EG UK Theory and Practice of Computer Graphics (2006)
M. McDerby, L. Lever (Editors)

Expressive and Dynamic Deformation of Animated Computer-Generated Characters

Paul Noble and Wen Tang

School of Computing, University of Teesside, Middlesbrough, U.K.

Abstract

3D computer animated characters often lack the communicative power and the expressiveness of traditional animation, especially when used in non-photorealistic graphical representations. Inspired by a talk by Pixar Animation Studios at Animex 2005 [WR05] and based upon the animation and cartooning concepts of lines of action and motion, we present an expressive animation tool that emphasizes and accentuates the intentions of the animator derived from the skeleton-driven animation. Motion and pose-based shape deformations are automatically generated to quickly add an emotive layer to an animated character. This warping of the character geometry also provides a novel solution to the problem of implied motion in non-photorealistic still images.

Categories and Subject Descriptors (according to ACM CSS): I.3.7 [Computer Graphics]: Animation

1. Introduction

In the early 1930s, animators at the Disney Studios began studying their subjects for the first time in an attempt to create the 'illusion of life'. They pored over film reels, frame by frame, and analysed the anatomy of their characters; even going so far as building a jointed armature of a young deer for reference purposes when making Bambi [TJ81].

Traditional animators have, however, always had a rather flexible view of bone structure. An understanding of anatomy is crucial but of paramount importance are the fundamental principles of traditional animation [TJ81, Wil01], which maintain that bones are there to be squashed and stretched, and joints can be broken [Wil01] if it makes for a more appealing image or dynamic motion. As a result, the limbs of hand-drawn characters are often distorted to accentuate a motion or imply an emotion (see Figure 1).

These deformations encapsulate four of the eleven [Las87], or twelve [TJ81], fundamental principles of traditional animation, defined by Lasseter [Las87] as:

- Squash and Stretch – Defining the rigidity and mass of an object by distorting its shape during an action.
- Anticipation – The preparation for an action.
- Staging – Presenting an idea so that it is unmistakeably clear.
- Arc – The visual path of action for natural movement.

However, it is important to note that, with the exception of 'Squash and Stretch', limb-bending serves to merely accentuate any use of these principles rather than employing them directly.

Figure 1: *Examples of expressive deformation of a character's limbs. ©Disney*

The primary cause of these distortions is the use of what animators and cartoonists refer to as 'Motion Lines' and 'Action Lines' [LB78, Har94, Har97, Bla94, Whi86, BPE01]. Action Lines are "the basis for rhythm, simplicity, and directness in animation." [Bla94].

As can be seen in Figure 2, these two types of line are closely related and are typically drawn as smooth curves or arcs. The action line is often thought of as an extension of the spine and, in traditional animation, indicates the overall pose and direction of a character. Motion lines indicate "the direction of the most accentuated movement of the pose." [Har97]. When applied to the motion of humanoid characters, these motion lines usually define the motion of the limbs and, whether consciously or subconsciously, often lead to the distortion of these limbs in the final illustration (see Figures 1 and 2).

Figure 2: *Lines of action and motion. ©Christopher Hart (top) ©Marvel (bottom).*

Whereas an artist can intuitively add deformations to a hand-drawn character, the anatomy of most 3D computer-generated characters is more rigid and, rather than a loose sketch of a skeleton, animators work with virtual joints and bones. These skeletal structures incorporate transformation restrictions based on reality: joints have rotation limits, bones have fixed lengths, and the associated geometry of the character must follow these rules. In this respect, skeleton-driven computer animation struggles to compete with the flexibility and expressiveness of traditional hand-drawn animation.

In this paper, we present an animation tool that dynamically deforms the limbs of computer-generated characters based on the pose and motion of their virtual bones. Our aim is to enable computer animators to quickly add a layer of expressiveness to an animation and possible applications include use in the computer games industry or as a tool for creative animators.

