Seventh Irish Workshop on Computer Graphics (2006)
Catherine Noonan (Editor)

A Survey of Stylisation in Computer Animation

John Dingliana

Interaction Simulation and Graphics Lab (ISG), Trinity College Dublin.

Abstract

Stylistic animation and rendering approaches intentionally produce output that diverges from physical reality in order to enhance the quality of graphical output. In this paper, we discuss the domain of stylistic animation, with particular emphasis on techniques for stylising dynamics and motion synthesis. We present a survey of previous work and introduce some of the major issues in stylised animation. We conclude with a discussion of open challenges and present some of our own current and ongoing work in the field.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Animation, I.3.7 [Computer Graphics]: Colour, shading, shadowing, and texture, I.3.7

1 Introduction

For years since the first computer generated images were created, realism has been the predominant, if not fleeting, goal of computer graphics. Most early research therefore has focused on synthesising output that is as close to reality as possible. An informal, but widely accepted, Turing test for most graphical systems has been to assess if the output generated is discernable by human viewers from reality recorded in the form of photographs or videos.

The goal of extreme reality, even in the present day world of hyper-optimised graphics technology is not only difficult but quite as often futile. For, no matter how mathematically complex our models of physics and light transport and no matter how intricately detailed our geometric models, in most cases they are still but approximations of the infinitely complex universe which they emulate.

On the other hand, persistent research and development of faster hardware and more efficient algorithms has brought us to the stage where synthesised output has indeed become indiscernible from recorded video and imagery for a number of specific applications. Largely, these are in the domain of offline rendering and animation, although advances in GPU optimised techniques bring us ever nearer to creating interactive output that is indiscernible from reality to the untrained human eye. Focus has shifted significantly to optimising computational efficiency and accuracy in simulation and animation.

Another recent trend in graphics research has been in the area of Non-photorealistic Animation and Rendering (NPAR), a classification of techniques that are driven, not by the goal of simulating physical laws of light transport and dynamics to extreme mathematical accuracy, but instead utilise strategies that optimise the aesthetics, emotional impact or expressiveness of an image or an animation. A significant amount of existing approaches attempt to use automated computational techniques to emulate methods used by human artists, who exploit their intuitive understanding of how we interpret visual cues, in order to enhance renderings and animation. However, there are also a number of original approaches that are intrinsically dependent on automated computational processes and not derived from traditional art and animation.

Our discussion in this paper will focus on stylistic and NPAR techniques for enhancing synthesised motion. Section 2 provides an overview of generic NPAR, including a brief discussion on the stylised rendering problem. However the rest of the paper will discuss animation issues, including rendering optimisation for animation in Section 3, visualising motion in Section 4, and stylistic distortions to

Animation in Section 5. Section 6 discusses the role of input constraints in stylised animation. Finally we discuss open challenges and our plans for future work in section 7.

2 NPAR

Although much work in early Computer Graphics research, which delivers less than photorealistic results due to limitations in computational power, could loosely be categorised as non-photorealistic, we distinguish these from NPAR techniques that deliberately deviate from physical reality in order to achieve a specific output objective. The original study of NPAR is largely derived from techniques long established by human artists, who enhance the effectiveness of images through various strategies including abstraction of subjects, removal of extraneous information and emphasis on key features. NPAR brings together diverse knowledge in art, science and human perception with the goal of increasing the effectiveness of images at communicating information or emotional impact.

Figure. 1: A scene from “A Scanner Darkly”, which employs abstraction of video data to generate NPR animation. © Warner Independent Pictures

2.1 Stylistic Rendering

There is a diverse range of work in simulating artistic still image rendering styles, from simple approaches such as toon-shading to highly detail models for simulating not only the techniques used by real-artists but the physical properties of the media used in real art. Although the rendering problem is not the major focus of this paper we will provide a brief overview of some of the most relevant techniques here.

Toon-shading [27][32][33] is a simple but widely popular rendering style, which emulates techniques used in cartoon cel-animation and comic books. Toon-shaded images are characterised by images with a limited colour palette usually consisting of bright accentuated colours and a prevalent use of silhouettes to represent significant boundaries. Toon-shaded applications including games are fairly widespread due to the simplicity in implementing the rendering style.

