Interactive Walk-Through by a Simple Bicycle Fitness Facility on a GPU-Accelerated Terrain Visualization System

Hung-Kuang Chen
Dept. of Electronic Engineering
National Chin-Yi University of Technology
Taichung, Taiwan 40109
Email: hankchentw@gmail.com

Wea-Sung Chen
Information and Design Department
Asia University
Email: weasung@gmail.com

***Abstract*—Current bicycle fitness facilities mostly do not provide interactive visual feedback. People usually feel boring after using such kind of facilities for a few minutes. To cope with this, we presented a simple system that combines a simple bicycle fitness facility with a terrain visualization system. With the new system, the user may use the bicycle fitness facility as an input device to our terrain visualization system and interactive experience of riding on the custom load terrain in the virtual world.**

I. Introduction

As a pollution-free, quiet, energy-efficient, and space-saving transportation tool, riding bicycle has become popular again in these years. In comparison with driving a car, bicycle riding is advantageous in many ways: for example, a bicycle is much cheaper than a car, easy to maintain, no oil or gas consumption, free of license and taxes, etc. Furthermore, bicycle riding is a very food aerobic sport, which helps us improving our healthy in many ways.

Many people enjoy riding bicycle outdoors in the holiday on various types of terrain. However, in stormy or snowy days, riding bicycle outdoors is not comfortable and safe. People prefer staying home and doing in-door activities. For bicycle fans, it could be a very hard time. Current bicycle fitness facilities mostly equipped only with simple meters such as the tachometer, odometer, timers, or some systolic or blood-pressure sensors. People simply use them for fitness reason and rarely have fun in using these facilities.

To address this problem, we presented a simple system that combines a simple bicycle fitness facility and a terrain visualization system. In the new system, the user may interactively ride-through a custom load terrain by the bicycle fitness facility as if he is riding on a bicycle on the terrain in the virtual world. To increase the interactivity by accelerating the rendering speed and increase the realism, we have implemented a two-pass special-effect shader on the GPU to visualize the terrain with ripple and environmental reflection effects.

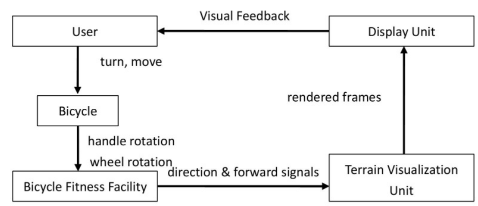

Fig. 1. The interactive terrain visualization bicycle fitness system.

II. The Interactive Bicycle Controlled Terrain Visualization System

As shown in the Fig. 1, our system comprises 5 major components: the bicycle, the fixation site, the terrain visualization system, the display system, and the user. In this system, we adopted an LCD TV or projector as our major display device. To increase the degree of immersion, it is allowed to substitute the display device with a head mounted display(HMD) device or to tile two or more LCD TV or projectors. We begin the discussion with a briefly introduction to the fixation site then on the terrain visualization system we have implemented.

A. The simple bicycle fitness system

To provide feedback signals from the bicycle to the terrain visualization system, we have adopted a simple interactive bicycle fitness system developed by the Cycling & Health Technology Industry R&D Center, Taiwan. The simple system, as shown in Fig. 2, is a portable bicycle fixation site built-in with two type of sensors. The user can attach or release their bicycle of arbitrary size to the fixation site by simply screwing or unscrewing the fixation slots beside the back wheels, respectively. At the front end of the fixation site, a variable resistor is build in the turn table underneath the front wheel to detect the turn of the handle. At the rear end, a rotation sensor is built inside a roller placed underneath the back wheel to detect the bicycle movements. Both sensors are wired to the circuitry connecting the host computer through the USB interface.

978-1-4577-2119-9/12/$26.00

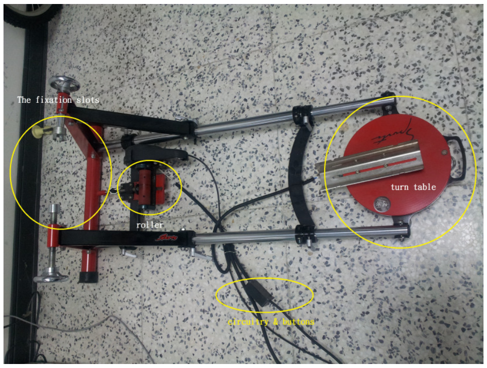

Fig. 2. The bicycle fixation site.

