Al-Rousan, R, Sunar, MS, Kolivand, H and Alhajhamad, H

Interactive non-photorealistic rendering

Article

Jurnal
Teknologi

Full Paper

INTERACTIVE NON-PHOTOREALISTIC RENDERING

Riyad Al-Rousan, Mohd Shahrizal Sunar*, Hoshang Kolivand, Hasan Alhajhamad

MaGIC-X (Media and Games Innovation Centre of Excellence) UTM-IRDA Digital Media Centre Universiti Teknologi Malaysia, 81310 UTM Johor Bahru, Johor, MALAYSIA

Article history
Received
3 December 2013
Received in revised form
2 July 2014
Accepted
25 November 2014

*Corresponding author
shahrizal@utm.my

Graphical abstract

Abstract

Due to increasing demands of artistic style with Interactive Rate, we propose this review paper as a starting point for any person interested in researching of interactive non-photorealistic rendering. As a simple yet effective means of visual communication, interactive non-photorealistic rendering generates images that are closer to human-drawn than are created by traditional computer graphics techniques with more expressing meaningful visual information. This paper presents taxonomy of interactive non-photorealistic rendering techniques which developed over the past two decades, structured according to the design characteristics and behavior of each technique. Also, it covers the most important algorithms in interactive stylized shade and line drawing, and separately discussing their advantages and disadvantages. The review then concludes with a discussion of the main issues and technical challenges for Interactive Non-Photorealistic Rendering techniques. In addition, this paper discusses the effect of modified phong shading model in order to create toon shading appearance.

Keywords: Non-photorealistic rendering, interaction techniques, shading, line drawings

1.0 INTRODUCTION

Non-Photorealistic Rendering is a branch of computer graphics which concentrate on the methods that generate images in a diversity of artistic and illustrative styles [1, 2, 3]. Although the big progress over the last two decades, the NPR field is still an active area of research [4]. Historically, producing images which are indistinguishable from the original is the main area of computer graphics. In many cases the Photorealistic is not the optimal solution for rendering the scene [5]. In contrast, non-photorealistic images are important for conveying the information and adding the simplicity of complex objects [6].

One of the most common goals of non-photorealistic rendering is to produce images which appear to be made by an artist [7]. Non-photorealistic rendering is considered more effective than photorealistic rendering in many cases such as physical structure or phenomena. In addition, non-photorealistic rendering will be more expressive when is made at interactive rate.

Non-photorealistic rendering typically generates static images which prevent the user making any editable or modifying operation [10]. On the other hand, non-photorealistic rendering has as a set of parameters can be varied over the time such as light direction, zoom and position. Tuning these parameters by the user enhances the spatial structure and clarifies the shape. Also, tweaking the rendering parameters produces many styles of the resulting images. Consequently, combining non-photorealistic rendering with interactive methods enhances the artistic shape [8, 9].

Recent research about non-photorealistic rendering proved that is affirmative to involve the user's input into the process of creating a Non-photorealistic System [11]. Shading model and edge lines are the most vital features in Interactive NPR system which enhances the capability to interactively display. In addition, complex geometric models must be rendered at interactive rate. This research describes many techniques for achieving these goals.

1.1 Motivation

Interactive rendering allows users to select rendering appropriate parameters for their application. Often, Non-photorealistic Rendering software is not used by the artists. The user maybe a scientist who would like to produce a specific style of images that demonstrates the structure he is studying. The user might not recognize which Non- photorealistic rendering technique is appropriate for his project. In this case, the tweaking process of rendering parameters at interactive rate leads the user to find quickly the appropriate final image.

Humans depend on hand for hundreds of years to create illustrations for medical, artistic, scientific, and entertainment purposes. Computer graphics fields such as Non-Photorealistic Rendering tried to create images such illustrations. Although a huge number of Non-Photorealistic Rendering techniques exist, they have not yet been approved by illustrators. Recently, Non-Photorealistic Rendering research is focusing on demonstrating these techniques with providing the illustrator the freedom freedom to modify the parameters for getting the desired image [12].

Artists commonly use tweaking shading behavior to depict objects, this is because tweaking shading has an exceptional ability to convey the shape characteristics. However, a lot of the editing job is still worked by hand [13, 14, 15].In order to provide the users of computer graphic applications with similar efficiency for conveying shapes, several rendering techniques for controlling the shading behaviors have been presented. Consequently, Shading-based techniques are now popular for depicting 3D object shape.