More complex models add details, such as key contours and feature-lines to enhance the understanding of shape depicted in the image [57].

Approaches for empirical and physically-based simulation of pigments and media include [58][59][60], which not only emulate the artistic process of choosing which features to draw, but also simulate the evolution, degradation and interaction of pigments with the medium.

Illustrative visualization is one particularly relevant application domain, where 3D volume or surface data must be rendered in a clear and meaningful manner such that it communicates shape, gradient and material characteristics efficiently. Data in many real-world applications, such as medical imaging, is visually too congested to render usefully, using traditional methods. There is a clear need to remove extraneous data and enhance task-specific features [19][49].

In the more generic interactive graphics domain, a notable piece of work is the system developed by Mohr and Gleicher [48] for intercepting, interpreting and transforming graphical calls from standard OpenGL applications and generating stylised output based on customizable styles. Another compelling application is using non-photorealistic strategies to augment regular computer rendering techniques with emphasizing cues similar to those used in NPR [44][45].

For a more detailed treatise of the rendering problem we direct the reader to [46] and [47].

3 NPR for Animation

Given the abundance of Non-photorealistic rendering techniques for still images, requiring little or no user intervention, a naïve notion would be that it should be a simple extension to use these for generating full animations that are non-photorealistically rendered. Unfortunately, this is usually not the case in practice due to temporal inconsistencies between images generated by many automatic NPR techniques. Although individual still frames may look reasonably robust, lack of coherence in the strokes (or other graphical primitives) used to render each frame can lead to flickering, an effect referred to by some stop motion animators as “boiling”. There are some approaches that address the frame-to-frame coherency issue [38][39][40][41][42] but they are not without computational cost, or at the expense of image complexity.

In addition, when we attempt to bring NPR to the domain of interactive applications, the problem is exacerbated. With processing time at a premium, we not only have to deal with faster approaches to rendering each stylised frame but we need to address the coherency problem on a much tighter time schedule using wholly automatic techniques. Because of this, most current real-time NPR applications are confined to relatively basic toon-shading or use an ad-hoc method of applying static textures that are pre-rendered with the relevant NPR style. However, relatively simple rendering can be used to great effect with proper input and this is most evident in image-based video abstraction techniques for Stylistic Animation [43].

4 Rendering Motion

A common illustrative technique in still image rendering is to use visual cues to represent the motion and incipient behaviour of objects. This is achieved in traditional art by motion lines and dynamic poses. In computer generated imagery, we can reproduce some of these effects and extend them with additional techniques, specifically designed for and facilitated by digital media.

4.1 Motion Blur

Motion blur is an effect caused in photography by the rapid movement of an object during the time the camera shutter remains open to capture an image. Due to the discrete nature of rendering in computer graphics this artefact of photographic media does not automatically manifest itself in computer imagery. However, the influence of analogue media such as film and video is so widespread that it has become natural to simulate even the defects in the technology. There is an argument for suggesting that simulating motion blur also has beneficial effects in computer animation as it hides the detrimental effects of discrete timesteps on fast moving virtual objects and indeed motion blur as a post-process is a desirable effect to make synthesised motion seem more natural.

Figure 2: Simulated motion blur and ghosting in live video © Collomosse et al

Potmesil and Chakravarty [28] are amongst the first to present a formal method for simulating motion blur in computer imagery. Korein and Badler [30] address the issue specifically for animation, referring to the motion blur as temporal anti-aliasing. Although hardware limitations at the time prevented real-time realisation of their technique, similar strategies are used for many current implementations of interactive motion blur. In more recent work Brostow and Essa use an automated image-based technique to apply motion blur as a post-process to stop-motion animation [29].

4.2 Motion Cues

Hsu et al [11] depict fast motion in a still image using speed lines, which, in still sketches, is analogous to motion-blur in film. However, unlike motion blur related techniques to the problem, which largely generate raster data, their approach operates on 2D vector data, generating Catmull-Rom splines interpolating the movement of key features of the object as it moves from one configuration to the next. Masuch et al [10] and Lake et al [12] address the issue for stroke rendered 3D animation, presenting a method for depicting motion by redrawing a reduced outline and feature contours of transformed previous states of an object. Haller et al [13] extend this by combining speed lines, multiple image motion blur and squash and stretch to depict motion for computer games.

