PASCAL: Physics Augmented Space Canvases for Animating Locomotion

Journal:	*Computer Animation and Virtual Worlds*
Manuscript ID:	draft
Wiley - Manuscript type:	Special Issue Paper
Date Submitted by the Author:	
Complete List of Authors:	Fei, Guangzheng; Animation School. Communication University of China Lee, Won-Sook; University of Ottawa, SITE, Faculty of Engineering Xin, Zijun; Communication University of China, Computers School Dong, Huikai; Communication University of China, Computers School Joslin, Chris; Carleton University, School of Information Technology
Keywords:	Sketch based animation, Space canvas, Physical simulation

PASCAL: Physics Augmented Space Canvases for Animating Locomotion

Guangzheng Fei, Animation School

Communication University of China

Won-Sook Lee, SITE, Faculty of Engineering

University of Ottawa

Zijun Xin, Computers School

Communication University of China

Huikai Dong, Computers School

Communication University of China

Chris Joslin, School of Information Technology

Carleton University

Abstract

We describe an animation creation system called PASCAL that supports sketch based modeling and physics augmented locomotion simultaneously. The system uses sketches and reconfigurable space canvases as basic modeling primitives and uses physics to improve the expressiveness and efficiency of several animation techniques to obtain controllable and plausible locomotion animation. The usability evaluation of the system was conducted both with professional and novice animators.

Keywords: Space Canvas, physical simulation, free-form deformation, inverse kinematics

Introduction

Sketches have been highly employed both in 2D and 3D animation production process. While they are considered the basic primitives for describing scenes and characters in traditional 2D animation production, they are also used by animators in 3D animation production as an aid to visual thinking during character design and scene design phase.

While an animation created by a professional animator with much effort attempts to evoke a certain response in the viewer, an animation created by a novice animator often serves as a communication aid. In both cases, animators demand a system to help them to quickly create convincing animations. Professional animators spend much time studying how things move, and understanding concepts such as anticipation and squash-and-stretch that help create expressive motion. Some systems [1, 2, 3] are able to automatically generate animated movements of 2D/3D characters through sketches. Others [4] use sketches to define motion paths and motion notations in animation process. Compared with these systems, PASCAL is less concerned with creating polish animation but more concerned with accomplishing expressive animation more quickly.

Though 3D animation has been very popular recently, traditional free-hand animation provides more artistic results. Therefore in the area of 3D animation, it is still true that many artists and designers keep their habit to work with free-hand when they need to form their idea. The current 3D technique equipped with 3D motion of object and camera is still comparably less popular in the early design phase due to its requirement of a good sense of

3D space and a good patience in tuning the tedious settings and parameters.

From the observation of 3D animation production, we question ourselves why the free-hand animation has been less and less popular and constantly replaced by 3D animation while many artists still want to use the free-hand version which is also more artistic and more expressive. We consider one of the biggest reasons is the convenience of 3D manipulation and free-view visualization of 3D scene. Therefore we take an approach starting from 2D drawing to the direction of 3D animation by helping the traditional designers to keep their skills with intuitive 3D interface. We consider that the traditional art production method must be kept with certain help from computers instead of replacing them by pure 3D animation.

In this paper we present a system that uses space canvas and its associated strokes as the foundation for a representation of a scene. Our system provides the user the capabilities to organize the drawn primitives in 3D space, while still retaining the essence of pen-and-paper based sketching manner to utilize his/her strong 2D drawing skill. To ease the animation design process, we enhance the functionality of key-framing, inverse kinematics (IK) and free-form deformation (FFD) techniques by introducing physics into the animation system.

The rest of the paper is organized as follows. We begin with a discussion on how our system relates to previous work. We then give an overview and a following detail description of the system in two aspects: sketch based modeling and physics augmented animation. We conclude with future directions in this research, based on the experience of animation art students using the system.

Related Work

The goal of our research is not about how to transfer 2D free-hand drawing into a 3D model, but about how to increase the flexibility and extend the capability of the free-hand drawing system while still retaining the essence of pen-and-paper based drawing. Thus, our review focuses on previous research work that directly uses the strokes without trying to convert them to 3D models, while enhancing 2D design productivity and flexibility by means of introducing 3D capabilities. Most sketching animation systems provide either modeling or animation approach, while our system considers the both approaches in a whole animation creation process. In the following paragraph we will discuss the sketch based modeling and physics augmented animation approaches respectively.