2. Related Work

With non-photorealistic rendering (NPR) techniques maturing, interest in non-realistic and expressive computer graphics has increased in recent years as 3D computer animators look to the world of traditional animation for inspiration and understanding [SS02, NT04, CPIS02]. The techniques used in traditional animation are just as relevant to computer animation [Las87] but transferring these principles and practices is not always a straight forward matter.

One challenge facing many 3D computer animators attempting to produce cartoon-style animations is the fact that most hand-drawn characters simply cannot be modelled in 3D. Traditional animators distort their characters to maximize their aesthetic appeal depending on the direction from which they are being viewed. “Spatially, a hand-drawn character is often ‘cheated’ to emphasize traits of the face or body - to improve legibility of an expression, recognition of character (through silhouette features), and/or design (composition). The character's features are often drawn slightly out of proportion or off-perspective as a result.” [Van01]

A sense of three-dimensional solidity is crucial in 3D computer animation, which is often presented by a realistic geometric model. However, a single 3D model cannot encompass all of the possible distortions required. To avoid the need for a new model for each new viewpoint, it is possible to deform a single model depending on the current direction of view [Rad99]. Several deformed models, each linked to a different key viewpoint can be used to warp a base model. At each frame of an animation the base geometry is distorted by interpolating these key deformations to produce geometry unique to a particular viewpoint. This technique produces excellent expressive distortions of animated characters but the initial modelling work is highly labour-intensive and is also character-specific.

Tools can be provided to assist the animator with the task of creating the key deformations directly from drawings [LGXS03]. Again, this technique relies heavily on the animator creating the distortions and, furthermore, alters the underlying skeletal animation which may alter the original intensions and expressiveness of the initial animation.

Capturing the essence of traditional animation can also be taken literally [BLCD02]. The motion-capturing of cartoons allows the work of master animators to be retargeted to 3D models. The inherent drawback of this technique is that no new animation is actually created and there is a finite supply of suitable source material. The creation of a coherent piece of animation based upon the retargeted sequences of old films would be almost impossible.

Another problem encountered when trying to produce cartoon-style animations is how to imply motion in a non-photorealistic still image without using motion blur. Simulating motion blur in photorealistic animations is a relatively simple matter [PC83] but these techniques can rarely be used with Non-Photorealistic Rendering (NPR). Traditional animators and cartoonists use many visual cues and techniques to convey the motion of objects [Bla94, Whi86, BPE01]. Speed-lines, after-images, and jagged distortions have been applied successfully in computer graphics [KHK03, HL94, LMHB00, SPR94] but the deformation of the whole object can also imply motion. As clearly illustrated by the tennis racket in the third frame of Figure 1, the exaggerated distortion of the racket shows its velocity and implies the impact of its motion. This aspect of implied motion in traditional animation, which has been largely overlooked until now, is reflected in our expressive animation system.

3. Overview

The following sections describe a tool created for 3D animators that dynamically bends the limbs of computer-generated animated characters to create the appearance of a more stylized motion. Our algorithm first determines prospective 'motion-lines' based on the pre-existing animation and then accentuates this line by bending the geometry of the character's limbs. These deformations are applied to the geometry associated with the eight bones connected to the knees and elbows (for the sake of simplicity, the forearms and lower legs are judged to consist of one bone rather than two).

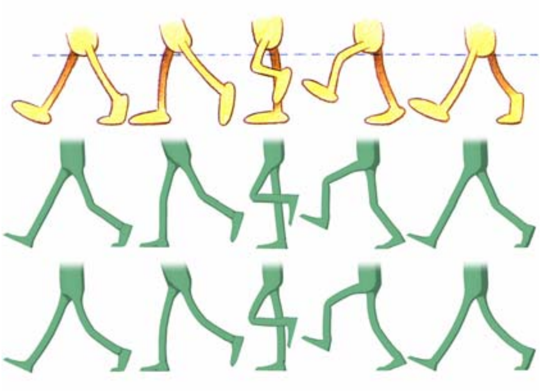

Figure 3: *Top - A traditionally animated walk cycle ©Tony White [Whi86], middle - reproduced using 3D Animation software, bottom – the same animation with limb-bending applied.*

