Graphics Interface 2010

Ottawa, Ontario, Canada
May 31 to June 2

Poster Session Proceedings

Edited by
Won-Sook Lee and Mauricio Vines

Sponsored by the Canadian Human-Computer Communications Society

The papers in this volume comprise the proceedings of the meeting mentioned on the title page. They reflect author’s opinions and are published as presented without change. Their inclusion in this publication does not necessarily constitute endorsement by the editors or the Canadian Human-Computer Communications Society.

Contents

Graphics Posters

Human Computer Interface Posters

Message from the Chair

It is a pleasure to present the posters for Graphics Interface 2010. This is a very exciting session, which consists on 15 poster presentations on Computer Graphics and Human Computer Interfaces. Topics in graphics include non-photorealistic rendering, physics simulation, human animation, and image processing. Presentations in computer interfaces cover topics of visualization, interfaces for immersive and mobile environments, videogames, accessibility and usability. The poster presentations feature very innovative research in the field and I am confident this session will continue stimulating the advancement of the state of the art in computer graphics and computer interfaces.

I would like to express my gratitude to the Canadian Human-Computer Communications Society and the University of Ottawa, for their support and making this session possible.

Won-Sook Lee
Posters Chair

Graphics Interface 2010

Ottawa, Ontario, Canada
May 31 to June 2

Poster Papers

Oil Painting Rendering Changeable by Light Effect

Sungkuk Chun[*] Keechul Jung[†]

HCI Lab., Soongsil University

Abstract

Traditional oil painting works enable the spectators to feel the various impressions because it can be shown differently by the changes of light effect. The reason of oil painting's distinguishing feature is that it contains the texture and volume of color expressed by thickness of used pigments and brushing. In this paper, we present a novel method that reproduces oil painting-like image from a source picture based on a virtual light and non-photorealistic rendering technique. To generate the oil painting-like image as an output, the system first performs stroke distribution, which is to determine where a brush is located for stroking, using edge detection and image segmentation on an input picture. And the intermediate image is constructed with the suitable color, orientation and size of brush at each stroke point. At last, the system applies light effect to the intermediate image and generates the oil painting-like image.

Keywords: Non-photorealistic Rendering, oil painting, aesthetic

Index Terms: I.3.3 [Computer Graphics]: Picture/Image Generation — Display algorithms; I.4.0 [Image Processing and Computer Vision]: General

1 Introduction

In the recent decade, artists express their creativity not only through the intuitive expression but with the help of computer technologies such as image processing, computer vision, and computer graphics. And using these research fields, a more aesthetically evolution of digital art is being represented into a fascinating digital form. The aim of these researches, such as non-photorealistic rendering techniques [1][2], is to make use of computer techniques to reproduce an aesthetic digital art representation from a still image.

The lots of existing non-photorealistic rendering methods have a tendency to focus on intrinsic and technical aspects of how to paint the image similarly to real painting works. In these methods, it is important to determine the order, the direction, and the number of strokes for painterly rendered image generation. However, for the oil painting works, the extrinsic and environment points such as light effect also acts essentially, because the texture and the volume variable by the different light conditions enable to give spectators various impressions.

In this paper, we present a novel method that reproduces oil painting-like image from a source picture based on a virtual light and non-photorealistic rendering technique. The system first performs stroke distribution to determine the point to be stroke, by using edge detection and image segmentation [3] on an input picture. And the intermediate image is constructed with the suitable color, orientation and size of brush at each stroke point. At last, intermediate image transformation based on the light effect generates the oil painting-like image as an output of our system.

We give a detailed description about our algorithm in Section 2. The experimental results are illustrated in Section 3. We then conclude the paper in Section 4.

2 Proposed System

This paper proposes a non-photorealistic rendering system to create an oil painting-like image from an input picture. The system consists of three modules, stroke distribution, painterly rendering, and intermediate image transformation. Figure 1 shows the process of proposed system.

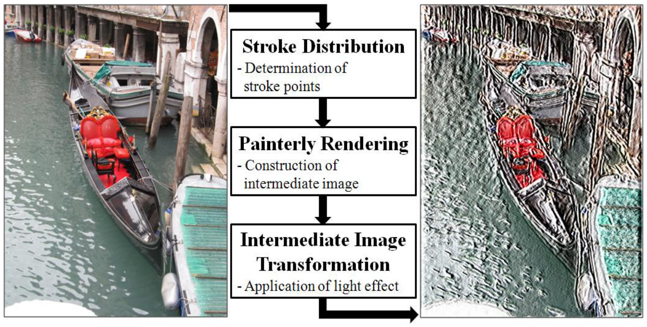

Stroke Distribution
- Determination of stroke points
Painterly Rendering
- Construction of intermediate image
Intermediate Image Transformation
- Application of light effect

Figure 1. Process of proposed system

Stroke distribution process as the first step is to decide the stroke point where the brush texture defined by user is located. For the determination of stroke point, the system analyzes the local complexity of an input image using edge detection and image segmentation.

In painterly rendering process based on the existing method [2], the system draws an intermediate image from an input picture by using local image moments [4] and the stroke distribution.

Intermediate image transformation retouches the intermediate image by using light effect and the number of stroke times at each pixel. After this work, the oil painting-like image that changeable by light direction and light power is created.

2.1 Stroke Distribution

As artists decide initially where they paint, the proposed system also determines by first the stroke point to be painted. In order for stroke distribution, the system analyzes the local complexity of the input picture using edge detection and image segmentation. This process is based on two assumptions; 1) complicated region must be painted using lots of small and delicate brushes, 2) simple region must be painted using a suitable brush to the region.

Edge detection is used to extract the location where a rapid and complex color change between neighboring pixels is appeared. Through this method, it is possible to recognize the complicated region having large color variation.

In case of simple region extraction, the system applies image segmentation, which is generally used for grouping the neighbor pixels that consists of similar colors. And then the central points of segmented regions are defined as the stroke point.

[*] e-mail: k612051@ssu.ac.kr

[†] e-mail: kcjung@ssu.ac.kr

2.2 Painterly Rendering

Painterly rendering as second process is to generate a painting-like image as an intermediate image based on the stroke distribution computed from the previous step and local image moments. For rendering the image, the following three properties must be defined; 1) brush texture, 2) stroke properties, 3) stroke order.

Brush texture that means the style of brush is defined by user. And stroke properties, such as suitable brush color, location, orientation, and size to each stoke point, can be obtained by local image moments which are used for calculating the centoid, width, height, orientation of local image. Stroke order is in order to paint large regions first, and depict small regions on the painted large regions based on the brush size at each stroke point.

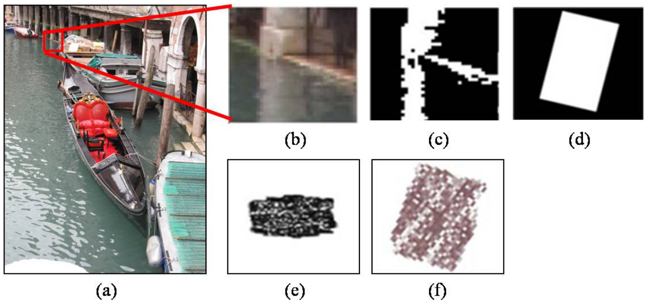


Figure 2. Stroke process: (a) input image, (b) local image, (c) local binary image, (d) equivalent ellipse based on local image moments(width: 33, length: 26, θ: 1.4358), (e) defined brush texture, (f) rendered brush style.

Figure 2 describes the image moments based painterly rendering process. For the calculation about image moments, the local image is converted to the binary image (Figure 2(c)). And then the width, the height, and the orientation of brush are computed by image moments (Figure 2(d)). After that, the color of brush is applied to brush texture (Figure 2(f)). All of these steps are computed at every stroke points.

2.3 Intermediate Image Transformation

To create the oil painting-like image, the system transforms the intermediate image by using light direction, light power, and the number of stoke times at each pixel.

For application of light effect, the system utilizes two kinds of data, depth map (*D*) and gradient map (*G*). Depth map contains the number of stroke times at each pixel, and gradient map is obtained by differentiation of depth map along the defined light direction. Through adding the gradient image to the intermediate image, a transformed image (*T*) as the oil painting-like image is completed. The following equations represent intermediate image transformation. In these equations, $T(x,y)$, $G(x,y)$, and $P(x,y)$ mean respectively a pixel value in the transformed image, the gradient image, and the intermediate image.

$$T(x,y) = \alpha \times G(x,y) + P(x,y), 0 \leq \alpha. \qquad (1)$$

Here, α is a light power parameter and $G(x,y)$ is calculated by the following equation,

$$G(x,y) = \begin{cases} D(x,y) - D(x+1,y) & \text{if left light} \\ D(x,y) - D(x-1,y) & \text{if right light} \\ D(x,y) - D(x,y+1) & \text{if bottom light} \\ D(x,y) - D(x,y-1) & \text{if top light} \end{cases}, \qquad (2)$$

where $D(x,y)$ is a pixel value in the depth map.

Figure 3(a) and (b) are the results by using right light and top light respectively. As shown in Figure 3, the system enables to generate the oil painting-like image changeable by the light effect.

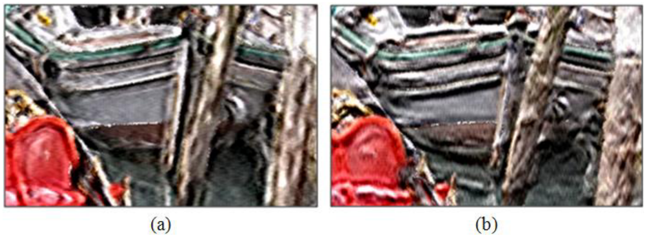


Figure 3. Results of different light directions; (a) right light, (b) top light

3 Experimental Results

We have tested 50 pictures obtained from the web for the experiments. Two examples of them are shown in Figure 4.

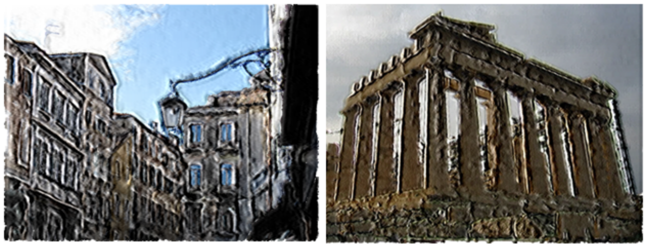


Figure 4. Result examples of our system.

4 Conclusion

This paper proposed a novel method of reproducing oil painting-like image from a source picture based on virtual light effect and non-photorealistic rendering technique. Computational methodology such as edge detection, color quantization based image segmentation, and image moments serves as the core engine for stroke distribution and painterly rendering. And through applying light effect into the intermediate image, the system generated oil painting-like image changeable by virtual light condition. For the future work corresponding to this, we will try to apply the more reasonable light effect through analysis of light flow on an input picture.

Acknowledgements

This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency)(NIPA-2009-(C1090-0902-0007)), and the Soongsil University BK 21 Digital Media Division.

References

[1] A. Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. *Proceedings of SIGGRAPH '98*, Computer Graphics Proceedings, Annual Conference Series, pages 453–460. ACM SIGGRAPH, ACM Press, 1998.

[2] M. Shiraishi and Y. Yamaguchi. An algorithm for automatic painterly rendering based on local source image approximation. *Proceedings of 1th International Symposium on Non Photorealistic Animation and Rendering*, pages 53-58, 2000.

[3] A. H. Dekker. Kohonen neural networks for optimal colour quantization. *Network: Computation in Neural Systems*, volume 5, pages 351-367, 1994.

[4] C.H. The and R.T. Chin. On image analysis by the methods of moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 10, pages 496-513, 1988.

Statistical Fluid Flow Synthesis

Mauricio Vines* Won-Sook Lee[†]

University of Ottawa

Abstract

We present our ongoing work on developing techniques for real-time simulation and visualization of fluids. We propose synthesizing fluid motion from simulation examples obtained with a full numerical solver. We segment our simulation data in order to capture the properties of local fluid behavior and apply statistical techniques to obtain the main modes of variation of the flow interacting with solid obstacles. Most research in fluid simulation has been focused on techniques for solving the Navier-Stokes equations for fluid motion, obtaining physically accurate simulations. We aim to obtain visually plausible flows by combining the results of physically accurate simulations preserving the general flow behavior. Our technique is suitable to be applied for visual effects design in movies as well as real-time applications such as videogames and virtual environments with haptic interaction, where efficient computations are required.

Keywords: Fluid simulation, principal component analysis, visualization.

Index Terms: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality; I.6.8 [Simulation and Modeling]: Types of Simulation—Animation

1 Introduction

Simulating fluids remains one of the most challenging problems in computer graphics. Recent advances in computer applications demand more realism and interactivity. Several techniques have been developed to provide accurate and realistic fluid simulations, however, at a great computational cost. This is due to the interplay of different complex phenomena involved, such as advection, diffusion and turbulence, which are difficult to simulate. Fluid behavior is characterized by the well known Navier-Stokes equations which are a system of partial differential equations for which no analytical solution is known.

In computer graphics, the main goal is to achieve visual plausibility, even at the cost of physical accuracy. Several approximations to fluid behavior have been developed, including particle-based simulations [4] and height field approximations for liquid surfaces [3]. Applications of fluid simulations include visual effects in movies and videogames, as well as real-time interaction with virtual environments [1][7].

We propose a different method to efficiently obtain visually plausible simulations using fluid motion examples. These examples are obtained by running a standard fluid simulation under different flow conditions and the resulting motion data is spatially partitioned into different fragments. We apply principal component analysis (PCA) on the fragments and we employ this information to synthesize new fluid motion. Fluid motion databases have been used in the past for computing fluid forces on immersed solid objects [1], however we focus on visualizing the fluid behavior in real-time. Composing fluid behavior from a set of simulation components or *tiles* has been applied with model reduction [6]. Here each component contains a base of possible fluid velocities. In our scenario, we do not use model reduction for our simulations, but synthesize motion from examples obtained by numerically solving the Navier-Stokes equations.

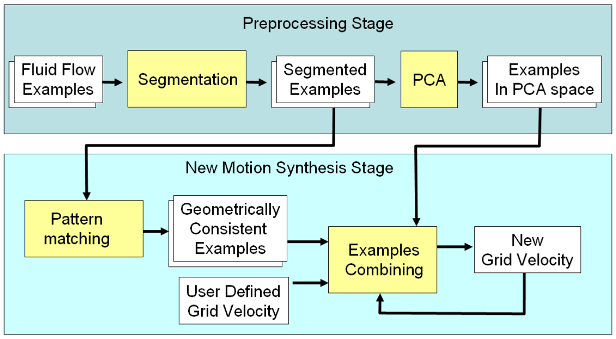


Figure 1. Schema of our approach

2 Methodology

In this section we discuss the two stages of our simulation framework: the construction of the examples database and the synthesis of new fluid motion from these examples. We represent the fluid using an Eulerian approach, both in the preprocessing and motion synthesis stages. The simulation space is divided into a regular grid of cells. In each cell the fluid velocity and pressure are stored. Solid objects are assumed to be static and they are voxelized as in [2], conforming to the simulation space configuration. In the preprocessing stage simulation examples are produced and analyzed. In the synthesis stage we find velocity values for each grid cell to approximate fluid behavior by combining the flow information from the simulation examples. A schema of our approach is shown in Figure 1.

2.1 Database construction

Fluid flow examples are obtained from simulating the fluid on the grid in the pre-processing stage. For simulating the fluid we employ a finite difference method based on operator splitting and semi-Lagrangian advection [5]. We run several simulation rounds for different solid configurations in order to have a rich set of fluid motion examples. In a simulation round the obstacle geometry is defined, and the fluid is simulated under different flow conditions, varying the direction and magnitude of the fluid velocity. Velocity values for each time step of simulation are stored for segmenting and statistical analysis.

Once the examples have been obtained, the results of each time step are segmented in fragments of equal size. A fragment consists on several grid cells, each storing velocity and pressure values. For each fragment we keep track of its neighboring fragments as well as the corresponding fragment in the next time step. As we consider only the case of static solids, the velocities in the grid cells occupied by a solid are zero. An example of the fluid examples generated and its fragments is shown in Figure 2.

* e-mail: mvine059@uottawa.ca
[†] e-mail: wslee@uottawa.ca

We apply PCA on the fragments to capture the main modes of variation of the fluid velocity. We construct a covariance matrix from the fluid velocity data in each fragment and we obtain the eigenvectors and eigenvalues of this matrix. Then we transform our simulation data into the PCA space, this is, we represent the data in terms of the principal components, which are the eigenvectors with highest eigenvalues associated. The fragments spatial and temporal associations are stored, enabling to quickly determine the local evolution of the fluid for a specific fragment.

2.2 Flow synthesis

Synthesizing new fluid motion in the simulation grid is performed using the data from the examples and the current velocity in the grid. A specific velocity field may be defined by the user as initial conditions for the simulation, or as result of real-time interaction. The new simulation data is segmented into fragments the same way as the examples and the velocity data for each fragment is translated into the PCA space. For fragments that contain solid cells, or that share a boundary with a solid, a pattern matching step is performed in order to determine the best example fragments that resemble the specific local solid geometry.

Let *f* be a fragment in the new simulation and *S* the set of example fragments that are geometrically compatible with *f*. The fluid velocity in *f* is determined by finding a combination of examples such that it matches the behavior observed in *f*. Formally, let $e_1, e_{2},,...,e_n$ be the first *n* principal components derived from the examples, and let $B=[b_1, b_2,...b_n]^T$ be the vector of coefficients corresponding to *f* in the PCA space. For each example $s^i \in S$, let $C^i=[c^i{}_1,..., c^i{}_n]^T$ be the vector of coefficients in the PCA space. Then synthesizing new motion reduces to solving the following system:

$$\begin{bmatrix} C^1 & \cdots & C^k \end{bmatrix} \cdot x = B \qquad (1)$$

The coefficients of vector *x* indicate how the samples must be combined to match the current motion in the current time step. As the matrix in equation (1) may not be square, the system may have to be solved in the least squares sense. If no external force is added by the user, then solving the fluid velocity in *f* for the next time step consists on finding the corresponding fragments in *S* for the next time step and combining them using the coefficients in *x*. When an external force is added, it is necessary to solve the system in equation (1) as the fluid motion may have drastically changed. Notice that as we assume the solid objects are static, determining the set *S* of geometrically compatible fragment examples for each new fragment is performed just once at the beginning of the simulation. Using this simple technique we can simulate fluids interacting with static rigid solids.

3 Discussion

Our approach may be used to simulate fluids at different detail scales, depending on the size of the simulation fragments. Smaller fragments will provide more detailed information, at a higher computational cost. We must note that the flow results in general will not be mass-conserving in the strict sense. The mass conservation principle expresses that any change in fluid mass in a region depends only on the net flux of mass in that region. This is usually enforced as an additional constraint when solving the Navier-Stokes equations. Although the examples used to construct our database are mass-conserving, linear combinations of these examples may not. While this is not a big problem when simulating a gas, care must be taken when simulating liquids using our technique. Liquid simulation requires the proper definition of a free surface. A non-mass conserving fluid will appear as evaporating, hence losing mass. To properly simulate liquids using our technique, examples of the free surface have to be generated and statistically analyzed as well.

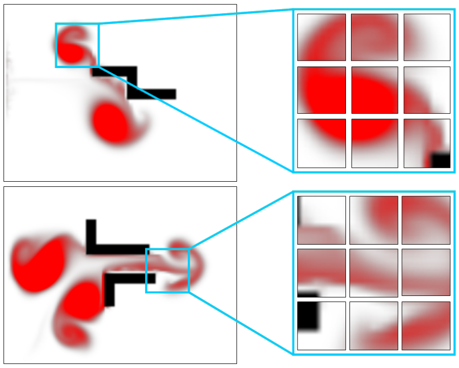


Figure 2. Frames of fluid simulation examples and a detail of fragments for each

We note as well that the different visual effects that can be obtained by this technique are limited to the fluid motion variations in the database. Extending the examples database would allow simulating more diverse solid-fluid interactions.

4 Conclusion

We present an approach to synthesize fluid motion from examples. Statistical analysis is performed on the examples and they are combined in order to approximate fluid behavior. This approach is suitable to be used for real-time applications such as videogames and interaction with virtual environments. In the future we aim to extend our method to properly simulate fluid interaction with moving solids. Also we aim to provide more animation control by allowing the user to specify the velocity in specific fragments in different locations and times, so fluid motion must be synthesized to match the user constraints.

