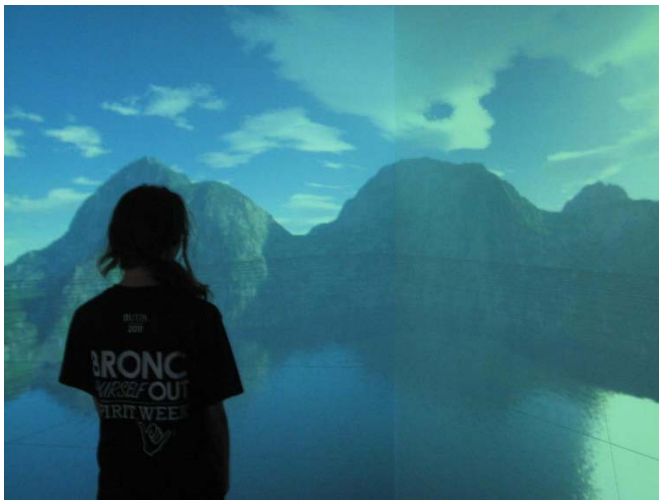


Figure 2. Three sided display. Several viewers can stand across the front area to view the scene.

single 12'x7' image with a total display resolution of 2560x1440. Conventional ceiling mounting of a single projector illuminating a 12'x7' wall accommodates 12 viewers.

A single computer system providing four stereoscopic outputs is used. The computer (HP Z800) and two graphics cards (nVidia 5000), each with two stereoscopic outputs, use nVidia's SLI (Scalable Link Interface) architecture. The architecture provides transparent gpu scalability and eliminates much of the programming required for multiple displays. Using a single computer avoids display synchronization issues and their associated software and hardware costs. Results with the single computer system have been satisfactory, and suitably high frame rates are possible due to the gpus' handling of the graphics processing. Other

systems utilizing less expensive graphics cards and stereoscopic monitors are also available for development and educational use.

The principle software tools for virtual environment development are VTK (Visualization Toolkit) [28, 29] and UDK (Unreal Development Kit) [35]. Each is used for both educational and research efforts. Both are freely available. VTK serves a wide range of needs. It is the principle software package used in the visualization class supported by the laboratory. Students typically execute programs that are developed on other systems to explore stereoscopic and immersive display in the lab. In research efforts its wide range of libraries, together with its extensibility using native code, supports efficient iterative design. VTK also provides a simple means to create programs for multiple displays through its multiple viewport capabilities. VTK's support for multiple languages, e.g., TCL, Python, allows prototyping and experimentation with modest programming effort. UDK is a game engine with its own scripting language, supporting a range of interaction and high level functionality. A significant element in the context of the laboratory is the ease with which moderately complex textured scenes can be created with UDK so that additional tools for scene creation are not needed. In addition to VTK and UDK, VR Juggler, VRPN, and OpenGL are extensively used.

System cost was a primary factor in design decisions and component selection. We chose components that would provide a reasonable compromise between the very lowest cost and satisfactory performance. During component identification a game development laboratory with less demanding computing and display requirements was also under development, which provides comparison of lower cost components.

Total cost for the system was about \$20,000. The largest cost was for the four stereoscopic DepthQ projectors, \$10,000. The single computer and two nVidia Quadro 5000 graphics cards were \$5,000. Six pairs of nVidia shutter glasses and rf emitter cost \$1,500. Tracking is performed by a relatively fast camera based system, NaturalPoint's OptiTrack S250, at a cost of \$3,500. It was selected to maintain compatibility with the CAVE system. For the relatively small space of the system, tracking could be done satisfactorily with game hardware based components, e.g., Kinect or Wii Remote [40]. Fabrication costs for projector stands, together with screen frame and materials, were less than \$1,000.

A similar monoscopic system with less expensive computer and graphics cards that would be quite suitable for many applications could be configured for less than half the cost of the system. Projectors of similar resolution and output would cost less than \$4,000. Computer and graphics controllers would cost about \$2,500 using, for example, nVidia GeForce cards. Six pairs of nVidia glasses and ir emitter would cost less than \$1,000. Tracking could be

performed using a game hardware based system for a few hundred dollars. Mountings and screen material would cost the same, less than \$1,000.

4 Conclusions

Current technology allows the construction of immersive display systems for virtual environments for orders of magnitude less cost than the earliest CAVE systems. This work describes one such system to be used for both research and education and details a relatively low cost solution to a set of design considerations common to many systems. The stereoscopic front projection display provides a means to create immersive environments that minimizes space utilization, while maintaining a relatively high degree of display immersion and configuration flexibility. The single computer, four graphics gpu system provides a computing solution that helps minimize both software configuration effort and cost. The use of publicly available software demonstrates viable low cost solutions to immersive display systems configuration and virtual environment development.

5 Acknowledgements

This work was supported by National Science Foundation grant MSI-CIEC, OCI-0636352, Department of Education grant P116Z0202159, and the University of Texas – Pan American Computing and Information Technology Center.

6 References

- [1] R. G. Belleman, B. Stolk and R. D. Vries, "Immersive virtual reality on commodity hardware," *Proc. 7th Ann. Conf. of the Advanced School for Computing and Imaging*, pp. 297–304, 2001.
- [2] A. Bierbaum, C. Just, P. Hartling, K. Meinert, A. Baker and C. Cruz-Neira, "VR Juggler: A virtual platform for virtual reality application development," *Proc. IEEE Conf. Virtual Reality*, pp. 89–96, 2001.
- [3] A. Bierbaum and C. Cruz-Neira, "ClusterJuggler: A modular architecture for immersive clustering," *Proc. Workshop on Commodity Clusters for Virtual Reality, IEEE Virtual Reality Conf.*, 2003.
- [4] D. A. Bowman and R. P. McMahan, "Virtual reality: How much immersion is enough?," *Computer*, July, pp. 36-43, 2007.
- [5] D. C. Cliburn and J. Krantz, "Towards an effective low-cost virtual reality display system for education," *Journal of Computing Sciences in Colleges*, vol. 23, no. 3, pp. 147–153, 2008.
- [6] C. Cruz-Neira, D. Sandin, T. DeFanti, R. Kenyon and J. Hart, "The CAVE: Audio visual experience automatic virtual environment," *Communications of the ACM*, vol. 35, no. 6, pp. 64-72, 1992.
- [7] C. Cruz-Neira, D. Sandin and T. DeFanti, "Surround-screen projection-based virtual reality: The design and implementation of the CAVE," *SIGGRAPH '93 Proc. 20th Ann. Conf. on Computer Graphics and Interactive Techniques*, pp. 135-142, 1993.
- [8] M. Czernuszenko, D. Pape, D. Sandin, D., T. DeFanti and M. Brown, "The ImmersaDesk and Infinity Wall projection-based virtual reality displays," *IEEE Computer Graphics*, vol. 31, no. 2, pp. 46–49, 1997.
- [9] B. Denby, A. G. Campbell, H. Carr, H. and G. M. P. O'Hare, "The LAIR: Lightweight affordable immersion room," *Presence: Teleoperators and Virtual Environments*, vol. 18, no. 5, pp. 409-411, 2009.
- [10] T. A. DeFanti, G. Dawea, D. J. Sandin, J. P. Schulze, P. Otto, J. Girado, F. Kuester, L. Smarra and R. Raoa, "The StarCAVE, a third-generation CAVE and virtual reality OptIPortal," *Future Generation Computer Systems*, vol. 25, pp. 169–178, 2009.
- [11] K. J. Fernandes, V. Raja and J. Eyre, "Cybersphere: The fully immersive spherical projection system," *Communications of the ACM*, vol. 46, no. 9, 2003.
- [12] H. J. Gardner, R. W. Boswell and D. Whitehouse, "The WEDGE emmersive projection theatre," *Proc. SimTect99*, 1999.
- [13] J. Jacobson, "Configuring multi-screen immersive displays with existing computer equipment," *Proc. Human Factors and Ergonomics Society 46th Ann. Meeting*, pp. 760-765, 2002.
- [14] J. Jacobson and L. Lewis, "Game engine virtual reality with CaveUT," *IEEE Computer*, vol. 38, no. 4, pp. 79-82, 2005.
- [15] A. Juarez, W. Schonenberg and C. Bartneck, "Implementing a low-cost CAVE system using the CryEngine2," *Entertainment Computing*, vol. 1, pp. 157–164, 2010.

- [16] O. Kreylos, "Environment-independent VR development," *International Symposium on Visual Computing 2008, LNCS*, vol. 5358, pp. 901–912, 2008.
- [17] W. Krueger and B. Froehlich, "The responsive workbench," *IEEE Computer Graphics and Applications*, vol. 16, no. 4, pp. 44-47, 1996.
- [18] P. Lin, Z. Pan, Y. Yang and J. Shi, "Implementation of a low-cost CAVE system based on networked pc," *Proc. Virtual Environment on a PC Cluster Workshop*, pp. 33-40, 2002.
- [19] T. Ni, G. S. Schmidt, O. G. Staadt, M. A. Livingston, R. Ball and R. May, "A Survey of large high-resolution display technologies, techniques, and applications," *Proc. IEEE Virtual Reality Conf.*, pp. 223-236, 2006.
- [20] D. Pape, J. Anstey and G. Dawe, "A low-cost projection based virtual reality display," *Proc. SPIE (International Society for Optical Engineering)*, vol. 4660, pp. 483-491, 2002.
- [21] D. Pape, C. Cruz-Neira and M. Czernuszenko, "CAVE user's guide," Electronic Visualization Laboratory, University of Illinois at Chicago, 1997.
- [22] A. Peternier, S. Cardin, F. Vexo and D. Thalmann, "Practical design and implementation of a CAVE environment," *Proc. Int. Conf. Computer Graphics, Theory and Applications GRAPP*, pp. 129–136, 2007.
- [23] N. Polys, S. Kim and D. Bowman, "Effects of information layout, screen size, and field of view on user performance in information-rich virtual environments," *Proc. ACM Symp. Virtual Reality Software and Technology*, pp. 46-55, 2005.
- [24] F. Pognadoresse, P. Bouvier, A. Herubel and V. Biri, "From research on the virtual reality installation," *Human-Computer Systems Interaction: Advances in Intelligent and Soft Computing*, vol. 60, pp. 335-345, 2009.
- [25] B. Raffin and L. Soares, "PC clusters for virtual reality," *Proc. IEEE Virtual Reality Conf.*, pp. 215-222, 2006.
- [26] M. Roth, G. Voss and D. Reinert. "Multi-threading and clustering for scene graph systems," *Computers and Graphics*, vol. 28, no. 1, pp. 63–66, 2004.
- [27] P. M. Sauter, "Vr2go: A new method for virtual reality development," *SIGGRAPH Computer Graphics*, vol. 37, no. 1, pp. 19–24, 2003.
- [28] W. J. Schroeder, L. S. Avila and William Hoffman, "Visualizing with VTK: A Tutorial," *IEEE Computer Graphics and Applications*, Sept./Oct., pp. 20-27, 2000.
- [29] W. Schroeder, K. Martin and B. Lorensen, *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*, 4th ed., Kitware, 2006.
- [30] T. Schou and H. Gardner, "A Wii remote, a game engine, five sensor bars and a virtual reality theatre," *Proc. ACM 19th Australasian Conf. on Computer-Human Interaction: Entertaining User Interfaces*, p. 234. 2007.
- [31] S. Schlechtweg, "miniCAVE—a fully immersive display system using consumer hardware," *Proc. EuroGraphics Virtual Environment EGVE Symposium*, pp. 73-90, 2008.
- [32] H. Seeley, "Game technology 2007: Cryengine2," *Proc. ACM SIGGRAPH 2007 Computer Animation Festival*, p. 64, 2007.
- [33] W. Sherman, "Commodity-Based Projection VR: Software for Virtual Reality," *ACM SIGGRAPH 2004 Course Notes*, ACM, New York, 2004.
- [34] W. R. Sherman, P. O'Leary, E. T. Whiting, S. Grover and E. A. Wernert, "IQ-Station: A low cost portable immersive environment," *Advances in Visual Computing, Lecture Notes in Computer Science*, vol. 6454, pp. 361-372, 2010.
- [35] UDK – Unreal Development Kit – Epic Games, <http://udk.com/>, 2012.
- [36] University of Minnesota, PowerWall, <http://www.lcse.umn.edu/research/powerwall/powerwall.htm>, University of Minnesota, 1994.
- [37] vrjuggler, <http://code.google.com/p/vrjuggler/>, 2012.
- [38] VRPN, <http://www.cs.unc.edu/Research/vrpn/>, 2012.
- [39] E. J. Wegman, J. Symanzik, J. P. Vanderluis, Q. Luo, F. Camelli, A. Dzubay, X. Fu, N. A. Khumbah, R. E. A. Moustafa, R. L. Wall and Y. Zhu, "The MiniCAVE: A voice-controlled IPT environment," *Proc. 3rd International Immersive Projection Technology Workshop*, pp. 179–190, 1999.
- [40] C. A. Wingrave, B. Williamson, P. Varcholik, J. Rose, A. Miller, E. Charbonneau, J. Bott and J. J. LaViola, "The Wiimote and beyond: Spatially convenient devices for 3D user interfaces," *IEEE Computer Graphics and Applications*, pp. 24-38, 2010.

Fact and Fiction – The Virtual World

The Past, Present and the Future of Computer Gaming

Shivansh Srivastava

3rd Year Undergraduate

BITS Pilani, India

Email - shivansh.bits@gmail.com

July, 2012

Abstract— This work reviews several aspects of the ever-increasing field of Computer Games. First, the Evolution of Computer Games is discussed. Two different classes of Principles by two great Pioneers of the field, interested in cognitive impact of Game Design, are reviewed. The study is a methodological framework, in which results & effects are compared, inferred, interpreted & generalized. The study takes on a unified research paradigm & methodologies to lay out the underlying principles of Gameplay, pursuing & achieving goals, consistency & fairness of gameplay, process of game designing. It takes a retrospect of the Interaction Model, analyzing the perspectives, comparing the Physical & Temporal Dimensions. The study also discusses the Game Business & the shape of the Industry. Finally, the Future of Game Development is concluded, deducing with respect to World of Mass Development (WMD) & HTML5.

Keywords— Game Engines, Graphics, Game Design, Gameplay, Game Business, WMD.

I. INTRODUCTION

The title of my Research Paper might be a bit deceiving, but what I intend to showcase here has a larger realm. Computer Game design and production is a fast paced, hit-driven, technology-based field, where Software is extensively improving, increasingly dependent on the high RAM & Graphic Cards; and the list of awe-inspiring & stunning features to incorporate is ever increasing. Computer games have a growing necessity of being recreational, & while novelty is good, what really matters is an enjoyable & an engaging experience. A sequel can be a hit while a strikingly original game may not. The following paper discusses gaming in the eras gone by, the gradual evolution & upcoming of different Game engines & development tools, the near Realistic games of the 21st Century, and what the future has in store for Computer Games Developers & Enthusiasts.

II. METHODOLOGY

A. *Empirical Research*

The data showcased in this paper has been taken from a Racing game project I worked on. The models used in the Racing game were designed in AutoDesk MAYA, which were exported as DirectX objects & integrated in the game, with XNA 3.0 Framework, using Microsoft Visual Studio 2008. The Physics & the Math of the development have descriptively been explained.

B. *Secondary Research*

The books I used for my Research were O'Reilly – Learning XNA 3.0, Packt Publishing 3D Graphics with XNA Game Studio 4.0 & websites include www.riemers.net & www.emunix.emich.edu/ and Top 10 Usability Recommendations by Nokia Series 40 Game Study^[1].

III. DATA USED IN THE PAPER

A. *Abbreviations and Acronyms*

XNA - XNA is Not an Acronym - XNA Framework is based on the native implementation of .NET Compact Framework for Xbox 360 development and .NET Framework on Windows. Its versions include Game Studio 2.0, 3.0, 3.1 & 4.0 (primarily used for Game Development on Windows Phone 7 platform (including 3D hardware acceleration), framework hardware profiles, configurable effects, built-in state objects, graphics device scalars and orientation, cross-platform and multi-touch input, microphone input and buffered audio playback, and Visual Studio 2010 integration.^[2]

AI – Artificial Intelligence – It refers to techniques used in computer and video games to produce the illusion of intelligence in the behavior of non-player characters. It may also refer to a broad set of algorithms that also include techniques from control theory, robotics, computer graphics and computer science in general.

SDK – Software Development Kit – It is a set of software development tools that allows for the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar platform. It may be something as simple as an Application Programming Interface (API) in the form of some files to interface to a particular programming language or include sophisticated hardware to communicate with a certain Embedded System.

B. Units

Metrics Systems here used are mainly in terms of pixels (px) of a screen. In case of Equations, SI (MKS) Metric System is used.

C. Equations

The Physics & Math of a Game is what defines the Virtual Reality. Taking the contemporary example of EA Sports FIFA, the reasons needed to know the physics equations are:

- Making a ball bounce (Co-efficient of restitution)

$$C_R = (v_b - v_a) / (u_a - u_b) \quad (1)$$

- Make a ball move in a direction with 'x' Newtons of force
- Distance travelled by a ball in give time period 't'

$$s = v*t + x_o \quad (2)$$

$$s = u*t + (0.5)*a*t^2 \quad (3)$$

- Make something have more or less friction so it will slide at a different speed.

$$f = u*M*g \quad (4)$$

Where, u = Coefficient of Friction

M = Mass of the object

g = Acceleration due to Gravity

- Collision responses - What direction will the ball be directed when hit. In such cases, Kinetic Energy & Linear Momentum will be conserved.

$$M_1*V_{i1} + M_2*V_{i2} = M_1*V_{f1} + M_2*V_{f2} \quad (5)$$

$$M_1*V_{i1}^2 + M_2*V_{i2}^2 = M_1*V_{f1}^2 + M_2*V_{f2}^2 \quad (6)$$

- Gravity – Force between two objects

$$\text{Force, } F = G*(M_1*M_2)/ D^2 \quad (7)$$

Where, G = Gravitational Constant

D = Distance between the two objects

Game development is as data driven as art production. Math is like technique, the more the better, as long as you can apply it.