The geometry of the character's limb is transformed using simple lattice deformers that act along the plane formed by the particular hinge joint and its two adjacent joints and perpendicular to the bone in question. For example, the bending of the lower leg would be in the plane formed by the hip, knee, and ankle joints and in a direction perpendicular to the shin. The deformation of each bone can, therefore, act in a concave or convex manner (see Figure 4). In this work we consider a concave bend to be one that bends in the opposite direction to the 'front' of the leg, which can be determined from the local rotation axis of originating joint of the bone; in the case of Figure 4, the knee.

The direction and magnitude of the deformation are dependent upon the aforementioned motion-line, which is itself defined by the existing animation or via several user-controlled attributes. Consequently, the tool's success depends upon the quality of the initial animation or the skill of the animator. A good piece of animation should possess inherent lines of action and motion and our task is to simply extract and help highlight them. The principle behind this work is not to attempt to replicate the creative process, but to act as an animation aid. We cannot hope to procedurally capture the skills used in the creation of a piece of animation, and nor do we hope to. This is, as always, the task of the artist, but we do aim to help emphasize and accentuate the animator's intentions.

The tool has been implemented in Autodesk's Maya software [May06] and utilizes this proprietary software's lattice and nonlinear bend deformer utilities.

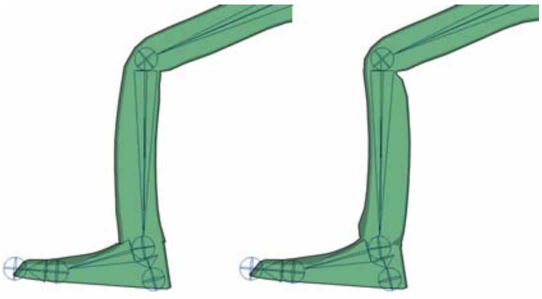

Figure 4: *Convex (left) and concave bending of the lower leg.*

4. Finding the lines of motion

In order to deform the limbs we must first find the lines of motion embedded within the animation of the character. The highly subjective nature of the limb-bending found in traditional animation makes any attempts to define it completely, or capture it algorithmically, bound to failure. Instead, we have determined empirically, that the limb deformations typically depend on two main factors:

- The bone's velocity
- The bone's associated child joints

4.1. Direction of Motion

The motion of the character is designed using a representation of a skeleton structure. The kinematic chains of the character are divided into five different parts representing the body and the limbs. The primary determining factor for the line of motion is the bone's direction of motion. This can be extracted from the animation by sampling the position of key joints in consecutive frames. Figure 5 shows three virtual joints connecting two bones, in two consecutive frames of animation, with j_0 stationary. When determining the bend of the bone joining j_1 and j_2, three vectors are required, and calculated by equation 1:

$$
\begin{aligned}
a &= \frac{j_{2,t} - j_{1,t}}{\left| j_{2,t} - j_{1,t} \right|} \\
b &= \frac{j_{2,t+1} - j_{2,t}}{\left| j_{2,t+1} - j_{2,t} \right|} \\
c &= \frac{j_{2,t+1} - j_{1,t}}{\left| j_{2,t+1} - j_{1,t} \right|}
\end{aligned}
\qquad (1)
$$

where t is the current frame of animation. The dot product of **a** and **b** gives the angle θ, and using simple linear algebra finding **c** indicating the precise location of $j_{2,t+1}$ relative to $j_{2,t}$, in the half-plane formed by j_0, j_1, and j_2.