References

[1] Y. Dobashi, M. Sato, S. Hasegawa, T. Yamamoto, M. Kato and T. Nishita. A fluid resistance map method for real-time haptic interaction with fluids. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology VRST'06* (Limassol, Cyuprus, November 1–3, 2006), pages 91–99, November 2006.

[2] N. Foster and D. Metaxas. Realistic animation of liquids. In *Graphical Models and Image Processing*, volume 58, issue 5, pages 471–483, 1996.

[3] M. Kass and G. Miller. Rapid, stable fluid dynamics for computer graphics. In *ACM SIGGRAPH Computer Graphics*, volume 24, issue 4, pages. 49-57, 1990.

[4] M. Müller, D. Charypar and M. Gross. Particle-Based Fluid Simulation for Interactive Applications. in *Proceedings of ACM SIGGRAPH Symposium on Computer Animation (SCA) 2003*, pages 154–159, 2003.

[5] J. Stam. Stable fluids. In *Proceedings of the 26th annual conference on Computer Graphics and Interactive Techniques SIGGRAPH '99*, pages 121–128. ACM Press/Addison-Wesley Publishing Co. 1999.

[6] A. Treuille, A. Lewis and Z. Popović. Model reduction for real-time fluids. In *Transactions on Graphics*, volume 25, issue 3, pages 826–834. ACM Press, July 2006.

[7] M. Vines, J. Mora and W.-S. Lee. Haptic Display of 3D Liquids for Interactive Applications. In *IEEE Consumer Electronics Games Innovation Conference ICE-GIC 2009* (London, United Kingdom from August 25-28), 2009.

A Hybrid Image-Based Method to Generate Sketching Portrait

Ling Xu* David Mould†

School of Computer Science, Carleton University, Canada

Abstract

In this paper, we describe a hybrid algorithm to create a sketch-like portrait from an input image. First, a user manually selects the face region from the input image. We use an image based method to render the subjects facial features and enhance the shading areas. A stroke based method is then applied to non-face regions to render the silhouette, creases and some dark regions. We show some examples of sketching portraits created by our method and make comparisons to original input images.

Keywords: Sketch, portrait, NPR.

Index Terms: I.4.3 [Image Processing and Computer Vision]: Enhancement—Filtering;

1 Introduction

Sketching is a classical drawing style that appears in a variety of contexts. Computer generated methods are often used to convert an input image into sketchy styles. But the results are not satisfactory due to the absence of shading variations, vital to express facial information.

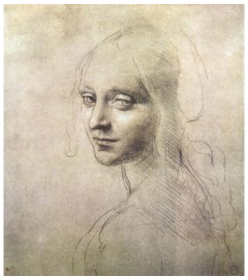


Figure 1: The sketch ”Angel for the Madonna of the Rocks”.

Commercial image processing tools concentrate on edge and silhouette features. Our method includes such features as well, but focuses on rendering a shading effect inspired by the sketching style of Leonard da Vinci (as shown in Figure 1) [3], which emphasizes on the shading and facial features and deemphasizes the surroundings. Our algorithm shares some characteristics with Brooks's method of mixed media portrait rendering [2] such as segmentation and interest in image detail, but our method involves feature and shading enhancement.

2 Algorithm

We chose human portrait as our sketch subject due to the great interest from the general public. The objective of our sketching style bears some similar characteristics as Figure 1:
- Exaggerated shading in the face region;
- Curved strokes describing the silhouette and creases;
- Straight-line strokes rendering the dark regions;
- Background blurred away from the face region with a smooth transition.

We implement a method for the above characteristics in the following two stages. In the first stage, based on the given input image, we use image processing to enhance the shading and feature information in the face region. We also blur the image to attain the objective of blurred background. In the second stage, we place curved strokes to render the silhouette and creases and use straight strokes to render the dark regions. We finally perturb the resulting image with a random value to simulate the granulated look of paper texture.

2.1 Shading and Feature Enhancement

In order to manage the shading and feature information, we use the tone management approach introduced by Bae et al [1]. For a given input image, we apply a bilateral filter to get the base layer and the detail layer. For the base layer which contains the shading information, we want to control the proportion of pixels with low intensity, medium intensity and high intensity; we achieve this with histogram matching. Since the detail layer bears the feature information which is generally contained in the low intensity pixels, we discard the high intensity portion by clipping the high intensity part of the histogram. With a similar method we get the feature enhanced detail layer. The process is shown in Figure 2.

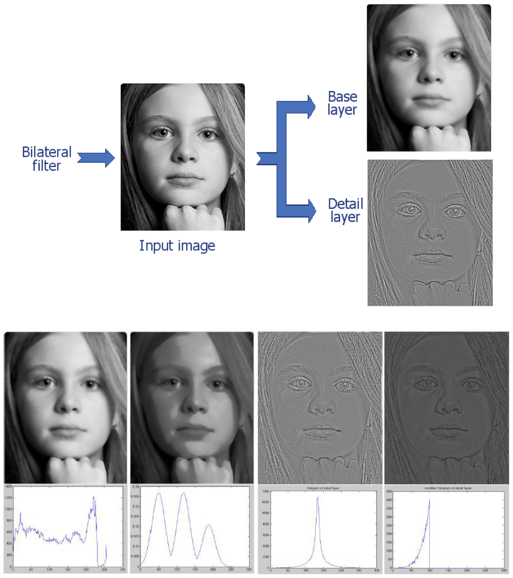


Figure 2: The shading and feature enhancement on two layers.

We then recompose the shading enhanced base layer and the feature enhanced detail layer. To simulate the effect of a background blurred away from the face region, we blurred the non-face region as shown in Figure 3. The face region and non-face region of the original image is segmented manually.

*e-mail: lxuc@scs.carleton.ca
†e-mail: mould@scs.carleton.ca

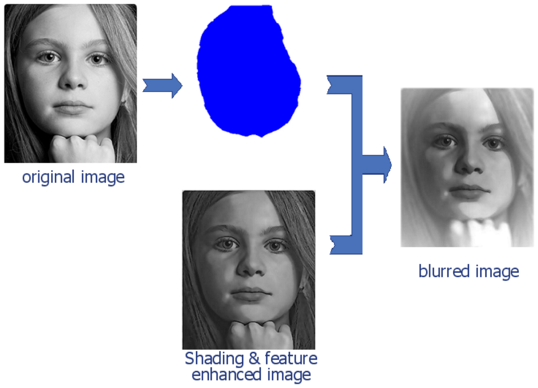


Figure 3: The process to blur the non-face region of the shading and feature enhanced image.

2.2 Stroke Rendering

The strokes of curves that describe the basic shape such as the silhouette and creases are obtained by Canny edge detection. The process is shown in Figure 4. Straight lines are used to render the shading and to imitate some quick pencil scratches which often appear in many sketch works. More strokes are placed in low intensity regions of the image. The result is shown in Figure 5.

3 Results and Discussion

Figure 6 shows some results obtained by the above method. For some input images with flat shading such as the first image, we successfully enhance the original flat shading to three distinct shading levels such as the dark shading in the areas of eyes and cheeks, medium dark shading in the nose, and highlights in the forehead and chin. For the input images with more obvious highlight regions and dark regions such as the second image, we keep the original well shaded information (such as the nose region) and enhance some features (such as the eye region).

Our current algorithm still has some drawbacks. The rendering result depends on the quality of the input image. If the input image is of low resolution, the shading and feature information is difficult to enhance. If the input image contains large areas of flat shading, say a large area of pure white or gray, it is hard for the algorithm to enhance the shading into distinct levels. Another drawback is the segmentation of input image into face region and non-face region. In the current implementation, users need to do the segmentation manually; in principle, face recognition could be used to extract a mask automatically.

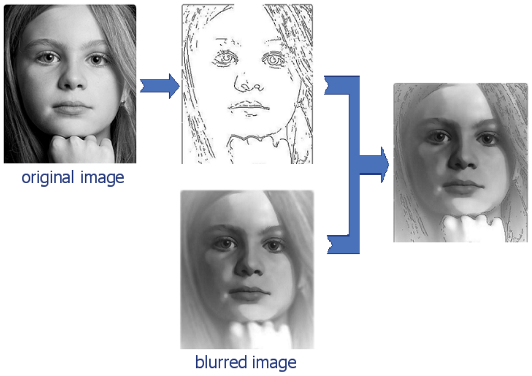


Figure 4: The process to render silhouettes and creases.

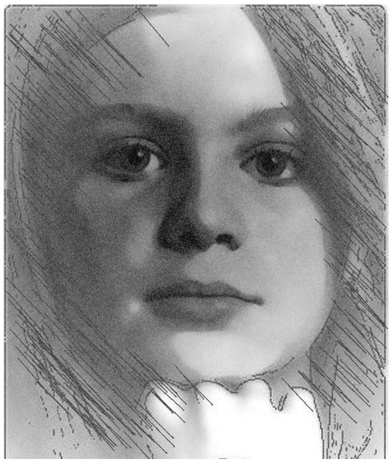


Figure 5: The process to apply strokes on a blurred image.

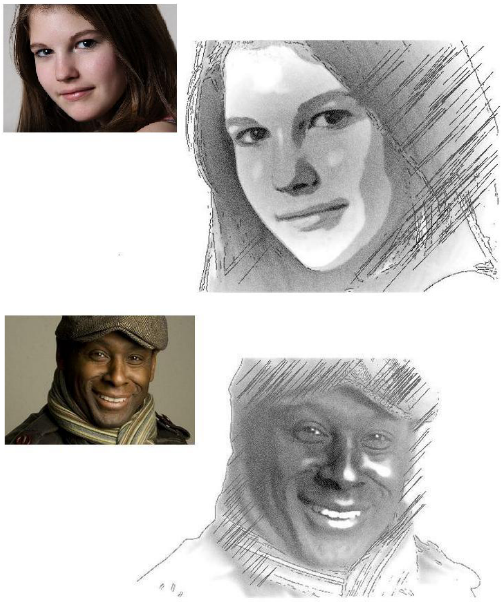


Figure 6: Input images and the output sketch-style portraits.

References

[1] S. Bae, S. Paris, and F. Durand. Two-scale tone management for photographic look. *ACM Trans. Graph. 25,3*, pages 637–645, 2006.

[2] S. Brooks. Mixed media painting and portraiture. In *IEEE Transactions on Visualization and Computer Graphics*, 2006.

[3] M. Clayton. *Seven Florentine Heads: Fifteenth Century Drawing from the Collection of Her Majesty The Queen.* Art Gallery of Ontario, Toronto, 1993.

Modeling Human Motion and Contact with Gaussian Processes

Olivier Rémillard[*] Paul Kry[†]

School of Computer Science, McGill University

Abstract

We introduce a simple approach for using Gaussian processes to model human motion involving contact. It comprises a low dimension latent space with dynamics augmented by switching variable. A Gaussian process models the time relationship of a motion while the switching variable helps model the discontinuities created by interaction with the environment.

1 Introduction

Generating realistic and lifelike animated characters from captured motion sequences is a hard and time-consuming task. The task is challenging due to the high dimensionality of human pose data and the complexity of the motion. Gaussian processes (GP) are useful for modeling the dynamics of human movement when combined with a latent variable model to approximate the lower dimensional manifold of human motion. These models alleviate the difficult problem of explicitly modeling physics and control, while providing a means of predicting behaviour, with applications in tracking and motion capture reuse.

The GP model will give good predictions if the latent trajectories are smooth. However, in many cases, the latent trajectories are not smooth because they include abrupt changes when the actor's body motion suddenly changes. It occurs often when a knee or an elbow joint reaches its full extent and locks or again when the actor interacts with the environment. These interactions can be as simple as stepping on the floor, pushing or pulling an object. Our solution borrows ideas from Switching Gaussian Process Dynamical Model (SGPDM) [1]. However, instead of using the switching variable to separate different motions, say walking and running, we use the switching variable to separate the nonsmooth dynamics acting on the motion into distinct sets. This separation further permits the use of simpler techniques such as principal component analysis (PCA) to reduce the dimension of the original problem.

2 Approach

We represent 3D human motion as a sequence of joints angles to describe how the pose changes over time, thus, a pose can be packaged in a vector, and a motion in a matrix. Assuming a first order Markov dynamic, modeling human motion is the task of computing

$$p(q_{t+1}|q_t),$$

where q_{t+1} denotes the next pose following the current pose, q_t. However, due to the high dimensionality of a pose, over 60 dimensions, most modeling techniques yield poor results. It is preferred to reduce the size of the input space.

We reduce the dimension of the space with a linear mapping

$$z_t = f(q_t),$$

where q is the pose and z its low dimensional latent coordinates. Since f is a linear transformation, we can chose f such that its inverse f^{-1} exists and use it to transform latent coordinate back to poses

$$q'_t = f^{-1}(z_t).$$

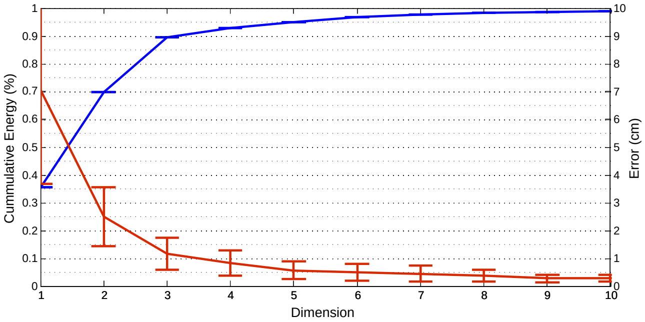


Figure 1: Effect of PCA Reduction on foot location over a walking sequence of 125 poses. The blue curve depicts the cumulative energy for a specific dimension. The red curve shows the mean error on the foot location of the reconstructed sequence in comparison to the original sequence. The error bars show 2 standard deviations.

The result, q'_t, is not equal to the original pose because f is a projection and the inverse does not reconstruct the original full space.

Figure 1 illustrates the trade off between the latent space dimension and the resulting reconstruction error at an end effectors of the body. Because of the hierarchical description of joints and angles of a pose, the error is accumulated as a body part is deeper in the hierarchy. Consequently, the error is greater at the feet and hands. As such, we use discrepancies of the foot location in q_t and q'_t as an indicator of the overall quality of the mapping. We choose to use the three first principal component of the observed pose data as transformation f because this captures 90% of the variation in the original motion (see Figure 1).

With dimension reduction, the model simplifies to

$$p(z_{t+1}|z_t).$$

We will model the time dependence between two consecutive poses with a GP. This is a non-parametric approach for solving regression problem.

Given a training motion sequence $Q \in R^{NxD}$, and its latent sequence $f(Q) = Z \in R^{Nxd}$, we can model the dynamics with

$$p(Z_+|Z_-,\theta) = \frac{\exp\left(-\frac{1}{2}\mathrm{trace}(Z_-^T K^{-1} Z_-)\right)}{\sqrt{(2\pi)^{(N-1)d}|K|^d}},$$

where K is the process kernel (covariance of the inputs), θ the kernel's hyper parameters, and $Z_+ = [z_2, ..., z_N]^T$ is the vector of states that follow $Z_- = [z_1, ..., z_{N-1}]^T$. The GP is maximized via

$$\theta = \arg\max_{\theta} p(Z_+|Z_-,\theta)$$

by optimizing the log likelihood of $p(Z_+|Z_-,\theta)$ using scaled conjugate gradient methods. Once trained we obtain

$$\begin{aligned} p(z_{t+1}|z_t) &= N(\mu(z_t), \sigma^2(z_t)), \\ \mu(z) &= Z_+^T K^{-1} k(z, Z_-), \\ \sigma^2(z) &= k(z,z) \;-\; k(z,Z_-)^T K^{-1} k(z,Z_-). \end{aligned}$$

[*] e-mail: remillardo@msn.com

[†] e-mail: kry@cs.mcgill.ca

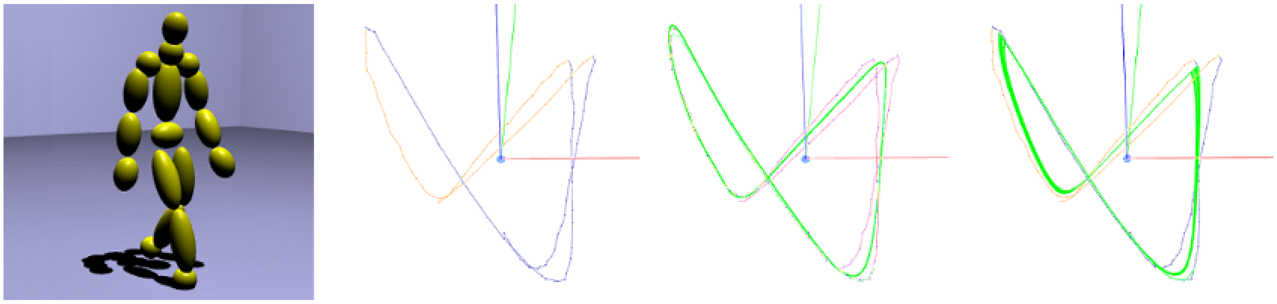


Figure 2: Walk. from left to right: a pose, the trajectory in latent space, inference of the model with no switching, inference with switching. The sequence consists of one and half cycles of walking for a total of 125 poses. The sequence is separated in 2 states S:contact with left foot (blue) and contact with the right foot(yellow).

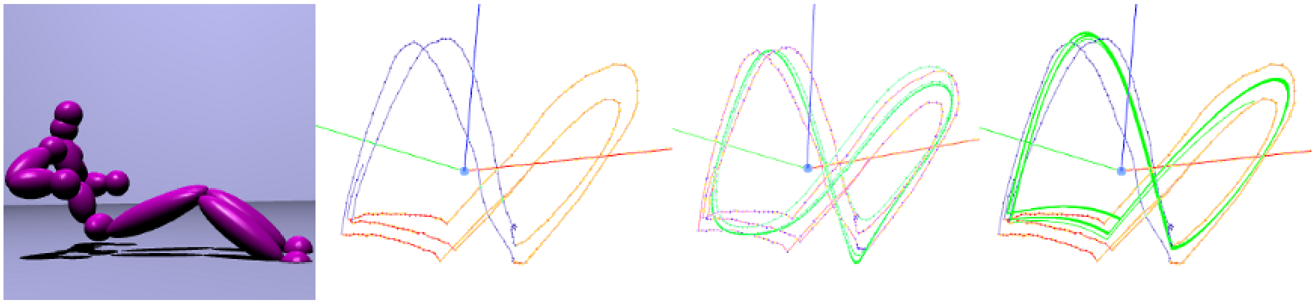


Figure 3: Rowing machine. from left to right: a pose, the trajectory in latent space, inference of the model with no switching, inference with switching. The sequence consists of two and half cycles of paddling for a total of more than 200 poses. The sequence is divided into 3 states S: pushing on the oars (yellow), pulling on the oars (blue) and pause between pushing and pulling (red).

where $k(z, Z_{-})$ is the covariance function applied to the input z and the training set inputs Z_{-}. Details of each step are found in [2, 3].

The linear transformation and the Gaussian Process described so far are sufficient to model some motions. Given an initial pose q_0, we seek q_t the tth pose following q_0. We start with $z_0 = f(q_0)$ and iteratively use the mean of the Gaussian process to obtain q_t by

$$q_t = f^{-1}(z_t),$$
$$z_t = \mu(z_{t-1}).$$

2.1 Switching Models at Contact

The problem with the usual formulation of the GP model is that it tends to smooth sharp turns in the trajectory. These discontinuities are characteristic to the contact forces acting on the human in motion, and those contacts should also be correctly modeled.

To cope with this predicament, we will add to the model a switching variable $s \in S$ to divide the motion into smaller subsets. Each variable should describe a situation where specific contact forces are in action. For example, in the walking situation we could have 4 values, $S = \{\text{no contact, left foot, right foot, both}\}$.

The switching variable permits the decomposition of $p(z_{t+1}|z_t)$ along the values of S, and the training of $|S|$ individual GP models [1]. Besides, a mapping from latent space coordinates to switching values can be expressed as a GP classification problem as explained in [2].

We can infer in this model the same way we did with the previous formulation. The main difference being the use of more than one Gaussian process. When stepping in the latent space $z_t = \mu(z_{t-1})$, we should only use the mean function of the GP trained along the switching value of z_{t-1}.

3 Results

Figures 2 and 3 show two examples where model switching is of benefit (a walking motion, and paddling motion on a rowing machine). The figures show the inference of 1000 poses using a model without the switching variable, and 1000 poses using the switching variable. The differences of both models reside in the ability to model the discontinuities of the trajectory. The main consequence of smoothing the discontinuities for the walking sequence is the accentuation of foot skating. For the paddling sequence, this consequence is reflected in the elimination of the pause (red) between the pushing and pulling movements.