- Arithmetic

- Linear Algebra (with geometric interpretations)

$$\text{Distance travelled} = \text{sqrt}(a^2 + b^2 + c^2) \quad (8)$$

- Calculus
- Combinatory (randomized levels)
- Probability (balancing)
- Statistics (Bayesian esp. for AI)

IV. SCOPE OF RESEARCH

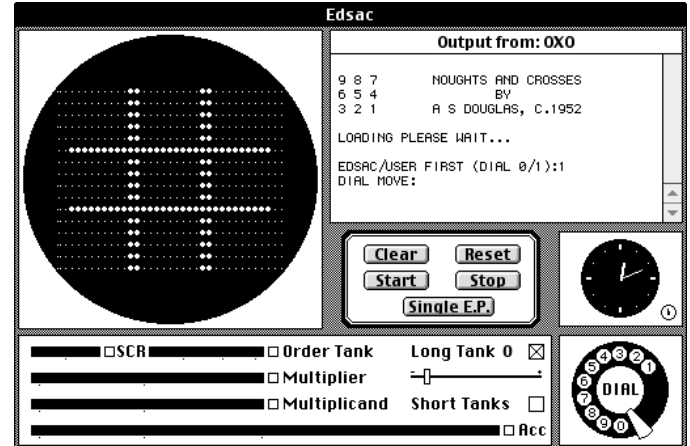
The world of Computer Gaming & its Development is vast; hence the study focuses on its History, Evolution, Design Principles, Game Design Process, Implementation of Physics, Types of Challenges, Game Business and its Future Expectations.

V. CONCEPTUAL FRAMEWORK

A. History of Computer Gaming & Industry

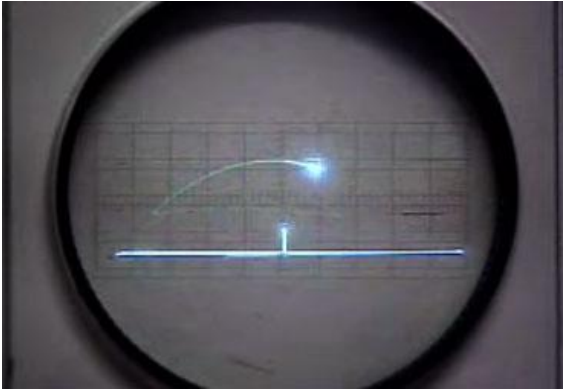
1) First "Games"

"OXO" (TicTacToe) was a computer game written for the EDSAC computer in 1952, an implementation of the game known as Noughts and Crosses in the UK, or tic-tac-toe in the United States. It was written by Alexander S. Douglas as an illustration for his Ph.D. thesis on human-computer interaction for the University of Cambridge. OXO was the first digital graphical game to run on a computer.^[3]



"Tennis for Two" was a game developed in 1958 on an analog computer, which simulated a game of tennis or ping pong on an oscilloscope^[4], created by American physicist William Higinbotham, it was one of the first electronic games to use a graphical display. He took an oscilloscope, what would be the first monitor, and a simple analog computer and put together a simple ping pong simulation using the computer to talk to the oscilloscope and put a bouncing dot of light and a bar which it bounced off of. The oscilloscope had been around for a little while but the computer was similar to the kinds which were used to coordinate the millions of bombs

dropped in World War II. Also, in order to program the physics of the game he used the current and much used protocol of missile trajectory plotting.



2) *First Generation Games*

The 70's saw a massive outbreak of ideas & implementation. One of the pioneers of such innovations was Ralph Baer. The idea began to form when he decided that he wanted to change the passive nature of the television into something more active, something that people could interact with. One day, while waiting for a tardy colleague in a New York City Bus Terminal, Baer sketched out his ideas for a TV based entertainment system. He later fleshed out his ideas in a four page paper. In the paper, he outlined his ideas for a low-cost system that would attach to a TV and display different games through the TV. His invention, the Light Gun, became the first video game in history to be filed for patent.

3) *Birth of Commercial Games*

In 1972, Nolan Bushnell started a company called Atari, & published a game by the name of Pong, which was hard wired into a wooden cabinet made for the sole purpose of playing Pong. It went on sale for \$1,200 per unit.

4) *Arrival of Color*

The next revolution was the advent of color. In 1974, a Japanese company known as Namco bought out the Japanese division of Atari and thereby instantly made a name for themselves in video games. In 1979, they develop what would be the first color video game. Up until this point, all the color on video games was faked by overlays on certain parts of the screen to give the appearance of color. The game was called Galaxian and it was rampantly successful.^[5]

5) *Subsequent Evolution*

In 1974, a company by the name Kee released the game Tank, the first game to use ROM. Tank featured far more advanced graphics than Pong in that it featured graphic memory which allowed for a higher degree of detail on screen. Also, in 1974, Atari launched the first racing game (Trak 10) & maze chase game (Gotcha).

In 1980, a designer by the name Moru Iwatani developed a game that would be Pac-man. This name soon became in household use and was the first video game to be popular

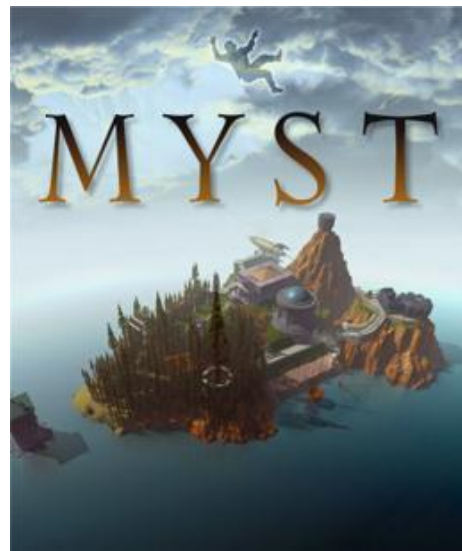
enough to warrant merchandising. There were Pac-man cereals, a Pac-man album, and even a Pac-man TV show.^[6]

6) *Console Wars*

In 1990, Nintendo released Super Mario 3 - all-time best-seller & PC's and Consoles are major game platforms.^[7] Electronic Arts acquired other game publishers. In 1991, Nintendo launched Super-NES (16 bit) & soon in 1992, PC gaming explodes as Nintendo has \$7 billion in sales (\$4.7B in US). It made higher profits than all U.S. movie and TV studios combined.

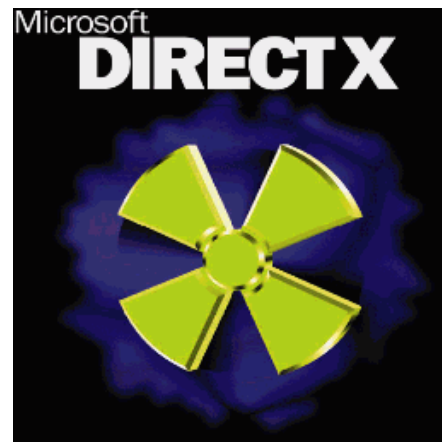
In 1994, Atari shipped Jaguar (64 bit), becoming one of the costliest consoles at around \$700 (>\$100/game).

Soon, DOOM, a first person shooter game was released by id Software and MYST, a graphic adventure video game, was released by Cyan Worlds, becoming the All time biggest selling PC game until 2002.^[8]



7) *32 Bit Wars*

In 1995, Sony introduced Playstation (32-bit) & Microsoft released Window 95, including the Game SDK – DirectX, bringing major game performance to Windows.^[9]



8) *Era of Playstations*

Playstation was launched in U.S. in the September of 1995, processing 300,000 polygons/sec at 30MIPS processor, with 4MB RAM & 2MB VRAM.^[10]



9) *Pre-Modern Gaming Era*

1996 saw Nintendo introduced Ultra 6, with multi-player gaming going commercial via modem and internet and network companies. It was 1997 that saw 3D acceleration starting to standardize on 3D-FX & games start to assume 3D acceleration. Pentium II's enhanced to 200Mhz.^[11]

In the September of 1996, Nintendo 64 is launched in US with 93.75 MHz & 64 Bit CPU with a 64-bit MIPS co-processor & over 500,000,000 - 16-bit operations/sec. It consisted of a built-in Pixel Drawing Processor (RDP) with 4.5MB RAM & 150,000 polygons/sec.^[12]

In 1999, the Maximum Score for Pac-Man was achieved, when Billy Mitchell achieved the highest possible score for Pac-Man, as he completed every board and wined up with a score of 3,333,360.^[13]

In 2000, development moves from PC to consoles as Playstation II is launched. Diablo II sold 1 million units in 1 week & SIMS sold 2.3 million units (\$95M). In 2001, Nintendo launched the Gamecube & Xbox was launched by Microsoft.

The launch of Sony Playstation 2 on May 4, 2000 in Japan became the best-selling console of all time. By Feb 2011, 1.52 billion PS2 titles had been sold since launch. Hardware used by it was a 128 Bit 300MHz processor, 3 Special purpose 150 MHz co-processors, 32MB DRAM: 3.2 GB/sec, DVD & CD drive, MPEG2 hardware, a Dual Shock 2 analog controller; processing at 66M polygons/sec geometry – 16M polygons/sec for curved. Evidently, Software development became tough.^[14]

10) *Modern Gaming Era*

November of 2001 saw the arrival of Direct X API, with Pentium IV 733 Mhz, a Custom 3-D 300Mhz GPU, 64MB Ram, 6.4 GB/sec, an 8GB hard drive; with sleek performance of 150 million transformed, 100+ million polygons per second sustained performance (shaded, textured), 300 million micropolygons/particles per second, full-scene anti-aliasing, 1920x1080 Maximum Resolution with HDTV support.

With arrival of more than ever PC games, in 2003, SIMS continued to grow, becoming the best-selling PC game of all time. WarCraft III, UT 2003 & GTA also took the market by a storm. Second Life and There.com are launched. EA bulked \$2.5B in 2003.^[15]

2004 became a year of sequels. Big Gaming firms increased their profits by launching SIMS 2, Halo 2, Half-life 2 & Doom.

2005 saw World of Warcraft take the whole Gaming Planet by storm, with 4 Million Subscribers at \$700M/year subscriptions.^[16]



11) *Post Modern Gaming Era*

November 2005 saw the arrival of Microsoft's Xbox 360, introducing an entirely new dimension to Gaming experience. The hardware included a Custom IBM PowerPC CPU with 3 symmetrical cores - 3.2 GHz each, with 2 threads/core, VMX-128 vector unit/core, Custom ATI Graphics Processor with 10MB DRAM, processing 500 million triangles/sec, 16 gigasamples/sec, 48 billion shader operations/sec, supporting 16:9, 720p or 1080i – HD output, 512 MB of 700MHz GDDR3 RAM – unified memory architecture, 22.4 GB/s interface bus bandwidth, 256 GB/s memory bandwidth to EDRAM & 21.6 GB/s front-side bus. It had an Overall system floating-point of 1 TeraFlop.



Playstation 3 influenced the market equally, with in-built Graphics by Nvidia, 550 Mhz GPU at 1.8 TeraFlops, 100 billion shader operations/sec, 51 billion dot products/sec. It consisted of 512MB RAM split between the CPU and graphics, with 512KB L2 cache & 7 AltiVec vector processing units.^[17]

The PlayStation 4 is expected to sport a 1.5 TB hard disk drive, which allows users to store a large number of games. It will have USB 3.0 ports along with an HDMI connect port. The Sony PlayStation 4 will have full 3D support with 4K2K compatibility, and supports 3D Blu-Ray.

B. Principles of Game Design

Games are a part of lives that have a story, one which is interactive & demands participation. Computer Games are a completely new form of entertainment, with completely new worlds to play in, allowing players to take on a new persona.

A game is a representation of physical objects – real or imaginary, such as a Terrain, Buildings (exterior and interior – walls, floors, etc.), Game objects (furniture, balls, fluids, weapons, vehicles, etc.), Animate objects (player, opponents, cars, etc), Providing dynamics to world, Physics, Behavior with Artificial Intelligence (AI), Supporting interaction, Graphics, Audio, such as dynamic sound, music & speech, and Networking.

ENGINEERING: TRADITIONAL VS. GAME

Traditional engineering	Game engineering
<ul style="list-style-type: none"> • High precision • Realism 	<ul style="list-style-type: none"> • High speed • Low memory • No spiking in resources • Scalability • Believability • Control • Low Cost Development

True for graphics, physics, AI, audio, etc.

1) Good Gameplay

- Challenges
- Interactivity
- Feedback about position relative to goals
- Interesting choices required to achieve goals
- Consistency and fairness
- Avoid repetition

2) Pursuing and Achieving goals

There is always something to achieve & the Character is always achieving something. Often, there are three levels of goals with rewards-

Long-term goal - “I can conquer the world.”

Medium-term goal - “I can take over a city.”

Short-term goal - “I can win a battle.”

For different genres of Games, there can be different Goals, such as-

Eliminate other players - Action games

Score points - Sports games

Get somewhere first - Racing games

Solve puzzles - Adventure games

Gain territory - Strategy games

Improve abilities - Role-playing games

Develop social relationships - Massively multiplayer games

Play God – Simulation

3) Gameplay Consistency and Fairness

There is consistency in the actions and associated outcomes for trying to achieve goals

Must be a reason for failure (or success)

Not arbitrary: Players know what to expect and can plan, e.g. - A pinball game uses “pinball” physics all the time.

Don’t solve problems by unique & unlikely actions.

Don’t break suspension of disbelief.

No “dead man” walking.

Fairness - Player should think they have a fair chance – game balance.

Can still be plot twists, but must be explainable.

4) Gameplay Sins

Poor production - Break the suspension of reality

Linear plot/gameplay - Player’s actions don’t affect how the plot progresses.

Micromanagement - Player is forced to perform menial tasks & AI should take care of all the obvious choices.

Repetition - Player must do same action over and over again, Player must sit through same cut scenes every time they play, have to replay 90% of level to fight boss, etc.

Doesn’t track user’s learning curve - Should start easy and get harder as game progresses.

Poor game balance - Same strategy always works & Trial and error is not fun Gameplay.

Not enough variety - Same graphics, objects, monsters, level design, sounds.

Awkward user interface

Limited feedback - Player is confused about goals, Player is confused about current progress to goals, no maps or help.

Inconsistency in story - There are no compelling and consistent goals for the player.

Dead and you don't even know it.

5) **Shigeru Miyamoto Design Principles**

Shigeru Miyamoto is mainly known for his work at the video game production company Nintendo, where he created some of the most successful video game franchises of all time, including *Mario*, *Donkey Kong*, *The Legend of Zelda*, *Star Fox*, *F-Zero*, and *Pikmin*.^{[18][19]} The Principles are-

- Start with a simple concept
- “running, climbing, jumping”
- Design around the computer's limitations
- Character wears dungarees so easier to see arms move
- Wears a hat because don't have to have hair
- Has mustache because couldn't draw nose and mouth
- Minimize the player's confusion
- What to do should be clear without consulting a manual
- The importance of play testing
- Incorporate a smooth learning curve
- Accommodate all skill levels

6) **Sid Meier Design Principles**

Sidney Meier is a Canadian-American programmer and designer of several popular computer strategy games, most notably *Civilization*. He has won accolades for his contributions to the computer games industry. Meier is a Director of Creative Development for computer game developer Firaxis Games.^[20] The Principles are-

- Player should have fun, not designer, programmer, or computer.
- Begin your game with a great first few minutes.
- Great game-play is a stream of interesting decisions that the player must resolve.
- The inverted pyramid of decision making lets the users make a few decisions to deal with first, and then let them multiply until the player is totally engrossed.
- Put the player in his dreams, where he/she is the hero.

7) **Nokia Series 40 Game Study: Top 10 Usability Recommendations**^[1]

- Provide a Clear Menu Structure
- Simplicity Is Key -If two solutions are equally valid, use the simpler.

- Provide Help When Needed
- Be Relentlessly Consistent - Use the mother tongue of the user; Be consistent with the phone's UI, with game industry conventions, and within the game itself.
- Don't Waste the User's Time
- Use Natural Controls
- Enable Save and Pause - Provide a simple save-game feature. Have the game auto-save when the user presses the red phone button - use the `destroyApp()` method to do this.
- Conform to Real-World Expectations – e.g. when jumping or throwing objects, the flight path should be predictable. There must be no invisible barriers that the player cannot pass or holes that he cannot reach. Do not end the game arbitrarily. Implement a realistic physics model, if relevant (e.g. racing games).
- Go Easy on the Sound
- Implement a High Scores List

C. **Game Design Process**

1) ***Idea for a Game***

Most games begin with a single idea. Idea can revolve around - A character [James Bond], Gameplay/Genre [A twitch FPS, an RTS game], a sport [Football, Baseball, Ice Hockey], a story/quest/goal [A time-travel adventure], a new technology [Motion capture of pro basketball players]. Along the same line, idea may be original, old, or hybrid – like SIMS, Civilization, FIFA

2) ***Inspiration***

For inspiration, mix existing ideas from other games. Steal ideas (but not characters) from other media, books, movies, comics, etc. Market Research - surveys, focus groups, sampling, case study, critical theory, etc. can prove helpful. Take a contemporary idea and make it better. Better technology - graphics, sound, AI, can be implemented. Better story should be devised in a different environment.