Collomosse et al [14] present a system for applying motion cues such as blur and ghosting, squash and stretch and speed lines to video data using a combination of computer vision and graphics techniques.

4.3 Flow Visualisation

Another application of representing motion by rendered output is in flow visualisation, where dynamic properties are visualised using geometry and colour. A well-known approach by Cabral and Leedom, involves visualising a vector field using linear and curvilinear filtering to blur textures along the vector field using Line Integral Convolution [16]. The technique is best suited to 2D flow and flow along surfaces of 3D objects. Interante and Grosch describe strategies to generalise this approach in [17] and [18], using a masking approach to remove uninteresting areas of the flow effectively generating a stream-lined texture effect to the flow volume. In addition, they apply chromatic variations and depth cues to enhance the expressiveness of the flow rendering. Ebert and Rheingans apply boundary and silhouette enhancement and tone shading to emphasize characteristics of flows visualised using regular rectangular grids [19]. Svakhine et al [15] extend this to tetrahedral meshes using pre-integrated GPU-accelerated ray casting. In their system illustrative techniques such as gradient enhancement, silhouetting and curvature enhancement are used to emphasize and communicate dynamic properties in visual output. Although these techniques have been developed for a very specific application domain and are largely non-interactive, similar methods could be derived to enhance more generic animation. For instance Cabral and Leedom demonstrate an application of their method to generate fake motion blur in images as a post process.

5 Stylised Animation and Simulation

O'Sullivan et al [1] categorise distortions in Physically Based Animation as being either *aesthetic* or *unavoidable*. Aesthetic distortion is a term used to describe errors or inaccuracies that are deliberately introduced to generate simulations that achieve a certain goal state or sequence of events. Unavoidable distortions, on the other hand, are forced into a system due to lack of precision due to faults in the computational models or because the system has to fall back to an imprecise approximated simulation process in order to achieve a target frame rate [3]. Many distortions are imperceptible to viewers, perhaps because we have an imperfect notion of dynamics events of high dimensionality.

5.1 Aesthetic distortions in Simulation

Barzel et al [2] are amongst the first to suggest that complete accuracy, the ubiquitous goal in physically based animation, is not necessarily a requirement. They suggest that there is a range of distortion in dynamic events that is imperceptible to the viewer, and that this uncertainty could be exploited in order to save on processing effort and also provide flexibility and controllability for animators. Furthermore, they point out that mathematical exactness in simulation is an improper goal, where the models used for simulation are simplified but geometrically perfect representations of objects, which in the real world have fine microscopic imperfections that lead to small random deviations in their behaviours.

Chenney and Forsyth [4] show how these observations can be taken a step further to deliver plausible simulations that satisfy goal states set by an animator. They use a simulation model which incorporates plausible sources of uncertainty, such as the texture of a rough table that is modelled geometrically as a perfect flat plane. To ensure plausibility they use a Markov chain Monte Carlo algorithm to sample a number of simulations that achieve the target goal state, applying a probability score to each simulation run and allow the user to pick the most appropriate one. This is a further improvement over previous simulation editing problems such as by Popovic et al [5], which calculates mathematically consistent simulation sequences that lead to a goal state specified by the animator.

All of the aforementioned methods generally deal with regular rigid body simulation, using conventional methods for integrating the trajectories of objects and only introducing uncertainty at discontinuities in the simulation process at the point of collision. A more general form of stylisation can be obtained by perturbing not only the motion paths and temporal characteristics of animation, but also the elastic nature of objects similar to the methods used by traditional cartoon animators [7].

5.2 Traditional Animation Strategies

Lasseter [22] discusses how traditional principles, developed to make animation more realistic and entertaining can be applied to computer animation. He describes how certain mechanisms, employed by Disney artists, involving shape deformation, trajectory distortion and re-timing can be applied to computer rendered offline animation.