1.2 Applications of Interactive NPR

In many applications, such as architectural, industrial, automotive and graphics design, Non-photorealistic is preferred than photorealism. Non-photorealistic conveys information better by omitting extraneous details, by focusing attention on relevant features, by clarifying, simplifying and disambiguating shapes, and showing parts that are hidden. It also provides more natural vehicle for conveying information at different level of details.

Computer games and Movies considered the main applications of Non-Photorealistic Rendering [16]. For Movies, 28 frames per second require to create animation which considers a big challenge for traditional animation [17]. Most of computer studio games tried to produce more realistic games. With the magnificent development with computer graphics hardware, Rendering the complex models or realistic scenes became existed issue. While the saturation point of realism was reached, the game studios started to produce games with a stylistic look such as XIII (2003), Prince of Persia (2008), and Borderlands (2009).

Computer Generated Caricature is considered another branch of Non-Photorealistic Rendering Applications which is developed in order to assist the user in producing caricature automatically or semi-automatically [18].

2.0 STYLIZED LIGHTING AND SHADING

Ample publications inToon-Shadingdiscipline started from 1996. This section elaborates these methods in details depending on chronological time

Decaudin *et al*. [19] presented a rendering technique that creates images with the appearance of a conventional cartoon from a 3D description of the scene. The 3D scene is rendered with multiple algorithms; the first one was used in order to send the outline and edges of objects to the back. This algorithm was presented at [20]. The second algorithm which described uniformly color the surface inside the outlines. The last algorithm was rendering the shadows (backface shadows and projected shadows) due to light sources. The shading model that used in this technique was based on the Phong shading model.

Gooch *et al*. [21] introduced Non-Photorealistic Rendering techniques that extended to imitate diverse stylistic appearances. Sloan *et al*. [22] presented the method defined as lit-sphere which depicts the details of view-independent cartoon, through a painted spherical environment map. Also they discussed the artistic problems in creating cartoon movies by hand and how the computer can assist to solve this problem.

Cartoon shaders are considered as the most important achievement of computer graphics methods that applied to the industry of cartoon animation [23]. Lake *et al*. [24] introduced numerous elementary real-time rendering algorithms for rendering 3D objects in stylized way, including a conventional toon-shader. Also he employed the geometry into scenes of cartoon to emphasize the motion of cartoon objects. The technique employed in this research based on the calculating of conventional diffuse lighting and texture mapping. The final result was a system that employed 3D model to produce 2D cartoon. This system presented many stylized results like cartoon shading [25], pencil shading and stylistic inking. Lake *et al*. [24] Calculates the light by the next equation.

$$\mathrm{Ci} = \mathrm{ag} \times \mathrm{am\ al} \times \mathrm{am} + \mathrm{Max}(\mathrm{L}\,.\,\mathrm{n}, 0) \times \mathrm{dl} \times \mathrm{dm} \qquad (1)$$

where Ci is the vertex color, ag is the coefficient of global ambient light, am is the ambient coefficient of the object's material, al is ambient coefficient of light, dl is diffuse coefficient of light, dm is diffuse coefficient of object's material, L is light unit vector from source to the vertex, and n is the normal unit vector surface at the vertex.

Instead of computing the colors per vertex, a texture map is created of a minimal number of colors. Often two or three colors are used to convey the shding effect. The main color of the texture map is computed by replacing the dot product term from equation 1 with a value of one (which is equivalent to an angle of zero degrees, and normally happens when the light is directed right at the vertex). The dark color of the texture map is calculated by replacing the dot product term in Equation 1 with a value of zero. Figure 1 illustrated how the light position and normal direction are employed to classify into the one-dimensional texture map.

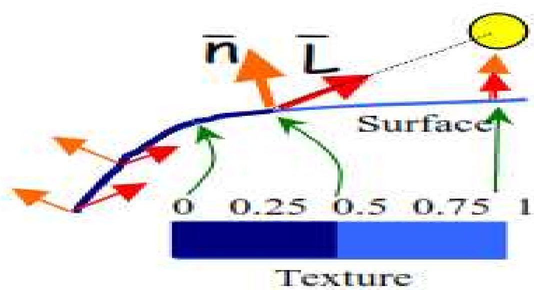

Figure 1 Texture coordinates generated from L.N, The dark color happened L.N threshold equal 0.5 [24]

Current research in Non-Photorealistic Rendering is directed towards improving interactive Toon-Shading [15, 23, 26, 27, 28] to accomplish a more cartoonish like rendering. The cartoon highlight shader in [15] lets the user modify directly through click-and-drag on the dark areas and highlighting areas in real-time on a surface to draw and animate them. Anjyo *et al*. [15] depends on the Lambertian shading rule to extract new light vector for the modified shaded area. Figure 2 explains the algorithm for modifying the shading area. They assumed the light source is a point or directional light.