3) ***Interactivity is the Raison d'être of Computer Games***

Ask “What is the player going to do?” - This question comes before all others.

Do not get sidetracked with story, character, core mechanics, artwork or ANYTHING else until you know the answer to this question.

4) ***The Player's Role***

Who is the player trying to be - Critical for representational/realistic games

In single game may have multiple roles/multiple modes – In Football – manager, coach, player

If you can't describe it clearly, it will be confusing for the player

5) ***Interaction Model***

As an avatar - A single character or object that represents the play, Player's actions are limited to the avatar's location

Omnipresence (by not necessarily omniscience) - Player can act in many or all places in the world, Chess is an obvious example

6) **Perspectives**

First-person - Doom, Half-Life, halo, Call of Duty

Third-person – Max Payne, Resident Evil 4

Side scrolling - Sonic

Aerial – isometric/top-down – HAWX, FIFA 12

Context sensitive - Resident Evil

DIMENSIONS: PHYSICAL VS. TEMPORAL

The Physical Dimension	The Temporal Dimension
<ul style="list-style-type: none"> • Dimensionality - 2D, 3D, 4D (multiple 3D spaces) • Scale - How big is the virtual world, How big are things relative to each other • Boundaries - What happens at the edge of the virtual world 	<ul style="list-style-type: none"> • <i>Is time meaningful</i> - Does the passage of time itself change the game • <i>Real time</i> or turn based • <i>Variable time</i> – eg.- in The Sims, time speeds up while people sleep. • <i>Anomalous time</i> - Time goes faster for some things than others. • Can the player adjust time – flight simulators and RTS game

7) **Types of Challenges**

Physical Challenges - Speed and reaction time (twitch games), Accuracy and precision (steering and shooting), Timing and rhythm (dance games), Learning special moves (fighting games)

Races – achieving something first

Logical challenges (puzzles) - Should be based on an underlying principle, Trial-and-error solution is a sign of bad design,

Exploration Challenges - Locked doors and traps, Mazes and illogical spaces, Teleporters

Conflict - Strategy, tactics, Logistics, Survival and reduction of enemy forces, Defending vulnerable items or units, Stealth

Economic Challenges - Accumulating wealth or points, Efficient Manufacturing, Achieving balance or stability in a system, Caring for living things within a system

Conceptual Challenges - Understanding something new, Deduction, observation, interpretation, Detective games offer conceptual challenges

8) **Positive Feedback**

Positive Feedback is an achievement that makes subsequent achievements easier, like taking an opponent's piece in chess. Without positive feedback, it is too easy to get stalemate. It must be controlled to avoid giving the lead player too much advantage. Examples of Positive Feedback include getting ahead in a race, more likely to get power-ups or special scores; in Monopoly – get houses, more likely to get even more money.

9) **Negative Feedback**

Negative Feedback increases the impact of chance - if chance is fair, it helps as much as hurts. Define victory in non-numeric ways - Chess is not won by taking away the most pieces. Increase the difficulty level as feedback kicks in, which happens in role-playing games. Examples of Negative Feedback – while getting ahead in a race, user is more likely to lose, due to Drafting Car Racing.

D. **Game Physics**

1) **Why Physics?**

Some games don't need any physics. Games based on the real world should look realistic, implying realistic action and reaction. Complex Games need more physics- such as sliding through a turn in a racecar, running and jumping off the edge of a cliff. [21]

For Newtonian physics with Rigid bodies, equations (1), (2) & so on can be applied.

2) **Position and Velocity**

Modeling the movement of objects with velocity - Where is an object at any time t? (Assume our metric is pixels)

Taking equation (2) into account -

$$\text{player_x}(t) = t * \text{x_velocity} + \text{x_initial}$$

$$\text{player_y}(t) = t * \text{y_velocity} + \text{y_initial}$$

Hence, for Computation, equation would be comprehended as -

$$\text{player_x} = \text{player_x} + \text{x_velocity}$$

$$\text{player_y} = \text{player_y} + \text{y_velocity}$$

Where, x_velocity & y_velocity are the vectors in the X & Y direction, travelled by the player from the Original Position.

3) **Acceleration**

For Computation, equation would be written as -

$$\text{x_velocity} = \text{x_velocity} + \text{x_acceleration}$$

$y_velocity = y_velocity + y_acceleration$

Changing acceleration can be computed as:

$x_acceleration = \cos(\Theta) * acceleration$

$y_acceleration = \sin(\Theta) * acceleration$

Acceleration comes into account in case of Racing Games such as Need for Speed Series.

4) Gravity

Taking equation (3) & (7) into account -

$$v(t) = 0.5 * g * t^2$$

$$g = 9.8 \text{ m/s}^2$$

For Computation, equation would be written as – (from equation (7))

$x_velocity = x_velocity + 0$

$y_velocity = y_velocity + gravity$

Gravity comes into account in case of Aerial games, like HAWX, Flight Simulator, or Space based Games.

5) Friction

Friction (Equation (4)) comes into account in case of Complex Racing Games; examples include Need for Speed Series, Billiards (Pool), FIFA.

6) Momentum and Energy

Conservation of Kinetic Energy & Linear Momentum will come in case of an occurrence of a Collision. Equation (5) & (6) can be used, accordingly.

E. Game Business

1) Shape of the Industry

Hardware - Sony, Nintendo, Intel, IBM, Microsoft

Software – Electronic Arts, Activision, Sony, Microsoft, Ubisoft, THQ, Vivendi, Warner Bros.

Internet - Sales, updates, multiplayer versions of games, massively multiplayer games.

2) A Hit-Driven, Entertainment Business

The interactive entertainment business is ENTERTAINMENT - It is NOT a packaged goods business. Games generate emotional responses, and are designed to fulfill fantasies, provide escape from reality, and stimulate the senses.

3) Top 10 Best-Selling Games of 2011^[22]

1. *Battlefield 3* (360, PS3, PC) Electronic Arts – 2 million
2. *Batman: Arkham City* (360, PS3) Warner Bros. Interactive – 1.5 million
3. *NBA 2K12* (360, PS3, PSP, Wii, PS2, PC) Take Two Interactive
4. *Rage* (360, PS3, PC) Bethesda Softworks – 550k
5. *Just Dance 3* (Wii, 360) Ubisoft

6. *Dark Souls* (PS3, 360) Namco Bandai Games
7. *Madden NFL 12* (360, PS3, Wii, PS2, PSP) Electronic Arts
8. *Forza Motorsport 4* (360) Microsoft
9. *Gears of War 3* (360) Microsoft
10. *FIFA Soccer 12* (360, PS3, Wii, PSP, PS2, 3DS) Electronic Arts

F. Game Releases

1) **Alpha** - When all the features are in, but not all bugs are out.

2) **Beta**- Development team believes all the bugs are out and No new features except ones to eliminate huge problems.

3) **Release** - No new features and Everything is done.

G. The Future of Game Development

1) World of Mass Development (WMD)

New platform for games creation that allows Developers to submit ideas to an active gaming community, raise the funds needed to develop them, get continual feedback from community team members that can play work-in-progress builds, use the WMD Portal to promote their project, get help from other developers, and ultimately launch their game to an eagerly-awaiting audience. It allows players to browse available projects and join any they are interested in, download and play regular builds of the game, participate in discussion & polls, speak directly to the developers, and eventually earn money back based on their contribution when the game is released.^{[23][24]}



Project CARS is the first of those titles to use the WMD development system and represents the 'Ultimate Driver Journey'.^[25]

- **Franchise Mode** allows you to carve out a personalized career starting in the Karting world and then progressing on to whichever motorsport specialization you prefer including Rally, Touring Cars, Open-Wheel, GT, Le Mans, and many more!
- **Play CO-OP** with a friend as Driver/Co-Driver
- **Full Team Management** - Have a large number of friends? Create, manage, and compete together!
- Experience the excitement of *Pit Stops* like you've never seen before.

- *10+ Game Modes* covering every form of motorsport
- *Cloud-Based Social Network* - allows you to connect with friends, compare times & scores, compete and challenge each other, and share content
- *User-Generated Content* – Create your own liveries, decals, tuning setups, and even events! Then share them with the world – either for free, in-game credits or even real money!
- *Pushing To The Limits* – Advanced physics, lighting, and AI

2) *Low level and API-free programming*

The limiting factor on PC is the performance overhead of the 3D API (mainly DirectX) while on consoles, game developers can use low level code to process more triangles than on PC. More render calls allow more creativity freedom for game designers. The solution would be to have a low level access to PC graphics hardware (direct-to-metal programming). Either way, it looks as though DirectX's future as the primary gateway to PC graphics hardware is no longer 100 per cent assured, especially when it comes to cutting edge graphics. Maybe there's an opportunity for an API like OpenGL- thanks to extensions, hardware vendors could offer new OpenGL extensions to have low level access to the GPU.^[26]

3) *HTML5 is the Future of Social Game Development*

HTML5 is the future of social game development, especially on Smartphones because just about every phone and browser out there supports the web development platform. If a game is developed with HTML5 on one platform, developers can take that game almost anywhere that supports the language. The iPhone, Android, WebOS, Backberry 6.0, Bada and Nokia's Symbian and Meego, all support HTML5 apps. Even Windows Phone 7 will roll out support in the next year. The highlight would be WebGL, Canvas and WebSockets, which have given developers the opportunity to flaunt their creativity by manipulating images, creating 3D environments and providing real-time interaction.^[27]

Acknowledgment

I owe my deepest gratitude and humblest appreciation to Prof. Matthew P. Evett, Professor, Department of Computer Science, Eastern Michigan University, whose Learning Material on Game Programming helped me elucidate various doubts in the vast realm of Game Development & Designing and inspired me to the overall success of the undertaking.

I would also like to thank my Mother for her support, both financially & emotionally.

And above all, to the Almighty God, who never cease in loving us and for the continued guidance and protection.

References

- [1] Nokia Series 40 J2ME™ Game Usability Guidelines and Implementation Model. (*references*), *Version 1.0; June 23, 2003.*
- [2] XNA Game Studio - http://en.wikipedia.org/wiki/Microsoft_XNA#XNA_Game_Studio_Express
- [3] First Game - <http://en.wikipedia.org/wiki/OXO>
- [4] First Digital game - http://en.wikipedia.org/wiki/Tennis_for_Two
- [5] First Coloured Game - <http://en.wikipedia.org/wiki/Galaxian>
- [6] First – Most popular Game - <http://en.wikipedia.org/wiki/Pac-Man>
- [7] All time best Seller - http://en.wikipedia.org/wiki/Super_Mario_Bros._3
- [8] History of Myst (1993) - <http://myst5.com/mysthistory.html>
- [9] History of DirectX & API's - <http://en.wikipedia.org/wiki/DirectX#History>
- [10] Emergence of Playstation 1 - <http://en.wikipedia.org/wiki/PlayStation>
- [11] Ultra Weekend 6 - http://en.wikipedia.org/wiki/Ultra.Weekend_6
- [12] Nintendo 64 - http://en.wikipedia.org/wiki/Nintendo_64
- [13] Highest possible score for Pac-Man achieved - [http://en.wikipedia.org/wiki/Billy_Mitchell_\(electronic_sports_player\)](http://en.wikipedia.org/wiki/Billy_Mitchell_(electronic_sports_player))
- [14] Emergence of Playstation 2 - http://en.wikipedia.org/wiki/PlayStation_2
- [15] PC Games in 2003 - http://en.wikipedia.org/wiki/2003_in_video_gaming
- [16] Emergence of World of Warcraft - http://en.wikipedia.org/wiki/World_of_Warcraft
- [17] Emergence of Playstation 3 - http://en.wikipedia.org/wiki/PlayStation_3
- [18] The Art of Game Design, by Jesse Schell - http://books.google.co.in/books?id=Lp5xOYMjQKQC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false , p. 309-329.
- [19] Principles of Game Design - <http://ai.vancouver.wsu.edu/~wallaces/courses/slides/game-design/GameDesignPrinciples136.pdf>
- [20] The Art of Game Design, by Jesse Schell - http://books.google.co.in/books?id=Lp5xOYMjQKQC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false , p. 171-206.
- [21] Game Physics, By David H. Eberly, Ken Shoemak – http://books.google.co.in/books?id=a9SzFHPJ0mwC&printsec=frontcover&source=gbs_ge_summary_r&cad=0#v=onepage&q&f=false , p. 13-82
- [22] Top 10 Games of 2011 - <http://www.videogamesblogger.com/2011/11/15/top-10-best-selling-games-of-october-2011-usa.htm>
- [23] WMD - <http://www.wmdportal.com/>
- [24] WMD-<http://www.slightmadstudios.com/world-of-mass-development/>
- [25] Project CARS - <http://www.wmdportal.com/projects/cars/>
- [26] <http://www.bit-tech.net/hardware/graphics/2011/03/16/farewell-to-directx/1>
- [27] <http://blog.games.com/2010/10/14/html5-is-the-future-of-social-game-development-moblyng-ceo-cl/>
- [28] External References – Game Development for the PC, Xbox 360 and Windows Phone 7 – Learning XNA 3.0 – Aaron Reed – by O'Reilly
- [29] External References – Game Development for the PC, Xbox 360 and Windows Phone 7 – Learning XNA 4.0 – Aaron Reed – by O'Reilly
- [30] External References – 3D Graphics with XNA Game Studio 4.0- Sean James.

SESSION
VISUALIZATION AND MODELING

Chair(s)

TBA

Graphics on Web Platforms for Complex Systems Modelling and Simulation

T.H. McMullen, K.A. Hawick, V. Du Preez and B. Pearce

Computer Science, Institute for Information and Mathematical Sciences,

Massey University, North Shore 102-904, Auckland, New Zealand

{t.h.mcmullen, k.a.hawick}@massey.ac.nz, {dupreezvictor, brad.pearce.nz}@gmail.com

Tel: +64 9 414 0800 Fax: +64 9 441 8181

April 2012

ABSTRACT

Graphical interactive simulations are a powerful means of exploring properties of complex systems models and aiding a development of model parameter spaces. Platform independence for graphics intensive simulations – and especially those involving 3D rendering – has been a difficult goal to realise until recently. We describe use of WebGL, JavaScript and associated graphics rendering interfaces to implement a range of complex system simulations for a Web platform. We experimented with various shading and rendering approaches. We compare the performance obtained using this approach for various compute devices. We discuss performance and implementation issues and the implications for future platform independence of complex systems simulations code development.

KEY WORDS

WebGL; JavaScript; portable shading; platform independent graphics

1 Introduction

Interactive graphical rendering is an important tool in exploring the parameter space of complex system simulations. Such simulations include models in physics, chemistry, biology, sociology and finance. All these areas have in common a need to adjust the parameters in a model to develop an intuition and understanding of what they mean and how they affect system behaviour [5].

Platform independent computing is a desirable goal for many software projects. Much work has been done using virtual machines and related approaches to platform independence [1]. The advent of widely available tablet computers has renewed interest in platform independence, particularly for high quality graphical rendering programs that can run with a (portable) web browser environment on a range of platforms. Such devices open up a range of potential applications that can use interactive graphics [12, 16].

New and emergent approaches to user interface design for such devices are also an exciting arena for interactive simulations [11].

Computational steering is the term introduced by both Smarr [14] and Fox [3], to describe the interactive exploration of parameter space – often prior to launching a series of non-interactive compute jobs to gather statistics from numerical experiments. The early stages of interacting with the parameter space can save a great deal of time and computational effort by identifying where to look and where to direct study.

High performance graphics is a huge aid to computational steering. High quality rendering of a complex system greatly aids in understanding the model. Graphical rendering of complex systems - particularly in 3D involves non-trivial amounts of computation however. Until relatively recently it has really only been practical to work with a high performance graphical rendering or shading system such as OpenGL or its proprietary equivalent. Although such code can sometimes be ported – with a greater or lesser ease depending upon the system involved – to different hardware and operating systems platforms, it has been difficult to realise the goal of complete platform independence.

Most recent work has focused on development of custom “Apps” for individual tablets or mobile phones [2]. Tools such as WebGL, JavaScript and associated software systems do open up a practical possibility of developing high graphical quality and reasonably high performance interactive rendering simulations that can be run through a web browser. This is obviously attractive since these will be supported on any platform that supports the appropriate web browser standard.

In this paper we develop a set of six computational simulation models that also have significant graphical rendering requirements. We implement these models using platform independent software technologies and investigate

their relative performance on web browser environments running on desktop PCs; Apple iPad and Samsung Galaxy tablet devices.

Our article is structured as follows: In Section 2 we describe some of the typical graphical simulation models that we might wish to port to run on various platforms and relate their key features. In Section 3 we present details of our implementation experiments. We present some graphical and performance results in Section 4 and a discussion of their implications in Section 5 in which we also summarise our findings and give some areas for future work.

2 Simulation Models

We summarise the main features of the simulation models we deployed, giving references where more details can be obtained.

Several models were used to test web platforms as a method for modeling complex systems and simulations. Firstly several 2D models were created, including Conway's Game of Life [4], Diffusion Limited Aggregation (DLA) [18], and an Ising model [9, 10] among others. These 2D models shared similar behaviour, and as such the methods for one can be used across a range of them. To test the use of 3D graphics on a web platform a 3D Ising model was constructed, with two different implementations. These have similar methods for the creation and manipulation of the simulation, but differ in how it is displayed.