Amongst these, a commonly used technique is *squash-and-stretch*, which applies exaggerated deformation to moving objects to accentuate their movements and visually convey mass and rigidity. This is explored in more detail by Chenney et al [23], who provide a mixed dynamics and kinematics model for applying cartoon-style squash and stretch in real-time. A number of previous approaches use different kinds of deformable modelling strategies to achieve the same effect [25][7][35][36][37] but are largely designed for non-real-time application or for relatively simple scenes. Furthermore most of these are purely for cosmetic effect, and only affect the visual appearance of the object without changing its behaviour. Chenney et al's approach not only generates real-time deformations for multiple simultaneous collisions, but the deformations are also utilised in generating the ensuing motion of the object.

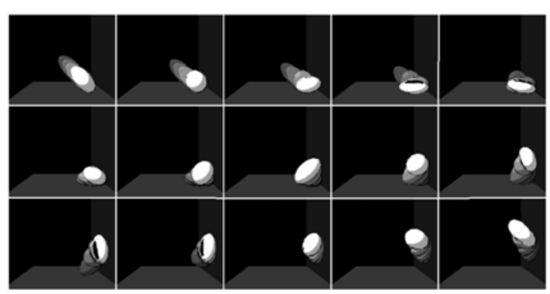

Figure 3: Squash and stretch simulated by Chenney et al © Stephen Chenney

Another common strategy used in traditional animation is to ensure that views of an object (characters in particular) are chosen so that the silhouette conveys an optimal amount of information about the object and its dynamic behaviour, sometimes necessary in toon-rendered animations which lack depth due to the simplified rendering style. With this objective, Rademacher [24] presents a method for applying view dependent geometric deformations to an object in order to optimise the object silhouette.

Wang et al [26] present a Cartoon Animation Filter, which takes an input motion sequence and retimes the motion to generate *anticipation and follow-through* in the animation. They combine this with squash-and-stretch deformations to generate cartoon motions from pre-generated input motion data.

5.3 Unavoidable distortions in Simulation

In order to achieve target interactive frame-rates in simulating complex scenes, many approaches adopt a strategy of simplification, trading-off complexity and resolution in order to deliver consistently smooth real-time output. The need for such strategies is advocated by O'Sullivan et al [1], who demonstrate that inconsistency in frame rate is a major set-back in animation and can affect users perceptions of dynamical properties of a scene. Simplification can be applied at the object level using multi-resolution models or we can apply simplified equations of motion in order to generate the animations. This type of simplification could be categorised as a form of stylisation which deliberately diverts from full accuracy in order to achieve a more acceptable output.

Hubbard [3] presents a strategy for optimising the collision detection part of the simulation pipeline using a refinable and time-critical approach. The system originally generates a very coarse approximation of contact, which can be iteratively refined until a computational time-limit is reached, at which stage we must fall back on the best completed solution so far. Dingliana and O'Sullivan extend this technique to achieve time-critical collision response through a similar iterative refinable method of modelling the contact manifold and thus the computed collision impulse [8][9]. Another approach, such as that by Carlson and Hodgins [31] is to use wholly different simulation models for generating motion depending on proximity to the viewer and hence importance in the scene. They refer to this as Simulation Levels of Detail (SLOD), using kinematic, dynamic or a combination of kinematics and dynamics as required for generating the motions of characters in the scene.

Although derived and evolved out of necessity, in order to make simulation more manageable, SLOD can be seen as a form of motion stylisation and it is conceivable that SLOD techniques could be applied in a more deliberate way, not for performance optimisation but as a stylistic means of abstraction and emphasis, in order to accentuate key motions and remove extraneous detail.

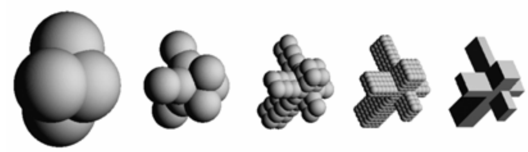

Figure 4: Simplification levels of a collision detection proxy used for SLOD © O'Sullivan et al

6 Input-Driven Stylisation

Although optimising the aesthetic or functional quality of output is the main objective in stylisation, there is much research that is also driven by input factors. Most animations, particularly in non-interactive applications are generated not by a random sequence of events, but in a much more deterministic manner with key events being scripted or configured by the animator. Such animations, which prevail in narrative media or movies, can be described as being goal driven – the primary objective is to reach a number of goal states through some sequence of events. Although, a plausible dynamic sequence leading up to these key states is desirable, in traditional animation, this intermediate motion is left to the skills and biases of the animator. This is essentially stylisation in its purest form, whether it be done on paper or in a computer animation system augmented with Inverse Kinematics solvers, to constrain the positions and orientations of different parts of an articulated character.