The multi-scale shading technique by [26] can also control the shape appearance detail by tuning parameters of the lighting model. Shimotori *et al*. [27] proposed an interactive way to directly alter shade by regulating the normal vectors on the triangles by intuitive mouse operations. The idea of altering the normal vectors assists the rendering of other frames as the normal vectors are not influenced by the camera movement. In case of editing concept, these editing systems are intuitive, providing animations several editing methods.

The work in [26] presented a novel way by modifying the lambertain shading to control the shade area for each object in the scene without affecting other objects.

A new lighted framework is presented for cartoon shadier by [29]. They have introduced a novel artistic lighting system to create 3D digital animation. The invented shader in this study concentrates on effects of stylized lighting including edge lighting, straight lighting and detail lighting that are considered as significant features in cartoon animation and very complex to accomplish by traditional methods. Figure 3 shows typical edge lighting effects created by the Edge Toon Mapping method.

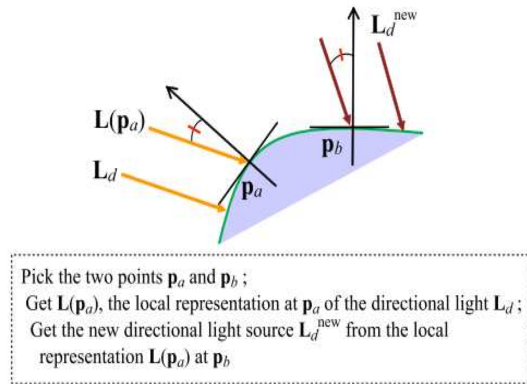

Figure 2 Tweaking the shaded area. Transferring the diffuse intensity from Pa to Pb means to change the directional light [15]

Lake *et al*. [24] proposed a proficient real-time rendering system contains a multiple algorithms which imitate a variety of cartoon styles. These real-time methods comprise a technique that can treat with highlighting of cartoon. Since this technique is fundamentally depends on a local shading model, it's hard to manage highlight shape. Other traditional approaches to create highlights would be of employing with projected textures, light maps, or virtual light sources.

Shading may differ based on different constructions of surface distance, character expressions, lighting, and action timing to explain storytelling or draw attention to a part of an object. Vanderhaeghe *et al*. [28] presented a technique which imitates a significant Non-photorealistic shading technique in dynamic 3D scenes, and that presented an easy and flexible means for stylized to draw and manipulate the shading appearance and its dynamic attitude. An another method for stylized shading is to Exploit an advantage of inverse rendering methods that permit illumination editing and direct material.

The Illumination Brush of [30] allows such control: beginning from a familiar BRDF, they deduce the environment lighting by painting on the object directly the wanted result of either diffuse or specular reflections.

The Exaggerated shading method of [31] locally supports light directions with controlling surface angles so that details are exposed through differences of a Half-Lambertian shader. The Apparent Relief technique [32] utilized the added dimension of the X-Toon shader to express shape features calculated through a combination of image-space and object measurements. The method has an ability to portray shape features at different scales, but displays artifacts with far objects due to the mixture of surface measurements.

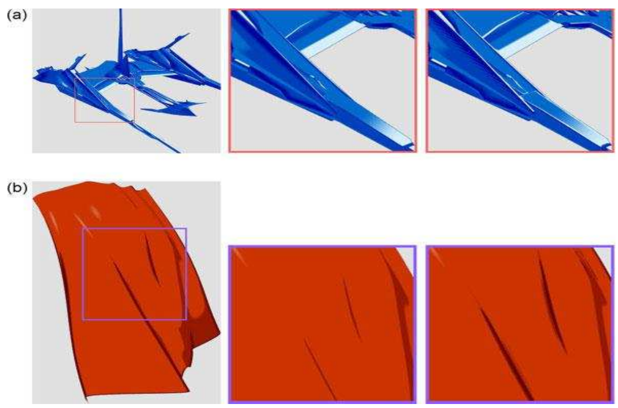

Figure 3 Comparison between conventional lighting result with stylized light result for Todo's system. The left image is Todo's system. The middle is close-up to the traditional light system and the right is a close-up of the Todo's Result System [29]

3.0 STYLIZED LINE DRAWING

This section describes the Line drawings of 3D shapes. The simplicity of Line drawings makes it commonly used in many computer graphics fields such as animations, sketches and technical illustrations. However, the existed line drawings techniques cannot be match the human artist expressiveness. In addition to the challenges that make the current techniques usefulness in interactive applications, which considered a vital and indispensible area in computer graphics.