4 Conclusion

We presented a way to model human motion and contact using Gaussian processes and a switching variable. The models we produce are useful for generating arbitrary length sequences of cyclic motion that can be adjusted to fit specific situations, and we have the added benefit that the switching variable helps model discontinuities due to contacts. One limitation is that we must label the switches in the training data. As future work, it would be interesting to use an unsupervised framework to choose labels that optimize the fit of the model.

References

[1] J. Chen, M. Kim, Y. Wang, and Q. Ji. Switching Gaussian process dynamic models for simultaneous composite motion tracking and recognition. In *Computer Vision and Pattern Recognition*, pages 2655–2662, 2009.

[2] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[3] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models for human motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30:283–298, 2008.

Deriving a 3D Femur from Multiple Radiographs

Gabriel Telles O'Neill* WonSook Lee†

The School of Information Technology and Engineering, University of Ottawa

Abstract

A Femoral Acetabular Impingement refers to a pathological condition where deformities of the hip bones deteriorate the joint's protective soft tissues. Unfortunately, the two primary imaging methods used to diagnose this condition (CT and X-Rays) both have major drawbacks. This paper describes on-going research towards a hybrid approach to eliminate these drawbacks by synthesizing CT-like results of the hip-bones by using multiple x-ray images as input. To accomplish this, Digitally Reconstructed Radiographs are used to create a mapping between the two imaging methods and Principle Component Analysis is used to express the variety of shapes for 3D femurs.

Keywords: Femoral acetabular impingement, digitally reconstructed radiographs, computed tomography, radiographs.

Index Terms: J.3 [Life and Medical Sciences]: Health; I.4.5 [Image Processing and Computer Vision]: Reconstruction—Transform Methods

1 Introduction

A healthy hip functions much like a ball-and-socket joint composed of two contact bones. These bones are the femur and acetabulum, located in leg and hip respectively. Hip motion is facilitated by layers of soft tissue between these two bones, such as the articular cartilage which covers the contact surfaces and the labrum which seals the hip joint.

A Femoral Acetabular Impingement (FAI) describes a pathological condition where there exists a bony bump-like deformity on either the femur's head (ball) and/or the acetabular rim (socket) which causes abnormal contact between the two bones during normal hip flexion. If this condition is left untreated, the joint's integrity will degrade over time due to increased damage to the soft tissue. FAIs have been associated with the development of cartilage damage, labral tears, early hip arthritis and lower back pain, among other complications. As such, quickly identifying and treating a patient suffering from a FAI reduces the likelihood of irreversible damage being incurred[1].

Physicians typically have two primary sources they can examine for diagnosing FAIs: (1) radiographs (x-rays images), (2) Computed Tomographic (CT) volumes. Unfortunately, both these sources have weaknesses to balance out their strengths. CT scans provide physicians with a 3D view of a patient's pelvis, thus no details of patient's hip can be obscured. However the CT machines themselves are quite large, expensive and can have a waiting period of three or more months for a scan. On the other hand, x-ray machines have a lower cost, higher availability and lower radiation exposure compared to CT machines. Conversely, radiographs suffer from having a single view and flattened details.

The goal of this research is to supply physicians with a new diagnostic tool that uses two or more pelvic radiographs as input and in return provides a 3D approximation of the femur and acetabulum. The advantage of this hybrid approach is to combine the specific strengths of the CT and radiographic imaging methods without carrying over their disadvantages.

Related work aimed at achieving the same goal tends to use parametric models to express the variety of femur shapes. One such approach used by Baudoin et al.[2] was to use parameters such as the femoral head radius, neck shaft angle, neck length, etc. to describe a 3D femur's form and orientation. Alternatively, Kurazumea et al.[3] generated their 3D femur from parameters obtained through Principle Component Analysis (PCA) of polygonized segmentations of CT data. What differentiates our research from similar work is that (a) our parametric models are generated from CT voxel intensity values and (b) our data set includes patients diagnosed with FAIs.

2 Methodology

The following subsections provide a brief overview of the currently implemented and forthcoming steps required to accomplish this research's goal. To begin with, this research has concentrated on recreating the femoral head only as an experiment to validate the intended methodology before recreating CT-like results for whole femurs and acetabulums.

2.1 Segmentation

The data-set used in this research originates from 20 patients, 17 of which were suffering from FAI in at least one hip at the data's time of taking and 3 from a control group. For each patient there is one pelvic CT scan, one digital radiograph taken from the patient's front and another taken from the patient's right.

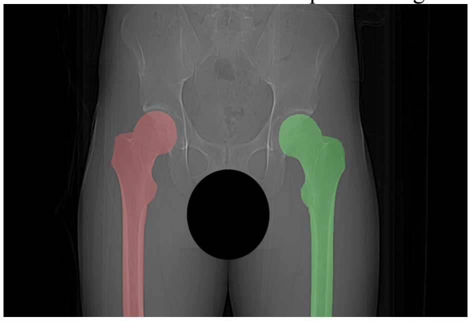


Figure 1. Segmented frontal radiograph. Pixels belonging to the patient's right/left femur are colored in red/green

Before this data can be interpreted, its relevant features had to first be segmented for extraction. Specifically, the pixels in the radiographic images and the voxels from the CT scans were labelled as belonging to the right femur, belonging to the left femur, belonging to both femurs or belonging to neither. Figure 1 shows an example of the pixels in a frontal radiograph that would be labelled as belonging to femurs. All segmentations were done semi-automatically using an in-house program and then corrected manually to eliminate possible machine errors.

*e-mail: gtell036@uottawa.ca

† e-mail: wslee@uottawa.ca

2.2 Digitally Reconstructed Radiographs

In order to create CT-like shapes from radiographs, a mapping scheme must be established between the 2D and 3D segmented objects. To accomplish this mapping, Digitally Reconstructed Radiographs (DRRs) are used. DRRs describe a number of volume rendering approaches which attempt to generate synthetic radiographs from CT data. In this research, DRRs are used to provide a flattened view at various angles of a CT volume. The femoral contours from these flattened views can be compared to those found on the corresponding radiographs. These overlapping contours (as well as their interiors) can be used to provide the necessary mapping.

Of the many approaches available for generating DRRs, raycasting is considered the standard[4] because it returns the highest fidelity results. As such, a raycasting scheme was adopted for this project and implemented to exploit the latest graphics hardware in order to offset its associated high computational cost. This scheme naturally lends itself to simulating the radiographic process inside a 3D graphics environment:

1. A point in space is chosen to represent the source of x-rays
2. The location of the radiographic plate is replaced with a similarly sized texture
3. For each pixel on the texture, a ray is cast from the point in space to that pixel
4. If a ray traverses the CT volume, the pixel associated with the ray accumulates the intensity values of all the voxels touched by the ray
5. The resulting texture is the DRR

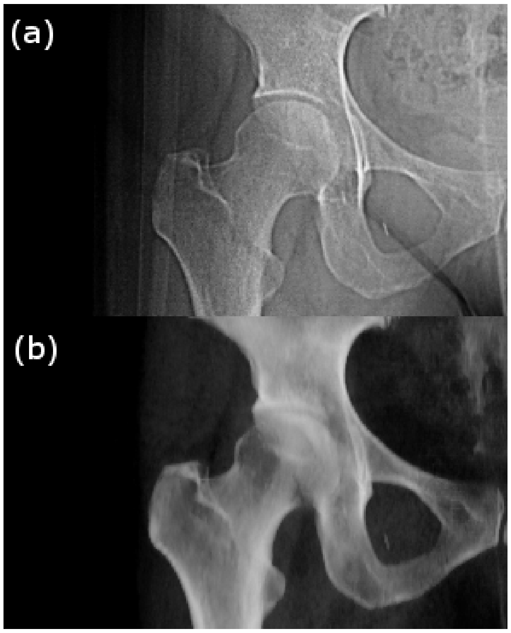


Figure 2. (a) Radiograph of a patient's femoral head (b) DRR of the same patient's femoral head with soft tissue subtracted

An additional complication exists in that CT data is captured using a different process than radiographs which results in there being a registration error between the DRRs and the radiographs along the patient's vertical axis. This is caused by a CT scan's slice-based approach, which is different from capturing the whole volume at once from a perspective point like an x-ray machine. Thankfully, this registration error can easily be removed by:

1. Resizing the CT volume along the patient's vertical axis according to the volume's distance from the "x-ray source"
2. Modifying the raycasting algorithm to use orthographic projection along the CT volume's vertical axis and perspective projection for the other two axes

An example of a resulting DRR can be seen in Figure 2.

2.3 Statistical Analysis of Femur Shapes

This step marks the research's current stage in development. The objective is to discover the parameters that best describe the variations in 3D femur shapes- especially those that might suggest the presence of FAIs. This phase will be performed as follows:

1. **Region of Interest Selection.** The femoral heads found in each patient's segmented CT data are fitted with a 3D bounding box
2. **Normalization.** The contents of each bounding box are converted into a standardized format, dimension, position, orientation, etc.
3. **Vectorization.** The contents of the normalized bounding boxes are converted into 1D vectors of voxel intensities
4. **PCA.** Principle Component Analysis is used to analyze patterns in the set of 1D vectors
5. **Parameterization.** The average femur shape is found in the PCA space along with the most influential parameters

2.4 Creating a 3D Femur from Two Radiographs

Creating the 3D femur shape from the 2D radiographs is the final phase of the proposed procedure. It will be done as follows:

1. Starting from the average femur shape, the PCA space is searched until the contours (and contents) of the parameterized femur's DRRs match best with those from the segmented radiographs
2. Reverse any normalization of the parameterized femur to obtain the corresponding 3D femur shape

3 CONCLUSION

This paper describes a process by which, two or more pelvic radiographs can be used to recreate the 3D shape of a patient's femur. This work's aim is supply physicians with an additional tool to diagnose FAIs. Areas of future work include reconstructing 3D acetabulums along with whole femurs using the same radiographs as source material.

4 ACKNOWLEDGEMENT

Funding for G. Telles O'Neill's research was provided from the Natural Sciences and Engineering Research Council (NSERC).

REFERENCES

[1] M. Hossain, J. Andrew. Current management of femoro-acetabular impingement. *Current Orthopaedics*, Volume 22, Issue 4, pages 300-310, 2008.

[2] A. Baudoin, W. Skalli, J. A. de Guise, D. Mitton. Parametric subject-specific model for in vivo 3D reconstruction using bi-planar X-rays: application to the upper femoral extremity. *Medical and Biological Engineering and Computing*, Volume 46, Number 8, pages 799-805, 2008.

[3] R. Kurazumea, K. Nakamuraa, T. Okadab, Y. Satoc, N. Suganoc, T. Koyamad, Y. Iwashitaa, T. Hasegawa. 3D reconstruction of a femoral shape using a parametric model and two 2D fluoroscopic images. *Computer Vision and Image Understanding*, Volume 113, Issue 2, pages 202-211, 2009.

[4] B. Russakoff, T. Rohlfing, D. Rueckert, R. Shahidi, D. Kim, C. R. Maurer Jr. Fast calculation of digitally reconstructed radiographs using light fields. *Medical Imaging 2003: Image Processing*. Proceedings of the SPIE, volume 5032, pages 684-695, 2003.

Learning Control Policies for Virtual Grasping Applications

Sheldon Andrews* Paul Kry† Doina Precup‡

School of Computer Science, McGill University

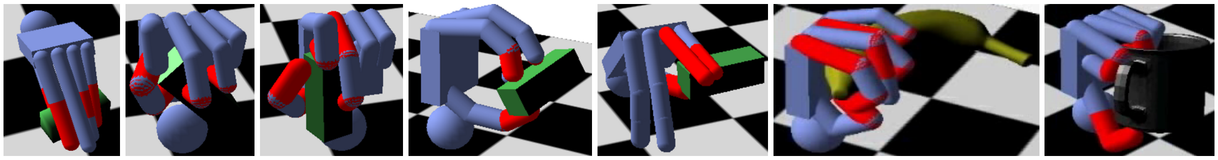


Figure 1: Grasps synthesized by our reinforcement learning framework, including two at right on objects not included in the training episodes.

1 Introduction

Human grasping is one of the most challenging problems in computer animation. Physically based grasp synthesis involves coordination and contact, and depends on many variables such as the shape, size, texture, and physical properties of the target object. The traditional approaches have originated in the robotics community and employ a combination of motion planning (to move within reach of a target) and contact planning (to achieve a stable grasp configuration). For example, the GraspIt! platform [4] has been used to develop such solutions to the grasping problem.

Grasping in computer animation has typically focused on control algorithms that involve the solution of carefully designed optimization problems [2], or in other cases only addresses the reuse of examples under specific initial conditions [5]. In contrast, we present a novel application of reinforcement learning (RL) to grasp synthesis in a physically based virtual environment. A set of basis controllers is used to move the hand along coordinated joint trajectories and a control policy is learned for choosing among them with the goal of automatically synthesizing grasps. Our approach is straightforward and preliminary results show success in learning stable grasps that generalize across objects (see Figure 1).

2 Experimental Setup

We use a hand model (depicted in Figure 2) that approximates the shape and kinematics of a human hand. Each finger has 3 phalanges and 3 joints, corresponding to finger segments and "knuckles", respectively. For simplicity, the visual and physical representations use capsules to model the finger segments and a box for the palm. The joint connecting the first segment of each finger to the palm is modelled as a universal joint, whereas the others are rotary joints with a single degree of freedom.

The state representation is relative to a coordinate system centred at the palm and aligned with the hand (the coordinate system denoted O_H in Figure 2). A palm centric frame is used because the objective is stable grasp synthesis and we assume that the goal is always to grasp a target object. The state consists of 29 continuous state variables:

- $\vec{p}$ - 3D position of the target object in frame O_H
- $\vec{v}$ - 3D linear velocity of the target object in frame O_H
- $\vec{o}$ - 3D orientation of the target object in frame O_H, represented by Euler angles

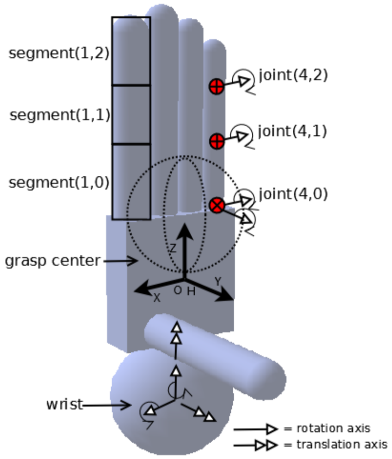


Figure 2: The hand model with reference frame, O_H, centred at the palm. A few selected finger segments are highlighted to show ordering, and joint rotational axes are shown, including the wrist. The wrist axis is shown (lower) and translation directions are denoted using double arrowheads.

- $\theta_{i,j}$ - hinge angle of the jth joint on the ith finger
- ϕ_i - abduction angle of finger i

The hand and graspable objects are simulated using a physics engine (CMLabs Vortex). Hand posture is managed by a PD controller (with stiffness resembling that of a human hand) that actuates joints according to coordinated motion of joint angles or joint velocities, e.g., open, close, adduct the fingers, pinching pose.

3 Reinforcement Learning

We use a combination of value iteration and SARSA to explore the state-action space and compute an optimal control policy approximation (see [6] for details). The value iteration algorithm begins by selecting one of the user-defined states from an *initiation set*, which is user-defined, and rigorously explores all possible state-action pairs by running the agent until a maximum number of steps n is reached. The initial state set is user-defined and allows the user to direct the agent toward regions of the state space which are relevant. The pseudo-code for the recursive function used to perform value iteration is given in Algorithm 1. Here, $\vec{s}$ represents the current state of the task, as described in Section 2. An action, a, is represented by a PD controller that moves the hand in some coordinated motion, also described in Section 2.

*e-mail: sheldon.andrews@mail.mcgill.ca
†e-mail: kry@cs.mcgill.ca
‡e-mail: precup@cs.mcgill.ca

Algorithm 1 valueIteration($\vec{s}, a, n$)

```
if $n > 0$ then
    save($\vec{s}$)
    for all $a' \in A$ do
        $\vec{s}' \leftarrow$ nextState($\vec{s}, a$)
        $R \leftarrow$ reward($\vec{s}, a$)
        $v \leftarrow$ valueIteration($\vec{s}', a', n-1$)
        $Q_{\vec{s},a} \leftarrow R + \gamma v$
    end for
    return $\max_a Q_{\vec{s},a}$
end if
return 0
```

This value iteration step is followed by training episodes of SARSA and ε-greedy exploration, essentially "rounding out" the value function. The capacity to inject strategies into the learned policy is useful for leading the policy toward grasping behaviours that are desired in a given scenario. We provide the user with this ability by making direct changes to the value function, which are further refined in subsequent training episodes leading to desirable behaviours in nearby states.

The value function, $Q_{\vec{s},a}$, is represented using a nearest neighbour function approximator, constructed incrementally during learning (see Algorithm 1). The current state is added to the value function if its distance to the nearest neighbour exceeds a *novel state* threshold η (similar to the trusted state policy used by [1]). Otherwise, the current state is aliased by its nearest neighbour, and the value of its neighbour is updated. At each step, the current state is stored and used to query the agent for an action. The new action is given control of the hand and the simulation is advanced using forward dynamics. Post step, a reward (or penalty) is given to the agent based on the new state.

As recommended by [3], the reward function, *reward*($\vec{s}, a$), is a combination of *immediate* and *delayed* rewards, chosen to accelerate the learning process. Immediate rewards encourage the agent to take actions that will likely lead to good grasping configurations in the short term; delayed rewards, which are given more sparsely, are large bonuses given to the agent when it finally achieves some milestone.

Immediate rewards. An immediate reward, R_p, is given when the agent moves toward the *position* of the target object and is calculated as the decrease in Euclidean distance between the target object and grasping centre compared to the previous time step. The reward R_c is given when the agent chooses an action that increases the number of finger segments in *contact* with the target object; a penalty is given if the number of finger segments is reduced. The value is simply equal to the number of contacts. Typically, large relative velocity between the hand and object will not lead to successful grasping, so penalty, R_v, proportional to the magnitude of the linear *velocity* of the target is given. Another negative reward, R_e, equal to the *effort* required to performing an action, is used to dissuade the agent from choosing irrelevant actions. The effort is estimated as the sum of the magnitudes of the torques at each joint integrated over the simulation time step.

Delayed rewards. A very large delayed reward, R_q, is given when the hand finally achieves a grasp, and is based on the *quality* of the grasp. The reward is calculated as the minimum distance from the origin of the set of *wrench* vectors (combined force and torque) due to contact between the hand and the target object. The grasp is stable when the *convex hull* of the available contact wrenches contains the origin of the wrench space; the ability of the grasp to resist perturbations improves as the distance from the origin to the surface of the hull increases. We filter transient grasps by disregarding the grasp quality metric when the velocity of the target object projected onto the palm-aligned axis (see Figure 2) is positive.

The total reward R is calculated at every simulation time step as

$$R = w_p R_p + w_c R_c + w_q R_q - w_v R_v - w_e R_e,$$

where the w factors denote the weight of the reward components. For our experiments, we used $w_p = 0.1$, $w_c = 0.5$, $w_q = 1000$, $w_v = 0.2$, and $w_e = 0.001$. The weights for the reward function were manually adjusted until the agent performed well during preliminary trials. This was largely a trial-and-error process based on observation and required some intuition.

4 Results

Our experiments were performed using a 2.6 GHz Intel quad-core processor and 4 GB of memory. The time required to run a typical episode was ≈ 2 seconds. The simulation was allowed to run faster than real-time, achieving frame rates of $120-500$ frames per second, and a mean of ≈ 250 frames per second.

The average time required to query the control policy was ≈ 2.9 ms, and involved performing a nearest neighbour search of about 10000 states. The average time for computing the grasp quality reward R_q was 21.4ms.

Figure 1 shows grasps synthesized using our method, including two experiments (at right) where the policy was tested with objects not seen in the training episodes. The agent was trained with an initiation set of 6 states using the test object (green box). After freezing the control policy, the agent was tested using a series of unseen test objects, in this case, a coffee mug and a banana. The agent was able to successfully grasp each of these objects, however a limitation of the approach is that the agent sometimes produces grasps that are either marginally stable or aesthetically awkward.

5 Conclusion

The preliminary results suggest that it is possible for an RL agent to learn a control policy enabling a virtual hand to do stable grasping of target objects. The agent successfully controlled the hand to perform grasp synthesis, including grasping moving objects and objects of unseen shape and size. While the learning process is automated (the user need only select an initiation set), it also allows for user interaction at different stages through the injection of example strategies.