Another prevalent method is to use motion capture to control the movement of characters and there is evidence to suggest that this works reasonably well when the captured data is applied to simplified or stylised renderings [34]. There is a growing requirement for retrieval and reuse of data from motion capture databases in cases for which the specific motion was not explicitly captured. For instance, motion capture data might be applied to a character that was dynamically different, in size, weight or features from the captured subject. Alternatively, a motion capture of a simple walk might have to be reused for animating a character walking up a hill or uneven terrain; or the animator might wish to use a captured kicking motion to generate a slightly different kick. All of these are variations of motion re-targeting [50][51][52], which may involve rescaling, retiming or more extreme modifications to captured data, for instance blending motion capture with dynamic simulations. In all of these techniques, we are in essence applying stylisation to an essentially data-driven method for animation.

In an inverse problem, Forbes and Fiume [53] present a technique where lower dimensionality input is used to match and retrieve more detailed motion captured at an earlier time. Less Direct approaches involve specifying motion using sketches [54], textual commands [55] or voice [56]. Indeed, lower dimensional output can be combined with motion re-targeting to not only provide an intuitive interface for the animator to retrieve captured data, but simultaneously add stylised modifications to the motion.

7 Future Directions

Computer generated NPAR is certainly not a new field. There is a large body of previous work particularly in the field of rendering scenes emulating traditional art styles in order to increase effectiveness in visualising shape and important features of virtual objects. There is a smaller number of published works in stylised animation, although several well-known techniques in traditional animation have been implemented for both off-line and real-time animation. We believe that there is significant scope for further work in several areas.

Perceptual evaluation of computer graphics is a relatively new area rapidly gaining momentum in the graphics research community. While early computer graphics work was evaluated fairly rigorously on mathematical accuracy and computational efficiency, the visual element, if evaluated at all, was left to very simple *ad-hoc* comparisons with real images or output from previous related work. Recent studies have begun to examine techniques for using perceptual metrics to measure the quality of images and this has recently been extended to animation. Perceptual discourse is particularly relevant to NPAR, which proposes to introduce non-photorealistic distortions with the intent of increased effectiveness or improved aesthetics. Perceptual metrics for NPAR will not only allow us to evaluate how much increased effectiveness we can gain from a rendering or animation strategy, but they can also be used to drive the abstraction or simplification processes themselves, in a perceptually adaptive system.

There is very little work in developing enhancement methods that are intrinsically computational in nature, and not derived from previous traditional approaches. Drawing parallels in the rendering field, there is reason to believe that there is an opportunity to utilise the automatic nature of computer simulation in order to derive brand new techniques for enhancing animation and perception of virtual environments. In particular, simulation tasks that are, as yet, intractable for real-time animation might be solved by heuristic methods that don't simply strive to optimise the computational simulation problem but utilise stylistic approaches that are perceptually plausible, for instance, based on the SLOD type techniques discussed in section 5.

Interaction with the virtual environment through traditional interfaces or through haptic and force feedback devices, using stylised distortions to force and motion calculation, is another area that is relatively unexplored. Whilst stylised input is already widely used in generating offline animations, it may be feasible to extend such approaches to real-time interaction using lower dimensional intuitive input interfaces e.g. gestures, to drive complex animations.

Ongoing work at the Interaction, Simulation and Graphics Lab, will extend on previous studies which dealt primarily with unavoidable distortions using refinable dynamic simulation techniques [20][21][8][9] and strategies to evaluate the fidelity of physically based animations [1]. We have also studied some key factors that contribute to the perception of collision events. Further work in motion synthesis is planned, which will explore how various rendering fidelities, including non-photorealistic rendering of simulated scenes, affect a user's ability to perceive distortion anomalies in dynamic simulation. In addition, we introduce the notion of a functional distortion, which is a deliberate deviation introduced into dynamics in order to accentuate key behavioural features of an interactive scene. We will investigate how methods used in motion rendering and aesthetic distortion can be used to improve the feedback of a virtual scene, visually and also in the presence of other modalities of input such as haptic feedback and collision sounds. Finally, we will explore different methods of accelerating coherency preserving non-photorealistic rendering styles using GPU and LOD acceleration techniques, in order to provide real-time NPAR in various styles, as a viable visualisation method in interactive virtual environments.