If $j_{2,t+1}$ lies in either shaded quadrant as shown in Figure 5 then the bend is convex, otherwise it is concave. Hence, there should be minimal distortion if **a** is perpendicular or parallel to **b**.

$$
|a \bullet b| = \begin{cases} 0, or 1 & \text{Minimal distortion} \\ otherwise & \text{Convex or concave distortion} \end{cases} \qquad (2)
$$

Hence, |**b**| gives us the velocity of the joint which can be used in conjunction with θ to determine the magnitude of the bend. The calculation of the direction and magnitude of this bend are covered in section 5.3.

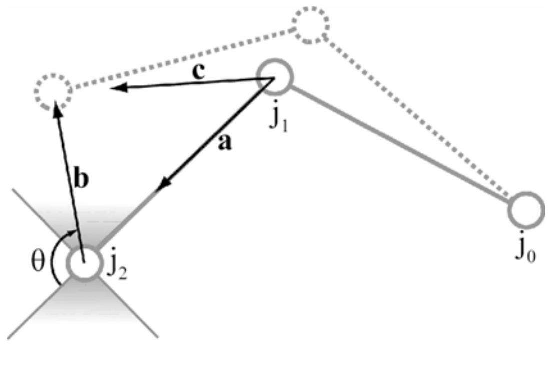

Figure 5: *Joint vectors. Solid lines indicate bone and joint locations at time t, dashed lines at time t+1.*

4.2. Secondary Deformation Influence of Associated Child Joints

The second influence on the line-of-motion we considered was the role of child joints. A child joint is defined to be one that is one step further down the skeletal hierarchy. For example, the ankle would be the child of the thigh bone. Child joints affect the bend of the parent bone because motion-lines are generally drawn as smooth curves and, as a consequence, any distortion that helps the overall shape of the limb approximate a curve is beneficial. For instance, when the foot is fully extended, we have observed that the lower leg is more likely to be drawn with a concave bend. The thigh is, more often than not, drawn with a convex bend because the knee is usually bent.

To mimic this effect in our system, we assigned each joint a threshold value. If the angle between a particular joint and its associated child joint exceeds this threshold, then the joint axes (shown in Figure 5) are skewed to increase the likelihood of a particular distortion. Whether this distortion is concave or convex will depend on the orientation of the hinge joint in question. Some joints rotate forwards, (for example, the ankle) and some rotate backwards (e.g. the knee). By decreasing the range of the shaded quadrants, the joint becomes more conducive to deformations that will help the limb approximate a curve to convey the expressiveness of the motion.

4.3. Hyperextensions

Hyperextension of the knee or elbow is undesirable in everyday life but is common in cartoon-style animations (see the four corner images in Figure 3). This ‘breaking of the joints’ [Wil01] occurs when the limb is fully extended and is treated in our system as a special case.

In order to hyperextend the joint we deform the limb as a whole rather than each individual bone. If the joint reaches its rotation limit then the two adjoining bones are treated as one. Any deformations of the individual bones are therefore omitted but this allows us to avoid moving the joint. Thus there is no alteration of the underlying animation which avoids the risk of losing some inherent animated information.

5. Deforming the limbs

The transformation of the geometry is dependent on the particular character-rig and the software being used. Most 3D animation software supports skeleton-driven deformation and a host of other tools to warp geometry. We have implemented our system in Maya and we utilize its lattice and nonlinear bend deformer utilities.

5.1. Lattice deformers

A lattice deformer allows the warping of an object or the components of an object. The lattice surrounds its associated geometry and any transformations applied to the lattice are propagated to the geometry. This allows us to manipulate complex geometry using relatively few control points. Furthermore, we can deform the lattice itself using other deformers. In our system we apply lattice deformers to the associated geometry of each bone and then manipulate the lattice using nonlinear bend deformers.

