Observer Dependent Deformations In Illustration

D. Martín *
Dpto. de Lenguajes y S. I.

S. García †
Dpto. de Dibujo

J. C. Torres ‡
Dpto. de Lenguajes y S. I.

Universidad de Granada, 18071 Granada, Spain

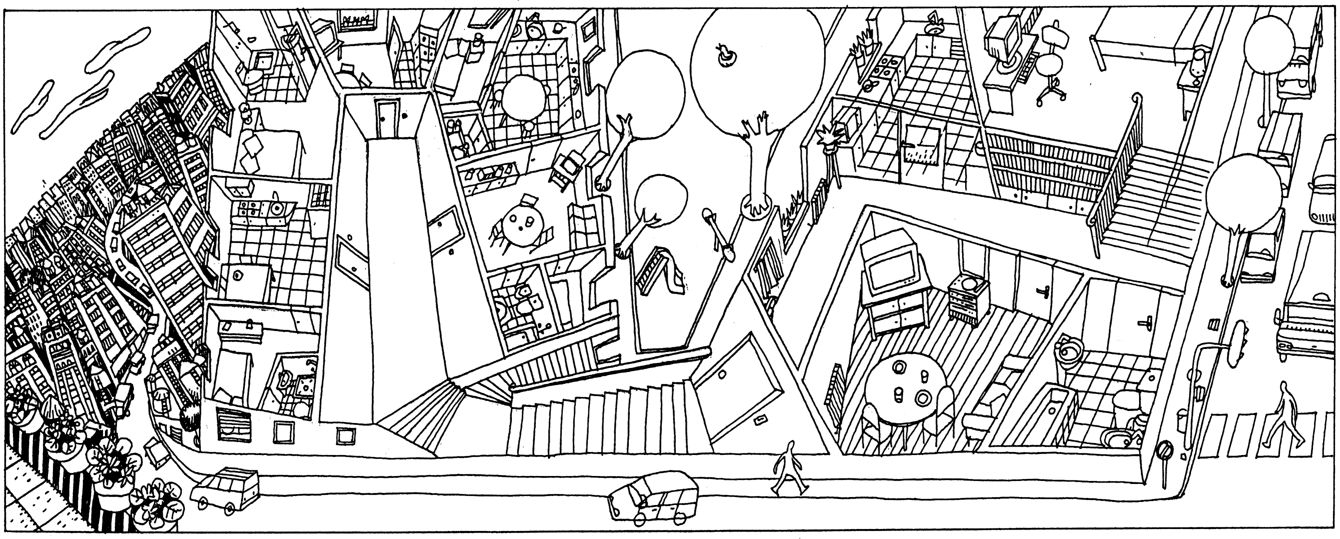

Figure 1: Example of a hand-made illustration

Abstract

Illustration has some visual characteristics that are very interesting yet also very difficult to obtain with a computer. Whilst the simulation of the tools used for painting and drawing is successful enough, expressive capabilities have not been developed to the same extent. The deformations of objects and space is a major element in the expressiveness of illustration. The use of Hierarchical Extended Non-Linear Transformations is presented as a powerful tool for obtaining such kind of expressivity.

Keywords: deformations, hierarchical extended non-linear transformations, illustration, animation, non photo realistic rendering

* dmartin@goliat.ugr.es
† sgarcias@platon.ugr.es
‡ jctorres@goliat.ugr.es

1 INTRODUCTION

Since the beginning of computer graphics there has been a great deal of interest in producing images that are highly realistic, in physical terms, which is called photo realistic rendering. The main goal is to obtain images that are equal to the real ones. Over the last few years we have seen that new methods have been developed for displaying images in a non-photo realistic manner. This kind of rendering is called non-photo realistic rendering or expressive visualisation. The main goal is to simulate alternative forms for representing visual information. The interest is in how the visual information is represented, in a plastic way, as well as what is represented. For example, we can use methods to transform models or images in a pictorial way, or we may be interested in reporting about the internal structure of a mechanical component.