Line Drawing term is referring to any type of drawing focused on the outlines of shapes. Despite the huge number of line drawing methods, the researchers can be divided Line drawing techniques as 3D model or image based methods. The researchers also classified the 3D model into three categories: Image Space Methods, Object Space Methods and Hybrid Methods, as shown in Figure 4.This section focuses on 3D Model approach and discusses the three subcategories in detail [61].

Here we are going to focus on the direction 3D Model Based Line Drawings: detection lines algorithms. A huge number of approaches utilized to extract the borders for 3D models. Isenberg *et al*. [34] divided these techniques into three main categories: object space algorithms, image space algorithms and hybrid algorithms.

3.1 Image Space Methods

Image space method depends on rendering different techniques such as cosine-shaded model or depth map, then checking the image buffer to extract the operating lines only and supply a silhouette represented as features in a pixel matrix.

Saito *et al*. [20] employed the z-buffer algorithms for detecting the silhouettes such as Sobel edge algorithm that has the propriety of finding object-relevant edge such as contours and silhouettes. Hertzmann *et al*. [35] expanded this approach by employing a normal buffer instead.

Some methods extract curves from the silhouette pixels such as introduced by Loviscach, who fit Bezier curves to the pixels [36]. Raskar *et al*. [37] introduced a new Non-Photorealistic Rendering camera that notice depth edges using multi-flash images. They employed the location of the shadows depth discontinuities as a robust cue to produce a depth edge map in both dynamic and static scenes

Lee *et al*. [38] introduced a novel method to extract lines automatically at suitable scales from abstract shading. Then they render lines along tone boundaries or thin dark areas in the shaded image. An efficient line drawing produced by this algorithm can effectively express both materials cues and shape.

3.2 Object Space Methods

At this type of artistic lines methods, all calculations are executed in object space, and provide the resulting lines represented by an analytic description which meet the requirement of applying further stylization to the lines.

Buchanan *et al*. [39] proposed a data structure named an edge buffer to boost the silhouette edge detection process. However, this method calculates the silhouettes by the brute-force method, so it required checking each edge whether has one adjacent back-facing and one adjacent front-facing. McGuire *et al*. [40] introduced GPU hardware methods for extracting silhouettes from 3D meshes, and explains how to employ hardware to produce thick contours demonstrated on the screen with end-caps that connect line segments with adjacent thick.

There is a lot of research on acceleration methods that attempt to reduce running time. Sander *et al*. [41] introduced a magnificent approach for silhouette edge detection which constructs a hierarchical search tree to store the mesh's edges. Another interesting acceleration approach encompasses the Gauss map that is utilized by [25, 39]. Benichou *et al*. [42] and Hertzmann *et al*. [43] presented a work where every mesh edge matches to an arc on the Gaussian sphere, which links the normal's projections of its two adjacent polygons. A very different sort of acceleration technique, most suitable to interactive systems, is proposed by [44].

Who introduce a stochastic algorithm to extract silhouettes? They examine that only a few edges in a polygonal model are actually silhouette edges.

A large amount of diverse definitions on object space lines demonstrated in the past decade, including suggestive contours [45, 46], ridges and valley lines [47, 48],suggestive highlights [49], photic extremum lines (PELs) [51] and demarcating curves [52]. Cole *et al*. [33, 53] have demonstrated that various families of lines should be merged to describe a large variety of shapes effectively. According to that, Grabli *et al*. [54] described an innovative data structure, the view map, which merges topology, geometry, and the rest of the properties of feature lines.

Laplacian-of-Gaussian (LoG) edge detector makes which proposed by Zhang *et al*. [55] discussed a new object-space line drawing approach that can represent the shapes with view dependent feature lines in real-time.

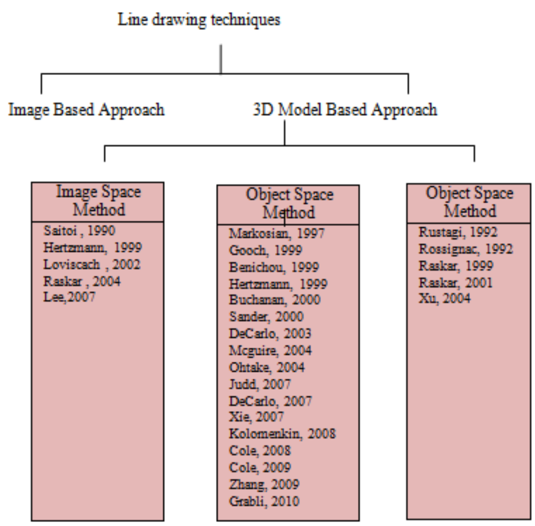

Figure 4 Taxonomy of stylized line drawing techniques

3.3 Hybrid Methods

Hybrid methods combine the image space and object space methods which execute the manipulations in object space and getting the lines in image buffer. Silhouettes in the result are represented in a pixel matrix similar to image space algorithms.