A system called 3D6B editor [5] was based on an idea to project strokes onto a user-definable 3D grid. The data that represents the created scene stored as 3D line segments (called strokes). The rendering of the scene is done with a line-based renderer in this system. With the functionality of moving the canvas while drawing strokes, it is able to create non-planar 3D strokes in their system. We found it very useful in creating conceptual models. However, as the number of 3D lines grows when representing a complex scene, it becomes very confusing and hard to control for designers. The inconvenience of manipulation of the canvases is another problem of this system that limits its usability. In addition, since no region-based occlusion is considered in this system, it is hardly applicable for a system where spatial layout reviewing is a major requirement, such as the 3D storyboarding system.

By examining the 3D6B editor[5], Dorsey et al. [6] realized its difficulty in organizing canvases and its lack of post-creation transformation or alteration of strokes, which makes the creation of a 3D sketch a tedious task. In order to overcome the deficiencies, they presented a system called mental canvas that uses strokes and planar "canvases" as basic primitives with the basic mode of input being traditional 2D drawing. The system allowed a designer to transfer strokes between canvases. They also introduced methods for a user to control stroke visibility. In addition, they provided a few built-in 3D assemblies of canvases in common arrangements: axial cross-sections, parallel stacks, and a circumferential ring.

The above sketch-based modeling systems have achieved some functionality for 3D freehand drawing by either drawing 3D curves or drawing 2D curves on reconfigurable canvases. However, the shortcomings are also obvious. All of the above systems deal with simple lines that can carry a few stroke properties thus make it less comfortable for artists to be artistically creative. An artistic stroke contains rich properties, such as variant width, texture and transparency.

To our knowledge, Baecker's Genesys [7] was the first informal animation sketching tool to use sketching for both creating objects and demonstrating motion. Genesys demonstrated that a computer could make an animator's work easier by capturing in-between frames from sketched input.

Davis et. al [8] present a general-purpose sketching system called K-Sketch that allows ordinary computer users to create informal 2D animations from sketches. The system is simple enough for novice animator to create explanatory animation, but it is far from a professional tool to create expressive animation.

Chenney et. al [9] have presented a mixed dynamic and kinematic model for simulating cartoon style squash-and-stretch motions in real time. They allow designer to easily specify particular styles of the motion.

In next section, we will give an overview of our system that is designed for creating expressive animation, where both sketch based modeling and physics augmented animation are supported.

System Overview

Since the complexity of conventional animation design tools mainly comes from their focus on precise details, one might be willing to sacrifice some precision and detail for ease of use and shorter completion time. Sketching is such an approach that has often been used to simplify the animation process in some design tools in order to make animation more accessible to non-professionals or to let professional animators focus on more creative part of their works. The sketching operations are often used as basic design tools for modeling objects and/or creating motions in these animation creation systems.

Our aim is to provide an animation tool, which is convenient for artists who prefer to free-hand drawing to create 2D like artistic works. In our system a user is allowed to create planar and curved canvases and place them in 3D space with six degrees of freedom. Free-hand strokes are drawn on each canvas by the user. In this way, our system extends the 2D artistic sketching to 3D, while strives to preserve the degree of expression, imagination, and simplicity of use in 2D drawing. Shortly, our system allows a user to draw on manipulable canvases in 3D space.

Our system also supports some physics augmented animation functionalities to achieve plausible and controllable object movements in the scene. Camera movements in 3D space are also supported in our system.

While sharing some basic idea with 3D6B editor [5] and mental canvas [6] in the modeling approach, our system defers from them in the following aspects:

- **Stroke representation:** Strokes in our system are represented as triangle strips rather than curve lines in the other systems. Our representation of strokes is more suitable for carrying various artistic properties.

- **Region occlusion:** Since a stroke is defined as a triangle strip in our system, user can easily create a region by overlapping wide strokes with various transparencies. All of the above mentioned systems except the mental canvas overlooked this functionality, thus might cause visual confusion when viewing a complex scene. The mental canvas system used a rasterized occlusion map for region occlusion, which would suffer from resolution problem when viewing the content on a canvas from a close distance.