Generating a robust control policy involves protracted computing time. Our training times could be improved by increasing the efficiency of the convex hull algorithm used to calculate the grasp quality, and the value iteration and SARSA algorithm could be parallelized and offloaded to multiple CPUs. Learning the reward function, e.g., by inverse reinforcement learning, may lead to better trade-offs between grasp quality and other aspects of the performance, and allow for variations on the grasping task. Finally, to improve the aesthetic quality of the synthesized motion, we are developing methods to generate controllers from a motion capture corpus.

Acknowledgements

This work was supported in part by grants from NSERC and GRAND NCE.

References

[1] S. Coros, P. Beaudoin, and M. van de Panne. Robust task-based control policies for physics-based characters. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 28(5), 2009.

[2] C. K. Liu. Dextrous manipulation from a grasping pose. *ACM Transactions on Graphics*, 28(3):1–6, 2009.

[3] M. J. Mataric. Reward functions for accelerated learning. In *Proc. 11th International Conference on Machine Learning*, pages 181–189, 1994.

[4] A. Miller and P. Allen. Graspit! a versatile simulator for robotic grasping. *IEEE Robotics and Automation Magazine*, 11(4):110–122, 2004.

[5] N. S. Pollard and V. B. Zordan. Physically based grasping control from example. In *Proc. 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 311–318, 2005.

[6] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

Real-time Seismic Wave Modeling and Visualization

Aaron Maynard* Minglun Gong†

Memorial University

Abstract

Current rising energy demand/supply ratio trends are forecast to continue into the foreseeable future. As a consequence, effective and efficient seismic surveying and processing techniques are required now more than ever. Unfortunately those outside of organizations with the resources to develop their own processing and/or visualization packages, or license proprietary third-party ones, must make due with text-based processing systems driven by shell scripts. While these systems perform many of the important algorithms in the seismic processing pipeline with adequate accuracy, the slow performance and limited 3-D processing support along with the learning curve associated with the command-line interface leaves much to be desired in terms of usability .

To address these concerns, an interactive seismic simulation and visualization system has been designed and developed as a solution to enable geophysicists to process and manipulate seismic data in a more natural and intuitive manner. Seismic event simulations are accelerated on the graphics processing unit (GPU) to framerates suitable for real-time visualization, which permits fine control over simulation and visual parameters, allowing them to be modified by the user while the simulation is running. This allows the geophysicist to view the results of any modifications instantaneously, without having to re-start the simulation and generate a pre-computed video file as in prior systems. This paper will focus on the algorithms used to implement this system as well as the resulting improvements in geophysicist work-flow and data gathering efficiency that result from employing real-time simulation techniques; preliminary performance benchmarks will be given along with a brief conclusion.

1 Introduction

Mechanical waves originate when a sudden stress is applied to a suitable medium, causing a lack of equilibrium which is propagated from the source of impulse[3]. For seismic waves, the propagation medium is the Earths crust. Thus the characteristics of seismic waves, such as amplitude, frequency, and wave velocity; are determined by the laws of physics governing waves, and the composition of the Earths sub-surface. Inversely, if details of the seismic event are known, information regarding certain features of the subsurface can be deduced.

For over two decades[11], the leading open-source seismic processing toolkit has been Seismic Unix (*SU*). Developed at the Center for Wave Phenomena-Colorado School of Mines, SU has been adopted by universities as a research tool and by small to medium-sized companies to process the results of their seismic surveys. While it is robust in the sense that it is highly configurable and supports many seismic algorithms, it is also limited in several areas. SU runs in terminal mode only, meaning that any non-trivial processing sequence must be configured through shell-scripts; a feature that is useful to expert users but sharply increases the learning curve for geoscientists and others who may not have a programming background. SU has no 3D processing support, and although its results can be viewed as animations or images (as appropriate for the algorithm), the results must be pre-computed so animations are non-interactive and limited to a small 2D slice of the domain. This can be alleviated somewhat through clustering[10] as many of the algorithms are highly parallelizable. The fact remains that SU and many commercial packages[1] do not feature real-time processing concurrent with interactive visualization.

The aim is to not only demonstrate the aptness of GPUs for seismic processing and visualization, but also to improve the efficiency of the geophysicists workflow through its easy-to-learn user interface, specifically in the case of forward wave propagation modeling. Forward modeling is used for velocity model verification; that is, the spatial-domain models derived from raw field data, which have been subject to post-processing and depth-conversion[4]. The methods used to generate these models are highly sensitive to noise and random pertubations introduced by imperfections in the data recording device (geophone), thus the models are verified by simulating the seismic event that generated the data and comparing the simulated and field results. If the results are well-correlated then the model is considered to fit the data; if not, adjustments must be made based on the level of deviation from the measured field data. For geophysicists working on land or marine surveys, the cycle of data-gathering and model generation/verification can quickly become time-consuming and expensive for sufficiently complex structures; which is exasperated by the fact that processing resources are not typically located on-site, meaning a slight error in geophone array orientation, etc., can lead to numerous delays as data must be transported physically or accross slow satelite connections to the processing center for interpretation. These issues are addressed by using standard, consumer-grade programmable GPUs to accelerate the modeling pipeline; and providing an intuitive user interface which allows the geophysicist to simultaneously verify, visualize, and interpret the candidate model on-site with hardware compact enough to fit into laptop-sized forms, thus saving considerable time and associated financial resources.

2 Seismic Wave Propagation

2.1 Modeling Algorithm

$$\frac{1}{C(x,y,z)^2}\frac{\partial^2\Psi}{\partial t^2} = \frac{\partial^2\Psi}{\partial x^2} + \frac{\partial^2\Psi}{\partial y^2} + \frac{\partial^2\Psi}{\partial z^2} + src(t) \qquad (1)$$

Equation1: Acoustic Wave Equation

The Earth is an effective propagator of several forms of wave energy, as it is both an elastic and acoustic medium. Geophysicists conducting seismic surveys are mainly concerned with acoustic waves, as their higher propagation velocity ensures they are recorded first by the geophone array[2]. Acoustic waves can be modeled with the second-order partial differential equation given in (1), where C is the wave velocity at a point in the domain of interest; *x, y, z,* and t are spatial and temporal coordinates respectively; src(t) is the seismic source function; and Ψ is the wave pressure field. Solving this equation accross a domain Ω will gradually disperse the wave energy introduced by src(t) over a given Δt.

Because computing exact derivatives is far too costly to be considered for an interactive application, a set of finite difference (FD) approximations are used to calculate the wave propagation vector; more specifically an order-8 in space and order-2 in time kernel. The domain is decomposed into a uniform mesh; then at every

[*] e-mail: thomasm@mun.ca

[†] e-mail: gong@mun.ca

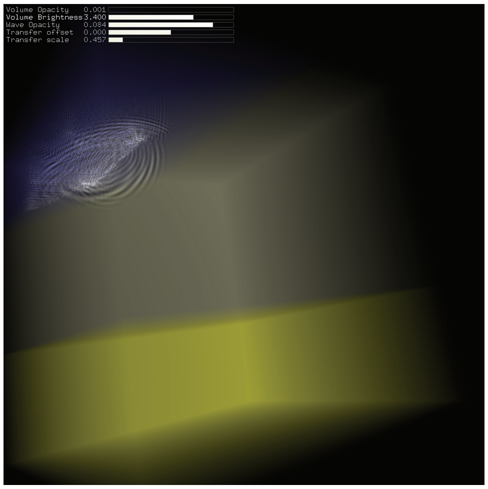
mesh element, the FD kernel is evaluated, which takes the half-order number of neighbouring elements in each spacial direction and the previous (in time) value of the current element, computes the differences between them, then sums the results modulated by a particular weight value, determined by the size of the kernel[6].

2.2 GPU Implementation

Seismic problems such as wave modeling are particularly suitable for GPU optimization, especially since the development of GPGPU oriented languages has eased the task of implementing parallel scientific algorithms on the GPU. The FD method is by its nature, a data-parallel algortihm in that each mesh element can be computed in any given order, which seems to fit nicely within the stream processor paradigm of the GPU. However, there does exist a memory dependency since each node needs to access its spatial and temporal neighbours. Temporal neighbours are handled by double-buffering the pressure field; i.e. the entire current and previous simulation frames are in memory at all times, once the value is read from the relevant node in the previous timeframe, it is no longer required and the results for the new frame are stored in its place, effectivly swapping the buffers at the end of each frame.

A slightly modified version of the method presented by Micikevicius[8] performs memory optimization for accessing the spatial neighbours. Each thread iterates along the z-axis, storing the current node in shared memory. Following a syncronization, neighbouring threads in the x-y grid can access each others node value through the low-latency shared memory, rather than having to read from global memory for each access. The modifications employed relate to integrating proper FD weights into the computation, and an implementation of energy-absorbing boundary conditions along the edges of the dataset (except for the top boundary, which represents the Earth's surface and should generate a reflection), optimized to execute only when required along the model boundary.

3 Model Visualization



```
Volume Opacity    0.001
Volume Brightness3.400
Wave Opacity      0.084
Transfer offset   0.000
Transfer scale    0.457
```

Figure 1: Seismic wave visualization featuring refraction and reflection at velocity interface

The simulation is visualized using a volume rendering method. Volume rendering allows varying transparency or color to be encoded into a transfer function that highlights interesting portions of a dataset and also allows the rendering of these variations to proceed efficiently. The physical-approximation approach of volume rendering enables high-quality image results and permits easily mixing surface, volume, and other primitives in a scene, while accounting for many of their optical interactions. This allows the geophysicist or other interpreter to resolve features with greater detail than traditional rasterization[7]. To aid perception and interpretation of data, the user is given abillities to interactively adjust viewing location and orientation, extract and isolate volume slices along any axis, and to adjust the transfer function scale in order to clearly visualize data accross a dynamic range of scales.

4 Preliminary Results

The system has been benchmarked on several classes of machines. On an NVIDIA GTX 285 GPU, the simulator solved the FD equations at a rate of approx. 130 GFLOP/s while rendering at a framerate of approx 25 FPS. The system was also tested on a mobile class GPU, an NVIDIA GTX 260M, which due to a lower shader count and power saving design performed considerably worse; about 5.5 GFLOP/s for the simulation and between 3-5 FPS.

5 Conclusion

The development of programmable GPUs over the last decade and inherent increases in performance and flexibility has opened up a new opportunity to advance interactive user interfaces in ways that were previously unfeasable. It has provided the means by which the default rendering pipeline can be altered to implement the direct volume rendering algorithm; and also to offload intensive computations onto the parallel platform to acheive performance increases vs. serial and distributed implementations, and real-time visualization speeds. The system presented here can also be generalized in order to model other scientific phenomena approximated by FD, such as weather/climate modeling[5] and heat diffusion[9]; and, along with the appropriate transfer functions, can visualize them so that the improvement in information conveyal and user interaction applied here to the geosciences can be extended to these and other scientific domains.

References

[1] *Visualization Reference for the Oil and Gas Industry*. Gulf Publishing Company, 2006.

[2] R. P. Bording. *Wave equation difference engine*. PhD thesis, University of Tulsa, 1995.

[3] J.-P. Cordier. *Velocities in Reflection Seismology*. Kluwer Academic Publishers, 1985.

[4] E. L. Etris, N. J. Crabtree, J. Dewar, and S. Pickford. True depth conversion: More than a pretty picture. *Canadian Society of Exploration Geophysicists Recorder*, 26:11–22, 2001.

[5] J. W. Finch and J. H. C. Gash. Application of a simple finite difference model for estimating evaporation from open water. *Journal of Hydrology*, 255(1-4):253 – 259, 2002.

[6] B. Fornberg. Calculation of weights in finite difference formulas. *SIAM Rev*, 40:685–691, 1998.

[7] H. F. Gerd Marmitt and P. Slusallek. Interactive volume rendering with ray tracing. Eurographics 2006 STAR Report, 2006.

[8] P. Micikevicius. 3d finite difference computation on gpus using cuda. In *GPGPU-2: Proceedings of 2nd Workshop on General Purpose Processing on Graphics Processing Units*, pages 79–84, New York, NY, USA, 2009. ACM.

[9] A. S. Mokhtar, K. A. Abbas, M. M. H. M. Ahmad, S. M. Sapuan, A. Ashraf, M.A.Wan, and B. Jamilah. Explicit finite difference solution of heat transfer problems of fish packages in precooling. *American Journal of Applied Sciences*, 1(2):115–120, 2004.

[10] A. E. Murillo and J. Bell. Distributed seismic unix: a tool for seismic data processing. *Concurrency: Practice and Experience*, 11:169–187, 1999.

[11] S. U. Website. *http://www.cwp.mines.edu/cwpcodes*. Colorado School of Mines - Center For Wave Phenomena, 2010.

Ambient and Artistic Visualization of Residential Resource Use

Johnny Rodgers*   Lyn Bartram†

Simon Fraser University, School of Interactive Arts + Technology

Abstract

Supporting sustainable resource use in the home requires a range of feedback techniques to enable informed decision-making. These techniques can include traditional screen-based interfaces, but these tools often require too much effort and attention from already-busy residents. An alternative approach is the provision of ambient and artistic visualizations integrated into the domestic environment. This method reduces the attention required of residents, increases aesthetic interest and coherence with the home, and enables situated and timely feedback on resource use. We present the theoretical basis of our research, discuss how we have applied it to the development of prototypes in two green home projects, and detail our ongoing efforts to evaluate techniques within this domain.

Keywords: Aesthetics, ambient visualization, residential resource use, sustainability.

Index Terms: H.5.0 [Information Systems: Information Interfaces and Presentation — General]

1 Introduction

Supporting sustainable resource use in the home requires a range of feedback techniques to enable informed decision-making. Sophisticated sensing and data collection mechanisms are making a wealth of information about resource use available to residents. Visualizing this data in ways that are meaningful and contextually appropriate will help to bridge the gap between data and decisions about how to sustainably use resources in the home.

However, as others have pointed out [1][12][13], and as we have argued previously [2], we cannot just import established visualization techniques to non-work environments. Contextualizing visualization for these environments means considering a range of user attention from ambient to attentive, appropriate placement of displays to support daily activities, and balancing aesthetic appeal and usefulness [13]. After all, we cannot expect residents to refer to graphs on their computer every time they make a resource use decision. By integrating ambient and artistic feedback into the home, we can increase awareness of resource flows subtly and beautifully.

2 Context

Our work in this area has grown out of two high-profile sustainable housing projects. The first, North House, is a net-zero home that placed 4th at the U.S. Department of Energy Solar Decathlon 2009, an international competition to design and build the most energy efficient solar-powered home. We designed and built the Aware Living Interface System (ALIS), an interactive visualization, control and social networking system to support informed energy and water use choices in North House. It includes embedded displays with both passive and interactive modes, web-accessible tools, mobile visualizations, and ambient informative art.

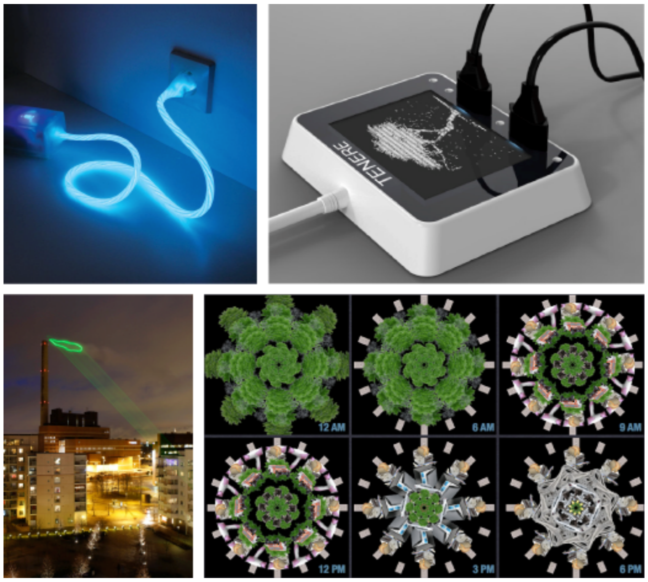


Figure 1. Clockwise from top-left: The *Power-Aware Cord* glows and pulses as electricity passes through it [4]. *The Ténéré* depicts a tree that dynamically changes to indicate power use [8]. *7000 Oaks and Counting* conveys carbon footprint as a kaleidoscopic artistic visualization, with trees mapped to low consumption and electronics to high [5]. *Nuage Vert* visualizes the energy consumption of a region of Helsinki. Decreased demand leads to a larger 'green' cloud projected onto the emissions of the local power plant [3].

West House, our second and current project, is a small footprint sustainable laneway home developed in partnership with the City of Vancouver and displayed at the Vancouver 2010 Olympic Winter Games. The second implementation of ALIS for West House includes new prototypes in informative art and ambient displays informed by our experiences with North House.

3 Related Work

Significant research has been done to understand ambient and artistic approaches to information visualization. Pousman and Stasko state that ambient information systems "display information that is important but not critical; can move from the periphery to the focus of attention and back again; focus on the tangible (representations in the environment); provide subtle changes to reflect updates in information (should not be distracting); and are aesthetically pleasing and environmentally appropriate [11]." Ambient approaches make up one facet of *Casual InfoVis*: "the use of computer mediated tools to depict personally meaningful information in visual ways that support everyday users in both everyday work and non-work situations [12]." Pousman et. al. draw attention to the potential *Casual InfoVis* represents for users outside of the traditional InfoVis paradigm, allowing practitioners to "develop for [the] idiosyncratic, private [...], and delicate nature of people's lives outside of focused episodes of work [12]."

Examples include Miller and Stasko's InfoCanvas, which artistically conveys peripheral personal information in order to

* e-mail: jgr3@sfu.ca
† e-mail: lyn@sfu.ca

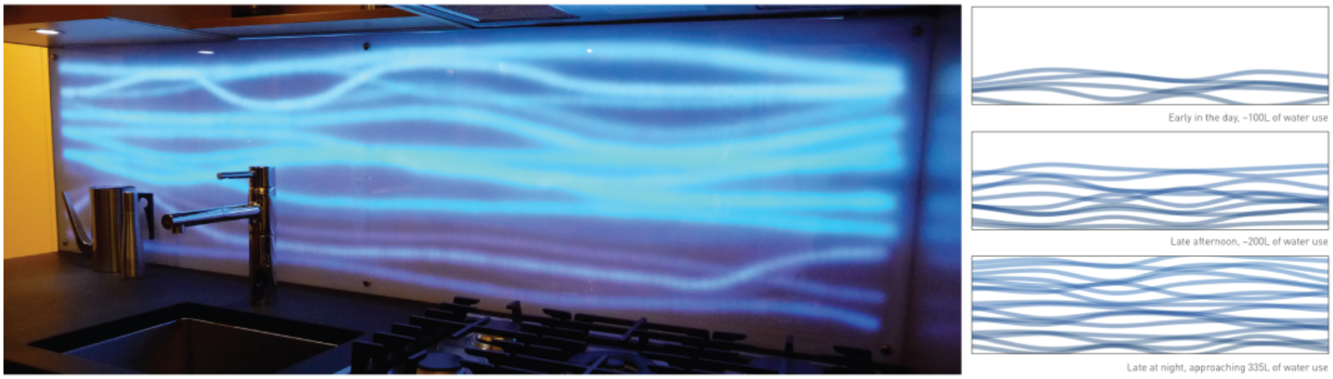

Early in the day, ~100L of water use
Late afternoon, ~200L of water use
Late at night, approaching 335L of water use

Figure 2. The Ambient Canvas is an ambient artistic visualization, pictured here embedded in the kitchen backsplash in West House. The canvas conveys relative levels of resource use in the home, as compared to past averages. As shown at right, one mode of display visualizes water use over time. Using the metaphor of a volume filling with water, the LED strings are lit and 'fill' the canvas over the course of the day as residents use water.

reduce information overload [10], and Skog et. al.'s informative artworks [7][13]. Skog et. al. highlight the importance of balancing aesthetic appeal and practical usefulness when applying these techniques. Along with Pousman et. al. [12], they note that traditional methods of evaluating the effectiveness of a display — typically considered in a work context — may not be the most important criteria to use when evaluating artistic displays. They argue that aesthetic factors and 'calmness' may be just as relevant, as these visualizations will be "lived with rather than used [13]."

We also draw on Kosara's distinction between artistic and pragmatic visualization. Where pragmatic visualization focuses on enabling the exploration and analysis of data sets, artistic visualizations primarily aim to "communicate a concern, rather than to show data [7]." Artistic displays may not be immediately recognizable as visualizations, and may require a period of learning to read accurately. Holmquist has proposed a framework for evaluating the comprehension of ambient displays over time, progressing from awareness *that* information is being visualized, to understanding *what* is being communicated, to finally comprehending *how* to make sense of that information [6].