Hybrid algorithms are much less common than the above types of algorithms, and are particular for e.g., contours. Rustagirt al. [56] introduced a simple technique utilizing the stencil buffer that brings the contour only, without the fully silhouette of an object sucg as [50, 54]. Then Rossignac *et al*. [57] presented a technique depends on z-buffer rendering that not only contours, but also silhouettes.

Raskarr *et al*. [58] portrayed a hardware-accelerated technique of rendering silhouette edges. They employed conventional z-buffering along with front- or back-face culling to determine automatically front- and back-facing polygons, respectively. Later same author introduced a one-pass hardware implementation [59] which depends on appending borders around each triangle [39]. And he is utilizing environment mapping in addition to standard shading which achieves a stylistic effect.

Due to the shortage of connectivity information, most current polygon-based silhouette creation algorithms cannot be applied to point-based models. Xu *et al*. [60] introduced a new technique to create silhouettes not only avoiding this connectivity requirement, but also adopted point-based models with sparse non-uniform sampling and inaccurate/no normal information. They render the points as expanded opaque disks in the first pass to get a visibility mask, while as normal size splats/disks in the second pass. In this way, edges are described automatically at depth discontinuities, more often than not at the silhouette boundaries

3.0 RESULTS AND DISCUSSION

This study proposes a discussion of shape depiction through interactive Non-Photorealistic rendering. Non-Photorealistic rendering techniques distinguished over conventional computer graphics with two aspects: ability to introduce a vast body of styles in addition to provide computer-human interactions. In particular, this study has presented two groups of interactive Non-Photorealistic techniques for improving the shape depiction. The first group of strategies is based on shading approach, while the second group focuses on line-based rendering.

This paper gives an extended description of the toon shading [24]. The main contribution is exploiting graphics processing unit (GPU) to allow interactivity. Our algorithm was tested with an NVIDIA GtForce 525m display card and an Intel(R) Core(TM) 2 Duo CPU. Several models are rendered through our algorithm, are shown in Figures 5, 6, and 7, respectively. Figure 5.a shows the original teapot model, while figure 5.b illustrates the toon shading effect of our algorithm. Figure 6 and 7 illustrates more complicated models compared with teapot.

4.0 CONCLUSION

Rendering of a user-interactive cartoon animation with high frame-per-second in real-time is considered to be a crucial problem in Computer Graphics. A lot of factors should be overcome to implement an Interactive Non-Photorealistic rendering such as the necessity of adding some calculations during the rendering process. For example, the rendering of silhouette edges requires more calculations compared with extraction of silhouettes. Furthermore, achieving the interactivity at Non-Photorealistic rendering requires additional resolution to render the objects.

In this study we proposed an overview of the current interactive non-photorealistic rendering techniques which combines the shading effect and stylized line drawing. In addition, this paper discusses the effect of modified phong shading model in order to create toon shading appearance. This illumination model can be applied on a wide variety of 3D models with more ability of controlling the shading appearance over the surface of 3D model. This ensures better shape depiction and faster solutions to reach the desired results. However, the model we have presented is tailored to convey the material appearance as well as the depiction of complex details. The further limitation concerns to aliasing which occurred on the hard boundaries between colors.

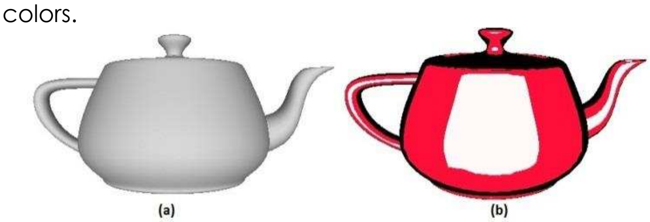

Figure 5 Visual comparison between: (a) original modal and (b) Toon shading model

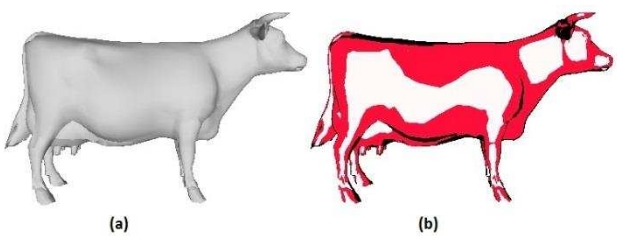

Figure 6 Visual comparison between: (a) original modal and (b) Toon shading model