The first model implemented was Conway's Game of Life. This was as it is simple to implement with a well defined rule set. To start with two 2D lattices are created, and one initialized with a random or predefined state. From this state the program will work through that lattice and write the updated value for each cell onto the other lattice. The rules for updating a location on the lattice is that so the program will look at the eight closest neighbors, and count the number which are alive. Based on the amount of alive neighbors the cell will become dead or alive or stay in its current state. This state is represented by the colour of the cell, with white been dead while red if alive. Once the lattice has been updated, the program will then use this data to update the other lattice. This switching occurs in many of the simulations, where the lattice is worked though in a liner fashion, otherwise the results created are not accurate. This process of working on one lattice and writing to another then switching between them continues throughout the duration of the simulation.

The Game of Death [13] is another cellular automaton, similar to that of the game of life, but introduces a new third state. This addition state is called the zombie state, which adds a new phase into the transition between life and death. This model was implemented using the game of life simulation as a template to be based off of. In the

game of death the switch between life and death still have the same rules applied as they do within the game of life, but now once a cell is dead, if it does not come back alive it will become a zombie. From the zombie state a cell is only allowed to become alive if two of its neighbors are alive. As the transition between all states is still only based off of the number of alive neighbors, this simplifies the rules and creation of the simulation.

Another simulation implemented in the 2D is a Diffusion Limited Aggregation model. With this simulation only one lattice is used, and is initialized with one cell in the middle. Another cell (Walker) is then randomly placed on the lattice, which will move about in a random direction till it touches the cell in the center. Once the walker has touched the center cell it will join and start creating a cluster, and a new walker is spawned. This will continue till the cluster reaches the boundaries of the lattice.

A simple 2D Ising model was implemented, once again only using a single lattice. This was initialized by giving each cell within the lattice a random state, been one or zero. The simulation would then be passed a value for the temperature, as this would effect the over all energy of the system. Once the initialization is completed the simulation will pick a spot at random and check to see the state of the neighbors, based on the state of the surrounding cells the selected cell will flip or stay the same based on which would result in the lowest energy of the system. If a flip did not occur then there is a chance based on the temperature of the system that the cell would flip, with a high temperature there is more energy so it is more likely a flip will occur.

The Invasion percolation model [17] was also created, this involves having a walker start at one location on a 2D grid, and move its way to another location, with materials blocking the way. The walker will start moving in a semi-random manner, with a weight added to give it some form of direction. The simulation will create a barrier between where the walker starts and the end point. The created barrier has a small gap in it, along with a small chance of allowing the walker to pass through it.

The last 2D simulation implemented was a Sznajd opinion model [15]. This starts on a single 2D grid, which has each cell within it initialized to a random opinion. The simulation will select a point on the grid randomly and then also select on of its surrounding neighbors. If both cells share the same opinion, then all cell connected to those two will change to match them.

To test how these simulations ran in 3D two implementations of the Ising model were created, as this has the most specific rule set for use within a 3D environment. Though the drawing method differs between each one, the computation used is the same. This lead to the only deference between each one been that off how each is visualized, this is important to factor in as visualizing the data within a

simulation can be the most intensive part if not taken care of correctly.

Regardless of the dimension, be it 2D or 3D all these simulations rely heavily on the use of random numbers, along with a method used to visualize the data, in this case WebGL or HTML5 canvas. These models all come with their own problems in regards to how the code is interpreted on each platform.

3 Implementation Method

To create these models several things needed to be decided first, A language was needed that would be supported on multiple platforms, and be able to visualize the different dimensions, for this JavaScript was used, as it is needed to be able to work with WebGL. This is different from traditional rendering approaches for 3D simulations using OpenGL [6, 8]. To create each 2D simulation an HTML5 canvas was used as the developer is able to have direct access to each pixel. To compared the was in which to optimize the code for it to run on multiple devices was needed, as this was to designed to be as platform independent as possible. Another area looked at was a way in which to optimize the rendering of 3D simulations using different methods in WebGL.

The simplest way to create platform independent simulations is to build them upon web based technologies. The most appropriate of which was JavaScript, as it is well supported within all modern web browsers, and would work on a range of different devices as a result. Another upside of using JavaScript is how it allows for the use of WebGL, which is a JavaScript API for rendering 3D graphics. By using these technologies we are able to write directly to an HTML5 canvas to create a 2D image or use WebGL to create 3D images within supporting browsers. By using WebGL we are able to take advantage of the graphics processing abilities which are normally reserved for use with native applications, although it is not fully supported on all devices currently.

For creating the 2D models the HTML5 canvas is used, as this allows for direct access to each pixels data. By storing the data within the image itself there is no longer a need to transfer data from buffer to another. This allows for a increase in performance as it removes unnecessary steps from the program, although now to access the data the locations in which the data is stored has been changed as each pixel takes up 4 spots in the array, one red, green, blue and alpha value are given for each pixel, and this leads to a slight decrease in performance.

For creating the 3D implementation of the Ising model, two different methods were created to allow for the performance of different rendering methods to be compared. The first one to be implemented was one which rendered the cubes on the other edge of the simulation, and when

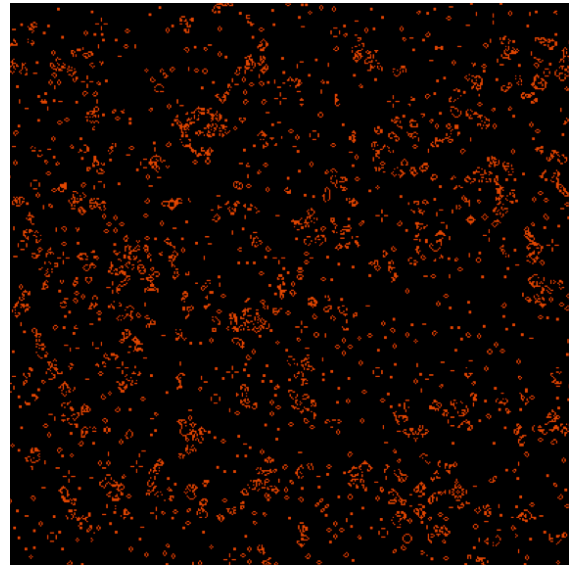


Figure 1: A 200x200 Game Of Life grid, where red cells are alive.

expanded all the inside cubes will also be rendered. This allows for the creation of a lattice of any give size in the x y and z dimension, along with giving the ability to see the inner workings of the simulation when it is expanded. To create this WebGL was used to draw a cubes to represent a location on the simulation, with it changing color based on the state. By doing this essentially one cube was been draw over and over again, with only a change in color.

To create the second 3D Ising model, a single cube was created, and used textures to display the state of the Ising model. To do this the values of the outer points on the lattice are written to a data array, which is arranged to be the same as image data, with the four locations for each pixel. This image is then passed through WebGL and creates a texture to be applied to the cube. To apply a texture the cubes faces had to be labelled so the correct image would be displayed, this was done by the use of a vertex and fragment shader. By creating the model with this method the size of the simulation on the x,y and z axis all had to be of a power of two, as WebGL only supports textures of that size.

4 Results

The resulting web applications from this are a range of 2D simulations which can be run on a range of devices, along with 3D models which are able to take advantage of the processing power of modern GPUs.

Figure 1 shows the result of the game of life simulation running, which is able to be run on a range of devices, with no difference in presentation. The screen can be enlarged on a tablet, such as the iPad to make the data more visible, though this in turn does effect the speed in which the simu-

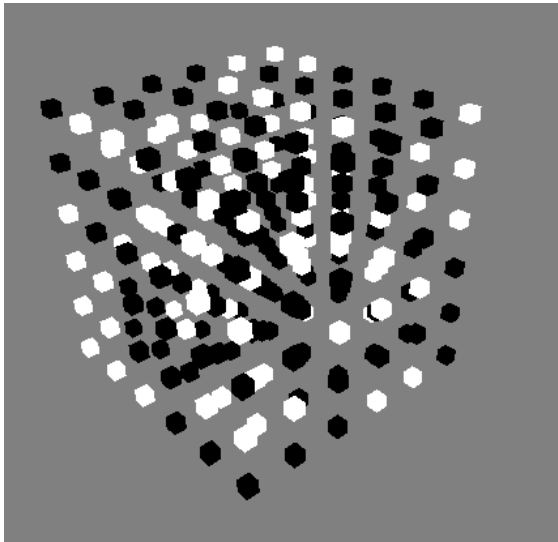


Figure 2: Expanded view of a 3D Ising model with a size of 6^3 .

lation is run. This simulation seemed to run slightly faster than the Game of Death, and when compared to other simulations provided a solid base to compare to.

The graphs show the time taken for each device to complete the simulations in various grid sizes. What is visible in all of them is how each one seems to follow the pattern of the DLA, Invasion and Sznajd simulations been the fastest while the Ising was the slowest in all cases. Though is is note worthy that the Invasion model did take considerably longer on when ran on the Samsung Galaxy.

Figure 2 displays one of the possible views of the 3D Ising model created using WebGL. This method involved drawing each point within the Ising model with a cube. When the model is not expanded the model will only draw the outer points, as the inner ones are not needed and would slow down the simulation. While when expanded all the points within the system are visible, allowing the user to see what is happening within the simulation as it runs, and is not restricted to only viewing the outer workings.

Figure 3 shows the workings of the simulation running but by using textures mapped over a single cube, replacing the need to render multiple cubes. The method for producing this kind of rendering is shown in algorithm 1, although it has been greatly simplified the core mechanics are present. The process begins by creating the images for each side of the cube based on the simulation data, this data is then placed into an array and arranged into that of RGBA. Once the image has been created it is then mapped to the appropriate edge of the cube. These textures are drawn to the cube when the drawing function is called.

The performance difference between the two methods of rendering the 3D Ising model become clear when viewing figure 7, it shows that at the lower sizes that both methods have a similar time for 1000 iterations. At higher grid

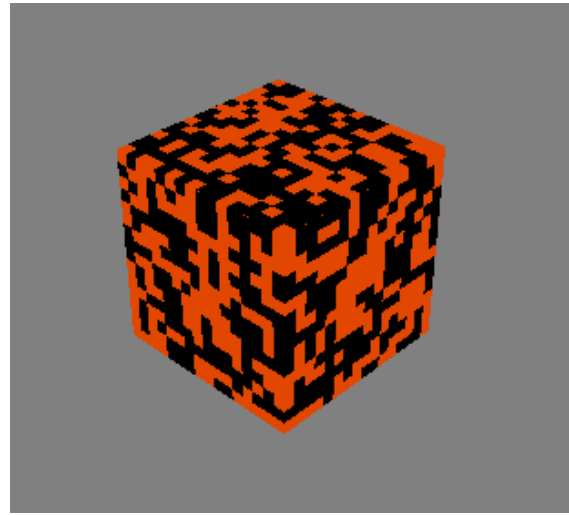


Figure 3: A 3D cube, rendered with textures to visualize the outer working of an Ising model.

Algorithm 1 The process of creating the texture and running the simulation while keeping that texture up to date.

```

declare textures
bind textures to gl.Texture_2d
pass image data to textures
while running do
    Draw Scene
    Update Simulation
    Update Textures
end while

```

sizes the texturing method remains at a constant rate, while to produce each cube takes much longer. There are two down sides to the Texturing method, firstly the images dimensions must be of a power of 2, thus the tests been of sizes 8, 16, 32 and 64 cubed. Secondly using the textured method it is not possible to view the inner workings of the simulation.

Figure 4 shows some performance data obtained using the Chrome browser on a desktop PC. The Ising model requires floating point calculations and has the poorest performance rating. The Invasion model is the computationally cheapest model to run, with DLA and Sznajd closely matching. This is largely due to domination of the graphical rendering.

Figure 5 shows corresponding times for the same simulations run on an iPad2. The curves have the same profile, but are scaled back by a factor of only around 0.75, reflecting the lower processor power of the tablet device.

Similarly Figure 6 shows performance times obtained for the same models running on a Fire Fox browser on a Samsung Galaxy tab 10.1, which has very similar performance profile to the iPad.

Figure 7 demonstrates the very considerably enhanced

performance of the Ising model rendering when texturing of the 6 polygonal surfaces of the simulated cube is performed, rather than forcing a rendering (and associated hidden surface removal) individually for each cell in the model. When texturing is done, the performance profile flattens showing that it is then dominated by computational performance of the model simulation rather than by the graphical rendering.

5 Discussion & Conclusions

We have described a set of complex systems simulation models that can be simulated in both two and three dimensions, and which have interactive parameters that the user can usefully adjust to develop an intuition on their meaning. We have used these as specific examples to explore their graphical rendering requirements including 3D shading. We have developed and experimented with platform independent versions of these graphical simulations using WebGL and associated software systems. The platform independent implementation of the models run smoothly on both desktop and tablet based platforms, albeit with different performance features.

The performance data suggests that web platforms using systems such as WebGL/JavaScript are of sufficient computational performance capability that it is quite feasible and practical to develop simulation demonstrations that run on them, even for quite complex three dimensional models that have sophisticated graphical rendering requirements including shading. We have demonstrated that a few techniques such as texturing the surfaces of a cube manually rather than forcing hidden surface removal of a large number of independently invoked polygons does speed the rendering up. In fact for the sophisticated models we have used here, this approach was necessary.

There is scope for considerable further work in experimenting with other three dimensional models. Our immediate plans are to develop a software library incorporating the techniques and best practices uncovered during this prototyping work, and incorporating domain-specific code generation techniques [7]. We have worked with various web browsers on desktop computer systems. We anticipate continued increases in WebGL platform performance on various devices and especially on tablet computer systems in the future, to the point where they will be viable platforms for this sort of simulation demonstrations too, without the need to develop customised and platform specific Apps.

References

- [1] Bowman, D.A., Hodges, L.F., Bolter, J.: The virtual venue: User-computer interaction in information-rich virtual environments. *Presence* 7(5), 478–493 (1998)
- [2] Charland, A., LeRoux, B.: Mobile application development: Web vs native. *Communications of the ACM* 54(5), 49–53 (May 2011)
- [3] Fox, G.C., Williams, R.D., Messina, P.C.: *Parallel Computing Works!* Morgan Kaufmann Publishers, Inc. (1994), ISBN 1-55860-253-4
- [4] Gardner, M.: *Mathematical Games: The fantastic combinations of John Conway's new solitaire game "Life"*. *Scientific American* 223, 120–123 (October 1970)
- [5] Hawick, K.A.: Simulated worlds: Educating students in doing science with computers. In: *Proc. WORLDCOMP 2009 International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS 09)*, Las Vegas, USA. No. CSTN-080 (July 2009)
- [6] Hawick, K.: 3d visualisation of simulation model voxel hyperbricks and the cubes program. Tech. Rep. CSTN-082, Computer Science, Massey University (October 2010)
- [7] Hawick, K.: Engineering domain-specific languages for computational simulations of complex systems. Tech. Rep. CSTN-123, Massey University (June 2011)
- [8] Hawick, K., Playne, D.: Spectral and real-space solid representations and visualisation of real and complex field equations. Tech. Rep. CSTN-101, Computer Science, Massey University (August 2009)
- [9] Ising, E.: Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift fuer Physik* 31, 253?258 (1925)
- [10] Leist, A., Playne, D., Hawick, K.: Interactive visualisation of spins and clusters in regular and small-world Ising models with CUDA on GPUs. *Journal of Computational Science* 1, 33–40 (2010), www.elsevier.com/locate/jocs
- [11] Meixner, G., Paterno, F., Vanderdonck, J.: Past, present, and future of model-based user interface development. *i-com* 3, 2–11 (2011)
- [12] Narayan, M.A., Chen, J., Perez-Quinones, M.A.: Usability of tablet pc as a remote control device for biomedical data visualization applications. Tech. Rep. TR04-26, Virginia Tech (2004), <http://eprints.cs.vt.edu/archive/00000703/01/TR04-26-Jian-Michael.pdf>
- [13] Resnick, M., Silverman, B.: Exploring emergence: The brain rules. <http://llk.media.mit.edu/projects/emergence/mutants.html> (February 1996), <http://llk.media.mit.edu/projects/emergence/mutants.html>, MIT Media, Laboratory, Lifelong Kindergarten Group
- [14] Smarr, L., Catlett, C.E.: Metacomputing. *Communications of the ACM* 35(6), 44–52 (June 1992)
- [15] Sznajd-Weron, K., Sznajd-Weron, J.: Opinion evolution in closed community. *Int. J. Modern Physics C* 11(6), 1157–1165 (2000)
- [16] Tesoriero, R., Montero, F., Lozano, M.D., Gallud, J.A.: Hci design patterns for pda running space structured applications. In: *Proc. 12th Int. Conf. on Human-Computer Interaction: Interaction Design and Usability* (2007)
- [17] Wilkinson, D., Willemsen, J.F.: Invasion Percolation: a new form of percolation theory. *J.Phys.A.* 16, 3365–3376 (1983)
- [18] Witten, T.A., Sander, L.M.: Diffusion Limited Aggregation, a Kinetic critical Phenomenon. *Phys.Rev.Lett.* 47(19), 1400–1403 (Nov 1981)