References

[1] Carol O'Sullivan, John Dingliana. Thanh Giang and Mary K. Kaiser. "Evaluating the Visual Fidelity of Physically Based Animations". In ACM Transactions on Graphics. 22(3), Proceedings of SIGGRAPH 2003, July 2003.

[2] Ronan Barzel, John F. Hughes, and Daniel N. Wood. "Plausible Motion Simulation for Computer Graphics Animation". In *Computer Animation and Simulation '96*, pages 184–197, 1996. Proceedings of the

EurographicsWorkshop in Poitiers, France, August 31-September 1, 1996.

[3] Philip M. Hubbard. “Approximating polyhedra with spheres for time-critical collision detection” In ACM Transactions on Graphics. Volume 15 , Issue 3. July, 1996. Pages: 179 - 210. 1996.

[4] Stephen Chenney and D. A. Forsyth. Sampling “Solutions to Multi-body Constraint Problems”. In *Proceedings of SIGGRAPH 2000*. ACM SIGGRAPH, July 2000.

[5] Jovan Popovic, Steven Seitz, Michael Erdmann, Zoran Popovic, and Andrew Witkin. Interactive Manipulation of Rigid Body Simulations. In *SIGGRAPH 2000 Conference Proceedings*.ACM SIGGRAPH, July 2000.

[6] A.H. Barr “Global and local deformations of solid primitives.” In Computer Graphics (Proceedings of SIGGRAPH 84), ACM SIGGRAPH, pages: 21 - 30. 1984.

[7] Frank Thomas and Ollie Johnston. “Disney Animation - The Illusion of Life”. Abbeville Press, New York, 1981.

[8] John Dingliana. Carol O'Sullivan. “Graceful Degradation of Collision Handling in Physically Based Animation”. In Computer Graphics Forum. Vol 19(2000), Number 3, pp 239-247. 2000.

[9] John Dingliana, Carol O'Sullivan. “A Voxel Based Approach to Approximate Collision Handling”. In Journal of Graphics Tools. Volume 10. Number 3. December 2005.

[10] Maic Masuch, Stefan Schlechtweg and Ronny Schulz. “Speedlines - Depicting Motion in Motionless Pictures”. In SIGGRAPH'99 Conference Abstracts and Applications, S. 277, ACM SIGGRAPH, 1999.

[11] Siu Chi Hsu and Irene H. H. Lee. "Drawing and Animation using Skeletal Strokes". In Proceedings of ACM SIGGRAPH 94, pages 109-118. 1994.

[12] Adam Lake, Carl Marshall, Mark Harris and Marc Blackstein. “Stylized Rendering Techniques For Scalable Real-Time 3D Animation”. In Proceedings of NPAR '00: The 1st international symposium on Non-photorealistic Animation and Rendering, pages 13 – 20. 2000.

[13] Michael Haller, Christian Hanl, and Jeremiah Diephus. “Non-Photorealistic Rendering Techniques for Motion in Computer Games”. In ACM Computers in Entertainment, Vol. 2, No. 4, October 2004, pages 11- 24. 2004.

[14] John P. Collomosse, David Rowntree and Peter M. Hall “Cartoon-style Rendering of Motion from Video”. Proceedings of the 1st International Conference on Vision, Video and Graphics (VVG), pages 117-124. 2003.

[15] N. A. Svakhine, D. Ebert, E. Tejada, T. Ertl and K. Gaither. “Pre-integrated Flow Illustration for Tetrahedral Meshes” In Volume Graphics 2006.

[16] Brian Cabral and Casey Leedom. "Imaging Vector Fields Using Line Integral Convolution", Computer Graphics Proceedings, Annual Conference Series, 1993, pp. 263-269. 1993.

[17] Victoria Interrante and Chester Grosch. "Strategies for Effectively Visualizing 3D Flow with Volume LIC", proceedings of Visualization '97, pp. 421-424. 1997.

[18] Victoria Interrante and Chester Grosch. “Visualizing 3D Flow”. In IEEE Computer Graphics and Applications. Volume 18, Issue 4. July 1998. pp. 49 – 53. 1998.