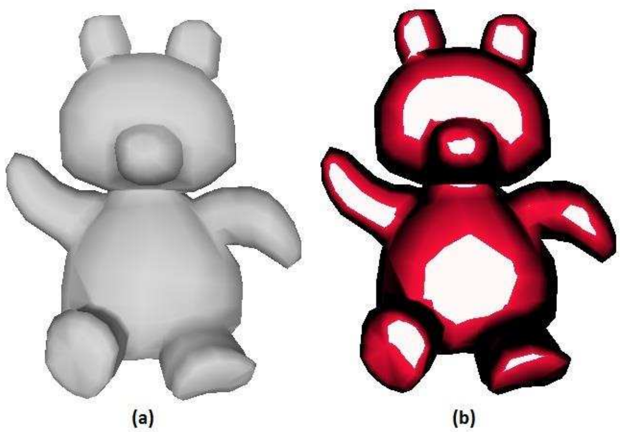

Figure 7 Visual comparison between: (a) original modal and (b) Toon shading model

Acknowledgement

This work was supported by the UTM MaGICX

References

[1] Y. Kawagishi, K. Hatsuyama, and K. Kondo. 2003. Cartoon Blur: Nonphotorealistic Motion Blu. *In Computer Graphics International, 2003. Proceedings*. 276-281.

[2] C. de Juan and B. Bodenheimer. 2004. Cartoon Textures. In *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. 267-276.

[3] R. Raskar, R. Ziegler, and T. Willwacher. 2006. Cartoon Dioramas in Motion. In *ACM SIGGRAPH 2006 Courses.* 6.

[4] N. Nasr and N. Higgett. 2002. Traditional Cartoon Style 3d Computer Animation. In *Eurographics UK Conference, 2002. Proceedings. The 20th*. 145-146.

[5] W. Van Haevre, F. Di Fiore, and F. Van Reeth. 2005. Uniting Cartoon Textures with Computer Assisted Animation. In Proceedings of the 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia. 245-253.

[6] J. Claes, F. Di Fiore, G. Vansichem, and F. Van Reeth. 2001. Fast 3D Cartoon Rendering with Improved Quality by Exploiting Graphics Hardware. *Proceedings of Image and Vision Computing New Zealand (IVCNZ)*. 13-18.

[7] B. Gooch, P.-P. J. Sloan, A. Gooch, P. Shirley, and R. Riesenfeld. 1999. Interactive Technical Illustration. In Proceedings of the 1999 Symposium on Interactive 3D graphics. 31-38.

[8] E. B. Lum and K.-L. Ma. 2002. Interactivity is the Key to Expressive Visualization. *ACM SIGGRAPH Computer Graphics.* 36: 5-9.

[9] E. B. Lum and K.-L. Ma. 2002. Hardware-accelerated Parallel Non-photorealistic Volume Rendering. In Proceedings of the 2nd International Symposium on Non-photorealistic Animation And Rendering. 67-ff.

[10] D. Sýkora, J. Dingliana, and S. Collins. 2009. As-rigid-as-Possible Image Registration for Hand-drawn Cartoon Animations. In Proceedings of the 7th International Symposium on Non-Photorealistic Animation and Rendering. 25-33.

[11] R. D. Kalnins, P. L. Davidson, L. Markosian, and A. Finkelstein. 2003. Coherent Stylized Silhouettes. In ACM Transactions on Graphics (TOG). 856-861.

[12] P. Neumann, T. Isenberg, and S. Carpendale. 2007. NPR Lenses: Interactive tools for Non-photorealistic Line drawings. In Smart Graphics. 10-22.

[13] D. Bandeira and M. Walter. 2009. Automatic sprite shading. In Games and Digital Entertainment (SBGAMES), 2009 VIII Brazilian Symposium on. 27-31.

[14] F. Thomas, O. Johnston, and F. Thomas. 1995. *The Illusion of Life: Disney Animation*. Hyperion New York,

[15] K.-i. Anjyo, S. Wemler, and W. Baxter. 2006. Tweakable Light and Shade for Cartoon Animation. In *Proceedings of the 4th International Symposium on Non-photorealistic Animation and Rendering*. 133-139.

[16] J. Suarez, F. Belhadj, and V. Boyer. 2010. Comics stylizations of 3D scenes using GPU. In Advances in Visual Computing, ed: Springer. 524-533.

[17] H. Winnemöller. 2013. NPR in the Wild. In Image and Video-Based Artistic Stylisation, ed: Springer. 353-374.