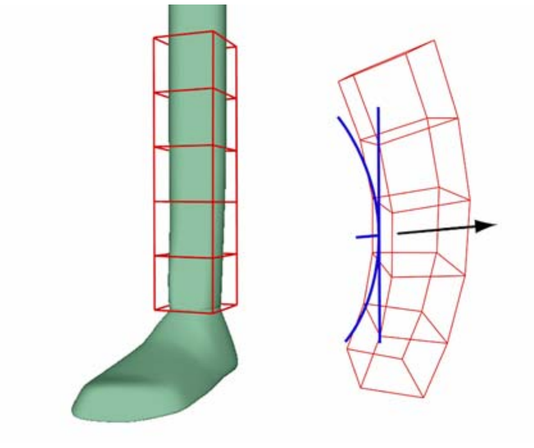

Figure 6: *Lattice and non-linear bend deformers.*

5.2. Bend deformers

Bend deformers warp geometry along a circular arc. They are controlled via a handle which defines the radius of the deforming arc (see Figure 6). In our system the orientation of the deformer is defined by the bone itself and the bend handle is controlled by a vector, **d**, the calculation of which is covered in the following section.

By default the deformer bends around its centre, so an expression is used to translate the lattice in the direction of **d** to ensure the geometry maintains its continuity at the joints.

5.3. Direction and magnitude of bend

As discussed in section 3, the direction of the bend is always perpendicular to the bone in question and in the plane of the triangle j_0, j_1, j_2 (see Figure 5). We can calculate this vector **d**, as:

$$\mathbf{d} = \mathbf{a} \times (\mathbf{a} \times \mathbf{b}) \tag{3}$$

The magnitude of the bend is dependent on user-defined parameters, b_{max}, v_{max}, and the value of θ. b_{max} is the maximum bend value and v_{max} is the threshold value of the velocity of the joint. As seen in section 4.1, the bend should be minimal if **a** is perpendicular or parallel to **b**, and at its maximum halfway between these angles. A simple trigonometric function is used to interpolate the bend between these points and define the magnitude of the bend vector **d** as Equation 4:

$$|\mathrm{d}| = b_{\max}\left(\frac{-\cos 4\theta + 1}{2}\right) \tag{4}$$

The dashed line in Figure 8 shows this interpolation curve with two peaks representing the maximum concave and convex distortions.

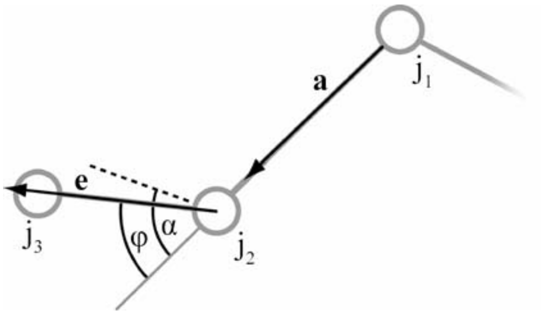

Figure 7: *Calculating the influence of child joints.*

5.4. Influence of the child joint

As described in section 4.2, the role of child joints in influencing the expressive deformation must be considered. To calculate the influence of these joints we first find φ, the angle between the two bones by the dot product of vectors **a** and **e**, where **e** is the vector between the joint j_2 and the child joint j_3 as shown in Figure 7. Each joint is assigned a threshold value α and, where φ is less than α, equation 4 is modified to take into the account of the secondary deformation influence as equation 5:

$$|\mathrm{d}| = b_{\max}\left(\frac{-\cos 4(\gamma) + 1}{2}\right) \tag{5}$$

where

$$\gamma = \begin{cases} \theta + (\alpha - \varphi), if\ \dfrac{\pi}{4} < (\theta \bmod \pi) < \dfrac{3\pi}{4} \\ \theta - (\alpha - \varphi), otherwise \end{cases}$$

This has the effect of increasing the area of the concave peak and decreasing the area of the convex peak in our interpolation curve (see solid line in Figure 8). As mentioned in section 4.2, the scaling of these curves is related to the orientation of the hinge joint and bone being considered. The influence of some child joints will cause the bones to be more inclined to convex, rather than concave bending.