4 PROTOTYPES

Existing instances of ambient and artistic feedback on resource use have informed our approach [see Figure 1]. We have explored a variety of concepts in this design space during the development of ALIS, including ambient indicators embedded in smart appliances, informative art applied as thermo-chromatic ink to interior surfaces, digital photo frames displaying dynamic landscapes, and light-based feedback on water use.

Our first major prototype is an embedded display combining artistic and ambient approaches that we call the Ambient Canvas [Figure 2]. Embedded in the kitchen backsplash, the display combines LED lights and a transmission medium (Corian®, acrylic, or glass) for diffusion. Feedback on energy, water, and natural gas use is conveyed through shifting patterns and varying intensities of light. Baselines of resource use are gathered from logged system data in order to compute averages and convey relative levels of use to residents.

5 EVALUATION

The next step in our research agenda is formal evaluation of these techniques. Anecdotal feedback received during the Solar Decathlon and Olympic Games made it clear that visitors were drawn to the ambient feedback prototypes. Reactions indicated aesthetic appeal and curiosity, but also demonstrated the importance of carefully mapping data to representation to support understanding. For example, during public demonstrations, the Ambient Canvas in West House was fully lit. In our model, this would indicate a high degree of resource use — something to be avoided! However, visitors remarked on the appearance of this mode, indicating that they enjoyed the look of it when fully lit. This suggests that our model may be backward: visually active states should perhaps be mapped to low resource use in order to encourage conservation. These and many other questions remain to be explored.

We are currently translating the Ambient Canvas to a software emulator environment in order to study how different approaches using this format are perceived by participants. In a lab setting, we will not be able to assess the effectiveness of these approaches in reducing resource use. However, we will be able to assess the visualization's perceptibility in different conditions, and collect data on participant's impressions of the prototypes.

In addition to experimental evaluation in the short term, our research agenda includes plans for longitudinal studies. Now at its permanent location, West House provides us with the opportunity to study the effect of these techniques on resource use in the home over time.

6 CONCLUSION

Ambient and artistic visualizations represent a promising approach to the provision of feedback on residential resource use. The examples we have discussed address some of the drawbacks of traditional visualization methods in this domain, enabling cohesive integration with residential spaces and increased aesthetic appeal while supporting informed decision-making. Our ongoing research aims to identify and evaluate successful techniques within this design space and situate them within a larger ecosystem approach to supporting sustainable living.

REFERENCES

[1] R. Aipperspach, B. Hooker, and A. Woodruff, "The heterogeneous home," *Proc. UbiComp*, pp. 222-231, 2008.

[2] L. Bartram, J. Rodgers and K. Muise, "Chasing the Negawatt: Visualization for Sustainable Living," *IEEE Computer Graphics & Applications*, 30 (3), pp. 6-12, 2010.

[3] H. Evans, H. Hansen, "Nuage Vert," Ruoholahti, Finland. [Online]. Available at: http://hehe.org.free.fr/hehe/texte/nv/index.html, 2008.

[4] A. Gustafsson and M. Gyllenswärd, "The power-aware cord: energy awareness through ambient information display," *Proc. ACM CHI Extended Abstracts*, pp. 1423-1426, 2005.

[5] T. Holmes, "Eco-visualization: combining art and technology to reduce energy consumption," *Proc. ACM SIGCHI on Creativity & Cognition*, pp. 153-162, 2007.

[6] L. Holmquist, "Evaluating the comprehension of ambient displays," *Proc. ACM CHI Extended Abstracts*, p. 1545, 2004.

[7] L. Holmquist and T. Skog, "Informative art: information visualization in everyday environments," *Proc. GRAPHITE*, pp. 229-235, 2003.

[8] J. Kim, Y. Kim, T. Nam, "The Ténéré: Design for Supporting Energy Conservation Behaviors," *Proc. ACM CHI Extended Abstracts*, pp. 2643-2646, 2009.

[9] R. Kosara, “Visualization Criticism—The Missing Link Between Information Visualization and Art,” *Proc. InfoVis*, pp. 631–636, 2007.

[10] T. Miller and J. Stasko, “Artistically conveying peripheral information with the InfoCanvas,” *Proc. AVI*, pp. 43-50, 2002.

[11] Z. Pousman and J. Stasko, “A taxonomy of ambient information systems: four patterns of design,” *Proc. AVI*, pp. 67-74, 2006.

[12] Z. Pousman, J. Stasko, and M. Mateas, “Casual Information Visualization: Depictions of Data in Everyday Life,” *IEEE Transactions on Visualization and* Computer Graphics, 13 (6), pp. 1145–1152, 2007.

[13] T. Skog, S. Ljungblad, and L.E. Holmquist, “Between Aesthetics and Utility: Designing Ambient Information Visualizations,” *Proc. InfoVis*, pp. 30–37, 2003.

Target Pointing in 3D User Interfaces

Robert J. Teather* Wolfgang Stuerzlinger†

Dept. of Computer Science & Engineering, York University, Toronto

Abstract

We present two studies using ISO 9241-9 to evaluate target pointing in two different 3D user interfaces. The first study was conducted in a CAVE, and used the standard tapping task to evaluate passive haptic feedback. Passive feedback increased *throughput* significantly, but not speed or accuracy alone. The second experiment used a fish tank VR system, and compared tapping targets presented at varying heights stereoscopically displayed at or above the surface of a horizontal screen. The results indicate that targets presented closer to the physical display surface are generally easier to hit than those displayed farther away from the screen.

Keywords: Target selection, pointing, tapping, virtual reality.

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems – virtual reality. H.5.2: User Interfaces – input devices, interaction styles.

1 Introduction

Target pointing is a fundamental task in computer interfaces, and is a basis for direct manipulation interfaces. The WIMP interface paradigm (Windows, Icons, Menus and Pointing device) is a good example, as virtually all operations are accessible by pointing the cursor at interface widgets. Pointing in 2D interfaces has received a great deal of attention and is well modeled by Fitts' law [4].

Pointing is also required in 3D direct manipulation interfaces, but elementary 3D pointing tasks have not received the same attention. Few attempts have been made to model 3D pointing tasks. Most research in virtual object selection and manipulation instead focuses on high-level techniques. Experimental designs vary between studies, thus it is difficult to generalize findings.

The ISO 9241 standard part 9 [6] describes a method for evaluating pointing devices. It is based on Fitts' law and ultimately computes throughput, which represents information capacity (in bits per second), which enables direct comparison between devices. We propose using this standard for 3D pointing, too. There have been few, if any, previous attempts to employ this methodology in evaluating 3D input devices. We examine some of the issues in extending the standard for use in 3D user interfaces in the context of the two experiments described below.

2 Target Pointing

The VR community has studied target pointing, i.e., object selection, extensively [1-3, 5, 7, 9]. Yet, elementary pointing tasks have not been formalized or modeled as well as in the 2D user interface domain. Most VR object selection techniques are based on either ray casting or virtual hand metaphors. Ray-based techniques cast a virtual ray into the scene from the user's hand, finger, or cursor and selects objects hit by this ray. The virtual hand metaphor requires users to intersect their hand representation with objects. Both paradigms use rapid aimed movement, as modeled by Fitts' law [4]:

$$MT = a + b \cdot ID , \quad \text{where } ID = \log_2\left(\frac{A}{W}+1\right) \qquad (1)$$

MT is the movement time, and *a* and *b* are determined via linear regression for a given technique. *ID* is the index of difficulty (in bits). *A* is the movement distance, and *W* is the target width. *ID* represents the task difficulty based on the target size and distance. Hence, small, far targets are harder to hit than large, near targets.

ISO 9241-9 employs a standardized pointing task (Figure 1) based on Fitts' law [6]. The standard uses throughput (*TP*) as a primary characteristic of pointing devices [6], which is given in bits per second as:

$$TP = \frac{ID_e}{MT} , \quad \text{where } ID_e = \frac{A_e}{W_e} \qquad (2)$$

where ID_e is the *effective* index of difficulty, and *MT* is the measured average movement time for a given condition. ID_e uses *effective* scores to account for the tasks users really performed, as opposed to presented task. Effective width is defined as:

$$W_e = 4.133 \cdot SD_x \qquad (3)$$

where SD_x is the standard deviation of the over/under-shoot projected on to the task axis (line between targets) for a given condition and A_e is the averaged actual movement distance.

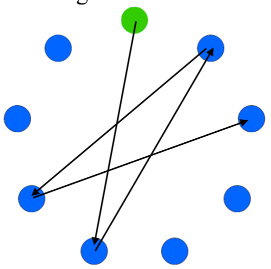


Figure 1. ISO 9241-9 reciprocal tapping task with nine targets. Arrows depict target order.

Throughput incorporates both speed and accuracy and is unaffected by speed-accuracy trade-offs [8]. It may also account for device noise common to 3D tracking technology [9].

3 Passive Haptic Feedback Study

It is generally accepted that haptic feedback improves the usability of immersive virtual environments. The goal of this study was to determine if throughput would elicit this effect in a 3D pointing task based on to the ISO 9241-9 task.

Twelve participants took part in the study. The study was conducted in a 6-sided CAVE, using an Intersense IS-900 tracked stylus as the input device. Participants' heads were positioned on a headrest to ensure consistency. Thirteen spherical targets were stereoscopically presented 0.3 m in front of the participants, arranged in a vertically oriented circle. Passive haptic feedback was provided by co-locating a transparent plastic panel with the targets. The target positions conformed to the ISO task (Figure 1). Participants were instructed to click the highlighted target as quickly and accurately as possible. Figure 2 depicts the setup.

* e-mail: rteather@cse.yorku.ca

† e-mail: wolfgang@cse.yorku.ca

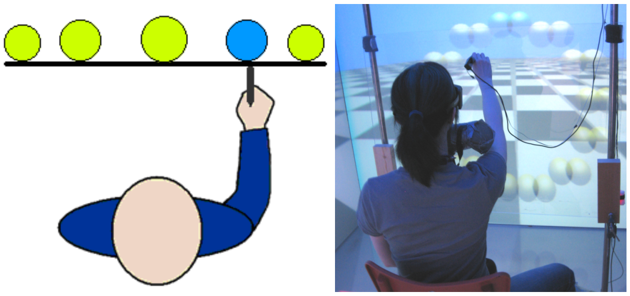


Figure 2. (Left) The pointing task, from above. (Right) Participant performing the task in the CAVE.

The experiment employed a 2 × 3 × 3 × 3 within-subjects design. The independent variables were haptic feedback (present or absent), target size (sphere diameter 2.8 cm, 4.0 cm, and 5.2 cm), distance between targets (circle diameter 22 cm, 27 cm, and 32 cm), and block (1 to 3). The dependent variables were movement time (ms), error rate (percent), and throughput (bps). Results were analyzed with repeated measures ANOVA.

The average movement time was 1.60 s (*SD* 1.17) without haptic feedback and 1.59 s (*SD* 0.99) with haptic feedback. The difference was not significant ($F_{1,11} = 0.04$, ns). The average error rate without haptic feedback was 13.3% (*SD* 7%). With haptic feedback, it was 11.1% (*SD* 6%). This difference was also not significant ($F_{1,11} = 0.69$, ns). However, throughput, which incorporates both speed and accuracy, was significantly different between conditions ($F_{1,11} = 6.47$, $p < .05$). The throughput without haptic feedback was 2.37 bps (*SD* 0.74), and haptic feedback increased it to 2.56 bps (*SD* 0.76).

4 Fish Tank VR Study

VR systems often use stereo graphics to project targets in front of, or behind, the display surface [1-3, 7]. Unlike volumetric displays [5], these displays introduce conflicts between the vergence and accommodation depth cues. The goal of this study was to evaluate the effect of these conflicts with the ISO 9241-9 task and also to compare the standard 2D tapping task with pointing in 3D space.

Twelve paid participants took part in the study. All had normal or corrected vision, and could perceive stereo depth. The study used a fish tank VR system consisting of a CRT monitor positioned horizontally, and a stylus tracked by a NaturalPoint *OptiTrack* tracker. The participants' heads were tracked using the same system, and the virtual camera position was coupled to the head position. Targets were stereoscopically presented either at the surface of the screen, i.e., without disparity, or at varying heights above the screen surface. Target height did not vary within a set of targets. Targets were on top of cylinders and textures were used to enhance depth perception. Participants were asked to click the highlighted target disk as quickly and accurately as possible. Figure 3 depicts the task and setup.

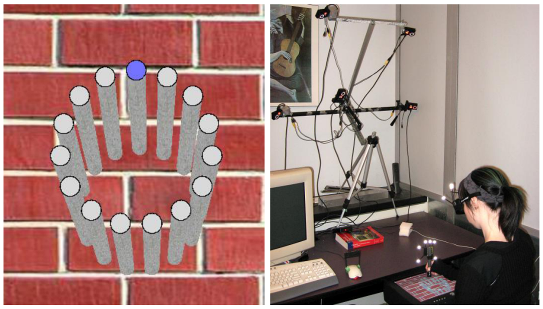


Figure 3. (Left) The pointing task, as viewed by the participant. (Right) Participant performing the task.

The grand mean movement time was 1053 ms. There was a significant main effect for target height ($F_{3,11} = 7.34$, $p < .001$) and block number ($F_{3,11} = 24.8$, $p < .0001$) on movement time. Higher targets took longer to hit than those at or near the screen. The overall error rate was 14.3%. There was no significant difference in error rate for repetition ($F_{3,11} = 0.90$, ns), or target height ($F_{3,11} = 0.14$, ns). The mean throughput was 4.77 bps. There was a significant main effect for target height ($F_{3,11} = 8.17$, $p < .0005$) and block ($F_{2,11} = 48.13$, $p < .0001$) on throughput. Linear regression of *MT* on *ID* indicates that Fitts' law best modeled movements at the surface of the screen ($R^2 = 0.88$), possibly due to the presence of haptic feedback there. Conversely, the 5 cm height was worst modeled by Fitts' law ($R^2 = 0.75$).

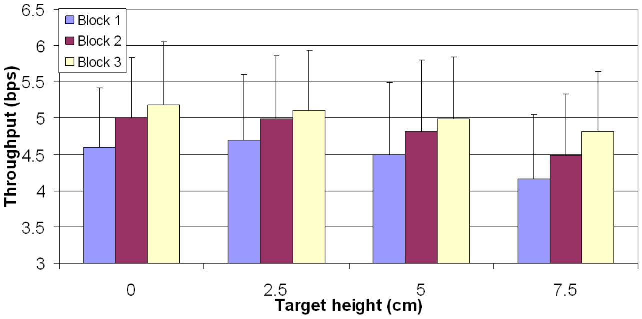


Figure 4. Throughput by target height and block.

5 Conclusion

We presented two studies evaluating 3D motions using variations on the ISO 9241-9 standard pointing task. The results of the first study indicate that passive haptics significantly improved pointing throughput. Throughput also helped elicit differences between conditions that were not detectable with standard speed or accuracy measures. The results of the second study indicate that pointing at targets presented stereoscopically above a display surface tends to be harder than pointing at targets presented near or at that surface. Increasing target height also degraded the correlation between movement time and task difficulty in Fitts' tapping tasks.

References

[1] Boritz, J. and Booth, K., A study of interactive 3D point location in a computer simulated virtual environment, in *VRST '97*, 181-187.

[2] Bowman, D., Johnson, D., and Hodges, L., Testbed evaluation of virtual environment interaction techniques, in *VRST '99*, 26-33.

[3] Bowman, D. and Hodges, L., An evaluation of techniques for grabbing and manipulating remote objects in immersive virtual environments, in *SI3D '97*, 35-38.

[4] Fitts, P. M., The information capacity of the human motor system in controlling the amplitude of movement, *Journal of Experimental Psychology*, 47, 1954, 381-391.

[5] Grossman, T. and Balakrishnan, R., Pointing at trivariate targets in 3D environments, in *CHI 2004*, 447-454.

[6] ISO 9241-9 Ergonomic requirements for office work with visual display terminals (VDTs) - Part 9: Requirements for non-keyboard input devices. International Organization for Standardization, 2000.

[7] Liu, L., van Liere, R., Nieuwenhuizen, C., and Martens, J.-B., Comparing aimed movements in the real world and in virtual reality, in *VR 2009*, 219-222.

[8] MacKenzie, I. S. and Isokoski, P., Fitts' throughput and the speed-accuracy tradeoff, in *CHI 2008*, 1633-1636.

[9] Teather, R. J., Pavlovych, A., Stuerzlinger, W., and MacKenzie, I. S., Effects of tracking technology, latency, and spatial jitter on object movement, in *3DUI 2009*, 43-50.

Two New Mobile Touchscreen Text Entry Techniques

Ahmed Sabbir Arif* Mauricio H. Lopez[†] Wolfgang Stuerzlinger[‡]

Department of Computer Science & Engineering, York University, Toronto, Canada

Abstract

This article introduces two new mobile touchscreen text entry techniques. One is timeout-based and the other is pressure-based. Also, this work examines the effects of tactile feedback on text entry techniques. Empirical comparisons between conventional and proposed techniques show that the new techniques, as well as tactile feedback, enhance overall text entry performance.

Keywords: Text entry, error prevention, touchscreens, virtual keyboard, delay, timeout, pressure.

Index Terms: H.5.2 [User Interfaces]: Haptic I/O

1 Introduction

Recently, touchscreens have become one of the dominant interaction modality for handheld devices. Many of these devices replace physical keyboards with virtual ones, which permit larger displays, less weight or size. It also enables adaptation to different layouts and orientations. However, virtual keyboards are more error prone [3], mainly due to smaller key sizes [8] and the absence of tactile feedback [8], . To counteract these issues, we present two new techniques that are timeout-, respectively, pressure-based. We also examine if providing synthetic tactile information can improve overall text entry performance.

2 Related Work

MultiTap is the dominant technique for standard 12-key keypads on mobile devices. In MultiTap, keys are pressed repeatedly until users get the intended character. Then, one can proceed to the next character, assuming that it is on a different key. If not, users have to either wait for a *timeout* for the system to accept a character on the same key, or have to press a predetermined *kill* button.

McCallum *et al.* [6] introduced a pressure-based technique for 12-key mobile keypad with three pressure levels. Their technique was shown to have higher *expert* entry speed compared to MultiTap, but at the expense of higher error rates. Likewise, Tang *et al.* [10] developed a 3-key chorded keyboard with three pressure levels, which again yielded higher error rates. Hoffmann *et al.* [2] designed a physical keyboard that used pressure to prevent errors. This reduced mistyped characters by 87% and correction attempts by 46%. Brewster and Hughes [1], used pressure-based techniques to switch between upper and lower case. This technique was faster and more accurate than standard touchscreen techniques.

3 New Techniques

The main idea of this work is to generate a list of potential next characters based on the preceding input in real-time. Then we identify those letters including space characters that have less than .01% probability of appearing after the preceding input. Such unlikely characters are then made to be more difficult to enter. In practice, we use a digram frequency table [5] for letter-pairs of the English language to calculate the probability ρ of a character C_n's appearance given the preceding character C_{n-1}, using Equation (1):

$$\rho(C_n \mid C_{n-1}) = \frac{Total(C_{n-1}, C_n)}{Total(C)} \qquad (1)$$

There, $Total$ (C) is the total number of characters including space and $Total$ (C_{n-1}, C_n) is the total number of a specific digram ($C_n \mid C_{n-1}$) in the table. We use digrams mainly for simplicity here. However, *n*-grams, a dictionary, or grammar rules could also be used to identify less probable characters more accurately.

In the timeout-based technique, we force users to press unlikely keys longer than 0.5 seconds, to make them harder to input. In other words, users will have to press-hold those keys for longer than usual. In the pressure-based technique we use pressure and users will have to apply more force on keys that are unlikely.

4 Measuring Pressure Workaround

All present handheld touchscreen devices do not provide hardware support for measuring pressure. Therefore, we detect pressure by measuring the movements of the touch centre over time, which identifies different levels of contact force [7], .

4.1 Pilot Study

We created an application with the iPhone SDK on an Apple iPhone 3G at 320×480 pixel resolution for our pilot study. The application's virtual Qwerty keyboard was practically identical to the default one, see Figure 1, and provided users with auditory and visual feedback via clicks and highlighting during a press.

Three participants aged from 22 to 24 participated in the pilot study. One of them was female, two of them had prior experience with touchscreens, and all of them were right-hand mouse users.