[18] S. B. Sadimon, M. S. Sunar, D. Mohamad, and H. Haron. 2010. Computer Generated Caricature: A Survey. In Cyberworlds (CW), 2010 International Conference on. 383-390.

[19] P. Decaudin. 1996. Cartoon-looking Rendering of 3D-scenes. *Syntim Project Inria.* 6.

[20] T. Saito and T. Takahashi. 1990. Comprehensible rendering of 3-D shapes. In ACM SIGGRAPH Computer Graphics. 197-206.

[21] B. Gooch and A. Gooch. 2001. Non-photorealistic Rendering. 201: AK Peters Wellesley.

[22] P.-P. J. Sloan, W. Martin, A. Gooch, and B. Gooch. 2001. The Lit Sphere: A Model for Capturing Npr Shading from Art. In Graphics Interface. 143-150.

[23] K.-i. Anjyo and K. Hiramitsu. 2003. Stylized Highlights for Cartoon Rendering and Animation. *Computer Graphics and Applications, IEEE*. 23: 54-61.

[24] A. Lake, C. Marshall, M. Harris, and M. Blackstein. 2000. Stylized Rendering Techniques for Scalable Real-time 3D Animation. *In Proceedings of the 1st international symposium on Non-photorealistic Animation And Rendering.* 13-20.

[25] B. Gooch, P.-P. J. Sloan, A. Gooch, P. Shirley, and R. Riesenfeld. 1999. Interactive Technical Illustration. In Proceedings of the 1999 Symposium on Interactive 3D graphics. 31-38.

[26] H. Todo, K.-i. Anjyo, W. Baxter, and T. Igarashi. 2007. Locally Controllable Stylized Shading. In *ACM Transactions on Graphics (TOG).* 17.

[27] Y. Shimotori, H. Nakajima, E. Sugisaki, A. Maejima, and S. Morishima. 2007. Interactive shade control for cartoon animation. In *ACM SIGGRAPH 2007 Posters*. 170.

[28] D. Vanderhaeghe, R. Vergne, P. Barla, and W. Baxter. 2011. Dynamic Stylized Shading Primitives. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Non-Photorealistic Animation and Rendering.* 99-104.

[29] H. Todo, K. Anjyo, and T. Igarashi. 2009. Stylized Lighting for Cartoon Shader. *Computer Animation and Virtual Worlds.* 20: 143-152.

[30] M. Okabe, G. Zeng, Y. Matsushita, T. Igarashi, L. Quan, and H.-Y. Shum. 2006. Single-view Relighting with Normal Map Painting. In *Proceedings of Pacific Graphics*. 27-34.

[31] P. Barla, J. Thollot, and L. Markosian. 2006. X-toon: An Extended Toon Shader. In *Proceedings of the 4th International Symposium on Non-photorealistic Animation and Rendering*. 127-132.

[32] R. Vergne, P. Barla, X. Granier, and C. Schlick. 2008. Apparent relief: a shape descriptor for stylized shading. In Proceedings of the 6th International Symposium on Non-photorealistic Animation and Rendering. 23-29.

[33] F. Cole, K. Sanik, D. DeCarlo, A. Finkelstein, T. Funkhouser, S. Rusinkiewicz, *et al.* 2009. How Well Do Line Drawings Depict Shape? *ACM Transactions on Graphics-TOG*. 28: 28.

[34] T. Isenberg, B. Freudenberg, N. Halper, S. Schlechtweg, and T. Strothotte. 2003. A Developer's Guide to Silhouette Algorithms for Polygonal Models. Computer Graphics and Applications. *IEEE.* 23: 28-37.

[35] A. Hertzmann. 1999. Introduction to 3D Non-photorealistic Rendering: Silhouettes and Outlines. Non-Photorealistic Rendering. *SIGGRAPH*. 99.

[36] J. Loviscach. 2002. Rendering Artistic Line Drawings using Off-the-Shelf 3D Software. In *Eurographics 2002*. 125-130.

[37] R. Raskar, K.-H. Tan, R. Feris, J. Yu, and M. Turk. 2004. Non-photorealistic Camera: Depth Edge Detection and Stylized Rendering Using Multi-flash Imaging. In *ACM Transactions on Graphics (TOG).* 679-688.

[38] Y. Lee, L. Markosian, S. Lee, and J. F. Hughes. 2007. Line drawings via abstracted shading. In *ACM Transactions on Graphics (TOG)*. 18.

[39] H. Kolivand, M. S. Sunar, K. S. Ray, B. K. Ray, I. K. Maitra, S. Nag, *et al*. 2011. To Combine Silhouette Detection and Stencil Buffer for Generating Real-Time Shadow. *International Journal of Computer Graphics*. 2: 1-8.