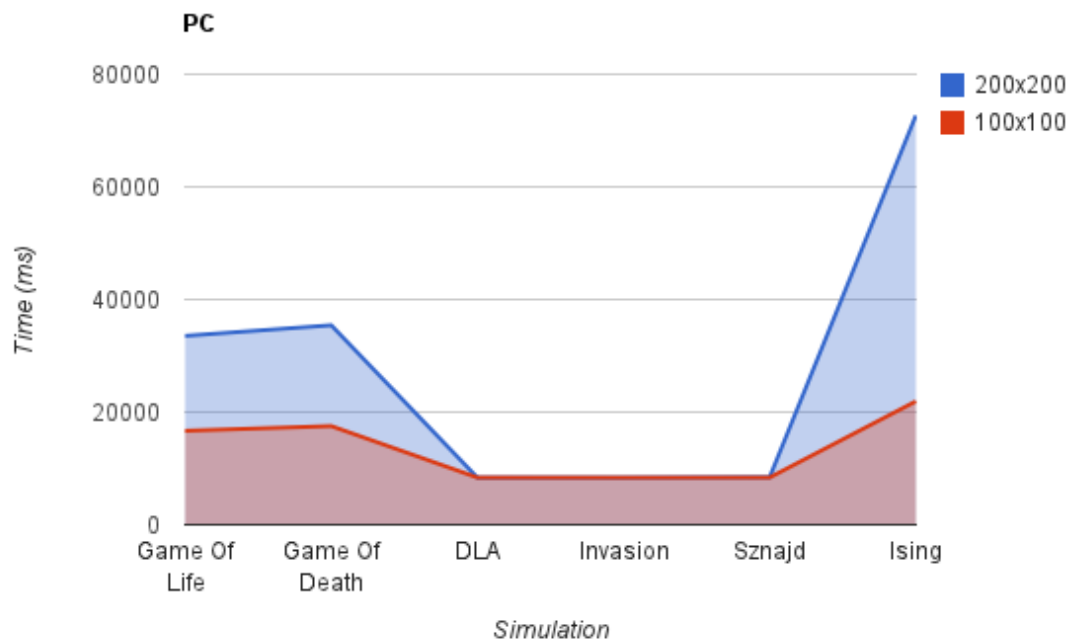


Figure 4: The average amount of time taken to run each simulation in milliseconds on a PC, using the Chrome browser Blue is when the simulation is run in a 200x200 size grid, while red is for a 100x100 size grid

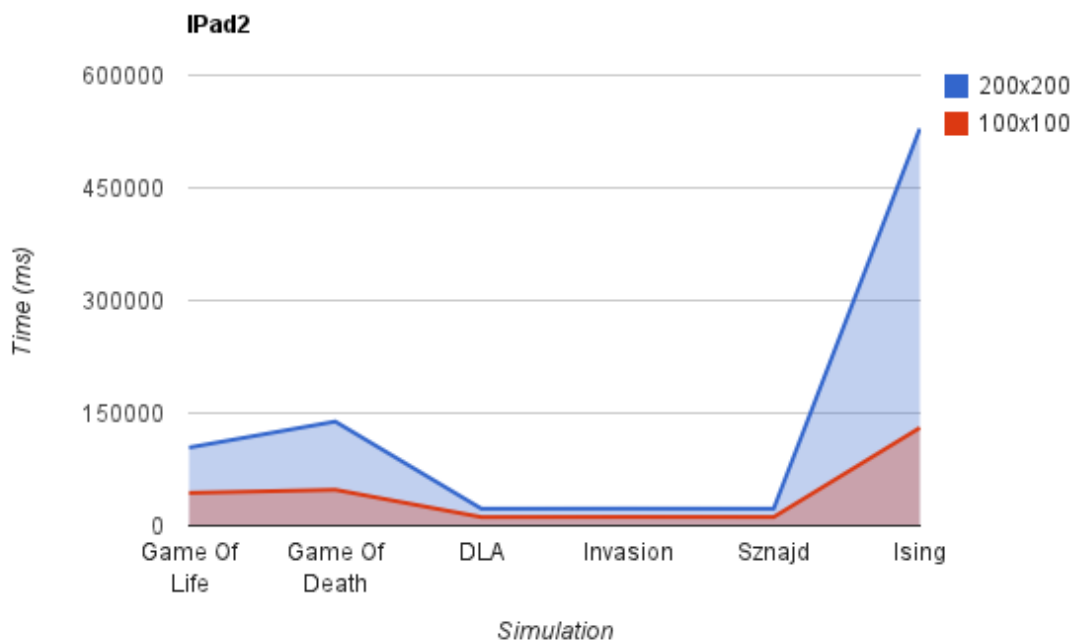


Figure 5: The average amount of time taken to run each simulation in milliseconds on a iPad 2, using the default browser

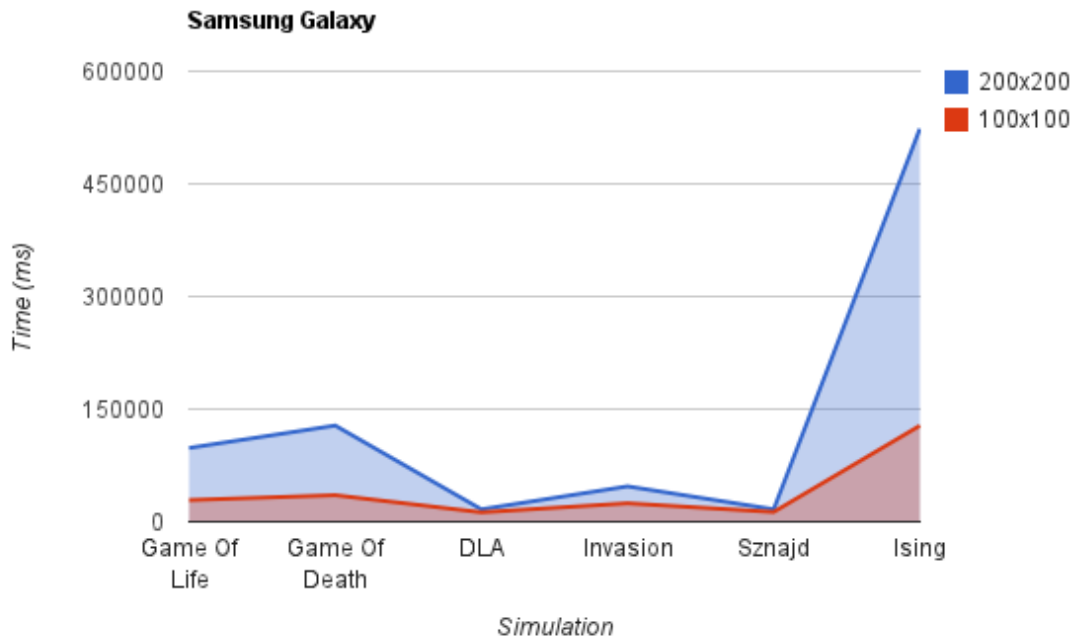


Figure 6: The average amount of time taken to run each simulation in milliseconds on a Samsung Galaxy 10.1, using the Fire Fox browser.

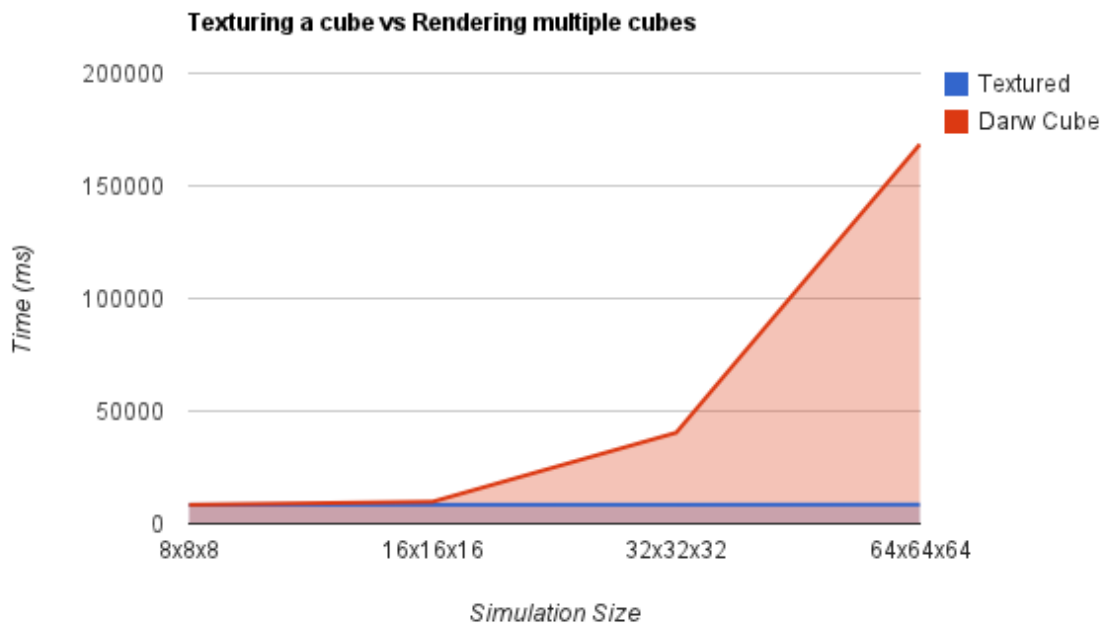


Figure 7: The time taken to complete 1000 steps of an Ising simulation, comparing the texturing methods(Blue) and rendering a cube for each of the outer points(red).

Simplification of mass spring models

M. Kot¹, H. Nagahashi¹, and K. Gracki²

¹Imaging Science and Engineering Laboratory, Tokyo Institute of Technology,
4259 Nagatsuta-cho, Midori-ku, Yokohama, Kanagawa 226-8503, Japan

²Institute of Computer Science, Warsaw University of Technology, Nowowiejska 15/19, 00-665 Warsaw, Poland

Abstract—*In this paper, we present an approach of simplifying mass spring models used for the representation of deformable objects. First we will briefly discuss the problem of constructing mass spring model (MSM) with well defined physical properties, next we will focus on the problem of generating lower resolution representations based on the reference model. Tested algorithms are based on classical LOD's (level of detail's) verticle merging approach. In case of LOD for physical interactions, a few new important aspects of the simplification emerge; when simplifying three dimensional objects we have to deal not only with the surface of an object, but also with its whole volume. Additional difficulty also arises when objects are composed of non-homogenous materials.*

Keywords: Level of detail, mass spring model, soft body deformation.

1. Introduction

Non rigid bodies are becoming increasingly popular in computer graphics and virtual reality applications. Computational power of processing units will soon allow for a wide use of shape changing objects in games and animations. Incorporating such objects in a simulated world increases its realism and allows for much richer viewing and interaction experiences.

Among many methods of modeling soft bodies [1], the ones based on physical laws offer the best automation and flexibility, at the same time keeping quality of an animation at a high level. Most popular techniques include modeling of deformable objects using FEM (*finite element method*) or *mass spring models* (MSM). In majority of physically based approaches, the representation of a deformable body is volumetric, therefore its complexity is relatively big. When presenting large scenes with many such objects, it is only natural to use some model reduction mechanisms, similarly as it is being done with meshes for visual representation (various *level of detail* – LOD techniques).

In this article we take under consideration mass spring models. MSMs are robust and computationally efficient, however they do not have a clear description in terms of their physical properties. Therefore, in majority of applications where physical correctness is either required or means of

controlling errors in a clear way are needed, FEM is being used. Consequently the attention of researchers dealing with model simplification of elastic objects has been focused on FEM based representations [3] [4].

Estimation of physical properties of MSM is less straightforward, but possible. We will briefly discuss this topic in section 3. Next, we will analyze a few techniques of generating lower resolution representations for a given model with well defined physical properties. The problem is in many aspects analogous to LOD generation for meshes [8]; we will try to adapt known simplification methods, to work with volumetric representations, and preserve not only shape, but also physical properties of modeled objects.

2. Related work

There exist many techniques of reducing computational time of a simulation of deformable objects. In general, improved efficiency can be achieved either by changing the simulation technique, which includes time integration algorithms and time step control (usually tightly coupled), or by reducing complexity of models. In this article we will focus on model simplification problem; popular techniques include adaptive refinement, model reduction by limiting the space of possible deformations [5], or geometric approaches such as shape matching [6] (which simplifies both – models and simulation technique). Domain embedding is very popular as well. Wojtan and Turk [2] use FEM representation with embedded high resolution surface mesh. Nesme et al. [4] use an embedding technique based on octree voxelization, that takes into account non-homogeneity of modeled objects. The problem of model simplification closest to ours has been analyzed by Kharevych et al. [3] for FEM representations. They formulate an elasticity on coarse resolutions by deriving effective elasticity tensors. The advantage of their method is an ability to handle heterogenous objects.

We aim to implement a method for the simplification of mass spring models. MSMs are very popular, in cloth modeling, however relatively few researchers are considering real volumetric models for 3D objects. The disadvantage of MSMs is a difficulty of defining their physical properties. There were several attempts to identify MSM parameters of particular materials, however most of them deal with very specific cases of materials and require additional modifications to mass spring systems such as adding custom

constraints or “angular springs”. The basic and general case that we would like to analyze is a MSM composed of mass points and linear springs only. The description of such MSM, was given by Lloyd et al. [10], however their equations are limited to models based on tetrahedral meshes. They obtain spring parameters by comparing MSM to a corresponding FEM-based model. Explicit equation is given for spring coefficients for regular tetrahedra. The description of MSM with mass points distributed on a cubic lattice was given by Ladd and Kinney [9]. They derive spring constants from linear elasticity equations. Additionally, more physically-oriented approach to the different types of MSMs as well as techniques of estimating their physical properties can be found in the review article by Ostoja-Starzewski [7].

One more problematic issue with model creation and simplification, concerns the judgement of the quality of a model and the quality of resulting animation. It is difficult to decide which behavioral artifacts are acceptable and will most likely not be noticed. Yeh et al. [11] give a good overview on this topic and explore a few approaches of identifying error tolerance of physical simulation. They propose the use of maximum percentage energy difference to evaluate the perceptual quality of the simulation.

3. Properties of mass spring models

Mass spring model represents an elastic material by means of discretization. It is important to note that in such representation there are two things that are being discretized. First is a mass distribution which in real materials is usually treated as continuous, the second, represented by springs, characterizes the interactions between the material points and thus defines elastic properties of a physical body. Both of these aspects are important and will affect the realistic appearance of simulated objects. If we increase the number of mass points representing an object, its resolution will become higher resulting in more detailed deformations. On the other hand, increasing the number of springs attached to each mass point allows one to represent the local elastic properties of the material in a more accurate way.

One of the most natural and commonly used MSM types is based on the distribution of mass points on a cubic lattice. We will refer to it as a *cubic lattice MSM* or *cubic MSM*. Typically, spring creation rules will assume closest and second closest neighbors to be connected by springs with the same spring constant k . In such case, elastic properties of resulting objects can be obtained from linear elasticity equations [9] and are given by:

$$\begin{aligned} E &= 2.5 \frac{k}{a}, \\ \nu &= 0.25, \end{aligned} \quad (1)$$

where E is a Young's modulus, ν Poisson's ratio, and a is the length of an elementary cube in the lattice. *Cubic lattice MSMs* do in fact correctly represent homogenous isotropic

material, which we have verified in our previous work [13]. We will use them as a high resolution reference models for the simplification algorithm.

More general description of MSMs can be obtained from its statistical properties¹. The equation

$$E = \frac{1}{12} \frac{N \bar{S} k}{V} \bar{R}^2 \quad (2)$$

gives the value of Young's modulus for a volume V of a material represented by MSM with N nodes connected by springs with spring coefficient k ; average number of springs attached to a node is equal to \bar{S} , and the average length of a spring is equal to \bar{R} .

Note that MSMs with *random* node placement are very flexible. They allow for arbitrary spring connection rules as well as node positioning. Equation (2) is approximate and the more homogenous and isotropic the considered volume of the material is, the better this approximation is expected to work. Our tests show, that it does in fact work very well for models having 14 or more springs attached to one node and reasonably uniform mass point distributions. It can be applied to non-homogenous objects as well giving averaged properties over considered volumes.

Both *cubic lattice MSM* and *random MSM* allow for constructing objects composed of isotropic materials with *Poisson's ratio* equal to 0.25, which cannot be changed. It is the limitation of models based on central forces only. Fortunately this parameter is of lesser importance, when considering applications such as games or animations. Very realistic models can be created just with appropriately chosen *Young's modulus*.

4. Model simplification

When creating an animation, very precise and detailed models are not always needed. In some cases it is desirable to generate a simplified version of an object, or at very least one that is not over-detailed. The level of detail in the description sets the scale of physical phenomena object can undergo without hurting accuracy of the simulation. If the level of detail is reduced greatly, only very simple deformations will be possible.

4.1 Global resolution scaling

The simplest approach of constructing lower resolution model of an object is to fill its volume with desired number of mass points – either based on *cubic lattice* distribution, or the *random* one. An example is shown in Fig. 1. Spring coefficients should be chosen according to equations (1) or (2). This approach however requires a prior knowledge

¹Detailed explanation is contained in the article that is currently being under review

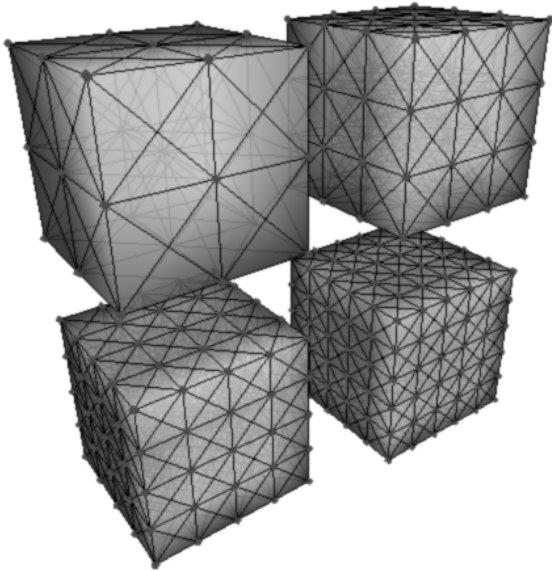


Figure 1: A cube represented by *cubic lattice MSM* with four different resolutions. Number of nodes: 27, 64, 125, 216.

about object's properties; it is in general analogous to global simplification methods of meshes with all their advantages and drawbacks. Its most noticeable characteristic is the fact, that different representations of the object generated this way are in principle not related to each other directly, which makes the problem of defining a clear correspondence between such representations more challenging. It is best applicable to homogenous models with simple geometry. In case of complex objects constructed of heterogenous materials, local simplification techniques are more suitable, because they allow for a better control of local features.