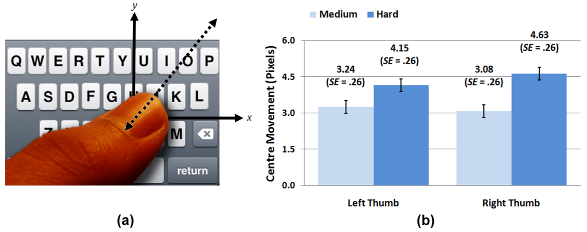


Figure 1. (a) Illustration of movement with contact force. (b) Touch centre movement during *medium* and *hard* presses.

During the pilot, participants entered all the characters on the keyboard holding the device in the portrait position. Two pressure levels, *medium* and *hard*, were tested. During the *medium* condition participants entered all characters using regular force from the top-left to the bottom-right, and then from the top-right to the bottom-left; using at first their *left* and then the *right* thumb. During the *hard* condition, participants repeated the same tasks, but applied more force than usual. We recorded the distances between the *initial* and the *release* touch centres. In total we

* e-mail: asarif@cse.yorku.ca
[†] e-mail: cs241053@cse.yorku.ca
[‡] e-mail: wolfgang@cse.yorku.ca

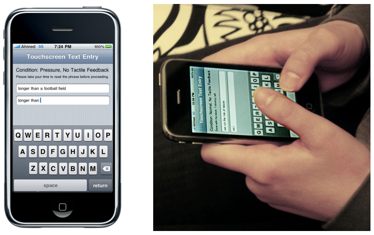
recorded 3 participants × 2 sessions (pressure levels) × 2 blocks (thumbs) × 27 keys (including space) = 324 presses.

An ANOVA showed that there was significant effect of different pressure levels on touch centre movements ($F_{1,2} = 21.19$, $p < .0001$). However, there was no significant effect of different thumbs ($F_{1,2} = 0.36$, ns). On average *left* and *right* touch centres moved 3.16 pixels ($SE = 0.19$) during the *medium* and 4.39 pixels ($SE = 0.19$) during the *hard* presses.

5 An Experiment

For our experiment, we used the same apparatus and software as for the pilot study. Based on the pilot results, we used a threshold of 4.4 pixels on touch centre movements to identify *hard* presses.

Twelve participants aged from 19 to 34, average 26 years, took part in the experiment. Five of them were female, four of them were touch-typists, and all of them were right-hand mouse users.



Touchscreen Text Entry

Condition: Pressure, No Tactile Feedback

Please take your time to read the phrase before proceeding.

longer than a football field

longer than

Figure 2. Screenshot of the application used during the user study.

5.1 Procedure and Design

We tested 3 conditions, namely the *regular*, *timeout-based*, and *pressure-based* techniques. Each condition was tested *with* and *without* synthetic tactile feedback. For the synthetic tactile feedback we activated the iPhone's vibration motor for 500 ms.

Participants were asked to enter a set of short English phrases [4], all in lowercase, as shown on the display. They held the device in a portrait position and were asked to take the time to read and understand the phrases, to enter them as fast and accurate as possible, and to press the *Return* key after completion of a phrase to see the next. Timing started from the entry of the first character and ended with the last. Participants were informed that they could rest between sessions, or before typing a phrase. They were asked to work normally, that is, to correct their errors as they noticed them. However, they had to use the *Backspace* button, exclusively, for editing. Based on the 3 × 2 = 6 techniques, we used a within-subjects, 6 × 6 balanced Latin Square design for our experiment. In summary the design was: 12 participants × 6 sessions (techniques) × 20 phrases =1440 phrases.

5.2 Results and Discussion

An ANOVA for the techniques showed that there was significant effect of entry techniques on *WPM* ($F_{5,11} = 3.21$, $p < .01$). There was, however, no significant effect of tactile feedback ($F_{1,11} = 0.12$, ns). The ANOVA on the *Total ER* also showed that there was a significant effect of entry techniques ($F_{5,11} = 2.38$, $p < .05$) and tactile feedback ($F_{1,11} = 7.57$, $p < .01$).

Deeper analysis showed that the *regular with tactile* (16.27 *WPM*, 9.46 *Total ER*) and *pressure with tactile* (16.08 *WPM*, 9.24 *Total ER*) conditions had better overall performance with higher text entry and lower error rates. *Timeout with tactile* was the slowest of all (14.91 *WPM*, $SE = 0.28$). A Tukey-Kramer test revealed that it was significantly slower than *regular with tactile* and *pressure with tactile*. However, it was, at the same time, the most accurate (8.0% *Total ER*, $SE = 0.70$).

From the results it is clear that tactile feedback reduces *errors* for all techniques without reducing the speed in a significant manner. The results also confirmed that pressure-based techniques have the potential to offer higher performance. We believe that with proper training the advantages will increase even more, as previous studies [6], , showed that response time increases with practice for different pressure levels.

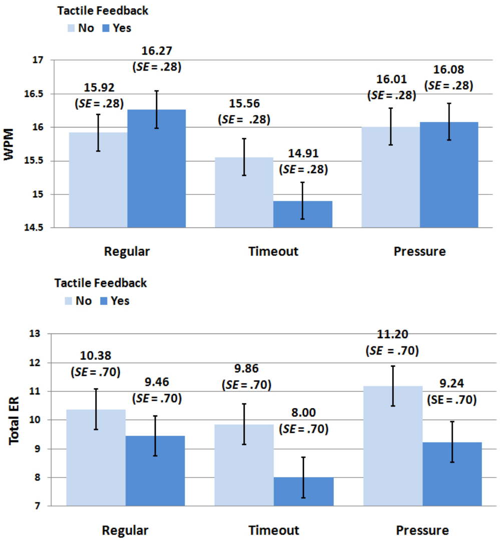


Figure 3. Average *WPM* and *Total ER* for different techniques.

6 Conclusion

Here, we presented and evaluated two new mobile touchscreen text entry techniques: one timeout-based and one pressure-based. The pressure-based techniques had better overall performance compared to the conventional one. Our results also showed that synthetic tactile feedback significantly reduces errors.

References

[1] S. A. Brewster and M. Hughes. Pressure-based text entry for mobile devices. *MobileHCI 2009*, ACM (2009), 1-4.

[2] A. Hoffmann, D. Spelmezan, and J. Borchers. TypeRight: a keyboard with tactile error prevention. *CHI 2009*, ACM (2009), 2265-2268.

[3] J. R. Lewis, K. M. Potosnak, and R. L. Magyar. Keys and keyboards. M. Helander, T. K. Landauer, and P. V. Prabhu, eds. *Handbook of Human-Computer Interaction*. Elsevier, 1997, 1285-1311.

[4] I. S. MacKenzie and R. W. Soukoreff. Phrase sets for evaluating text entry techniques. *Ext. Abstracts CHI 2003*, ACM (2003), 754-755.

[5] M. S. Mayzner and M. E. Tresselt. Tables of single-letter and digram frequency counts for various word-length and letter-position combinations. *Psychonomic Monograph Sup. 1*, 2 (1965), 13-32.

[6] D. C. McCallum, E. Mak, P. Irani, and S. Subramanian. PressureText: pressure input for mobile phone text entry. *Ext. Abstracts CHI 2009*, ACM (2009), 4519-4524.

[7] G. Ramos, M. Boulos, and R. Balakrishnan. Pressure widgets. *CHI 2004*, ACM (2004), 487-494.

[8] A. Sears, D. Revis, J. Swatski, R. Crittenden, and B. Shneiderman. Investigating touchscreen typing: the effect of keyboard size on typing speed. *BIT 12*, 1 (1993), 17-22.

[9] A. Sears. Improving touchscreen keyboards: design issues and a comparison with other devices. *IWC 3*, 3 (1991), 253-269.

[10] H. Tang, D. J. Beebe, and A. F. Kramer. A multilevel input system with force-sensitive elements. *IJHCS 54*, 4 (2001), 495-507.

[11] F. Wang, X. Cao, X. Ren, and P. Irani. Detecting and leveraging finger orientation for interaction with direct- touch surfaces. *UIST 2009*. ACM (2009), 23-32.

Coordination Policies for Tabletop Gaming

Joey A. Pape* T.C. Nicholas Graham†

School of Computing, Queen's University

Abstract

This paper explores how social interaction can be preserved in multitouch tabletop video games, when the turn-based gameplay of board games is relaxed in favor of a real time experience. In this paper we will present two games which we have built for the tabletop, as well as a classification of real-time coordination policies for board games and preliminary results from informal experience with users.

Keywords: Computer games, tabletop games, interactive surfaces.

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces - *input devices and strategies*

1 Introduction

Multitouch digital tabletop surfaces present the opportunity to design video games featuring novel styles of interaction. Similarly to traditional board and card games, tabletop games are played on a horizontal surface by small collocated groups, and players interact with the game by physically manipulating objects. With the release of Apple's iPad, multitouch displays are gaining mainstream attention, and being hailed as the perfect platform for digital versions of traditional board games [10]. Recent technologies such as the Microsoft Surface and the SMART Table allow us to move even closer to traditional tabletop play.

Board and card games are popular. For example, the Monopoly board game sells several million copies a year [1]. These games are social [2],[3],[6], played by a small group of players sitting around a table, where players can see and interact with each other. A new wave of cooperative board games has a strong element of group coordination, as players must work together and discuss strategy in order to succeed. Tabletops effectively support this type of interaction, as seen in other existing tabletop applications that support collocated collaborative tasks [7].

Board games are almost exclusively limited to some form of turn taking, in which only one player acts at a time. This can lead to significant downtime for players awaiting their turn. Turn-taking is often the only practical coordination policy for games, as more liberal policies might overly burden players with complex calculations to determine who is allowed to do what at a given time. Digital tabletops can use a computer to perform and enforce these calculations, opening the opportunity to design games with the streamlined gameplay and real-time coordination policies of video games, while preserving the sociality of board and card games. However, the transition from turn-based to real-time gameplay, risks speeding up the game to a point where social interaction is lost.

In this paper, we explore how games can be designed for digital tabletop surfaces to combine the social aspects of board and card games with the streamlined real-time gameplay of video games. We describe two games which we have built, and present preliminary informal observations from users playing these games. We also present a classification of real-time coordination policies for board games.

2 Related Work

Many games have now been developed for digital surfaces. A few examples are the following: Mandryk et al. [3] have shown how hybrid board/video games can enhance sociality by combining the tactile and tangible gameplay of traditional board games with the streamlined gameplay of computer-based games. *WeatherGods* [6] is a tabletop game designed to combine the advantages of both board games and tabletop technology. The game can use two different versions of tangible playing pieces: iconic or symbolic. *SIDES* [7] is a cooperative tabletop puzzle game, which uses board game design elements. It is designed to help adolescents with Asperger Syndrome to use effective group work skills. The *TViews* Table Role-Playing Game [8] and *SurfaceScapes* [9] are traditional tabletop role playing games implemented for touch surfaces. Both use tangible playing pieces as well as a tangible object to invoke a menu when placed.

The games RTChess [5] and Real-Time Chess [4] implement alternative real-time coordination policies for Chess. RTChess is a distributed game in which two teams of players play a standard game of Chess. Unlike standard chess, any player may move any piece at any time, and games are completed within tens of seconds. This is an example of how the transition to a real-time coordination policy can dramatically change gameplay. Certainly, in a ten second game, there is little opportunity for social interaction. In the tabletop game Real-Time Chess, up to four players move chess pieces around a Chess-like board. When a piece is moved, it cannot be moved again until some time has passed. This puts a time-based restriction on when players may make certain moves, slowing the pace of the game.

3 Early Results

The goal of our research is to explore how players interact with each other while playing cooperative board games, and how this changes when the game is played on a digital tabletop surface. Specifically we are interested in how social interaction and group coordination are altered when the rigid turn-based coordination policies of traditional board games are relaxed in favor of a real-time experience.

We have implemented two games for the tabletop. The first is a straight port of the two-player board game Checkers. The second is an implementation of the popular cooperative board game Pandemic. The following are preliminary results, including observations collected from informal experience with users, as well as a classification of alternative real-time coordination policies for cooperative board games.

3.1 Coordination Policies

A game's coordination policy restricts when players may take actions. We have observed that coordination policies fall into two categories: time-based restrictions and restrictions based on player actions. The two extreme coordination policies are turn-based and

* e-mail: pape@cs.queensu.ca

† e-mail: graham@cs.queensu.ca

free-for-all. In turn-based, only one player may take actions at a time; at the end of the current player's turn, the next player is allowed to play. In free-for-all, any player may take any action at any time and the actions are performed immediately.

3.1.1 Restrictions Based on Player Actions

We have identified two coordination policies where a players' ability to perform actions depends on the actions of other players. These are turn-taking (already described) and *barrier synchronization.* In barrier synchronization, each player is assigned a set of action points. Performing actions consumes these points. Once all players have consumed their action points, each player is granted a new set of points. Barrier synchronization allows players to take actions concurrently, but matches the overall pace of the game to that of the slowest player.

3.1.2 Time-Based Restrictions

Time-based restrictions pace the game by restricting how frequently players can perform actions. Two variants are *timed actions* and *trickle points*. In both approaches, players may take actions concurrently.

Under *timed actions,* actions take time to complete. For example, if a player moves a piece between distant points on the table, the movement may take 10 seconds to complete. Animation can be used to show the state of the action.

With *trickle points,* actions take place instantaneously as long as the player has action points available. Action points are periodically assigned to players over time. Players can bank action points, allowing a flurry of activity.

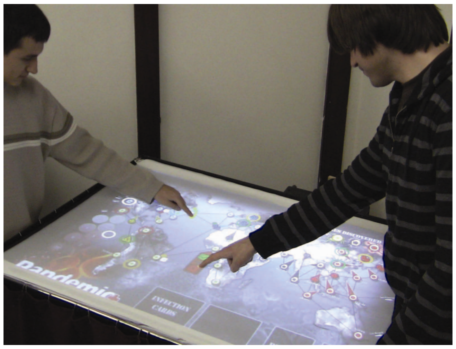


Figure 1. Pandemic Gameplay

3.2 Observations

Through informal user observations of our Checkers and Pandemic games, we have found the following playability issues with tabletop implementations of board games. Our current implementations provide only turn-based coordination. Experience with other policies represents future work.

Players expected the coordination state to be explicitly shown in the digital versions of the games. For example, testers of the Checkers game complained that the game did not show whose turn it was. This indicates that the digital format changed players' expectations – people have played Checkers as a board game for hundreds of years without requiring a turn indicator.

One advantage of digital games is that they can prohibit illegal actions; however, such automated enforcement of rules must be done carefully. Our games (initially) did not give feedback when players attempted to take actions which were not allowed. This led to players being unsure whether the game had registered the action. A particularly confusing case was Checkers enforcing the rule that players must take a piece if it is possible to do so. Most testers were unaware of this rule, and were confused as to why they were unable to make an alternative move.

In the Pandemic board game, random actions are taken following each player's move. These are carried out by a player by drawing a card. In the digital version of the game, players occasionally missed these actions, leading to confusion. Automated actions therefore must be clearly visible to players.

In general, these pitfalls suggest three design rules: anticipate that moving to a digital form will raise player's expectations; make it clear what players are allowed to do when game rules are being enforced; and ensure that automated actions are transparent.

4 Future Work

We have implemented the turn-based version of Pandemic. We will be implementing the other coordination policies. We will perform a study exploring how well these different real-time versions of Pandemic incorporate both the streamlined gameplay of video games and the social advantages of board games.

5 Conclusion

We have presented our research with the goal of exploring how the turn-based gameplay of board games can be relaxed in favor of real-time gameplay, in such a way that the social advantages of the original games are preserved. We have presented a classification of coordination policies for board games, as well as preliminary findings which reveal some of the issues with the transition from a board game to a tabletop video game.

References

[1] Elliot. S. "Would You Like Fries With That Monopoly Game?", *New York Times,* Sep. 12, 2006.

[2] Wolfe, J.D. Smith, T.C.N. Graham, "A low-cost infrastructure for tabletop games," in Proc. FuturePlay, 2008, pp. 145-151.

[3] R.L. Mandryk, D.S. Maranan, K.M. Inkpen. "False prophets: exploring hybrid board/video games," in *CHI '02 extended abstracts,* 2002, pp. 640-641.

[4] J. Chaboissier, F. Vernier. "Conception de jeux interactifs temps réel sur tabletop," in *Proc. Interaction Homme-Machine*, 2009, pp. 313-322.

[5] C. Gutwin, M. Barjawi, B. de Alwis. "Chess as a twitch game: RTChess is real-time multiplayer chess," demonstration at *ACM CSCW*, 2008.

[6] S. Bakker, D. Vorstenbosch, E. Van den Hoven, G Hollemans, T. Bergman. "Weathergods: tangible interaction in a digital tabletop game," in *Proc. Tangible and Embedded Interaction*, 2007, pp. 151-152.

[7] A.M. Piper, E. O'Brien, M.R. Morris, T. Winograd. "SIDES: a cooperative tabletop computer game for social skills development," in *Proc. Computer-Supported Cooperative Work*, 2006, pp. 1-10.

[8] A. Mazalek, B. Mironer, E. O'Rear, D. Van Devender. "The TViews Table Role-Playing Game," in *Proc. 4th International Symposium on Pervasive Gaming Applications,* 2007, pp. 127-134.

[9] J. Coldeway. "Hands-on: D&D on the Microsoft Surface." Internet: http://www.crunchgear.com/2010/02/10/hands-on-dd-on-the-microsoft-surface/, Feb. 10, 2010 [Apr. 12, 2010].

[10] C. Sapieha. "Apple's iPad is perfectly suited for a classic tabletop game." *The Globe and Mail,* April 10, 2010.

WidgetLens: Interaction Through The Looking Glass

Bhavna Agarwal * Wolfgang Stuerzlinger †

Department of Computer Science & Engineering, York University, Toronto, Canada

ABSTRACT

Computer and mobile device screens are increasing in size and resolution. This increase in resolution causes problems with graphical user interfaces designed for lower resolution screens, as all information gets smaller and smaller. We present two novel techniques to make graphical user interfaces on high-resolutions screens more accessible and usable. We introduce a new in-place, localized zooming technique that works on a per widget basis. We also present a novel widget magnification technique that implements special modalities for common user interface elements, which affords widget-dependent magnification.

KEYWORDS: Graphical user interfaces, widgets, accessibility, human performance.

INDEX TERMS: H.5.2 [User Interfaces]: Graphical User Interfaces

1 INTRODUCTION

Computer monitors are getting bigger and bigger and resolutions, i.e. the number of pixels, continue to increase. This trend exists both on handheld devices, such as smartphones, as well as traditional laptops and desktops. In general, bigger displays are preferred as users can display more information and see more detail simultaneously. More relevant to our current work is that the pixel density of screens, i.e. the number of pixels per inch (ppi), is also increasing steadily. Currently smart phones and portable book readers use screens with 150-300 ppi, laptops 100-250 ppi, and desktops 100-200 ppi.

However, most graphical user interfaces (GUI's) are designed for monitors with a constant pixel density, typically about 100 ppi, the lower end of today's displays. This makes all user interface elements, including text, appear (too) small on displays with higher densities. This causes problems for users with poor eyesight, but affects even average people when they upgrade their system. In many common GUI toolkits there is little or no support for scaling of widgets and their elements, such as images or text, to resize or render them differently depending on the resolution. Partial fixes to this problem exist in the form of a global, system wide scale factor or full screen pixel-wise zoom. However, this reduces either the information content per window, or provides less screen space for other programs. This quickly becomes a problem if the user is working with more than one application at a time. The zoom/pan facilities associated with the "canvas" widget used in browsers, text editors and drawing programs etc., address this problem for that type of widget, but this solution does not generalize. Reworking existing GUI toolkits to provide scalable widgets is labour intensive and time consuming and will likely require significant adaptation in existing applications. Hence, methods are needed to enhance the readability of existing GUI's without reducing content.

In this article, we present a novel technique to make user interface widgets more scalable for different screen resolutions and thus more accessible. We explore target expansion for GUI's composed of densely tiled widgets and present new widget-dependent magnification techniques.

2 PREVIOUS WORK

A GUI consists of many densely tiled targets, i.e. widgets. In such arrangements, the user needs to be able to quickly select and easily interact with any given widget. For this situation, dynamically expanding targets have been proposed based on their implications on human performance [1, 2, 3]. This established that dynamically changing a widget's size aids in selection tasks. Ruiz and Lank [2] presented a cost/benefit model for expanding targets in a tiled arrangement using kinematic endpoint prediction. They found a net benefit in expanding targets that occupy less than 4% of the display resolution.

Another area of relevant work is interface customization, i.e. adaptive and user-adaptable interfaces. Among this large body of work, Façades is particularly relevant [4]. This system permits seamless copy and paste of window regions using direct manipulation techniques, while maintaining full interactivity and without changes to the application code. Façades implements this by obtaining widget-related information through the accessibility API's supported by modern GUI toolkits and by redirecting events and screen output accordingly. Figure 1 shows the event management and image flow in Façades.