[40] J. W. Buchanan and M. C. Sousa. 2000. The edge buffer: a data structure for easy silhouette rendering. In *Proceedings of the 1st International Symposium on Non-photorealistic Animation and Rendering*. 39-42.

[41] M. McGuire and J. F. Hughes. 2004. Hardware-determined Feature Edges. In *Proceedings of the 3rd International Symposium on Non-photorealistic Animation and Rendering*. 35-47.

[42] P. V. Sander, X. Gu, S. J. Gortler, H. Hoppe, and J. Snyder. 2000. Silhouette Clipping. In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques. 327-334.

[43] F. Benichou and G. Eiber. 1999. Output Sensitive Extraction of Silhouettes from Polygonal Geometry. In Computer Graphics and Applications, 1999. Proceedings. Seventh Pacific Conference on. 60-69.

[44] A. Hertzmann and D. Zorin. 2000. Illustrating Smooth Surfaces. In Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques. 517-526.

[45] L. Markosian, M. A. Kowalski, D. Goldstein, S. J. Trychin, J. F. Hughes, and L. D. Bourdev. 1997. Real-time Nonphotorealistic Rendering. In Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques. 415-420.

[46] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella. 2003. Suggestive Contours for Conveying Shape. In *ACM Transactions on Graphics (TOG)*. 848-855.

[47] Y. Ohtake, A. Belyaev, and H.-P. Seidel. 2004. Ridge-valley Lines on Meshes via Implicit Surface Fitting. In *ACM Transactions on Graphics (TOG)*. 609-612.

[48] T. Judd, F. Durand, and E. Adelson. 2007. Apparent Ridges for Line Drawing. *ACM Transactions on Graphics (TOG).* 26: 19.

[49] D. DeCarlo and S. Rusinkiewicz. 2007. Highlight Lines for Conveying Shape. In *Proceedings of the 5th International Symposium on Non-Photorealistic Animation and Rendering*. 63-70.

[50] H. Kolivand and M. S. Sunar. 2011. Shadow Mapping or Shadow Volume. *International Journal of New Computer Architectures and their Applications*. 1: 64-70.

[51] X. Xie, Y. He, F. Tian, H.-S. Sean, X. Gu, and H. Qin. 2007. An Effective Illustrative Visualization Framework Based on Photic Extremum Lines (PELs). *Visualization and Computer Graphics, IEEE Transactions on*. 13: 1328-1335.

[52] M. Kolomenkin, I. Shimshoni, and A. Tal. 2008. Demarcating Curves for Shape Illustration. *ACM Transactions on Graphics (TOG)*. 27: 157.

[53] F. Cole, A. Golovinskiy, A. Limpaecher, H. S. Barros, A. Finkelstein, T. Funkhouser, *et al*. 2008. Where Do People Draw Lines? In *ACM Transactions on Graphics (TOG).* 88.

[54] S. Grabli, E. Turquin, F. Durand, and F. X. Sillion. 2004. Programmable Style for NPR Line Drawing. In *Proceedings of the Fifteenth Eurographics Conference on Rendering Techniques.* 33-44.

[55] L. Zhang, Y. He, X. Xie, and W. Chen. 2009. Laplacian lines for real-time shape illustration. In *Proceedings of the 2009 Symposium on Interactive 3D Graphics and Games*. 129-136.

[56] P. Rustagi. 1989. Silhouette Line Display from Shaded Models. *Iris Universe*. 42-44.

[57] J. R. Rossignac and M. van Emmerik. 1992. Hidden Contours on a Frame-buffer. In *Proceedings of the Seventh Eurographics conference on Graphics Hardware*. 188-203.

[58] R. Raskar and M. Cohen. 1999. Image Precision Silhouette Edges. In *Proceedings of the 1999 Symposium on Interactive 3D Graphics*. 135-140.

[59] R. Raskar. 2001. Hardware Support for Non-photorealistic Rendering. In Proceedings of the ACM SIGGRAPH/EUROGRAPHICS Workshop on Graphics Hardware. 41-47.

[60] H. Xu, M. X. Nguyen, X. Yuan, and B. Chen. 2004. Interactive Silhouette Rendering for Point-based Models. In *Proceedings of the First Eurographics Conference on Point-Based Graphics*. 13-18.

[61] L. Li, Y. Zhou, C. Liu, Y. Xu, and J. Fu. 2012. State-of-the-Art Line Drawing Techniques. In Computer, Informatics, Cybernetics and Applications. ed: Springer. 1249-125.