- **Supported canvases:** Both planar canvases and curved canvases are supported in our system. The system also supports user-definable polyhedral canvases to allow designers to choose from for some common arrangements. Furthermore, a user is allowed to create a group of canvases and bind them together to create a canvas assembly, or to export them for future use.

- **Animation:** Both object and camera movements are supported in our system. A novel physics augmented animation approach is presented in the system, which improves not only the performance but also the expressiveness in animation creation process. We will give a detail description of this method in the section of physics augmented locomotion.

Sketch Based Modeling

In this section, we mainly focus on the several aspects that are deferent from the other sketch based modeling systems as mentioned in the previous section.

Stoke Representation

Similar to the approach presented in [10], we generate a triangle strip to represent the stroke, where the medial axis is defined by the sketch curve and the width of the stroke is either previously defined by the user or automatically determined by the pressure value of the tablet pen. First, the 2D sketch curve is projected onto the selected canvas to become the medial axis. Second, two offset curves on both sides of the projected curve (medial axis) are generated according to the width value. Both curves are subdivided into line segments and the vertices of the segments are then connected to form the triangle strip. Texture coordinates for these vertices are automatically generated at the same time.

Although simply defined, designers are already able to create artistic graphic using these strokes just by changing the provided parameters: width, texture and transparency, as shown in Figure 2.

All the strokes can be transformed within canvases after creation. For example, they can be translated or scaled along the two axes of the canvas and rotated about the axis perpendicular to the canvas at a given point.

Our system supports stroke grouping. Once grouped, all strokes are transformed together with their relative positions preserved. As a comparison, in mental canvas system [6], grouping functionality is only supported in canvas level. Thus makes it less efficient to reuse strokes within canvas.

Strokes are allowed to intersect each other so as to give enough freedom for designers to draw arbitrary artistic works. Obviously, when the strokes are drawn on a same planar canvas, all the generated vertices lie in the same plane. This may cause serious aliasing problem at intersectional regions, as we can see from the left side of Figure 3. The system can automatically offset the top stroke vertices at the intersectional region a little bit above to eliminate this artifact.

Region Occlusion

Since each stroke can have an associated transparency, it is free for the designers to choose either to blend the current stroke with the previously drawn strokes or to paint on top of them in an overlay mode.

In either case, when the designer keeps drawing strokes in one particular region, especially when a wide stroke has been chosen, he/she may produce a region filled with strokes which can occlude the behind strokes. As our stroke is represented as a triangle strip, it is easier in our system to create a filled region than in the other systems, by providing a larger width for the stroke. In order to improve system performance, we provide a functionality to reduce the number of triangles in this region by first calculating an outline of the region, removing the original triangles and then re-triangulating the outline area. The region occlusion problem is solved in the way that the designer is able to avoid visual confusion when viewing a complex scene.

The view dependent opacity for strokes and one-sidedness for canvas is also useful as suggested in mental canvas system [6] to help deal with the stroke visibility issue.

Canvas Creation and Object Management

Canvas management is a basic functionality in our system to extend the 2D sketch to 3D. In our system, both planar and non-planar canvases are supported. Planar canvases are represented by their positions and orientations. Optional parameters such as texture, one-sidedness, and billboard information are also provided. There are two types of non-planar canvases. The first type is generated from a user-specified stroke. Once a stroke curve is drawn on a canvas and a width of the non-planar canvas is given by a subsequential mouse movement away from (may or may not be perpendicular to the canvas) the canvas, the stroke curve will be first subdivided into line segments with a pre-defined constant interval. To avoid narrow canvas, a width under a certain threshold is rejected; in our system this value is set to be three times of the pre-defined constant interval.

The second type of non-planar canvas is the polyhedral canvas, created by grouping a set of planar canvases or imported from any simple-shaped polyhedral object.

Similar to strokes, canvases can also be grouped and ungrouped. Canvases and the associated strokes in a same group will be transformed together.

Once a canvas is created in the system, it can be transformed in a way same as any CAD programs. For example, the user can transform canvases by moving along or rotate about their local coordinate axes, optionally using a single-axis constraint along any of the three coordinate axes. We provide interface widgets also similar to a CAD program to help perform these transformations.