So, we can see several methods that simulate the use of pencil, pen and ink, oil painting, and so on. These techniques have been used for obtaining images that are similar to the hand made ones, but they are done automatically. As in painting, we can observe that there is a similar evolutive process in computer visualisation. When photography appeared, the images that could be obtained with it were exact representations of the captured world. From then on, painters lost an objective; to represent as exactly as possible. Then, new methods and techniques for seeing and representing the world have been developed: cubism, expressionism, and so on. That is, techniques that could not be obtained with a machine. In the same sense, the expressive visualisation is appearing in computer graphics, once the realistic way is known enough. This is the stage in which we found ourselves at the moment. The computer techniques developed so far can simulate the use of painting and drawing tools

Figure 2: Example of a hand-made illustration

Figure 3: The same illustration but with a computer-made background

in a realistic way, in the sense that they may be considered to be hand made. So, we think that these methods and new ones that will be presented in the next years will be used instead of their real counterparts.

All this work is aimed at producing modifications in the visualisation of the elements included in the scene. That is, once the illustrator, animator or the computer have a model (mental or computer one), they can visualise it in several ways, changing its aesthetic characteristics. But the apparency can be also modified by changing the mode in which the model is seen, or by changing and deforming, the model itself (Figure 1). This approach adds another level of flexibility to visualisation. We talk about aesthetic and expressive capabilities so as to name both possibilities.

Though not as developed as the aesthetic part, there are some works aimed at obtaining the expressivity that can be found in classical animation and illustration. That is the goal of the paper: we present a method that allow us to obtain a high level of expressivity from 3D polygonal models, using the deformation of objects. The use of HENLT is presented as a tool for achieving the deformation of objects, in such a way that they can be used by plastic artists, specially illustrators.

2 RELATED WORKS

As we commented earlier, we may divide these works in aesthetic oriented and expressivity oriented ones, and also in 2D or 3D, and raster or vector based ones.

We are interested in the 3D expressivity and vector-oriented works, but the aesthetics is also important. The use of a 3D model, as shown in [3, 21], is necessary if there are camera movements, 3D characters, and so on. Using 3D models, there are papers such as [34, 28, 27, 30, 29, 14, 4, 7, 15] that concentrate on the aesthetic aspect, simulating the use of drawing and painting tools, using static models.

There are some works related to this one, most of them present methods for obtaining deformations, and how the latter can be used, specially in animation. On the other hand, we are interested in the use of deformations for obtaining the results needed by illustrators.

The development of expressivity is guided in animation by applying classical animation principles [32] to 3D models [18] with a 3D appearance, and [21, 22, 23], with a classical appearance. The expressivity is generally obtained by means of deformations of objects. Some methods have been developed to allow a certain degree of flexibility and modification with soft objects, for instance [33]. The best known representation methods are the Free Form Deformation [31], and the Non-Linear Transformation [1]. These methods have the advantage of being independent in terms of representation. FFD is commonly used because it allows local control of deformations. The use of control surfaces has been used before in the skeleton animation technique [2]. Litvinowickz [19] has also used FFD's in his $2\frac{1}{2}$ D animation system. Coquillart [5] presents a variation that allows the use of different control meshes, and in [6]