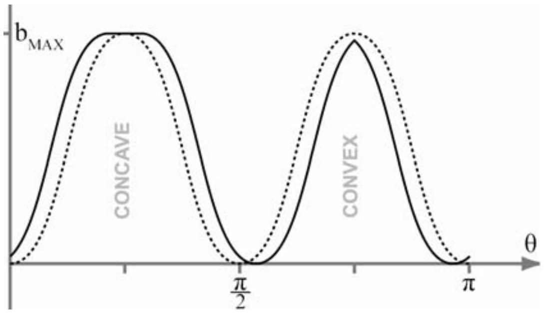

Figure 8: *Bend interpolation curves showing the standard curve (dashed) and the influence of child joints (solid line).*

5.5. Influence of velocity

As discussed, the primary factor on the deformation of limbs is motion, so the magnitude of the velocity of the joint is also influential. If the limb is stationary there should be no motion-based bending and the faster it moves the greater the deformation should be. We add a velocity threshold parameter, v_{max}, to equation 5 to incorporate the velocity effect as equation 6:

$$|\underline{d}| = \left(\frac{|\underline{b}|}{v_{\max}}\right) b_{\max}\left(\frac{-\cos 4(\gamma)+1}{2}\right) \qquad (6)$$

where $0 < |\underline{b}| < v_{\max}$

5.6. Smoothing

The calculation of the direction of the bend is based primarily on the existing animation. As a result, it is entirely possible that sudden changes in direction or pace between frames can result in a bone suddenly ‘popping’ from concave to convex in adjacent frames. Rather than complementing the animation and adding a layer of expressiveness, this bending would be a distraction.

To resolve this problem a simple smoothing algorithm is used. The type of bend for a particular bone is calculated for the current and subsequent frame and, if these differ, smoothing is applied. Concave bends are assigned a positive value and convex a negative. The sum of these values is applied to the bend at the current frame. Although this is a relatively crude solution, we have found that good character animations tend to possess a high level of continuity. This coherency of motion between frames renders the need for smoothing somewhat redundant.

6. Results

The system has been successfully implemented in Maya and the expressive shape deformations can be generated either automatically from the skeleton-driven animation or manually through a series of sliders. The maximum bend and the velocity threshold are user-defined parameters and Figure 9 shows differing degrees of bend generated for the same frame of an animation.

Figure 9: *Three variations of the same frame of animation using varying degrees of bend.*

7. Conclusions and future work

This paper has presented an animation tool that automatically adds expressive shape deformations to the limbs of animated characters. The direction and magnitude of the bending is derived from the originating animation. As such, these pose and motion-based deformations quickly add a layer of expressiveness to the 3D animation and help accentuate the underlying animation and highlight the animator’s intensions.

Figure 10: *An animated character running with expressive shape deformations applied.*

The primary area of future work is the extension of the tool to include what traditional animators refer to as the line of action. Closely related to lines of motion, they are often viewed as an extension of the spine of a character and so the extraction of this line from the animation and its analysis could greatly influence the bending of a character's limbs.

A relatively simple enhancement to the system would be the use of more complex bending functions to allow asymmetric bends or S-shaped deformations. This would enable adjoining bones to bend in opposite directions while still allowing the limb as a whole to follow a smooth curve.

A more sophisticated smoothing algorithm could also be easily developed and an investigation into bend interpolation functions could be carried out. The trigonometric function described in this paper yields acceptable results but the analysis of a variety of functions could be carried out to determine which, if any, generates the most visually pleasing results.

8. References

[Bla94] P. Blair, *Cartoon Animation*, Walter Foster Publishing, Laguna Hills, CA, USA, 1994.

[BLCD02] C. Bregler, L. Loeb, E. Chuang, and H. Deshpande, "Turning to the Masters: Motion Capturing Cartoons", P*roceedings of SIGGRAPH '02*, ACM Press, San Antonio, Texas, 2002, pp. 399-407.