Stroke-Canvas Intersection

Once an active canvas is selected, a user can sketch on 2D screen, which evokes the system to calculate the projective strokes on the active canvas. Since both the planar and the non-planar canvases are finally defined by one or more planes, the projection of a stroke to a canvas is always performed by calculating the intersection between a plane/planes and a ray shooting from each 2D point of the curve drawn on screen following view direction.

A planar canvas is considered to have infinite size, so that any intersection of the ray and the plane can be accepted as a resulted projection. A bounding-box for the selection purpose is regenerated once the stroke has been projected out of the extent.

A non-planar canvas consists of some planar facets. The intersection is performed first by checking the intersection between the ray and each plane that the planar facet belongs to, and then by checking whether the intersection point is inside the facet. Only the intersection point that lies within a facet will be accepted as a real intersection to be recorded as the projection of the 2D point on the canvas. All facets of a canvas will be traversed until a projection is found. To connect points across two adjacent facets, the intermediate point on the edge between these two facets is calculated to ensure the shortest path to connect these three points. Two successive points that do not lie within one facet or two adjacent facets due to an abrupt move in sketching are rejected to avoid connection problems. Once a projective curve is generated, it is converted into a triangle strip in the way discussed in the above subsection for stroke representation.

Physics Augmented Locomotion

An important functionality of our system is that we allow a user to define motions for both the objects and the camera. As for the object motion, we utilize the physics in a novel way to enhance the several traditional animation techniques, such as key-framing, inverse kinematics, and free-form deformation to obtain some improvements in expressiveness. In order to keep the “what you see is what you get” feeling for the designer, the camera does not visually appear in the scene. The transformation of the camera can be obtained implicitly when the user interactively changes the view.

Physics augmented key-framing

Traditional key-frame techniques require the designer to set the transformation for each key frame, thus make it very tedious in creating a compound motion. In our system, instead of starting from scratch to assign transformation for each keyframe, we allow the designer to first assign physical properties for the objects and then run the physics engine to obtain a motion sequence. Keyframes are then automatically extracted from the motion sequence, which can be further tailored by the designer for better artistic result.

With the help of physical simulation, the designer can have a plausible motion to start from without a tedious tuning process or an expensive motion capture equipment. The system utilizes the motion recording approach to capture the simulated motion sequence. The recorded motion curves are approximated by piecewise Hermite curves by solving a dynamic programming problem [11]. The control points of the Hermite curves are used to define the keyframes, which are then provided for the designer to modify the motion, or to add detail for artistic purpose. Moreover, a time-warping tool is provided to synchronize the movements when multiple objects exist in the system.

Physics augmented inverse kinematics

According to some animators, a natural arm swing contains a major swing of the upper and lower arms and a subtle accompanying hand motion. There is nice little overlap of the hands at each end of the swing, where the hand requires to maintain an arc, as shown in Figure 4.

As we know, principles of physics govern the dynamic behaviors of objects in the physical world. By observation of the hand movements carefully, we found that the arc of the hand movement at each end of the swing comes from the inertial property of the hand. Therefore, we combine physics and IK together to obtain higher efficiency and better performance in such case, where, for example, the motion of the upper and lower arms can be specified by IK while the accompanying motion of hands can be computed by the physics engine. In this way, we maintain both the convenient interaction and the correct physics at the same time to give designers an intuitive grasp of the object.

Physics augmented free-form deformation

Squash-and-stretch is a common technique applied to character motion in cartoon style expressive animations. The technique to automatically create the standard squash-and-stretch effects in computer animation helps bring life to the animated characters. The typical scenario is that when a character or a vehicle hit one surface or have a sudden stop, the object tends to perform a flexible deformation rather than a rigid one.

Free-form deformations (FFDs) [12, 13] are the most versatile and powerful tools for representing and modeling flexible objects. Faloutsos et al. [14] presented dynamic free-form deformations that extended the use of FFDs to a dynamic setting. Inspired by this approach, we apply a dynamic FFD to the free-hand sketch to obtain the squash-and-stretch effect, where mass, inertial properties and internal forces are assigned to the deformable object to allow it to perform free-form deformation. Some research suggests volume (or area in 2D case) preserving deformation methods for deformation of incompressible 2D object. Our system can be easily extended by implementing such an approach as introduced by Weng et. al [15].

Figure 5 gives an example of the motion sequence of a car runs in a road, drops off a cliff and makes a sudden stop at the lower ground, where we can see the squash-and-stretch effect clearly when the car hits the lower ground and when it makes the sudden stop afterwards.