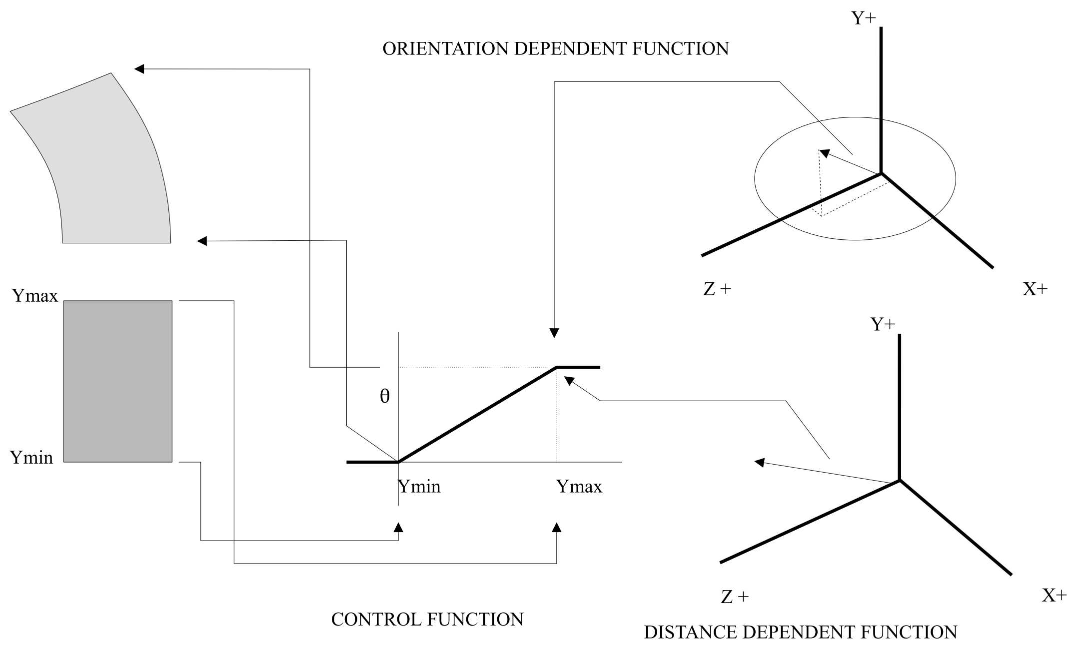

Figure 4: General scheme of orientation and distance dependent functions

the method is extended to develop an animation tool. Hsu [16] uses FFD's for obtaining sketches with arbitrary shapes in a 2D drawing system. Karan [17] uses control curves. Zorin [36] presents a method for the correction of the geometric perceptual distortion, which could be used just in the opposite direction, for expanding the distortions.

We have developed an extension of the Non-Linear Transformation as a way to produce deformations [22], the Hierarchical Extended Non-Linear Transformation.

Another possibility for obtaining the expressivity that is generally seen in illustration (Figure 1) is the use of multi-perspective. The work [35] presents a method for obtaining multi-perspective views of 3D models for use as backgrounds in animation, specially classical animation. Whilst the solution is good, it cannot be used in a general applications where there are many objects with their own form of being seen.

Rademacher [25] presents a method for deforming the geometry depending on the viewer. It is oriented to the production of images with a 2D appearance. This is achieved using a 3D model, from which key deformations are obtained matching the projection of the model and 2D images. The camera movements and the animation is achieve interpolating the key deformations (1D for camera movements and 2D for the animation of the character). The Whilst the goals are similar to our ones, the method is different. The differences will be commented when the method of HENLT's is explained.

3 AESTHETICS AND EXPRESSIVE CAPABILITIES IN ILLUSTRATION

We are interested in simulating the aesthetic and expressive capabilities found in illustration. The needs of the illustration are of two kinds: aesthetics and expressive. The aesthetics capabilities are represented by the use of tools such as pencil or pen and ink. With these tools the illustrator represents objects and characters by their shape lines and, normally, if necessary, flat colouring. Though the colouring is found in illustration, in most cases it is not as important as shape lines. Obtaining shape lines and colouring may be accomplished automatically, as is shown in [20, 14, 23]. As well as sketches and colouring, the use of textures is very important. Currently, our textures are hand made, which produces very good results.

The expressiveness is related to the skills that the artist has to represent his/her mental model on a piece of paper. Illustrators as well as animator have total flexibility in producing visual information. The use of time is important in both cases. Whilst the animator can transmit the information using the passing of time, the illustrator must condense the information in a few frames. This need for condensing information means that the illustrator develops and uses some plastic or expressive skills that are not so common in animation. For example, the use of multi-perspective views of the scene. This kind of expressive elements can be seen in the professional work by Sergio García [8, 9, 10, 11, 12, 13]. In Figures 1 and 2 there are elements that are projected in different ways. Thought there is an article relating to this kind of visualisation [35], its solution is based on adjusting several projections, and is oriented towards animation. In the case of the examples presented earlier, it is not possible to obtain the same results only with changes in the projection type, but it is also necessary to deform the objects. The solution proposed is based on the Alhambra system, using HENLT.