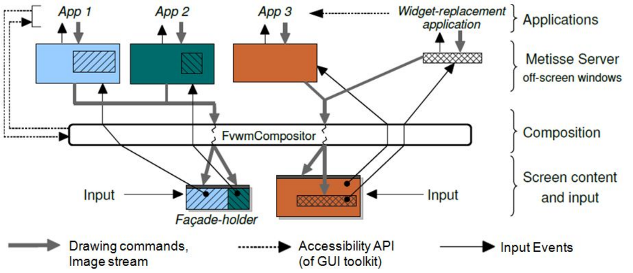


Figure 1: Façades architecture, from [4].

3 NEW TECHNIQUES

The basic idea behind WidgetLens is a localized widget zoom. Essentially, we map the widget under the cursor to a larger overlay window displayed over it. This design choice is motivated by the fact that we work in a (fixed) densely tiled arrangement of widgets, where there is likely no free surrounding space. Hence, we cannot move other widgets "away", similar to the OS X Dock.

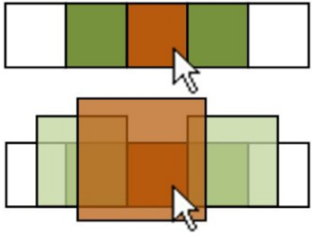


Figure 2: Illustration of widget displacement relative to cursor.

[*] e-mail: bhavna@cse.yorku.ca

[†] e-mail: wolfgang@cse.yorku.ca

For this localized widget zoom, we present two new techniques using pixel magnification and widget replacement. Both use the accessibility API of modern toolkits to retrieve information about widgets, such as position and/or interaction possibilities.

3.1 Local Widget Zoom

In this version of a WidgetLens, the pixel image of the current widget is used as a texture for an enlarged version, which is overlaid on top. The widget in focus is thus enlarged in each direction (by default a factor of two). We move this enlarged version proportional to the distance of the corresponding edge of the original widget from the cursor, which provides a sliding effect. Mouse motions are mapped to guarantee that as soon as the mouse cursor leaves the original widget, the local widget zoom switches to the corresponding next widget. Any mouse or keyboard interaction with the fully zoomed version is passed through to the original widget.

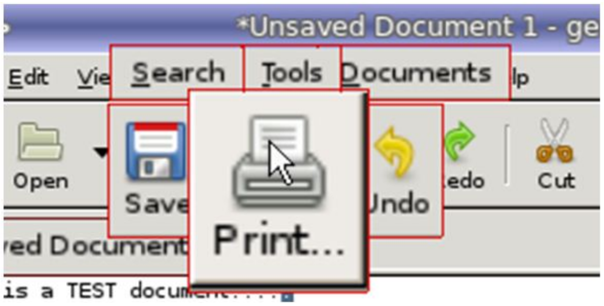


Figure 3: Widget in focus 2x zoomed and partially obscured widgets 1.5x zoomed. Red boundaries added for visualization.

As neighbouring widgets may be hidden by the expanded widget, we generate partially enlarged versions and overlay them at the respective corner closest to the cursor in an intermediate layer, which increases the visibility of these surrounding widgets. Figure 3 depicts the local widget zoom.

3.2 Smart WidgetLens

Straightforward magnification causes an undesirable blur in text and images. To address this, we present a novel form of widget lens that replaces widgets with higher resolution versions. This is implemented by traversing the widget tree of the underlying application and by duplicating each widget on a one-to-one basis in a separate window. The replication then uses larger fonts. If the system recognizes also the filename of an icon, it also uses a higher-resolution version of the corresponding image. Then, each of these enlarged versions is instantiated as a façade and overlaid on the original window as described in the local widget zoom (Figure 4). For simple widgets, such as buttons or menus, all interactions, such as mouse clicks are simply redirected to the original widget. The display mechanism, mouse events, and partially obscured widgets are handled as described above.

More complex widgets, such as combo-boxes or text fields, are managed differently. In this situation there are more interaction possibilities, such as selection, scrolling, or editing. We address this by allowing the user to interact directly with the overlaid widget and pass the resulting higher-level events, such as text changes or selection events, to the original widget. Another issue is that simple magnification may cause parts of the widget to fall outside of the monitor. Hence, we limit the number of lines, list items, and/or the number of characters to make sure that the widget stays within the boundaries of the monitor. We currently do not provide WidgetLens functionality for canvases. A naïve magnification would only lead to an explosion of white space. Moreover, we can assume that the application already provides zoom facilities for a canvas widget.

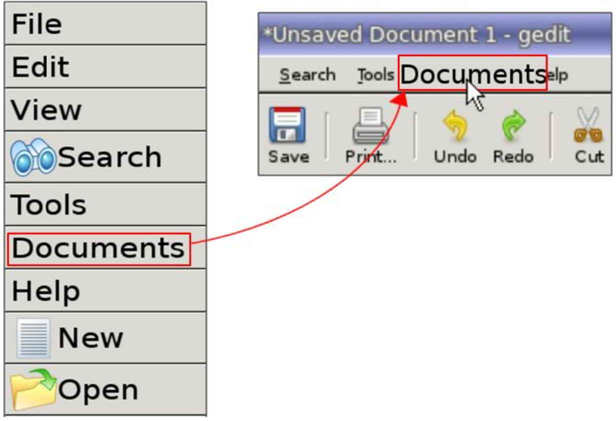


Figure 4 Left: Façade window with larger font (normally invisible), Right: User view.

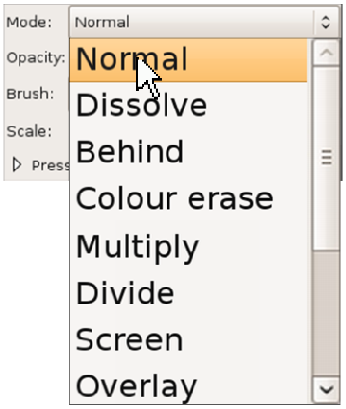


Figure 5: Combo-box with larger font and scrollbar.

4 Conclusion and Future Work

We presented a new method for end-users to interact with user interface elements on screens with high pixel densities. Based on the Façades system, we implemented two new methods that increase the visual accessibility and interactivity of GUI widgets.

We are working to implement various other effects, such as high-contrast display of widgets. Moreover, we are currently improving the animations to make the zoom effect less noticeable and dependent on cursor speed. Also, we are expanding the set of smart WidgetLenses further. Finally, we are considering a user study to analyze the benefits of WidgetLenses.

References

[1] A. Cockburn, P. Brock. Human on-line response to visual and motor target expansion. In *Proc. Graphics interface*, volume 137, pages 81-87, ACM GI, 2006.

[2] J. Ruiz, E. Lank. Speed pointing in tiled widgets: understanding the effects of target expansion and misprediction. In *Proc. Intelligent User Interfaces*, pages 229-238, ACM, 2010.

[3] M. McGuffin, R. Balakrishnan. Acquisition of expanding targets. In *Proc. SIGCHI*, pages 57-64. ACM, 2002.

[4] W. Stuerzlinger, O. Chapuis, D. Phillips, N. Roussel. User Interface Façades: Towards Fully Adaptable User Interfaces, pages309-318, ACM UIST, 2006.

Vibration Perception in Mobile Contexts

Idin Karuei* Zoltan Foley-Fisher† Sebastian Koch‡ Russ MacKenzie§ Mohamed El-Zohairy¶ Karon E. MacLean‖
Department of Computer Science, University of British Columbia, Vancouver, Canada

Abstract

Human sensitivity to vibration declines in mobile contexts. Designers of wearable haptic systems must account for the effects of movement and distraction so that tactile display information is perceived consistently. We compared the sensitivity of seven body sites in simulated mobile contexts, and found that the thigh is least and the wrists the most sensitive of the sites tested.

Keywords: Vibration, Sensitivity, Mobile Contexts, Movement, Distraction, Wearable Haptics.

Index Terms: H.5.2 [**Information Systems**]: User Interfaces – *Haptic IO.*

1 Introduction

Many body sites have been considered for wearable tactile displays, and vibratory information has improved performance of pilots and drivers. It is well known that some body sites are less sensitive than other areas, e.g. back versus wrist – a function of skin type and sensor density and composition. When a body part is in motion, it becomes less sensitive to stimuli [1] and vibration patterns may be misinterpreted or undetected. For wearable haptic systems, often used in ambulatory situations this is especially troublesome. Merely increasing vibration intensity is unsatisfactory due to concerns of power and comfort. In this paper we tackle unpredictable vibration sensitivity by seeking body sites that are less susceptible to changes in sensitivity, comparing diverse sites (a few studied elsewhere) in a single study.

2 Related Work

Wearable tactile systems have been the focus of many papers in the last decade due to its variety of applications. Ertan et al. introduced a wearable navigation system for guidance of blind users in unfamiliar indoors areas [2]. They used a vibrotactile display consisting of a 4-by-4 array of micromotors embedded in the back of a vest to communicate a stop signal or the four cardinal directions to the user. Bosman et al. developed a wearable haptic guidance system that could be attached to both wrists of a pedestrian to guide him inside unknown buildings [3]. Tsukada and Yasumura developed a belt with eight vibrotactile haptic displays to guide a pedestrian towards destinations, predefined locations, or valuables left behind [4]. Subjects could feel vibrations when stopped but often failed to recognize vibrations when walking; they could stop for a moment to recognize the direction of the vibration. This suggests that the effect of movement on detection of tactile stimuli which has been studied in the field of neural psychology [5][6] is in fact significant and ignoring it will harm the effectiveness of tactile user interfaces.

3 Methods

16 volunteers (8 male) took part. The participants (counterbalanced for gender and condition) sat in a chair in one condition, and walked on a treadmill in the other. A tall chair maintained a consistent view of the screen between conditions. We attached thirteen vibrotactile displays to seven body sites corresponding to wearable sites based on past studies and potential practicality: chest (left and right, directly below the collar bone), spine, outer thighs, stomach (left and right, halfway between navel and hip bone), feet (on the top surface of the foot), wrists, and upper arms.

During half of the trials in each condition, participants performed a workload task targeting vision, memory, and attention as shown in Figure 1. A screen four meters wide and three meters high displayed twenty-five blocks bouncing slowly around a three-dimensional room; one block was highlighted and participants counted the times the highlighted block hit any of the walls. The task was chosen for its controllable continuous workload typical of normal pedestrian activity, with distraction adjusted so that participants would not fall off the treadmill. Participants reported their collision count at the end of each workload condition.

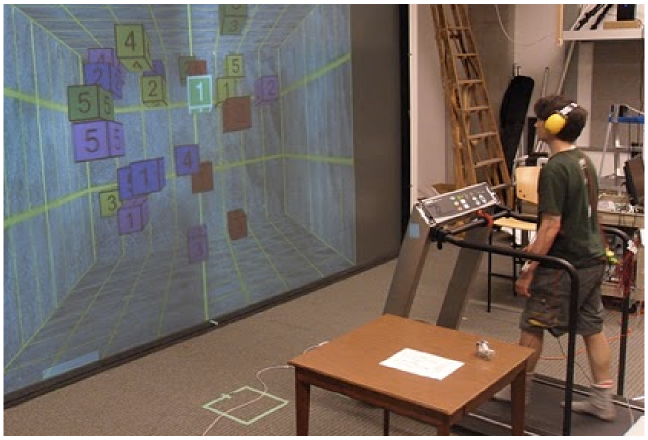


Figure 1: Experiment setup. Subject responds to vibrations while walking on the treadmill and doing the visual task.

During all conditions, participants pressed the right-hand button on a modified computer mouse when they detected vibration from any tactor. Vibrations were presented in randomized sites, intensities in randomized order, and the interval between 500 ms-duration tactor vibrations was randomized between four and six seconds. Reactions later than 3500 ms were discarded.

4 Results

As the dependent variable (detection rate) is dichotomous, we performed a logistic regression on five factors: Intensity, Task,

* e-mail: idin@cs.ubc.ca
† e-mail: zoltan@ece.ubc.ca
‡ e-mail: skoch@cs.ubc.ca
§ e-mail: rmacken1@cs.ubc.ca
¶ e-mail: zohairy@cs.ubc.ca
‖ e-mail: maclean@cs.ubc.ca

Movement, Body Site (within-subject) and Gender (between-subjects). Gender, Task, Movement, and Body Site (with spine as a reference point) are categorical variables. The omnibus test of the model coefficients is significant ($p < 0.001$). The regression results are listed in Table 1, where main effects Gender, Intensity, Movement and Body Sites are seen to be statistically significant.

Table 1: Results of logistic regression

	B	S.E.	Wald	df	Sig.	Exp(B)
Gender(1)	.215	.064	11.158	1	.001	1.240
Intensity	1.692	.036	2195.889	1	.000	5.429
Task(1)	.054	.064	.700	1	.403	1.055
Movement(1)	1.778	.071	625.023	1	.000	5.919
BodySite			649.684	6	.000	
BodySite(1)	-1.186	.121	96.079	1	.000	.305
BodySite(2)	.878	.125	49.060	1	.000	2.407
BodySite(3)	-.972	.121	64.962	1	.000	.378
BodySite(4)	-2.086	.125	279.697	1	.000	.124
BodySite(5)	.096	.121	.622	1	.430	1.100
BodySite(6)	-.102	.121	.715	1	.398	.903
Constant	-2.651	.117	511.428	1	.000	.071

The six Body Site levels in the table are Foot, Wrist, Stomach, Thigh, Chest, and Arm; Spine is the reference level. Foot, Wrist, Stomach, and Thigh are significantly different than Spine; Figure 2 further shows that the Wrist is more sensitive than the Spine, and the Foot, Stomach, and Thigh are less sensitive.

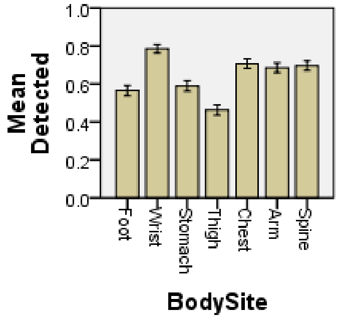


Figure 2: Detection rates at different body sites.

For Gender, males show a slightly higher detection rate of 65.3% compared to 63.0% for females. The Movement factor showed an important result: participants detected 73.9% of stimuli when sitting, but only 54.4% while walking. As suggested by the regression results, Task did not show significant differences; the detection rate with and without the visual distraction task was 64.4% and 63.8%, respectively. Predictably, there were strong results for Intensity: at intensity 4 (the strongest) almost all stimuli were detected, while at intensity 0 (weakest) only 16.7% were detected.

Interaction effects are difficult to analyze using regression, so we will present these results graphically rather than with tests of significance. There is a strong interaction between movement and intensity, as shown in Figure 3. At the highest intensity there is no difference between movement conditions, while at lower intensities the detection rate is much lower while walking.

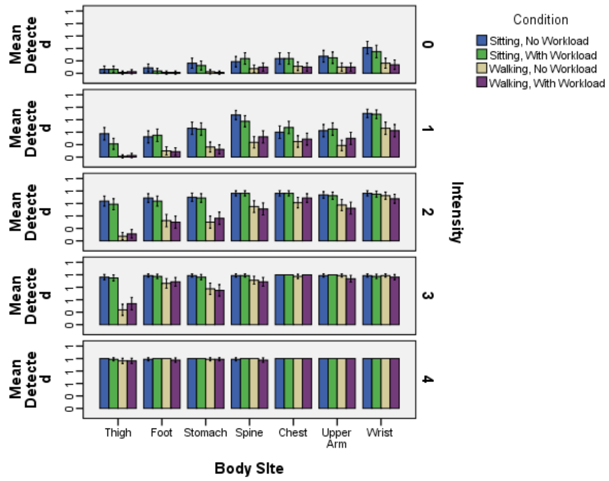


Figure 3: Sensitivities for five intensities across the four conditions.

All body sites are negatively affected by movement, but some sites more than others, as illustrated in Figure 3. Thighs, and to a lesser extent feet and stomach are particularly strongly affected. These are also areas of motion: the feet can feel heel strikes on the treadmill surface, while the stomach undergoes twisting motions as the arms swing.

5 Conclusion

The results of our experiment confirm the effect of body motion on detection of vibrations. We discovered that movement in a typical mobile context (i.e. walking) affects detection of vibrations on the thighs more than other body sites. Also, reaction times to vibrations are significantly reduced during walking. However, it appears that visual distraction in a mobile context may not have a significant effect on detection of vibration on any body site. In general, the thigh is not suited for applications that require discriminating among vibration patterns in everyday wearable haptics. This may be of interest to cell phone users who typically receive vibration notifications on the site most susceptible to movement effects. On the other hand, the data suggest that the chest, upper arm, and wrist are sufficiently sensitive to lower energy vibrations while the body is in motion.

References

[1] C. E. Chapman, M. C. Bushnell, D. Miron, G. H. Duncan, and J. Lund. Sensory perception during movement in man. In *Experimental Brain Research*, vol. 68, pages 516-524, 1987.

[2] S. Ertan, C. Lee, A. Willets, H. Tan, and A. Pentland. A wearable haptic navigation guidance system. In *Digest of the Second International Symposium on Wearable Computers*, pages 164-165, 1998.

[3] S. Bosman, et al. GentleGuide: An exploration of haptic output for indoors pedestrian guidance. In *Lecture Notes in Computer Science*, pp. 358-362, 2003.

[4] K. Tsukada and M. Yasumura. Activebelt: Belt-type wearable tactile display for directional navigation. In *Lecture Notes in Computer Science*, vol. 3205, pp. 384-399, 2004.

[5] R. W. Angel and R. C. Malenka. Velocity-dependent suppression of cutaneous sensitivity during movement. In *Experimental neurology*, vol. 77, pp. 266-274, 1982.

[6] L. Post, I. Zompa, and C. Chapman. Perception of vibrotactile stimuli during motor activity in human subjects. In *Experimental Brain Research*, vol. 100, pp. 107-120, 1994.

Effects of Latency Jitter and Dropouts in Pointing Tasks

Andriy Pavlovych* Wolfgang Stuerzlinger†

York University, Toronto, Canada

Abstract

Interactive computing systems frequently use pointing as an input modality, while also supporting other forms of input. We focus on pointing and investigate the effects of variations, i.e. jitter, in the input device latency, as well as dropouts, on 2D pointing speed and accuracy. First, we characterize the latency, latency jitter, and dropouts in several common input technologies. Then we present an experiment, where we systematically explore combinations of dropouts, latency, and latency jitter on a desktop mouse. The results indicate that latency and dropouts have a strong effect on human performance; moderate amounts of jitter in latency do not change performance in a significant way in most cases.

Keywords: Latency, jitter, Fitts' law, pointing, dropouts.

1 Introduction

Latency, or lag, is the delay in device position updates [2]. Latency and spatial jitter have been previously demonstrated to significantly impact human performance in both 2D and 3D tasks [3], [5], [6], [8]. Recent interest in remote application use (application as a service, [7]), as well as a renewed interest in interactive network gaming [4] highlights the need for systematic study of this phenomenon. Also, the pointing devices are affected to varying degrees in the reliability of position tracking. Any failure of the sensing gives rise to *dropouts* in the sequence of position reports.

We present two empirical studies that systematically investigate the effects of dropouts and latency jitter on human performance. The studies employ Fitts' law, a well-established model of pointing device performance. In our experiments, we used a mouse as an exemplary low-latency, low-jitter device, and artificially added latency and latency jitter to it, to match the range of latencies and jitter present in other commonly used devices, as well as in computer networks. We also varied the number of samples the system was omitting ("dropping") and the periodicity of such omissions (Experiment 1), or the number and the percentage of the omitted samples (Experiment 2). The main goal was to determine, all else being equal, the effects of dropouts and latency jitter on device performance at varying mean latencies.

As one can often trade some latency for a decrease in latency jitter, typically through time-domain filtering, and extrapolate the missing and delayed samples, knowing the interrelationships between the factors allows a designer to make an informed decision in choosing an appropriate filter and its parameters.

2 Background

Latency is the time from when the device is physically moved to the time the corresponding update appears on the screen. For technical reasons, it is hard to avoid latency. And it is known that latency adversely affects human performance in both 2D pointing [3] and 3D pointing [9]. Common LCD displays have update rates of only 60 Hz and may exhibit lags of 40 ms [5]. If a DLP projector is used, latencies as high as 100 ms may be encountered, and many computer games have significant delays, with 80–150 ms being most common [1].