B. The Terrain Visualization System

Terrain is usually the largest object in a 3D scene and plays a very important role in applications such as the landscape design, flight/battle-ground simulations, feature film special effects and computer games [1]. Terrain related issues such as the terrain representation, terrain generation, and terrain visualization etc., has been intensively studied [1]–[14]. Similar to other type of research problems, the representation issue is usually the most fundamental and controversial problem. Thus, we do not put much of our efforts on this issue and choose a common representation of the terrain, the height map or the digital elevation map(DEM). In the remaining of this section, we will give a discussion on the terrain generation and visualization techniques that we have employed in our system.

Terrain Generation: The height map or the digital elevation grid is an ordered dataset that defines a sampling of the elevations, or heights, over the surface of a terrain. Currently, there are at least two primary ways to generate a height map. The first is to generate the terrain from real-world information such as a range map from remote sensing, an aerial photograph, or a set of images [7]. Second, it can be custom made manually by a sketch system, automatically by procedural methods [2]–[4], [6], [9], [12], [13], or a combination of both techniques [1], [10], [11], [14], [15]. In our system, we have supported both ways. The user can either download a height map from the internet and use it in the system or to create an initial height map by a common painter software followed by applying the mid-point displacement method. The midpoint displacement algorithm works by iteratively perturbing the height similar to the way that the crustal movement creates the folding structure on the earth's surface. In our system, we adopted the diamond-square algorithm as described in [5], [8]. As illustrated in Fig. 3, the algorithm begin by assigning a random elevation value to each of the four grid points of the rectangular ground plane. To obtain each elevation value, an averaging and addition of a Gaussian random variable procedure, called the diamond-square, is iteratively performed to refine the terrain until a specified detail is achieved.

In each iteration of a diamond-square procedure, the original grid is subdivided to obtain five new grid points: namely,

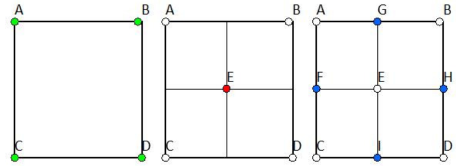

Fig. 3. The diamond-square procedure.

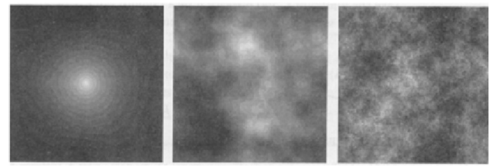

Fig. 4. An example of applying diamond-square method.

E, F, G, H, and J. The calculation of the center of the grid in Fig. 3 is called the diamond step, where a new grid point E on the center of the grid is introduced by subdividing the grid into four equal cells, whose elevation value is given as follows.

$$E = \frac{A + B + C + D}{4} + rand(), \tag{1}$$

Following to the diamond step, four new grid points are introduce by taking the center of the four boundary edges in the square step. The elevations of these four points: namely, the points F, G, H, and J in Fig. 3, can be given in the following calculations.

$$F = \frac{A + C}{2} + rand(), \tag{2}$$

$$G = \frac{A + B}{2} + rand(), \tag{3}$$

$$H = \frac{B + D}{2} + rand(), \tag{4}$$

$$I = \frac{D + C}{2} + rand(). \tag{5}$$

The two steps are recursively performed to each newly created grid, e.g., grid AGFE in Fig. 3, until a desired level of approximation is achieved. Note that the diamond step must be completed before the square step starts. An example of such subdivision is shown in the Fig. 4.

The Two-Pass Shader: As shown in Fig. 5, the visualization of the terrain mesh can be divided into the preparation stage and the two-pass rendering stage.

In the preparation stage, the terrain mesh is generated from a given 256 grey-level image, the height map. To shade the terrain, a two-pass renderer is employed, which combines a cartoon style shader and a special-effect shader that operates under three sub-stages: i.e, the pre-shading, feedback, and post-shading sub-stages.

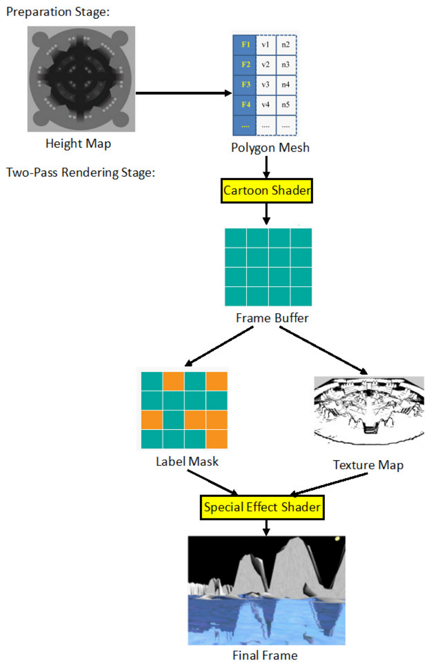

Fig. 5. The work flow of the two-pass terrain shading.