[19] David Ebert and Penny Rheingans. “Volume Illustration: Non-Photorealistic Rendering of Volume Models”. In *Proceedings of IEEE Visualization ’00* (October 2000, Salt Lake City, UT), IEEE Computer Society Press, pp. 195-202. 2000.

[20] Carol O'Sullivan and John Dingliana. “Real-Time Collision Detection and Response Using Sphere-Trees”. In Proceedings of the Spring Conference in Computer Graphics, Bratislava April 28-May 1st 1999 pp 83-92. 1999.

[21] Fabio Gannovelli. John Dingliana. Carol O'Sullivan. “BucketTree: Improving collision detection between deformable objects”. Spring Conference in Computer Graphics (SCCG2000), Bratislava, April 2000. pp 156-163. 2000.

[22] John Lasseter, "Principles of Traditional Animation Applied to 3D Computer Animation", Computer Graphics, pp. 35-44, 21:4, July 1987 (SIGGRAPH 87).

[23] Stephen Chenney, Mark Pingel, Rob Iverson, Marcin Szymanski. “Simulating cartoon style animation”. In Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering. Annecy, France.Pages: 133 – 138.

[25] D. Metaxas and D. Terzopoulos “Dynamic Deformation of Solid Primitives with Constraints”. Proc. of ACM SIGGRAPH 1992. pp 309 – 312. 1992.

[26] Jue Wang, Steven Drucker, Maneesh Agrawala and Michael Cohen. “The Cartoon Animation Filter”. ACM Trans. on Graphics (Proc. of SIGGAPH 2006), 2006.

[27] Ken Anjyo. K Hiramitsu. “Stylized highlights cartoon rendering and animation”. IEEE Computer Graphics Applications 23, 4, 54–61. 2003.

[28] Michael Potmesil and Indranil Chakravarty “Modeling Motion Blur in Computer-Generated Images”. In Proceedings of Siggraph 1983. Pages 389 – 399. 1983.

[29] Gabriel Brostow and Irfan Essa. “Image-based Motion Blur for Stop-Motion Animation”. In Proceedings of Siggraph 2001. Pages 561 – 566. 2001.

[30] Jonathan Korein and Norm Badler. “Temporal anti-aliasing in computer generated animation”. In Proceedings of Siggraph 1983. Pages 377 – 388. 1983.

[31] Deborah A. Carlson and Jessica K. Hodgins. “Simulation Levels of Detail for Real-time Animation”. In Proceedings of the conference on Graphics interface '97, p.1-8, May 1997, Kelowna, British Columbia, Canada

[32] A. Gooch, B. Gooch, P.S. Shirley and E. Cohen. “A non-photorealistic lighting model for automatic technical illustration”. In Proceedings of SIGGRAPH 98, Computer Graphics Proceedings, Annual Conference Series, 447–452. 1998

[33] Pascal Barla, Joëlle Thollot, Lee Markosian. “X-Toon: An extended toon shader”. In NPAR’06: International Symposium on Non-Photorealistic Animation and Rendering (NPAR). 2006

[34] Jessica K. Hodgins, James F. O'Brien, Jack Tumblin "Perception of Human Motion with Different Geometric Models". In IEEE Transaction on Visualization and Computer Graphics, Vol. 4, No. 4, Pages: 101 – 113. 1998.

[35] P. Faloutsos, M. Van De Panne and D. Terzopoulos. "Dynamic free-form deformations for animation synthesis." IEEE Transactions on Visualization and Computer Graphics 3, 3 (July - September), pp 201 - 214. 1997.

[36] A. Opalach and S. Maddock. "Disney effects using implicit surfaces". In Proceedings of the Fifth Eurographics Workshop on Animation and Simulation, Eurographics. 1996.

[37] Brian Wyvill. Animation and special effects. In Introduction to Implicit Surfaces, J. Bloomenthal, Ed. Morgan Kaufmann, ch. 8, pp 242-269. 1997.

[38] Ryota Hashimoto, Henry Johan, Tomoyuki Nishita, "Creating Various Styles of Animations Using Example-Based Filtering," cgi, p. 312, Computer Graphics International 2003 (CGI'03), 2003.