Spatial jitter is caused either by hand tremor or noise in the device signal or both. Some devices also exhibit additional noise during movements. Hand jitter only exacerbates this problem, especially in devices used in free-space. Temporal jitter, or latency jitter, refers to changes in lag with respect to time.

2.1 Characterizing Latency, Latency Jitter, Spatial Jitter, and Dropouts

To measure the latency, a video camera simultaneously filmed the motion of both the mouse and the cursor. The average delay of the mouse cursor motion relative to motion of the mouse was determined to be 8 ± 2.8 ms at the centre of the screen. More than 99.5% of the updates happened within 8–11 ms of the previous sample. Practically all of the remaining samples followed within 5–8 ms. We never observed a dropout in a mouse.

Optical sensing method employed by the mouse appears to filter the spatial jitter in hardware. Likewise, hand jitter, or hand tremor, does not appear to be an issue in our experiments, as resting the mouse on a physical surface largely eliminates tremor. Based on our measurements and the fact that our participants were young, we assume the input had no significant jitter of either kind.

3 Experiment 1

The first experiment compares effective throughputs under various magnitudes of latency, time jitter, and dropouts. Twelve students participated in the experiment. The study lasted 40–50 minutes. The software, implemented a standard Fitts' 2D task of 13 targets in a circle. The experiment was *within subjects*, and the order in which the various combinations of the factors were presented was randomized (without replacement), to compensate for asymmetric learning transfer effects. Each participant completed 100 "rounds" with different latencies, latency jitters, dropout durations, and dropout intervals, as described below.

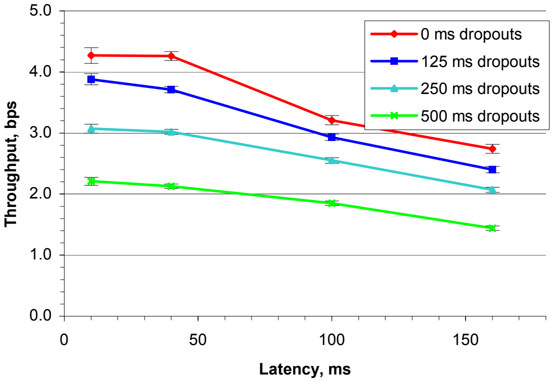


Figure 1. Throughput vs. latency and dropout duration. Here and further, error bars represent standard error.

The experiment had four independent variables in a (1×1 + 1×2 + 1×3 + 1×4) × (3×3 + 1) = 10×10 arrangement:

- *Latency* (constant part): 10*, 40, 100, and 160 ms;

[*] e-mail: andriyp@cs.yorku.ca

[†] e-mail: wolfgang@cs.yorku.ca

- *Dropout duration*: 0*, 125, 250, 500 ms;
- *Intervals between dropouts*: 0*, 500, 1000, 2000 ms.
- *Latency jitter* (normally distributed, in addition to the constant value above): σ = 0* ms for 10 ms latency, 0, ±20 ms for 40 ms latency, 0, ±20, ±40 ms for 100 ms latency, σ = 0, ±20, ±40, ±60 ms for 160 ms latency;

In the above list, * denotes the baseline condition, i.e., minimum latency, no latency jitter, and no dropouts. We chose a Poisson distribution for dropouts, as it is often used to model independent events, i.e., the time an event occurs does not depend on the previous occurrence. The indices of difficulty (ID), ranged evenly from 2.44 to 5.76 bits.

The dependent variable was effective device *throughput*.

3.1 Results

The effect of latency on throughput was significant, $F_{3,33} = 200.43$, $p < .0001$. The interaction between the latency and dropout duration was also significant, $F_{9,99} = 11.59$, $p < .0001$. Figure 1 illustrates the results.

4 Experiment 2

In this experiment investigate the effect of lower dropout percentages more thoroughly, to determine whether infrequent dropouts still have a measurable effect on throughput. Also, we aim to determine if there is a threshold for dropout duration, after which the throughput starts to drop progressively.

This experiment had three independent variables in a 4 × (5×5 + 1) = 4×26 arrangement, for a total of 104 combinations. In the following list, * denotes the baseline condition, i.e., minimum latency, no latency jitter, and no dropouts. The dependent variable was *effective device throughput* (in bits per second). All other aspects were similar to the preceding experiment

- *Latency* (constant): 10*, 40, 100, and 160 ms;
- *Dropout duration*: 0*, 10, 20, 40, 80, 160 ms;
- *Dropout percentage*: 0*, 1, 2, 5, 10, 20%.

5 Results and Overall Discussion

The effect of latency on throughput was significant, $F_{3,33} = 359.40$, $p < .0001$. No other significant interactions were observed. Figure 2 illustrates the results.

The effect of dropout duration on the throughput was significant, $F_{5,55} = 3.08$, $p < .05$. According to a Tukey-Kramer test, only the 160 ms condition was different from the others. The effect of dropout percentage on the throughput was significant, $F_{5,55} = 16.55$, $p < .0001$. According to a Tukey-Kramer test, no statistically significant difference exists between the 0, 1, 2, and 5% conditions. The interaction between the dropout percentage and duration was significant, $F_{16,176} = 2.18$, $p < .01$.

For low latencies, below approximately 40ms, we observed no significant differences in throughput, consistent with the first experiment and a previous study [5]. The significant interaction between latency and dropout percentages seems to be due to the 20% dropout condition, which has a significant drop of performance, $F_{1,11} = 8.17$, $p < .05$, even at low latencies, whereas the lower dropout conditions don't have such behaviour, $F_{1,11} = 0.09$, *ns*; see Figure 2.

For dropout durations of up to 80ms, there seems to be no significant effect on throughput, $F_{4,44} = 0.48$, *ns*. For dropout percentages up to 5% we observe no significant drop in performance, relative to the no-dropout condition. Looking at dropout durations of 160 ms we see a significant drop in performance above 5%, $F_{1,11} = 24.54$, $p < .0001$, and no drop before that, $F_{2,22} = 1.93$, $p = 0.16$. However, for lower dropout durations this transition happens at higher percentages, e.g., after 10%, for 80 ms-long dropouts, as can be observed in Figure 2.

One of the surprising findings (Exp. 1) was that latency jitter, that is, variations of latency with time, had little effect on performance, resulting in the worst case in an 8.5% drop in performance at 100 ms base latency and jitter with σ = 40 ms. Compared to the dramatic drops with increasing latency or dropouts, such a small drop is likely to be of little practical significance. Moreover, we can hypothesize that a higher, yet *constant*, latency could result in *worse* performance compared to just keeping the latency variations at their original level.

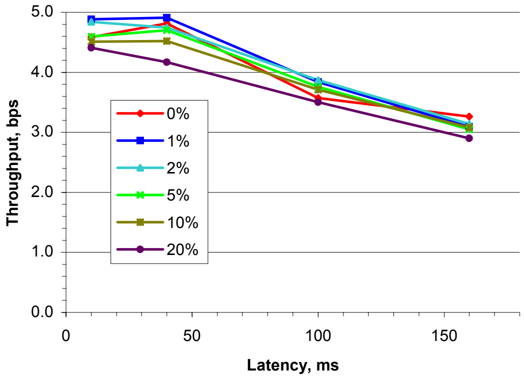


Figure 2. Throughput vs. levels of latency and dropout %.

For small dropout durations (up to 40 ms), dropout percentages can be relatively large (up to 20%), without noticeable effects on performance. On the other hand, longer dropouts (e.g. 160 ms) have significant effects even at low percentages (5% and more).

While long dropouts have a dramatic impact on performance, they are encountered in fewer situations, and, overall, their impact on performance is either similar to, or lighter than the impact of frequently encountered *latency* levels. Initial indications exist that interpolating dropouts by filtering may be of little or no use: for short intervals – because short dropouts have little effect on performance, and for large dropouts – due to this not being feasible. To summarize, while both latency and dropouts have detrimental effect on pointing performance, normally distributed latency jitter seems to have no noticeable effects. Filtering in order to combat latency jitter may actually be harmful, as the filter-added latency may outweigh any potential advantages.

Finally, we estimate that both latency and dropout duration are multiplicative factors for predicting the throughput. This suggests incorporating them into a homogeneous model for estimating the human pointing performance in the presence of latency and dropouts. This is a subject of future research.

References

[1] Console Gaming: The Lag Factor. http://www.eurogamer.net/articles/digitalfoundry-lag-factor-article.

[2] Foxlin, E. 2002 Motion tracking requirements and technologies. *Handbook of virtual environments: Design, implementation and applications*, Lawrence Erlbaum, 163- 210.

[3] MacKenzie, I. S., and Ware, C. 1993 Lag as a determinant of human performance in interactive systems. *ACM CHI '93*, 488.

[4] Online Gaming Sees Significant US Growth. Accessed Dec. 2009. http://www.edge-online.com/news/online-gaming-sees-significant-us-growth

[5] Pavlovych, A. and Stuerzlinger, W. 2009. The tradeoff between spatial jitter and latency in pointing tasks. In *Proc. EICS '09*. ACM Press , 187-196.

[6] So, R. H. Y., and Chung, G. K. M. 2005. Sensory Motor Responses in Virtual Environments: Studying the Effects of Image Latencies for Target-directed Hand Movement. *IEEE Engineering in Medicine and Biology Society*, 5006-5008.

[7] Software as a service. http://en.wikipedia.org/wiki/Software_as_a_Service

[8] Teather, R., Pavlovych, A., Stuerzlinger, W. and MacKenzie, S. 2009. Effects of tracking technology, latency, and spatial jitter on object movement, *IEEE 3DUI 2009*, 43-50.

[9] Ware C., and Balakrishnan, R. 1994. Reaching for objects in VR displays: lag and frame rate. *ACM TOCHI 1*, 4, 331-356.

Design of a Perceptual-based Object Group Selection Technique

Hoda Dehmeshki[*] Wolfgang Stuerzlinger[†]

York University

Abstract

Selecting groups of objects is a common task in graphical user interfaces. Current selection techniques such as lasso and rectangle selection become time-consuming and error-prone in dense configurations or when the area covered by targets is large or hard to reach. This paper presents a new pen-based interaction technique that allows users to efficiently select perceptual groups formed by the Gestalt principle of good continuity.

1 Introduction

Selecting groups of objects is a common task in graphical user interfaces and is required for many standard operations. Current selection techniques such as lasso and rectangle selection become time-consuming and error-prone in dense configurations or when the area covered by targets is large or hard to reach. Perceptual-based selection techniques considerably reduce selection time when targets form perceptual groups, as predicted by Gestalt principles of proximity and good continuity. However, they use heuristic and not-validated grouping algorithms. Also, they do not allow editing of a selection or selecting of groups with random configurations. Dehmeshki and Stuerzlinger developed a perceptual-based object group selection technique for mouse-based user interfaces [1]. In their system double-clicking on an object that is part of multiple (curvi-)linear groups selects all the groups. To deselect an undesired group, the user alt-clicks on its first non-desired object. Three key elements distinguish that system from the present work. First, clicks can specify only the location of a group but not the direction in which a group of objects extends. This makes selection less efficient when objects belong to multiple groups. Second, it provides no support for selecting non-perceptual groups. Finally, their system relies heavily on multiple-clicks, which is not appropriate for pen-based systems. This problem is shared by other techniques that use multi-clicking to cycle through different perceptual interpretations [3].

2 PerSel

This paper introduces PerSel, a new pen-based object group selection technique, which addresses the mentioned problems. PerSel consists of two components: The first component detects good continuation groups based on a neighborhood graph. The second provides a set of pen-based interaction techniques that use the detected groups to facilitate path-based selection.

2.1 Detecting Good Continuation Groups

The system first constructs a neighborhood graph. When the user performs a straight flick gesture starting from inside an object, the system examines all edges in the neighborhood graph that are connected to this object and picks the one which has the closest direction and distance to the gesture. The object and the edge are called the anchor object and the anchor edge, respectively.

PerSel is based on an implementation of Feldman's model [2] for linear groups which models paths as groupings of four objects combined with a sliding window paradigm. Given an anchor object 1 and anchor edge e, the algorithm finds all paths of length four starting from 1 and along edge e, see also figure 1. We call these paths primary paths. Then, for each primary path, the method computes a linearity coefficient (LC) that indicates how strongly the four nodes are perceived as a line:

$$\mathrm{LC} = \exp\left(-\frac{(a_1^2 + a_2^2 - 2ra_1a_2)}{2s^2(1-r^2)}\right),$$

where a_1 and a_2 are the angles between lines connecting the center of objects, and r and s are experimentally determined constants [2].

Primary paths with a LC smaller than a threshold are discarded as they are unlikely to be perceptually salient.If at least one primary path remains, the algorithm continues as follows: for each path it searches its potential continuations by identifying all neighbors of the last object in the path. Then, the four-object window is shifted to include each neighbor and the LC is computed in turn. If the new LC is smaller than a threshold, that completion is ignored, otherwise, the extension of the path with this neighbor is added to a stack. If all of the neighbors of the last node are unacceptable, the original path is kept; otherwise, the original path is discarded since at least one good extension has been found. The algorithm continues until all paths inside the stack have been processed.

Figure 1 illustrates this. In Fig. 1-a, the anchor node '1' and anchor edge 'e' are represented by thick borders. All four-vertex windows with small LC are visualized with ovals, while the ones with large LC are shown in rectangles. There are three primary paths, see Fig. 1-b, but only the one inside the rectangle is considered for further extension. After two more iterations, see Fig. 1-c and d, only the straight path is returned as being perceptually plausible, see Fig. 1-e.

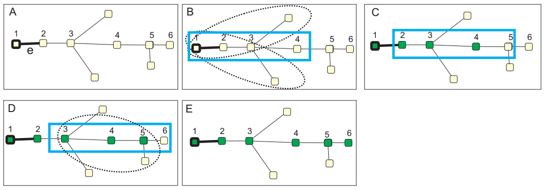


Figure 1: Illustration of Good continuity grouping algorithm. (a) shows the anchor object 1 and edge e. (b) the rectangle and ovals visualize primary paths (c) the path inside the rectangle is extended. (d) shows new potential primary paths. the one inside the rectangle is extended. (e) objects 1-6 are grouped.

For line gestures that cross an object, the gesture is first decomposed into two half-gestures, using the closest point of the gesture relative to the center of the object. Then, the linear groups corresponding to each half is found using the above methods, and then the two groups are merged.

Curvilinearity group detection: Similar to the linear case, when the user performs an arc gesture starting from inside an object, the system examines the edges in the neighborhood graph that are connected to this object. The edge closest to the gesture is picked, as defined by the sum of distances between gesture points and the edge. The rest of algorithm works similar to the linear case,

[*] e-mail: hoda@cs.yorku.ca

[†] e-mail: wolfgang@cs.yorku.ca

except that: (1) we use the curvilinearity coefficient that adapts the formula for LC by using the deviation from the average angle instead of the angles themselves, and (2) in the initial phase we only consider primary paths that turn in the same direction as the gesture.

2.2 Gestural Interaction

In this section we explain the gestural interaction techniques available in PerSel. As common in pen-based systems, tapping on a single object selects it. Also, PerSel cancels all selections whenever the user taps the pen on the background. This affords a simple and fast way to cancel erroneous selections.

Path Selection: Performing a straight gesture across an object selects the Good Continuation group aligned with the gesture direction. Similarly, an arc gesture across an object selects the curvilinear group that has a similar direction as the gesture. In both cases, the selected group is visualized by links connecting the successive objects, see Fig. 2.

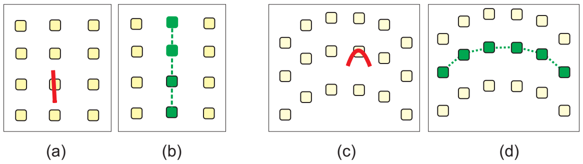


Figure 2: Good continuity group selection. Performing a line (a) or an arc (c) gesture, selects the corresponding linear (b) or curvilinear (c) group and visualizes it by links.

Partial Path Selection: There are two alternatives for partial selection of paths. The first way is to select the complete path and then cut undesired part(s) by drawing a flick gesture across one (or two) visualized links, see Fig. 3. The second alternative is to initiate the selection by a flick gesture from *inside* an object, see Fig. 4.

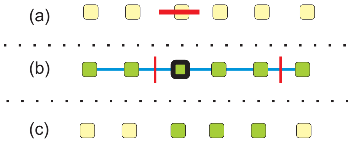


Figure 3: Partial group selection. (a) A flick gesture across an object selects the whole group. (b) Two flick gestures deselect all objects beyond these "cut" gestures.

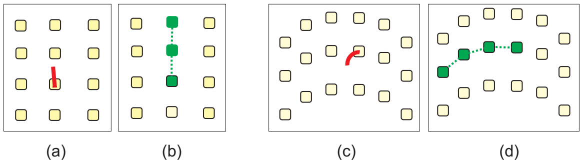


Figure 4: Partial group selection. In (a) respectively (c) a line and an arc gesture start from inside an object. Only objects that are on the same side of gesture are selected, as in (b) and (d).

Resolving Non-Perceptual Groups If a gesture corresponds to multiple potential curvilinear groups, all of them are selected. The user can then disambiguate the section by deselecting the non-desired groups. This is similar to the partial selection technique, in that the paradigm of "cutting" links is used to separate the non-desired objects from the targets, see Fig. 5.

Selecting Paths With Multiple Segments: More complex paths often consist of connected Good Continuity groups (segments). To enable selection of such paths, we introduce a new path editing feature. Assume that a path is already selected. If the user draws a gesture across an *already selected node* (called an anchor), a supplementary anchor is created. Then the selected path is modified by (1) automatically deselecting objects on the path beyond the new anchor, and (2) adding the new (curvi-)linear group corresponding to the new gesture and anchor to the selection, see Fig. 6.

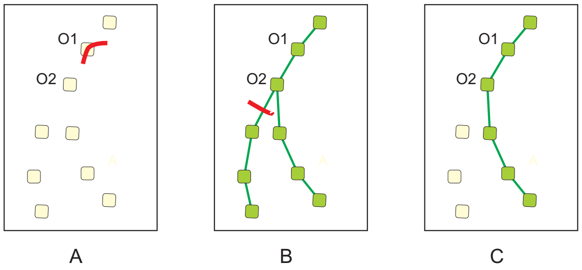


Figure 5: Resolving ambiguity. (a) An arc gesture on object O1 selects both curvilinear groups. (b) A "cutting" gesture disambiguates which objects to select. (c) Only desired objects remain selected.

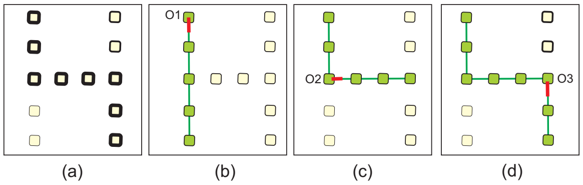


Figure 6: (a) Target objects have thick borders. (b) A line gesture over O1 selects the corresponding group. (c) A gesture from O2 guides the selection. (d) A gesture from O3 adds the remainder of the desired objects.

3 Conclusion and Future Work

This paper presented PerSel, a new gesture-based selection technique that is based on the Gestalt principle of Good Continuation. Performing a flick gesture crossing an object selects the (curvi-)linear group(s) that the object belongs to and is aligned with the gesture direction. PerSel also provides interaction techniques that allow users to perform partial group selection and selecting groups with arbitrary configurations. As future work we will include Gestalt principle of similarity and extend PerSel to deal with objects with different visual features such as shape and size.

References

[1] H. Dehmeshki and W. Stuerzlinger. Intelligent mouse-based object group selection. In *Smart Graphics*, 2008.

[2] J. Feldman. Curvelinearity, covariance, and regularity in perceptual groups. *Vision Research*, 37(20):2835–2848, 97.

[3] E. Saund and T. Moran. A perceptually-supported sketch editor. In *Proceedings of the ACM Symposium on User Interface Software and Technology–UIST'94*, pages 175–184, New York, 1994. ACM.

List of Authors

Bahvna Argawal
Sheldon Andrews
Lyn Bartram
Sungkuk Chun
Hoda Dehmeshki
Mohamed El-Zohairy
Zoltan Foley-Fisher
Minglun Gong
T.C. Nicholas Graham
Keechul Jung
Idin Karuei
Sebastian Koch
Paul Kry
Mauricio H. Lopez
Russ MacKenzie
Karon E. MacLean
Aaron Maynard
David Mould
Joey A. Pape
Andriy Pavlovych
Doina Precup
Olivier Rémillard
Johnny Rodgers
Wolfgang Stuerzlinger
Ahmed Sabbir Arif
Robert J. Teather
Gabriel Telles O'Neill
Mauricio Vines
Ling Xu