1) The Preparation Stage: In the preparation stage, the system reads in a height map and generates a triangle mesh for the terrain. Since the input is enforced to be a 256 grey levels image, pixel value 0 and 255 respectively represents the lowest and highest region. It is very often that the grid resolution of the terrain does not match with that of the height map. Therefore, the process of calculating the elevations of the grid points of the initial terrain mesh is like taking samples from the height map. If the resolution of the height map is larger than that of the terrain, an averaging filter is required; otherwise, an interpolation filter is needed.

To illustrate the process of the sampling, we assume that the the height map given is denote as $I[w, h]$ where its size is w pixels of width and h pixels of height; the terrain to be created is essentially a mesh containing $m \times n$ grids on the $X - Y$ plane, denoted as $T[m, n]$. Let the bounding volume of the terrain be $[(x_{min}, y_{min}, z_{min}), (x_{max}, y_{max}, z_{max})]$. Given a grid point of T, say $T[i, j]$, the planar location of $T[i, j]$ on $X - Y$ plane can be given as follows.

$$x = x_{min} + \frac{i}{m} \times (x_{max} - x_{min}) \tag{6}$$

$$y = y_{min} + \frac{j}{n} \times (y_{max} - y_{min}) \tag{7}$$

To calculate the height of the grid point, we have applied a simple nearest sampling followed by Gaussian smoothing as follows.

$$t_z = z_{min} + \frac{h[M_w(i), M_h(j)]}{255} \times (z_{max} - z_{min}), \tag{8}$$

where

$$M_w(i) = \lfloor \frac{i}{m} \times w \rfloor, \tag{9}$$

$$M_h(j) = \lfloor \frac{j}{n} \times h \rfloor. \tag{10}$$

and

$$h[i, j] = I[i, j] \cdot g[i, j] = \sum_{k=1}^{w} \sum_{l=1}^{h} I[k, l] \cdot g[i - k, j - l] \tag{11}$$

where

$$g[i, j] = e^{-\frac{(i^2 + j^2)}{2\sigma^2}} \tag{12}$$

On the other hand, if the resolution of the terrain required is larger than the height map, we can start with an initial grid with the same resolution as the height map followed by a series of mid-point displacements to achieve the desired resolution.

2) The Two-Pass Rendering Stage: As we have mentioned earlier, the two-pass renderer combines a cartoon style shader and a special-effect shader, which operates under three sub-stages: i.e, the pre-shading, feedback, and post-shading sub-stages. The pre-shading stage generates a rough outlook of the landscape. The result is used to label the ground planes and packed as a texture passing to the post-shading stage to simulate an iced and wet landscape with waves and environmental reflections.

In pre-shading sub-stage, the terrain is rendered with a cartoon style shader that uses the vertex processors to determine the visible surfaces and the silhouettes by taking the inner product of the view vector and the vertex normal. With this shader, the silhouette is shaded in black color and assigned different color to plains and slopes.

The display feedback stage has two major tasks. The first is to pass the results reside in the frame buffer to the texture memory. The other is to label the plains for the reflection shader to shade the environmental reflections and waves.

With the labels passed from the display feedback stage, a special-effect shader is used to shade the plains as waters or iced ground which increases aesthetic feel of the scene by introducing environment reflections and the effect of ripples.

III. Results

The overall rendering speed is maintained about 100 frame per second to 170 frame per second. A screen capture of the display generated by the terrain visualization system is shown in Fig. 6. The window on upper-right corner of Fig. 6 shows the height map used to generated the terrain; the window on lower-right corner shows the terrain grids by wireframe rendering. On the left side of Fig. 6, an fly-through view of the terrain without special rendering is shown.

In Fig. 7, the upper-left and the lower-right pictures respectively shows the result of shading the terrain with ripple water and iced ground effects; the upper-right and the lower-right pictures respectively shows the results of cartoon shading from aerial and ground view.

IV. The Conclusion and Future Suggestions

In this project, we have presented a novel system that combines a simple bicycle fitness facility and a terrain visualization

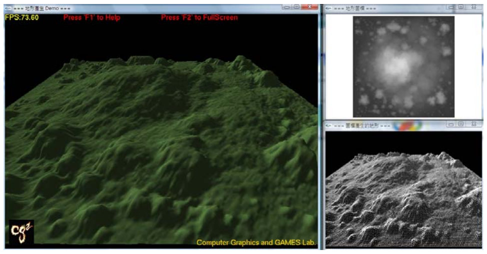

Fig. 6. The terrain visualization system.