Experimental Results

Our system runs on a desktop workstation optionally with a pressure sensitive input tablet. The system can be executed at interactive rates. A scene created in the system is rendered exactly the same as the designer has drawn. There is no change in shading when the view is changed, since no lighting information is applied to the scene. In other words, the scene will remain in the 2D drawing style when it is manipulated. As a result, the designer gains a “what you see is what you get” experience when using with our system.

Through several user studies over several months, our system has obtained feedback saying that it provides significant advantages over both 2D drawing systems and 3D modeling and animation systems. Compared with 2D drawing systems, our system has the freedom for designers to continuously change the viewpoint. As we can see from Figure 6, once the scene is drawn, we can view the scene from different view directions as in any 3D programs, while retaining a sketchy style all the time. Compared with 3D modeling and animation systems, our system keeps the sense of traditional 2D drawing style very well. Designers do not have to follow the tedious 3D modeling, texturing, and lighting process to obtain a final result. The sketchy style of objects provided by our system can hardly be achieved by rendering 3D models in any traditional 3D programs even with some powerful non-photorealistic rendering functionalities. Besides, the 2D performance of our system is not limited to the current implementation. Any advanced functionality in 2D painting system could be introduced into our system to improve the drawing quality. Moreover, the functionality of physics augmented locomotion promotes both the efficiency and expressiveness in animation creation process.

Figure 7 and Figure 8 show another two results created by the animation art students. The quality of the animation produced in our system is quite acceptable. The students were pleased with the results they were able to achieve using our system, with only a shallow learning curve. The results here were created by the users familiar with some 3D modeling programs in order to compare our system with the 3D animation creation process. However, some novice animators also tried our system and gave quite good feedback about the usability.

Discussion and Future Work

This paper has presented a novel system for 3D animation creation using space canvases for free-hand drawing and using physics for augmenting animation creation. To our knowledge, it is the first system that designers have been able to create a 3D animation in the way almost same as 2D drawing, while with 3D capabilities in manipulating both the objects and the camera.

The system is able to create rather complicated scene which consists of only free-hand drawn strokes. All the strokes are drawn in 3D space with the help of either planar or non-planar manipulable canvases. Designers are allowed to draw strokes without facing the difficulty to deal with a polygonal mesh or the inflexibility of a parametric pipeline. The novel use of physics in our system introduces more expressiveness to the system without requiring much effort from the designer.

Our approach suggests several interesting avenues for future work. The animation on strokes can be more articulated focusing on intuitive user gestures. Noticing that, when the camera moves around a fixed planar canvas, the sliced appearance of the canvas may be seen at certain angles which may destroy the 3D impression of the whole scene. We suppose an approach that automatically creates novel view out of existing views may be utilized to give bigger freedom for the designer to go for full-3D feeling.

Acknowledgements

We would like to thank Yi Zheng, Jing Jin and other animation art students for experimenting with the system and creating the examples in this paper. This project was partly supported by NSFC grant (No. 60403037).

References

[1] M. Thorne, D. Burke, M. van de Panne, Motion doodle: an interface for sketching character motion, Proc. of SIGGRAPH ’04, 2004, pp. 424–431.

[2] J. Davis, M. Agrawala, E. Chuang, Z. Popovic, D. Salesin, A sketching interface for articulated figure animation, In Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on computer animation, 2003, pp. 320–328.

[3] F. Sumi, M. Nakajima, A production method of reusing existing 2D animation sequences, In Proc. Computer Graphics International 2003, pp. 282–287.

[4] D. Bourguignon, M. -P. Cani, G. Drettakis, Drawing for illustration and annotation in 3d, In Proc. Computer Graphics Forum, Blackwell Publishers, A. Chalmers and T.-M. Rhyne, Eds., vol. 20, In Proc. EUROGRAPHICS, 2001, pp. 114–122.

[5] K. Kallio, 3D6B editor: Projective 3d sketching with line-based rendering, In Proc. Eurographics Workshop on Sketch-Based Interfaces and Modeling, 2005, pp. 73–80.

[6] J. Dorsey, S. Xu, G. Smedresman, H. E. Rushmeier, L. McMillan, The mental canvas: A tool for conceptual architectural design and analysis, In Proc. Pacific Conference on Computer Graphics and Applications, 2007, pp. 201–210.