[39] T. Haga, H. Johan, and T. Nishita, "Animation Method for Pen-and-Ink Illustrations Using Stroke Coherency", Proceedings of CAD/Graphics 2001, 2001, pp. 333-343. 2001.

[40] A. Hertzmann and K. Perlin, "Painterly Rendering for Video and Interaction", Proceedings of NPAR 2000, pp. 7-12. 2000.

[41] M. Kaplan, B. Gooch, and E. Cohen, "Interactive Artistic Rendering", Proceedings of NPAR 2000. pp. 67-74. 2000.

[42] B. J. Meier, "Painterly Rendering for Animation", Proceedings of ACM SIGGRAPH 96, 1996, pp. 477-484. 1996.

[43] Holger Winnemoller, Sven C. Olsen, Bruce Gooch. "Real-time video abstraction". In ACM Transactions on Graphics. 25(3). Proceedings of Siggraph 2006. pp 1221 – 1226. 2006.

[44] Szymon Rusinkiewicz, Michael Burns, and Doug DeCarlo. "Exaggerated Shading for Depicting Shape and Detail," In ACM Trans. Graphics (Proc. SIGGRAPH), Vol. 25, No. 3, 2006.

[45] Thomas Luft, Carsten Colditz, Oliver Deussen "Image Enhancement by Unsharp Masking the Depth Buffer". In ACM Trans. Graphics (Proc. SIGGRAPH), Vol. 25, No. 3, pp 1206-1213. 2006.

[46] Thomas Strothotte and Stefan Schlechtweg. "Non-Photorealistic Computer Graphics. Modelling, Rendering, and Animation" Morgan Kaufmann. 2002.

[47] B. Gooch and A. Gooch. "Non-Photorealistic Rendering". A.K. Peters. 2001.

[48] Alex Mohr and Michael Gleicher. "HijackGL: reconstructing from streams for stylized rendering". In NPAR '02: Proceedings of the 2nd international symposium on Non-photorealistic animation and rendering. 2002.

[49] Leif P. Kobbelt, Mario Botsch, Ulrich Schwanecke and Hans-Peter Seidel. "Feature sensitive surface extraction from volume data". In Proceedings of Siggraph 2001. Pages: 57 - 66 . 2001.

[50] M. Gleicher. "Retargetting Motion to New Characters". In Proceedings of Siggraph '98, p. 33-42, 1998.

[51] J.K. Hodgins. N.S. Pollard. "Adapting Simulated Behaviours for New Characters", ACM Siggraph 97, Computer Graphics Proceedings, p. 153-162, 1997.

[52] T. Komura, Y. Shinagawa., "Attaching Physiological Effects to Motion-Captured Data", Graphics Interface 2001, p. 27-36, 2001

[53] K. Forbes , E. Fiume, An efficient search algorithm for motion data using weighted PCA, Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation, July 29-31, 2005, Los Angeles, California

[54] Matthew Thorne, David Burke, Michiel van de Panne. "Motion doodles: an interface for sketching character motion" . Proceedings of the 2004 SIGGRAPH Conference. Pp 424 – 431. 2004.

[55] Okan Arikan, David A. Forsyth, James F. O'Brien. "Motion Synthesis from Annotations". ACM Transactions on Graphics (ACM SIGGRAPH 2003), Vol: 33, No: 3, pp 402--408, 2003.

[56] Zhijin Wang, Michiel van de Panne. " 'Walk to here': A Voice-Driven Animation System". In Proceedings of SCA 06: ACM SIGGRAPH and EUROGRAPHICS Symposium on Computer Animation. 2006.

[57] T. Saito and T. Takahashi. "Comprehensible Rendering of 3D Shapes." Proceedings of SIGGRAPH '90. pp 197-206. 1990

[58] M. C. Sousa and J.W. Buchanan. "Computer-generated graphite pencil rendering of 3D polygonal models". Computer Graphics Forum 18(3). P195-208. 1999.

[59] S. Takagi, M. Nakajima and I Fushiro. "volumetric Modelling of Coloured Pencil Drawing." In Pacific Graphics 1999.

[60] D. Small. "Simulating watercolor by modeling diffusion, pigment and paper fibres." In Proceedings SPIE'91. 1991