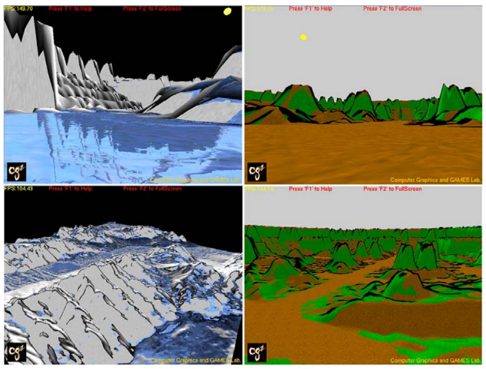

Fig. 7. Four shading outputs of the terrain visualization system.

system. In the new system, the user may interactively ride-through a custom load terrain by the bicycle fitness facility as if he is riding on a bicycle on the terrain in the virtual world. To increase the interactivity by accelerating the rendering speed and increase the realism, we have implemented a two-pass special-effect shader on the GPU to visualize the terrain with ripple and environmental reflection effects.

The system can be regarded as a novel application of terrain visualization on bicycle fitness. The interactivity is enhanced radically by substituting the traditional gamepad or mouse-keyboard input with the interactive bicycle fitness system. In compared with the commercial Bike gaming system developed by Sega Inc., our system is obviously has the advantages of lower costs, bigger screen, and more flexible or portable.

We have two suggestions for future researches. First, it would be useful and fun to show the street-side view of a city on the bicycle fitness system as a remote touring or sightseeing application. Second, the system can be further improved by introducing more sensors, buttons, or haptic feedback device to serve as a better gaming platform.

REFERENCES

[1] H. Zhou, J. Sun, G. Turk, and J. Rehg, "Terrain synthesis from digital elevation models," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, pp. 834 –848, July-August 2007.

[2] G. S. P. Miller, "The definition and rendering of terrain maps," *SIGGRAPH Comput. Graph.*, vol. 20, pp. 39–48, August 1986.

[3] A. D. Kelley, M. C. Malin, and G. M. Nielson, "Terrain simulation using a model of stream erosion," *SIGGRAPH Comput. Graph.*, vol. 22, pp. 263–268, June 1988.

[4] F. K. Musgrave, C. E. Kolb, and R. S. Mace, "The synthesis and rendering of eroded fractal terrains," *SIGGRAPH Comput. Graph.*, vol. 23, pp. 41–50, July 1989.

[5] E.-K. Koh and D. D. Hearn, "Fast generation and surface structuring methods for terrain and other natural phenomena," *Computer Graphics Forum*, vol. 11, no. 3, pp. 169–180, 1992.

[6] J. V. L. V. Pabst and H. Jense, "Dynamic terrain generation based on multifractal techniques," in *Proceedings of the International Workshop on High Per-formance Computing for Computer Graphics and Visualisation*, pp. 186–199, 1996.

[7] K. Martin, "Using bitmaps for automatic generation of large-scale terrain models," *Game Developer*, pp. 48–53, October 1999.

[8] C. Dachsbacher, *Interactive Terrain Rendering: Towards Realism with Procedural Models and Graphics Hardware*. PhD thesis, Universitat Erlangen-Nurnberg, Universitatsstrase. 4, 91054 Erlangen, 2006.

[9] K. R. Kamal and Y. S. Uddin, "Parametrically controlled terrain generation," in *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia*, GRAPHITE '07, (New York, NY, USA), pp. 17–23, ACM, 2007.

[10] R.-B. Zhao and M.-Y. Pang, "Synthesizing large-scale virtual terrain from image atlas," in *Education Technology and Training, 2008. and 2008 International Workshop on Geoscience and Remote Sensing. ETT and GRS 2008. International Workshop on*, vol. 1, pp. 242–246, Dec. 2008.

[11] G. J. P. de Carpentier and R. Bidarra, "Interactive gpu-based procedural heightfield brushes," in *Proceedings of the 4th International Conference on Foundations of Digital Games*, FDG '09, (New York, NY, USA), pp. 55–62, ACM, 2009.

[12] A. Peytavie, E. Galin, J. Grosjean, and S. Merillou, "Arches: a framework for modeling complex terrains," *Computer Graphics Forum*, vol. 28, no. 2, pp. 457–467, 2009.

[13] H. Hnaidi, E. Guerin, S. Akkouche, A. Peytavie, and E. Galin, "Feature based terrain generation using diffusion equation," *Computer Graphics Forum*, vol. 29, no. 7, pp. 2179–2186, 2010.

[14] R.-B. Zhao, J.-D. Wang, B. Yang, and P. Ming-Yong, "A method for generating large-scale terrain based on image set," in *Image and Signal Processing (CISP), 2010 3rd International Congress on*, vol. 1, pp. 16–19, Oct. 2010.

[15] J. Gain, P. Marais, and W. Strasser, "Terrain sketching," in *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, I3D '09, (New York, NY, USA), pp. 31–38, ACM, 2009.