[7] R. Baecker, Picture-Driven Animation, In Proc. AFIPS Spring Joint Computer Conference, 34 (1969), pp. 273–288.

[8] R. C. Davis, B. Colwell, J. A. Landay, K-sketch: a “kinetic” sketch pad for novice animators, In Proc. SIGCHI, 2008, pp. 413–422.

[9] S. Chenney, M. Pingel, R. Iverson, M. Szymanski, Simulating cartoon style animation, In Proc. NPAR 2002, Annecy, France, 2002, pp. 133–138.

[10] A. Hertzmann, Painterly rendering with curved brush strokes of multiple sizes, ACM Press/ACM SIGGRAPH, New York, M. Cohen, Ed., 1998, pp. 453–460.

[11] M. Plass, M. Stone, Curve-fitting with piecewise parametric cubics, In Proc. ACM SIGGRAPH ’83, 1983, pp. 229–239.

[12] T. W. Sederberg, S. R. Parry, Free-Form Deformation of Solid Geometric Models, In Proc. SIGGRAPH 86, 1986, pp. 151–159.

[13] S. Coquillart, Extended Free-Form Deformation: A sculpturing Tool for 3D Geometric Modeling, In Proc. SIGGRAPH 90, 1990, pp. 187–196.

[14] P. Faloutsos, M. van de Panne, D. Terzopoulos, Dynamic Free-Form Deformations for Animation Synthesis, IEEE Transactions on Visualization and Computer Graphics, 1997, vol. 3, pp. 201–214.

[15] Y. Weng, W. Xu, Y. Wu, K. Zhou, B. Guo 2D shape deformation using nonlinear least squares optimization. The Visual Computer 22(9-11), 2006, pp. 653–660.

Figure 1: A scene created in our system by an animation student using free-hand drawing.

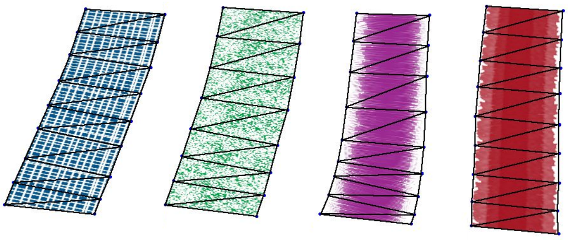

Figure 2: Strokes represented as triangle strips.

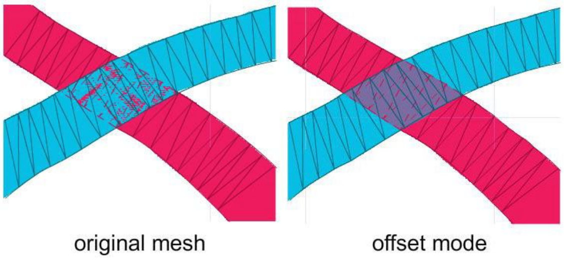

Figure 3: Aliasing caused by stroke intersection and the solution.

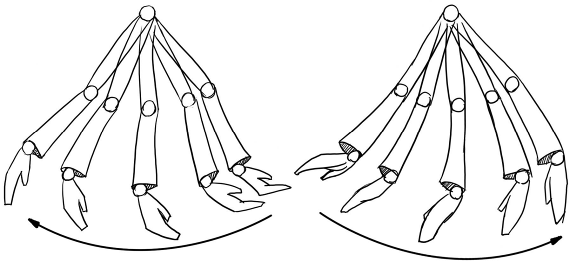

Figure 4: A typical sequence of an arm swing.

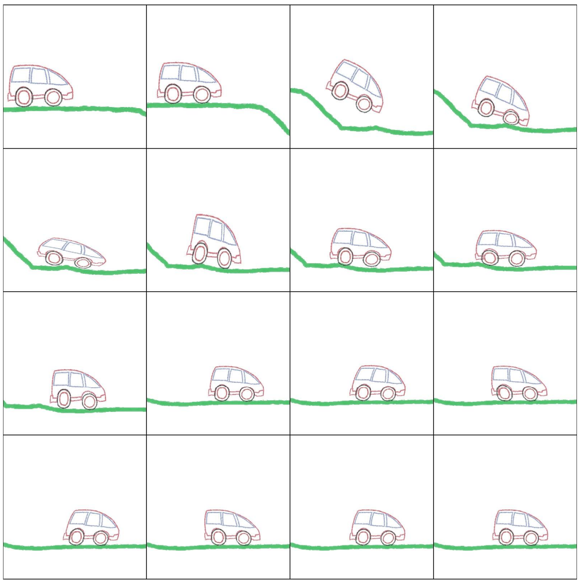

Figure 5: Squash-and-stretch effect of the car.