4.2 Simplification by node merging

A class of popular LOD generation algorithms, are based on local simplification operators that are capable of reducing details in a given area of an object. Such operators would be applied throughout the mesh, or the volume and result in its simplification. Methods based on local simplification operators allow to achieve better consistency between models with different resolutions, as more simple representations are based on high detailed ones. In classical LOD for meshes, local simplification operators include vertex removal, edge collapse, triangle collapse or vertex clustering. The basic ideas behind the last three of mentioned techniques are very similar – multiple primitives are being merged into one. We adapt this concept for 3D mass spring systems.

The core of analyzed approach is the procedure of merging multiple nodes of the mass spring system into one. The conditions of merging and the problem of choosing which nodes should be merged will be discussed in the next section. Currently we will focus on the merging step itself. Our goal

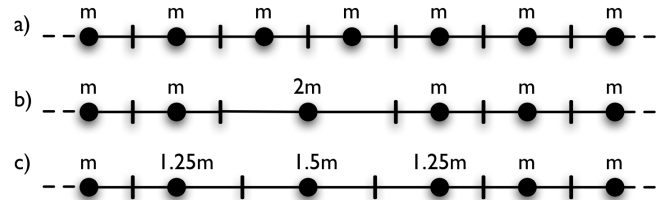


Figure 2: 1D example of mass discretization: a) uniform point distribution, b) two nodes merged creating $2m$ mass; volume represented by a mass point is proportional to the mass, c) volume representation based on closest proximity.

is to replace two or more mass points with a new one. The merging algorithm performs the following procedure:

- Set the mass of the new point equal to the sum of masses of old points.
- Set the position to the center of mass of old points.
- If old node has been connected by a spring with any other node, the new one will be connected as well (all old springs are being reattached to the new node). Each spring affected by this procedure changes its coefficient k proportionally to the inverse of its length L (following the rule of series $\frac{1}{k_{eff}} = \frac{1}{k_A} + \frac{1}{k_B}$):

$$\frac{k}{k_0} = \frac{L_0}{L}, \quad (3)$$

where k and L are the new values, k_0 , L_0 – old ones.

- Parallel spring connections are replaced by a single spring with appropriate k (by the rule of parallel connections, $k_{eff} = k_A + k_B$).
- All damping coefficients change proportionally to the changes of k ([12]).

In this approach springs between nodes that undergo merging are being discarded, therefore local properties of the material should not depend on singular springs, but should be reflected by coefficients of many springs. Additionally, we assume that the volume represented by a point is proportional to its mass. Alternatively, closest proximity representation can be used, where each point of a continuous object is represented by the closest node from a discretized model. In such case masses will be distributed also between neighboring nodes, that did not directly take part in the merging process. The difference between two approaches is shown in the Fig. 2. Both are equivalent in terms of static properties of an object, however choosing one over another will affect the dynamics slightly. In our tests we use the first approach because of its simplicity. Note that the simplification procedure changes the moment of inertia of the object slightly. This effect is reduced when the simplification is applied uniformly throughout the whole object.

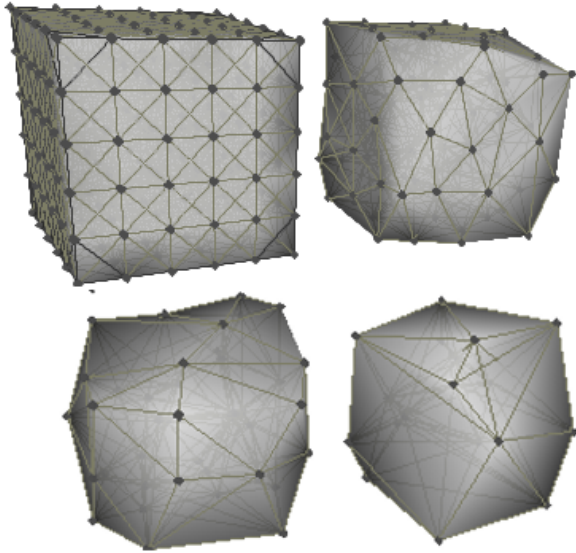


Figure 3: Standard node merging applied three times to the model of a cube. Number of nodes: 216, 95, 41, 18.

5. Node grouping for the simplification

In our tests, we take into consideration a few variations of *node merging* approach. First one is a simple merging of closest neighbors. More sophisticated techniques take into account heterogeneity of the object as well as try to preserve the border of the object. In all algorithms we merge only vertices connected by springs, which eliminates the potential problem of merging two mechanically separate parts of the object.

5.1 Standard merging

In this approach, groups of 2 or 3 mass points are chosen as candidates for merging based on geometrical proximity. Examples of LOD models of a cube generated by this technique are shown on the Fig. 3. The disadvantage of this method is the fact, that it does not take into account heterogeneity. When composites of different materials are simplified, high gradients of mass density or stiffness in the object are being smoothed out by this procedure. Also, the shape of the object may change dramatically which for many models is not acceptable.

5.2 Priority merging

In order to preserve heterogeneity of the object, we introduce a priority function into the merging procedure. Homogenous areas are simplified first, which results in a better preservation of physical properties of more complex objects. Improvement of the quality of LODs generated with this algorithm is noticeable especially in case of multiple simplifications of the same model. The priority function we use computes the gradient of *Young's modulus* (estimated with eq. (2)) as well as the gradient of mass density inside

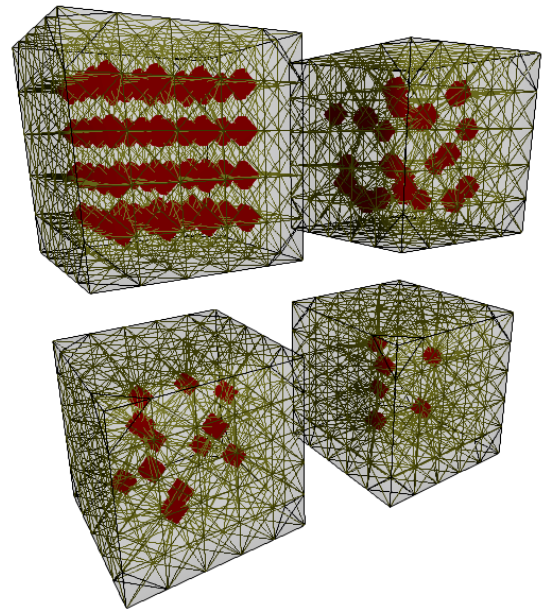


Figure 4: Node merging with border preservation applied three times to the model of a cube. Number of nodes: 216, 180, 164, 158. Nodes lying inside are shown in bold red.

the object. Areas with small gradients are given bigger priorities for merging.

5.3 Merging with border preservation

It is a modification of node merging algorithms that allows for minimization of object shape changes caused by simplification. We distinguish two types of border preservation.

Strong preservation

Prevents the points that lie on the border, from being merged. This results in no simplifications of outer areas of the object, therefore shape of original model doesn't change at all (Fig. 4). In many situations this fact will be crucial for object-object interaction, since border is the area of first contact; at the same time it imposes very serious limitations on simplification algorithms. For small objects, the area of the surface is relatively big compared to the volume, and if we forbid the surface regions to be simplified, there is not much left for the algorithm to work with (i.e. cube modeled by 5x5x5 lattice points consists of 125 nodes, 98 of which lie on the surface). It is important to note, that if the forces involved in an interaction are big, the shape of the object will not play a dominant role, therefore such strict border preservation method may not improve the overall behavior very much.

Weak preservation

It is another, less strict approach of shape preservation that imposes the following rule on node merging operator: nodes

that lie on the border can be merged only with each other. For any border node, if the best candidate for merging is a node that lies inside the object, it won't be merged at all (algorithm won't try to find another candidate, that may be further away, but satisfies the condition of being on the border as well). This prevents points that lie inside the object of becoming border points as the result of any merging operation. Narrow and high surface curvature areas are being preserved, whereas big and flat ones are being simplified.

6. Tests

In this section, we use the following notation:

- *LOD-n*: n-th level of simplification (each simplification reduces the number of nodes and springs by about 50%)
- *LOD-h-n*: LOD simplified with *priority merging*
- *LOD-b-n*: LOD simplified with *weak border preservation*.

Combinations of above are possible.

We do not show tests for LODs generated with *strong border preservation*. Because tested objects are relatively small, it imposes a limit on the number of nodes and springs that can undergo simplification, within which all the tested algorithms perform very well.

In previous sections, on all the figures, we show a cube model that undergoes simplification procedure. While it serves the purpose of illustrating how our algorithms work, it is not the best example for actual testing. Even though it is very tempting to perform the simplest tests of the behavior of different LODs such as a single bounce of a cube or sphere, one has to keep in mind, that such scenarios resemble an experiment of "throwing a dice". Slightest variations in shape (contact surface), internal mass distribution, or even a change of the simulation method will cause the object to change its trajectory significantly. The cube (or any other primitive) will start rotating, which reduces the amount of energy available for ascending making all comparisons of spacial positions between different LOD models meaningless. In such situations there still exist ways of estimating the quality of the simplification indirectly, such as a measurement of evolution of the object's energy, however they can be misleading, as different trajectories will cause the energy to dissipate in a different speed when the friction is present. Instead, in order to test the performance of generated LODs, we use moderately complex scenarios with heterogenous objects and observe their movement. We propose two tests in which the behavior of the object is driven by two different physical phenomena.

6.1 Scenario 1 – energy dissipation

In the first scenario a deformable spherical object with dense core and soft shell is being thrown horizontally just above a flat surface. It falls down and after a small bounce

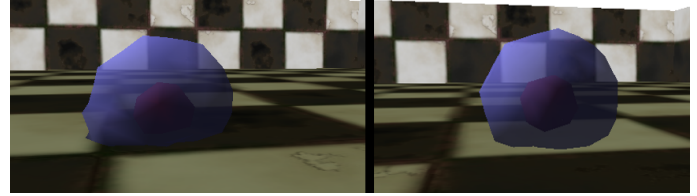


Figure 5: Energy dissipation test. Left: object falls on the ground. Right: object is rolling.



Figure 6: Dense core with soft shell. From the left: *cubic lattice MSM* model, $N = 281$, $S = 4094$, *LOD-hb-1*: $N = 114$, $S = 1596$, *LOD-hb-2*: $N = 51$, $S = 670$.

starts rolling until it stops due to energy dissipation that occurs mostly in the soft shell (Fig. 5). Examples of different LODs are presented in Fig. 6. We have measured and compared the distance each object travels. It is very volatile to the properties of the shell, as the overall movement is quite long, and all the differences in the behavior are accumulating with snowball effect.

The results are presented in the Table 1. It also contains information about the number of mass points and springs composing an object representation. As we can see, LODs generated with *priority merging* behave better than the standard ones. Surprisingly, enabling border preservation does not change the behavior of an object in a significant way. Originally this scenario has been designed specifically for testing the influence of border preservation on the quality of the model, as the behavior of this object is driven by deformations that occur in its outer regions. Visual quality of the animation is satisfactory for all generated LODs, and the most simplified ones have about 18% of the original number of nodes and springs.

model	distance	difference	nodes	springs
Base model	6.1	0	281	4094
LOD-1	4.5	1.6	123	1762
LOD-b-1	4.6	1.5	121	1784
LOD-2	5	1.1	55	722
LOD-b-2	4.5	1.6	53	780
LOD-h-1	5.6	0.5	114	1596
LOD-hb-1	5.6	0.5	114	1596
LOD-h-2	5.5	0.6	51	670
LOD-hb-2	6	0.1	51	670

Table 1: The distance a spherical object travels before it stops in the scenario 1.

6.2 Scenario 2 – complex movement

In the second test case, an object composed of two very stiff rods connected by rubber is being thrown at the wall, while rotating (Fig. 7 and Fig. 8). It hits the wall, bounces off, falls to the ground possibly still rotating until it finally stops due to friction. The exact movement is quite complex and the trajectory depends heavily on initial velocity of the object. In this test we use models generated by *standard* and *priority merging* algorithms, both with *weak border preservation*. We compare the final position of the center of mass of the object between different LODs. Each position is measured as a distance from the wall. The simulation has been carried out with variable time step and it was not deterministic. Every position presented in the table is taken as an average from three runs of the simulation for each model. We choose to ignore the final orientation of the bar as its distribution is too chaotic and may change significantly even between different runs of the same model. Fig. 7 shows the original model and an example of a simplified one. Snapshots of the animation are shown in the Fig. 8.

Results of this test are presented in the Table 2. They confirm that preserving non-homogenous areas of the object improves the quality of simplified representations. LODs generated with *priority merging* exhibit a behavior closer to the original. Both first level LODs (composed of about 45% of the base number of nodes and springs) allow to recreate the original movement of the object surprisingly well. Trajectories of second level LODs (20% of the original complexity) are noticeably different, however the behavior of these LODs is still realistic.

7. Conclusions

In this work we have presented an algorithm of simplifying mass spring model representations of deformable objects. We have shown, that the adaptation of classical LOD merging approach allows to obtain very good results with MSMs. In our test schemes, objects simplified by as much as 50% exhibit very small differences from the original behavior, comparable to divergences caused by the change of time step or simulation algorithm for the original non simplified model. Further simplification leads to noticeable changes in elastic properties, however the behavior of such objects is still plausible in most cases.

We have compared a few variations of presented algorithm including merging with border and heterogeneity preservation. Our tests indicate that preserving heterogenous areas has a noticeable, positive effect on the quality of simplified models. The influence of border preservation greatly depends on the shape of the object, and in our tests we did not observe significant changes in the behavior caused by enabling or disabling this modification.

Our tests show, that it is possible to achieve good behavior of simplified objects. In the next step of our work will will

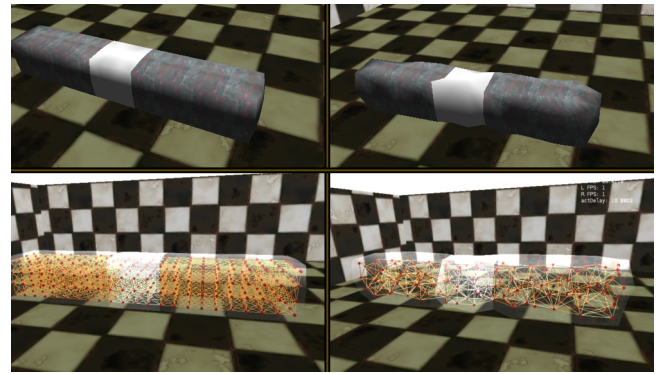


Figure 7: Two stiff bars connected by rubber (solid and wireframe). On the left *cubic MSM*, number of points $N = 368$, and springs $S = 4724$. On the right the same object simplified (LOD-hb-1), $N = 154$, $S = 1746$.

focus on incorporating a full physical LOD system with dynamical switching between object representations into a game or virtual reality application. It should allow to reduce the computational effort of the physical simulation significantly, allowing to create richer interaction environments.

References

- [1] A. Nealen, M. Müller, R. Keiser, E. Boxerman, M. Carlson: *Physically Based Deformable Models in Computer Graphics*. Comput. Graph. Forum, Vol. 25, No. 4. pp. 809-836 (2006).
- [2] C. Wojtan, G. Turk: *Fast Viscoelastic Behavior with Thin Features* ACM Transactions on Graphics, Vol. 27, No. 3, Article 47, Aug 2008.
- [3] L. Kharevych, P. Mullen, H. Owhadi, M. Desbrun: *Numerical Coarsening of Inhomogeneous Elastic Materials*. ACM Transactions on Graphics, Vol. 28, No. 3, Article 51, 2009
- [4] M. Nesme, P. G. Kry, L. Jeřábková, F. Faure: *Preserving topology and elasticity for embedded deformable models* ACM Transactions on Graphics, Vol. 28, No. 3, Article 52, 2009
- [5] P. Krysl, S. Lall and J. E. Marsden: *Dimensional model reduction in non-linear finite element dynamics of solids and structures* Int. J. Numer. Meth. Engng 51 479-504 2001
- [6] M. Müller, B. Heidelberger, M. Teschner, M. Gross: *Meshless Deformations Based on Shape Matching*. ACM Trans. on Graphics 24, 3 (Aug.), 471-478 2005.
- [7] M. Ostoja-Starzewski: *Lattice models in micromechanics*. Appl. Mech. Rev. 55, 35-61 (2002).
- [8] D. Luebke, M. Reddy, J. Cohen, A. Varshney, B. Watson, and R. Huebner: *Level of Detail for 3D Graphics*, Morgan Kaufmann, 2002.
- [9] A. J. C. Ladd, J. H. Kinney: *Elastic constants of cellular structures*. Physica A 240 349-360 (1997)
- [10] B. A. Lloyd, G. Székely, M. Hadders: *Identification of spring parameters for deformable object simulation*. IEEE Trans. Vis. Comput. Graph. 13(5) 1081-1094 (2007)
- [11] T.Y. Yeh, G. Reinman, S.J. Patel, P. Faloutsos: *Fool Me Twice: Exploring and Exploiting Error Tolerance in Physics-Based Animation* ACM Transactions on Graphics, Vol. 29, No. 1, Article 5, 2009.
- [12] B. A. Lloyd, S. Kirac, G. Székely, M. Hadders: *Identification of Dynamic Mass Spring Parameters for Deformable Body Simulation*. Eurographics 2008 - Short Papers, pages 131-134
- [13] M. Kot, H. Nagahashi, P. Szymczak: *Verification of physical properties of materials modeled with Mass-Spring Systems*. Technical report of IEICE. Multimedia and virtual environment 110(457), 201-206, 2011

Velocity at $t = 0$	Base	LOD-hb-1	LOD-hb-2	LOD-b-1	LOD-b-2
30	6.0 ± 0.5	6.0 ± 0.1	4.6 ± 0.1	4.2 ± 0.1	5.6 ± 0.1
31	3.6 ± 0.2	5.6 ± 0.1	4.8 ± 0.1	2.7 ± 0.4	3.7 ± 0.1
32	4.4 ± 0.9	5.2 ± 0.1	4.0 ± 0.3	3.2 ± 0.1	4.7 ± 0.1
33	3.4 ± 1.0	3.5 ± 0.1	4.4 ± 0.1	4.0 ± 0.1	3.8 ± 0.1
39	6.8 ± 0.3	6.4 ± 0.2	4.1 ± 1.0	7.9 ± 0.5	4.0 ± 0.2
40	5.7 ± 0.3	6.6 ± 0.1	5.2 ± 0.1	9.1 ± 0.1	3.8 ± 0.1

Table 2: Final position of the object, for different LODs and starting velocities. Base object $N = 368$, $S = 4724$; LOD-hb-1 $N = 154$, $S = 1746$, LOD-hb-2 $N = 67$, $S = 658$; LOD-b-1 $N = 154$, $S = 1828$, LOD-b-2 $N = 65$, $S = 640$.