[BPE01] B. Brooks, T. Pilcher, S. Edgell (editor), *The Complete Cartooning Course*, David & Charles Publishers, Newton Abbot, England, 2001.

[CPIS02] S. Chenney, M. Pingel, R. Iverson, and M. Szymanski "Simulating Cartoon Style Animation", *Proceedings of NPAR '02*, ACM Press, Annecy, France, 2002, pp. 133-138.

[Har94] C. Hart, *Everything You Ever Wanted To Know About Cartooning But Were Afraid To Draw*, Watson-Guptill Publications, New York, USA, 1994.

[Har97] C. Hart, *How to Draw Animation*, Watson-Guptill Publications, New York, USA, 1997.

[HL94] S. C. Hsu and I. H. H. Lee, "Drawing and animation using skeletal strokes", *Proceedings of SIGGRAPH '94*, ACM Press, New York, USA, 1994, pages 109–118.

[KHK03]Y. Kawagishi, K. Hatsuyama, K. Kondo, "Cartoon Blur: Non-Photorealistic Motion Blur", *Computer Graphics International 2003*, Tokyo, Japan, 2003, pp. 276-281.

[Las87] J. Lasseter, "Principles of Traditional Animation Applied to 3D Computer Animation", *Proceedings of SIGGRAPH '87*, ACM Press, Anaheim, July 1987, pp. 35-44.

[LB78] S. Lee, and J. Buscema. *How to Draw Comics the Marvel Way*. New York, USA: Simon and Schuster, 1978.

[LGXS03] Y. Li, M. Gleicher, Ying-Qing Xu, and Heung-Yeung Shum, "Stylizing Motion with Drawings", *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, Eurographics Association, San Diego, California, 2003, pp. 309-319.

[LMHB00] A. Lake, C. Marshall, M. Harris, and M. Blackstein, "Stylized rendering techniques for scalable real-time 3d animation", *Proceedings of NPAR 2000*, Annecy, France, 2000, pp. 13-20.

[May06] Maya, Autodesk, 2006.

[NT04] P. Noble, W. Tang, "Modelling and Animating Cartoon Hair with NURBS Surfaces", *Proceedings of Computer Graphics International '04*, IEEE Press, Crete, Greece, 2004, pp. 60-67.

[PC83] M. Potmesil, and I. Chakravarty, "Modeling motion blur in computer-generated images", *Proceedings of SIGGRAPH 1983*, ACM Press, Detroit, Michigan, USA, 1983, pp. 389-399.

[Rad99] P. Rademacher, "View-Dependent Geometry", *Proceedings of SIGGRAPH '99*, ACM Press, New York, USA, 1999, pp. 439-446.

[SPR94] T. Strothotte, B. Preim, A. Raab, J. Schumann, and D. R. Forsey. "How to render frames and influence people", *Eurographics '94*, Blackwell Publishers, Netherlands, 1994, pp. 455-466.

[SS02] T. Strothotte, and S. Schlechtweg. *Non-Photorealistic Computer Graphics - Modeling, Rendering, and Animation*. San Francisco, USA: Morgan Kaufmann, 2002.

[TJ81] F. Thomas, O. Johnston, *Disney Animation - The Illusion of Life*, Abbeville Press, New York, 1981.

[Van01] S. Vanzo, Interview: "3D Animation in Futurama", www.gotfuturama.com, 2001.

[Whi86] T. White. *The Animator's Workbook*. New York, USA, Oxford, England, Phaidon 1986.

[Wil01] Richard Williams. *The Animator's Survival Kit*. London, England: Faber and Faber Ltd., 2001.

[WR05] M. Walsh, R. Russ, Pixar Animation Studios presentation at Animex 2005 animation festival, University of Teesside, Middlesbrough, Jan 2005.