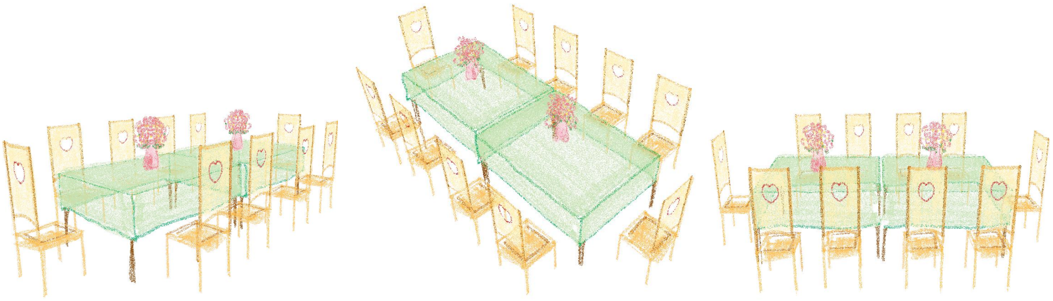

Figure 6: A glass table and several chairs shown in different views.

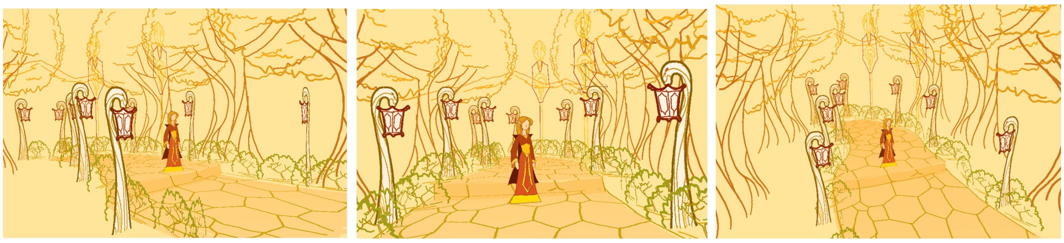

Figure 7: A lady in traditional dress walking in a marble road with a woody surrounding.

Figure 8: A fairy flying across a peaceful lake.

Dr. Guangzheng Fei is an associate professor in the Animation School of Communication University of China. He received his PhD from Institute of Software, Chinese Academy of Sciences, China. His research interests include computer graphics, computer animation, virtual reality and computer game. He has published over 50 peer reviewed conference and journal papers.

Dr. Won-Sook Lee, associate professor in the School of Information Technology and Engineering, Faculty of Engineering at University of Ottawa, Canada, received her PhD from the University of Geneva, Switzerland. She has wide experience in the field of computer graphics and related application areas such as computer animation, computer game, virtual reality, and medical imaging as well as haptic graphics. She has over 60 peer reviewed publications.

Zijun Xin is a graduate student majored in computer application at Computer School, Communication University of China. He received his bachelor degree in statistics at the Science School of Beijing Institute of Technology. His research interests are computer graphics, computer animation, and human-computer interaction.

Huikai Dong is a graduate student major in Computer Application at Computer School, Communication University of China. He received his B.S degree major in Electronic Information Engineering at College of Physics and Electronics, Shanxi University. His research interest includes computer graphics, computer animation, and human-computer interaction.

Dr. Chris Joslin is an assistant professor at the School of Information Technology, Carleton University. He received his PhD from the University of Geneva, Switzerland. He is a member of the Canadian Advisory Committee for Standards Council Canada, where he advises on matters arising for Canada's position for the International Standards Organisation (ISO) and also contributed methods on Device Benchmarking to the standard MPEG-21 Digital Item Adaptation (DIA). He has published numerous conference and journal articles in the area of media adaptation, medical preoperative planning, and collaborative virtual reality systems. His research interests have extended towards facial animation and media compression.