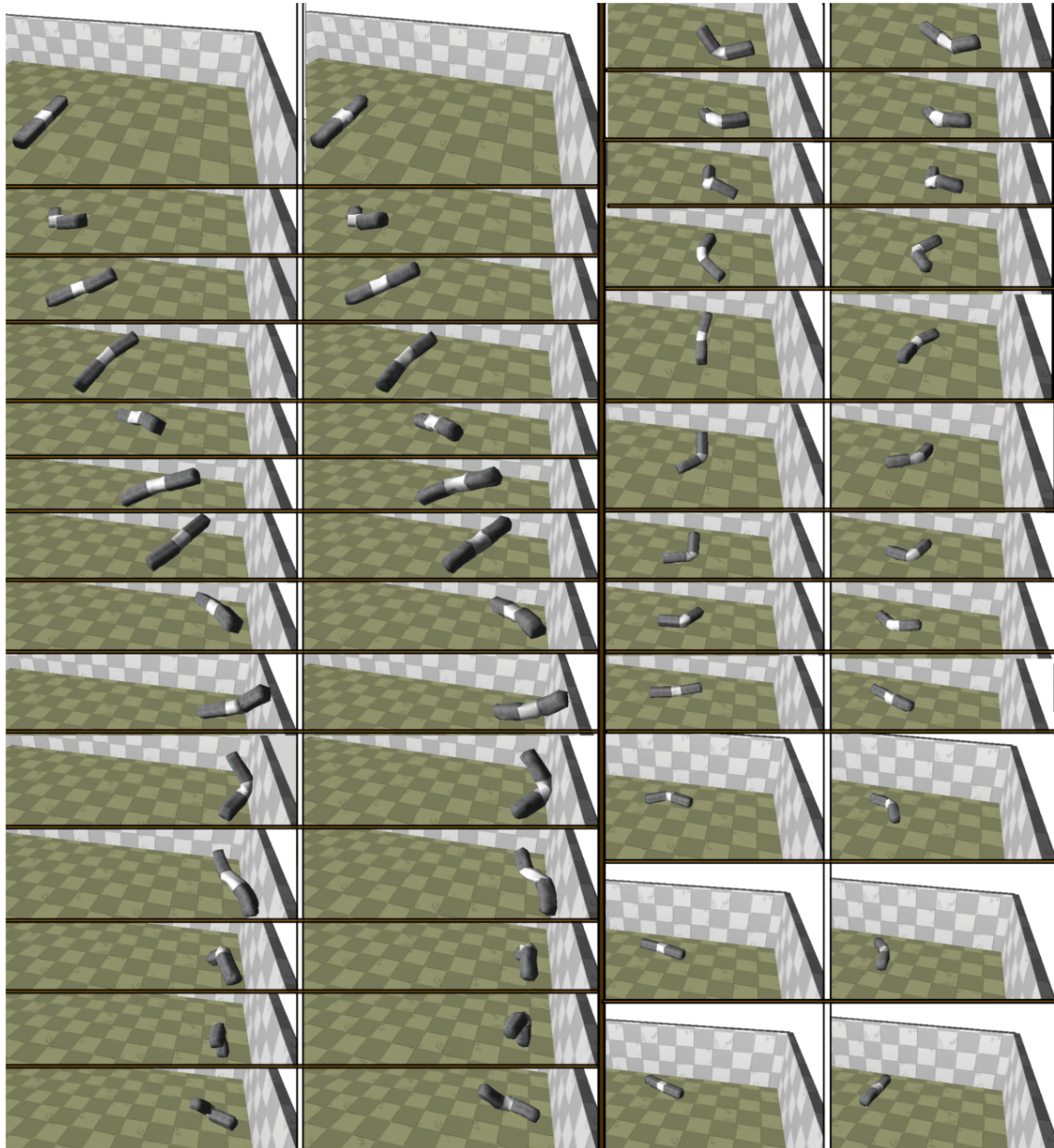


Figure 8: Animation for the test scenario 2. An object composed of two stiff bars connected by rubber rotates and hits the wall. Comparison between the base model (left) and LOD-hb-1 (right).

Make the Data Sing: Creating Audio out of Data

H.-P. Bischof^{1,2}, and L. Coy¹

¹Department of Computer Science,

² Center for Computational Relativity and Gravitation,
Rochester Institute of Technology, Rochester, NY, US

Abstract—*Visualization of data has a long history. The process converts data into a form which is easier to comprehend: something our eyes can see. Sight is one of our best-developed senses, therefore this process can provide a result which gives us faster and easier insight in a complex world. Visualization is, in mathematical terms, a conversion of n -dimensional data to k dimension(s); and k is significantly less than n . Precedented only by sight, hearing can be considered the second most developed sense. Therefore it makes sense to use audio as one additional dimension for the understanding of data. The sense of sight differs significantly from the sense of hearing in resolution and meaning. A visual can have many details while a sound describing data can only have broad strokes. A visual can be printed and can be used to publish visualization results. Sound describing a data set can be created, but it certainly is not easy to publish. This paper describes design patterns and a framework for generating audio as well as a method to describe sound in such a way that it can be published.*

Keywords: Visualizations, Sonification

1. Introduction

We are used to visual representations of data. Karl Pearson introduced the histogram to roughly access a probability distribution[13]. Maps depict relationships of objects. They are used as a navigational aid[7] or as a tool to organize knowledge, for example a map of the disciplines of Computer Science. Visualization of data is the art of converting n -dimensional data into a visual, accessible and comprehensible space. The dimensions of this space are less than n , which begs the question if it is possible to add one more dimension out side the visual spectrum.

We have a very good developed sense of sight and hearing that can be used to describe a situation. Our eyesight functions in very dim light and very bright light; Our ear can detect frequencies in the range of 20hz to 20000hz. This paper explores if we understand n -dimensional data better if we use additional audio information. We also present unexpected problems we faced in creating useful audio.

Visualization of data works because we are used to understanding visuals in our daily lives. Audification, the art of converting data to audio, has less application in our daily lives but still has very successful examples. Everybody understands what it means when a Geiger Counter[15] ticks

faster and faster. Imagine you watch a movie and you see heavy rain hitting the ground with and without sound. The sound made by the raindrops hitting pavement will give a better impression of how hard it rains, but imagine the same event when the rain hits a meadow. The sound of the raindrops in the second scenario will be significantly less dramatic. Lastly, imagine a group of people standing on the side of the road. They will always hear a change in the pitch of a police car siren, even if they don't understand what the Doppler effect is. It's obvious that audio adds information; the question is if the added information is decorative or is of substance.

2. Related Work

The Sonification Handbook is one of the most comprehensive works summarizing sonification. The book covers the theory behind it [16], parameter mapping[10], the navigation of data[5] and sub topics about audification[6]. Dombois and Eckel state that audification is the simplest form of sonification; nevertheless articles covering this field are rare. The book does not go into detail regarding how to describe the audio generated from the data, which is one of the initial problems we wished to solve.

Flowers[8] laid techniques which work and which won't work. In his paper he laid out a list of what works for Audification design principles. Representing numeric values through pitch, or using volume to represent changing numerical values are both effective techniques, Flowers claims. His work does not include positional changes of the origin of the sound source or movement of the sound source. Flowers concluded that he has concern with trusting the auditory system to do what it is not capable of. His last statement is "I believe our ears can indeed help us gain insight into complex data". We share his concern.

Herrmann[11] describes an environment which does not require a background in mathematics, but still allows data analysis. One of his examples maps Iris petal length to the starting point in time of the sound, Iris class to the stereo position, etc. The generated audio is interesting, but does not allow us to learn anything about the data. We are interested in audio which allows us to better understand.

The language Herrmann used to describe the conversion from data to audio does not allow the audio to be heard in the readers heads because it is, simply, a description of a

process. Mozart used sheet music, however, to describe Don Giovanni[14]. Sheet music describes how music sounds, and not the process of creating music. This is the idea we are following; We are describing an architecture which allows us to document audio in such a way that it can be generated platform independently and published in writing.

The following chapter narrows the problem statement followed by usable audio design patterns. Chapter 4 describes the language we chose for this work. Design patterns and examples are shown in chapter 5. The architecture is described in chapter 6. Chapter 7 discusses open question and unforeseen problems, followed by a summary of the results followed by our conclusions and future work.

3. Problem Statement

Audification is the art of converting scientific data to audio. The question is, does the combination of audio and video give a better understanding of the data? Most visualization system work in a very similar fashion - they use a data flow approach for the creation of images. The specific language and tool are unimportant. It does not help the community to create a specialized platform dependent architecture which will only work for one specific tool.

Most scientific data sets include the notion of time. Time moves typically in one direction. Audio can either be continuous or discrete. If it is discrete, the audio heard at a given point typically represents a delta t . The question is, when in this delta t will the audio be played; at the beginning, end, or middle?

An image or a movie can be evaluated independent of time. Each frame of a movie can be studied and analyzed for bugs individually. Audio, on the other hand, can only be evaluated over time because it has to be played. This makes the debugging difficult. We will describe in the architecture chapter how this problem can be solved.

Our brain processes visual and audio information differently, which results in different amounts of time required to process the information. This makes the synchronization of these to pieces of information challenging. As this leans towards cognitive sciences, we did not try to investigate this part.

This paper discusses a language describing audio in such a way that it can be printed, and an architecture that serves to create programs written in this language. We are also describing design patterns which can be used to solve sonification problems. In addition, we discuss problems we encountered in the process of creating audio.

4. Language

A language was needed as a medium between data and sound. This language's purpose would serve as the middle ground between data and sound - something that could house the data we needed and easily convert the data it contained into audio.

When searching for the appropriate parser, there were two requirements that needed to be fulfilled. The language firstly had to be extensible; extensibility defined as an implementation that has consideration for future development. If we wanted to add or change the way our data was represented audibly by adding an attribute like attack/delay or sound extent to our application, the language needed to be able to adapt, and represent these changes accordingly. The language secondly had to be documentable. The ability to add in comments about properties of the sound was crucial. The comments would not change the output, but simply remind the user of details to take note of.

GNU LilyPond[1] was the solution for our requirements. LilyPond, a \LaTeX [12] based typesetting program, receives text-based input from the user and outputs music engraved PDF files. The user simply writes LilyPond statements to a file and compiles for a generated PDF as well as a MIDI file to hear the final composition. Using LilyPond, we not only have a text-based medium for data to be converted to but also a visual manifestation of a sound generated output.

Although LilyPond is our parsing solution, it is not meant for human interaction when producing audio from data. The user should not have to modify, add to, or delete text from a LilyPond converted data set. Doing so could not only ruin the accuracy of the compiled audio file, but worst-case riddle the file with syntax errors, rendering it virtually useless. If the user is interested to see how the LilyPond syntax converts to engraved music or wants to simply observe the language, doing so is not difficult or ill-advised. The language's design strives to be simple for an entry level user for these reasons.

5. Design Pattern

History shows that given one independent variable, auditory expression through rate of change is an effective, uncomplicated way to help create data perception. One first mainstream approach to audification was the Geiger counter, made in 1908. The Geiger counter detects particles emitted by radiation by conducting electricity in an inert-gas tube whenever radiated particles or rays pass through. Whenever a pocket of ionized radiation is scanned by the detector, the conducted electricity is heard via auditory ticks; the denser the radiation, the higher the frequency of ticks in a given time interval. The Geiger counter is proof that through mono-temporal mapping, an element of data cannot only be conveyed through sound but can represent all the major patterns and details a visual component could, with less sensual attention given to the device itself.

Figure 1 is an example of a Geiger counter type-set to music. The rests after the notes indicate the various rates of change in radiation. The comments under the bars indicate the surround-sound position of the audio.

For certain tasks involving distinguishably patterned data, the Geiger counter's BPS *Beatspersecond* approach may prove sufficient. For more violent, rapidly changing data,



Fig. 1: Geiger Counter Example

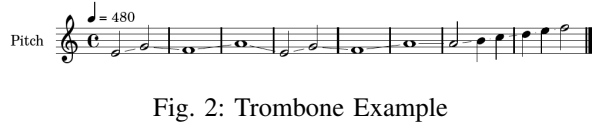


Fig. 2: Trombone Example

however, the methods output may become difficult to analyze. Pitch-based output may become a desirable alternative in these cases. Seismic datasets are a good example of a pitch-appropriate output; the data produced by earthquakes is usually incredibly large, depending on the time length of the recording. Days of recorded data can be frustrating to analyze, especially if the data's outliers are the only considerable portions. Using auditory seismic tools, what appeared to be infinitely long sets of data can be analyzed in a mere fraction of the time. When violent eruptions are analyzed, the data can become eccentric, going from peak to trough in a matter of milliseconds. Basing the output sound off tone rather than the time between notes gives the user more clarity for data that spikes and recedes hastily.

Figure 2 is an example of a pitch based data set. The notes are connected by glissando's and played high to low or low to high (like a trombone) depending on the upcoming note.

If location parameters are to be sonified, stereophonic audio, mapping visual location to audible location, may be a desirable option. Mapping a three-dimensional data set to a surround sound setting, for example can alert the user regarding what X-Y-Z vectors the data is becoming increasingly violent at. In the entertainment industry, the increasing amounts of films utilizing surround two-dimensional or three-dimensional sound technology are a good example of stereophonics that increases the spatial awareness of the user. Figure 1 is an example how this can look like. We used comments in the Lilypond file to describe location positions.

As these few examples are non-exhaustive, there are other common techniques such as duration, volume, and timbre which all have their respective uses.

Figure 3 is the program which creates Figure 2. The part shown describes the score. The score for realistic audio section will be generated by programs and not by humans.

6. Architecture

In the world of computer science, most solutions to problems are described with programs written in languages, which are a good fit for the problem domain. Program languages are designed to create platform independent solutions. We tried to follow the same idea and applied a

```

...
staffPitch = \new Staff {
  \tempo 4 = 480
  \time 4/4
  \set Staff.instrumentName = "Pitch"
  \set Staff.midiInstrument = "tubular bells"
  \key c \major
  \clef treble
  \relative c' {
    e2\glissando g2\glissando
      f1\glissando a1\glissando
    e2\glissando g2\glissando
      f1\glissando a1\glissando
    a2\glissando b4\glissando c4\glissando
    d4\glissando e4\glissando f2\glissando
  \bar "|."
}
}
...
}

```

Fig. 3: The Lilypond Program for the Trombone Example

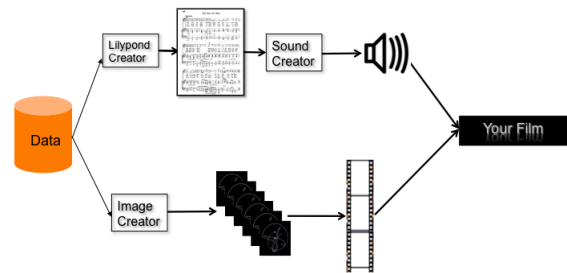


Fig. 4: Architecture World

data flow paradigm[9] to solve the problem. Figure 4 is a depiction of our solution. The data is read by and generates a program of the audio (.ly). The same data is read to create the images. The .ly file is platform independent and readable by a human. This allows us to debug the component which creates the .ly file. The .ly file is the translated to audio and then attached to the movie.

The architecture was used to extend the Spiegel visualization framework[4]. Figure 5 shows a visual representation of the program. The data is read and sent out on two different output streams. The upper data stream generates the audio and the lower data stream generates the images out the data. The same could easily be applied to other visualization systems like vish[2], or visIt[3].

7. Open Questions

A scientific data set created via simulation or an experiment is typically converted to a movie with 24 frames per second. The Geiger counter pattern requires, depending

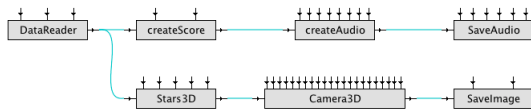


Fig. 5: Architecture World

on the data set, a high number of BPS. This causes a problem for the human ear and the audio system when a frequency greater than 50 BPS has to be generated. This leaves a narrow band compared to the color spectrum one has available for a visual.

A similar problem would present itself when working with pitch-based sound output. As previously stated, the human ear can hear between 20 Hz and 20 kHz – this is roughly a five octave range. Figuring out how to comfortably distribute this range evenly between a given data set could prove to be difficult because of the sheer range of possible output.

Although the applications were clear, the design had many challenges that should be a concern for any sound generating process. When programming with the Java Sound API, a decision had to be made when working with a velocity-quantified visualization; how the magnitude should correspond to the time between notes. At first, each magnitude of velocity was plugged into a formula that would output an integer, which would correspond to the number of milliseconds the audio would wait before producing another tick. This method produced distinguishable, yet prolonged, patterns that would be many times as long as the visualizations time length itself.

It is also important to note the scope that audio covers compared to its visual counterpart. Audio simultaneously covers an entire landscape of data for the duration of the track. Visualizations on the other hand are made up of pixels that can be analyzed separately from the rest if the user wishes to do so. There is no way to “split” generated audio up to look at a specific quadrant of data. The user must specifically compose the sound with this problem in mind.

8. Conclusion

Our platform independent architecture has allowed us to generate audio based on sheet music. The generation of the sheet music and audio is divided into two processes. We learned that the bandwidth which can be transmitted via audio is very limited, but it can be successfully applied to situations where simpler patterns work, like the Geiger Counter Pattern.

9. Future Work

A deeper study of more design patterns will be one of the next steps. These design patterns will include surround

sound, because it appears natural that this may help to represent the data better. Additionally we will study the effect of surround sound in combination of stereoscopic visualizations.

References

- [1] The lilypond project page @ONLINE. March 2012.
- [2] The vish project page @ONLINE. March 2012.
- [3] The visit home page @ONLINE. March 2012.
- [4] Hans-Peter Bischof, Edward Dale, and Tim Peterson. Spiegel - a visualization framework for large and small scale systems. In *In MSV 06: Proceedings of the 2006 International Conference of Modeling Simulation and Visualization Methods*, 2006.
- [5] Eoin Brazil and Mikael Fernström. Auditory icons. In Thomas Hermann, Andy Hunt, and John G. Neuhoff, editors, *The Sonification Handbook*, chapter 13, pages 325–338. Logos Publishing House, Berlin, Germany, 2011.
- [6] Florian Dombos and Gerhard Eckel. Audification. In Thomas Hermann, Andy Hunt, and John G. Neuhoff, editors, *The Sonification Handbook*, chapter 12, pages 301–324. Logos Publishing House, Berlin, Germany, 2011.
- [7] Eds. *The History of Cartography Volume 1: Cartography in Prehistoric, Ancient, and Medieval Europe and the Mediterranean*. University of Chicago Press.
- [8] J. H. Flowers. Thirteen years of reflection on auditory graphing: Promises, pitfalls, and potential new directions. pages 406–409, Limerick, Ireland, 2005. Department of Computer Science and Information Systems, University of Limerick, Department of Computer Science and Information Systems, University of Limerick.
- [9] Steve Goddard. Analyzing the real-time properties of a dataflow execution paradigm using a synthetic aperture radar application. *Real-Time and Embedded Technology and Applications Symposium, IEEE*, 0:60, 1997.
- [10] Florian Grond and Jonathan Berger. Parameter mapping sonification. In Thomas Hermann, Andy Hunt, and John G. Neuhoff, editors, *The Sonification Handbook*, chapter 15, pages 363–397. Logos Publishing House, Berlin, Germany, 2011.
- [11] Thomas Hermann. Sonifikation hochdimensionaler daten - funktionaler klang zum erkenntnisgewinn. In Georg Spehr, editor, *Funktionale Klänge*, number 2 in Sound Studies, pages 67–85. transcript Verlag, Bielefeld, 05 2009.
- [12] Donald Knuth. *The TeXbook*. Addison-Wesley, 1984.
- [13] Douglas C. Montgomery. *Introduction to statistical quality control*. Wiley, New York, NY [u.a.], 3. ed edition, 1997.
- [14] Wolfgang Amadeus Mozart and Lorenzo Da Ponte. *Don Giovanni / Wolfgang Amadeus Mozart ; [English translation by Norman Platt and Laura Sarti]*. J. Calder ; Riverun Press, London : New York :, 1983.
- [15] E. Rutherford, H. Geiger, and J. Harling. An Electrical Method of Counting the Number of α -Particles from Radio-Active Substances. *Royal Society of London Proceedings Series A*, 81:141–161, August 1908.
- [16] Bruce N. Walker and Michael A. Nees. Theory of sonification. In Thomas Hermann, Andy Hunt, and John G. Neuhoff, editors, *The Sonification Handbook*, chapter 2, pages 9–39. Logos Publishing House, Berlin, Germany, 2011.

Design and Application of the Visual Model Pool of Mechanical Parts based on Computer-Aided Technologies

Xiaoying Dong, Xia Ye, Qinxian Jiang, Xianghua Zhang and Wei Bi

School of Mechanical and Automobile Engineering, Jiangsu Teacher's University of Technology, Changzhou, Jiangsu Province 213001, P.R. China

Abstract - *To meet the need of the integration of computer-aided technologies (CAX) and the teaching of CAX courses, three dimensional (3D) bodies were modeled with Auto CAD, the rotation, intersection and animation were made with Solid Works and 3Ds Max, the model and the scene were real-time rendered with VtmlPad, the images were post-processed with Photoshop, the source elements were merged and linked on the platform of Dreamweaver, and finally, the visual model pool of mechanical parts was established. The pool consists of four components: the integrated parts, the single parts, the assembled parts and the standard parts. Users can see animations and different views of a 3D body as well as observe a body by zoom, drag and rotation.*

Keywords: Computer-aided technologies, Mechanical parts, Visual model pool, SolidWorks, Dreamweaver, AutoCAD

1 Introduction

It is a long established tradition that the training of engineering students' design capability begins with engineering graphics courses. To enhance students' recognition, large amounts of real models are needed to

assist both class teaching and homework. The use of these models has obvious positive effects on the training space thinking capabilities, but it is practically limited and difficult to realize in the teaching of engineering graphics [1,2]. In addition, problems such as fragile, non-portable and irreversible greatly hinder the application of real models. Traditionally, the freshman begins with real models and finishes with drawing sheets, followed in subsequent courses from drawing sheets to parts, which reduces efficiency. Computer-aided technologies (CAX) course series has become an important component of mechanical engineering majored courses. CAX also meets the demand of various industrial applications. Based on the characteristics of CAX, this research constructed a visual model pool of mechanical parts, with computer-aided design (CAD) three dimensional (3D) model and educational modernization technologies. The visual model pool meets both demands from teaching and industry, which provides a multi-functional, efficient, and portable tool for a novel learning mode of students and a novel working mode of engineers.

2 Structure

More than 5000 models are stored in the visual model pool, which consists of single parts, integrated parts, assembled parts and standard parts. Some industrial products are also available. The pool is an open system with parameterized design. Administrators can easily change, add or delete models from the pool, which not only facilitate the design of new product, but also substantially reduce the development cycle of new industrial products. Figure1 illustrate the general scheme of the visual model pool of mechanical parts (Fig. 1).

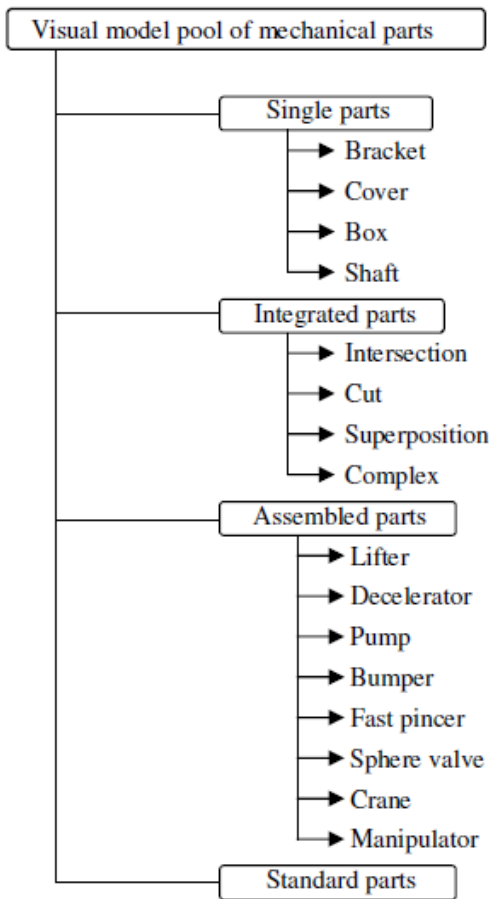


Fig. 1 Schematic illustration of visual model pool of mechanical parts

2.1 Homepage

The homepage is an introduction of the visual model pool of mechanical parts, available on the website of Jiangsu Teachers' University of Technology, P.R. China

(www.jstu.edu.cn). Functions and characters of the four categories of the pool are briefly introduced. Browsers can leave a message to developers, or choose to enter a category main page (Fig. 2).



Fig. 2 Mainpage of the visual modeling pool of mechanical parts

2.2 Secondary Mainpage

2.2.1 Single Parts Mainpage

More than 1000 single parts here consists of four types, the bracket, the cover, the box and the shaft (Fig. 3).



Fig. 3 Mainpage of the single parts model pool

2.2.2 Integrated Parts Mainpage

More 1000 integrated parts here consists of four types, intersected parts, cut parts, superposed parts and complex parts (Fig. 4).



Fig. 4 Mainpage of the integrated parts model pool

2.2.3 Assembled parts Mainpage

About 100 product models here include the lifter, the decelerator, the pump, the bumper, the fast pincer, the sphere valve, the crane and the manipulator (Fig. 5).

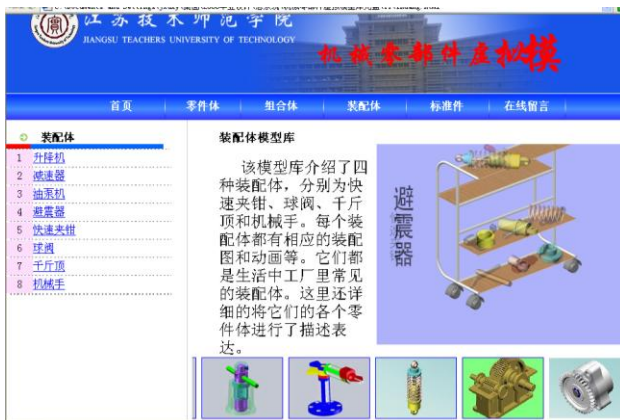


Fig. 5 Mainpage of the assembly model pool

2.2.4 Standard parts Mainpage

The construction of a standard parts pool is a prerequisite for the fast upgrade of products. Standard parts in the visual model pool include more than 1000 keys, pins, nuts, bolts, shafts and shims (Fig. 6).

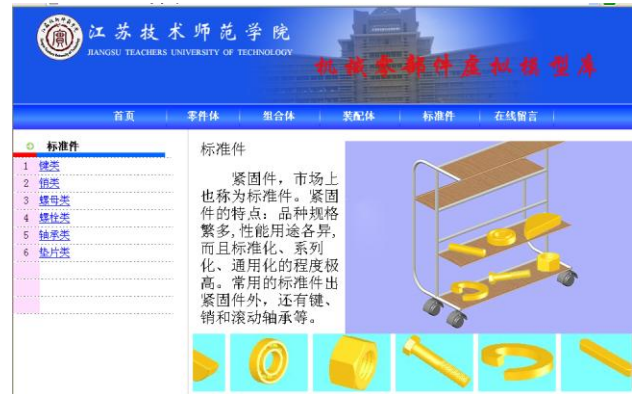


Fig. 6 Mainpage of the standard parts model pool

2.3 Assembled Parts Category

As the most important category of the pool, the assembled parts encompass more than 100 mechanical products, which can be easily modified and updated. Figure 7 shows a model pump.



Fig. 7 Mainpage of the pump model pool

Among others, the functions of the assembled parts navigation module include animation, product introduction, multi-view representation, free observation and seeing single parts, etc. The “animation” shows the dismounting and assembly of the pump (Fig. 8).

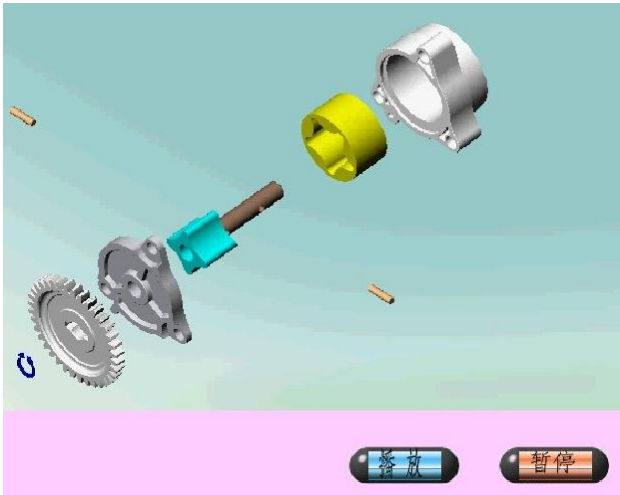


Fig. 8 Page of pump dismounting and assembly animation

By “free observation”, users can rotate, move, cut and zoom each single part, to help understand the internal structure (Fig. 9). Users can also “see each part” to go to corresponding single part’s main page (Fig. 10).

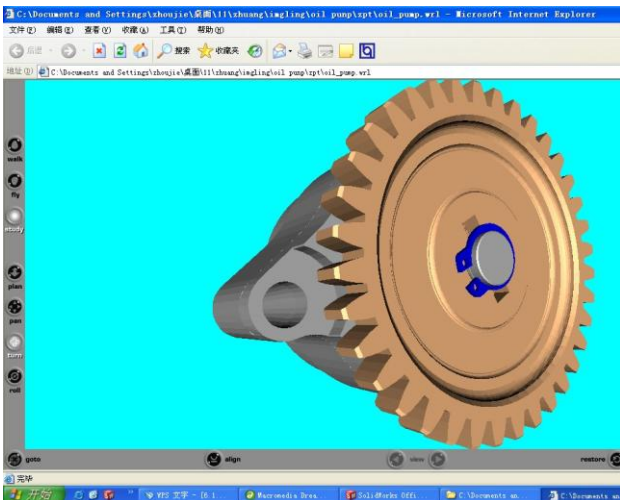


Fig. 9 Free observation function of the pump



Fig. 10 Page of “see each part” of the pump

3 Design and development

3D models in the visual model pool of mechanical parts (version 1.0) were constructed with SolidWorks and read by 3Ds Max in Standard Template Library (STL) format. In 3Ds Max, models were rendered for different view, light and materials. AVI animations were exported after rotation, cut and intersection. VRML Codes were added with VrmIpad to carry out in-situ rendering to the model and scene. Images were post-processed with photoshop. Finally, the elements were integrated and linked on the platform of Dreamweaver.

Based on the application of the above technologies, the system reaches the design requirements, with the following characteristics:

1) Visual reality is based on the VrmI. Browsers can observe a mechanical product comprehensively. Enterprises can manage their products simply and orderly.

2) The networked system facilitates data transfer and product exhibition. The included on-line trade system shortens the transaction time.

3) Researcher in industry can directly design products through the parameterized system, thanks to the integration of 3Ds Max and VRML codes with design softwares such as CAD and Solidworks.

4) To meet the need of teaching, great flexibility is enabled in the visual model pool; from students’ point of view, relevant professional knowledge is expended, to train the 3D thinking capability and enhance the understanding of engineering graphics.

4 Applications

4.1 Application in Education

The visual model pool of mechanical parts (version 1.0) is applied in teaching. The class survey shows that the pool is impressed as accurate, vivid and colorful. Complicated structures are fully represented, such as the outer shape and the inner cavity of complex bodies, intersection lines and the assembly of single parts. Animations are easy understood. The interaction between the users and the on-line pool is supported by VRML. Multi-view representation and free observation reach the effect of usage of real models. Inner structures of models are also clearly seen.

4.2 Application in Industry

Feedbacks from industrial partners show reduced product design periods, lowered costs and an increased efficiency. The development, research, exhibition and trading of products are effectively integrated. The on-line environment facilitates the sales and publicity of the products and enhances the interaction with customers.

5 Conclusions

With combined computer and multi-media technologies, the visual model pool of mechanical parts is designed and developed for educational and industrial applications, which substantially optimizes the quality of teaching and the profit of enterprises. The realization of parameterized design with the visual model pool show advantages in fast modification and supplementation of models, which has promising future applications.

6 Acknowledgments

The development of the visual model pool of mechanical parts is funded by the Jiangsu Educational Bureau project "Chinese-German cooperative educational research and practice in engineering graphics" (KYZY07040)

7 References

- [1] Z. Wang. Journal of Guangdong Industry Technical College, Vol. 3, 25-28, 2008.
- [2] X. Chen, L. Li and X. Wang. Journal of Chengdu University of Technology, Vol. 32, No.4, 80-82, 2005.
- [3] X. Dong. Journal of Jiangsu Teachers' University of Technology, Vol. 13, No.7, 57-60, 2007.
- [4] B. Lu. China Mechanical Engineering, Vol. 15, No.11, 987-900, 2004.
- [5] G. Zhou, K. Feng, Z. Zuo, et al.. Chinese Journal of Engineering Graphics, Vol. 34, No.6, 185-188, 2005.
- [6] M. Fan. Journal of Guangdong Polytechnic Normal University, Vol. 28, No.3, 114-115, 2006.
- [7] Q. Li and X. Liu. Journal of Changchun University, Vol. 4, 1-3, 2005.
- [8] X. Ni, Y. Yang, J. Liu. Mechanical Design and Manufacture, Vol.7, 169-171, 2007.