4 THE ALHAMBRA SYSTEM

The Alhambra system is the software developed based on the Alhambra model. It is oriented mainly towards animation. The main components are divided into visual components and animation components. The visual components are related to the capability for obtaining shape lines, which we call sketches, colouring, which can be flat, in two or more tones, and shading. There are two kinds of sketches: silhouettes or exterior sketches and interior sketches. Sketches are lines that limit some visual characteristics, more common, the limit between the visible and invisible parts of the object, that is, its silhouettes or external sketches, and the limit between

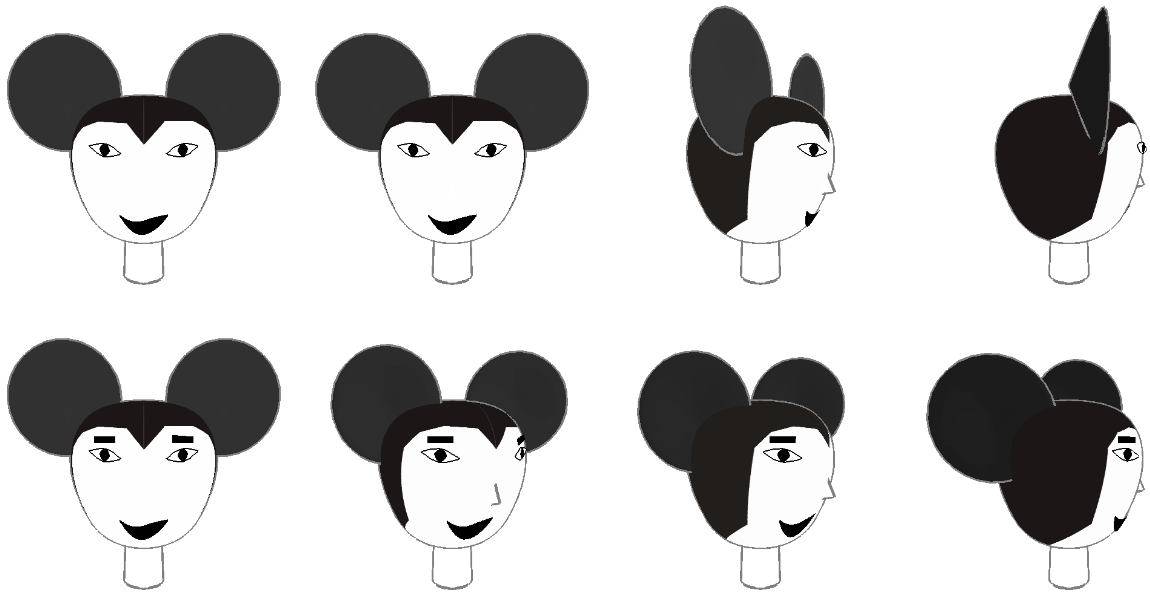

Figure 5: The model changes as the observer moves around it. See eye, mouth and ears

other visual attributes, normally a change of colour or shade, the internal sketches.

In order to obtain sketches, the *Virtual Lights* model is developed. It allows the designer to decide where, when and which geometric elements of the model will appear as sketches. Both types of sketches are treated in the same way with *Virtual Lights*. Although this model may be used with other types of geometric models, the implementation uses a 2-manifold polygonal model, based on triangles. In this case, the geometric elements are edges. Unlike other methods for extracting shape lines, which are raster based [26, 30], our method is 3D, which allows us more flexible control over the appearance of sketches, for example, changing the width, style, an so on.

The other component is the animation. With animation as main goal, the technique developed allows us to produce the deformations that are generally found in it. Though the commonly used method is Free Form Deformation, FFD, we have extended the Non-Linear Transformation method, obtaining the Hierarchical Extended Non-Linear Transformations, HENLT. It can be shown that FFD's are a subset of general HENLT's [24]. Whilst a FFD deformation is always defined by a 3D control function, a volume, the HENLT is defined using a 1D control function, which is simpler and most times may produce good results. It adds several new capabilities, including the use of any geometric transformation (rotations, scaling, translation, and so on), the use of an application axis as well as a selection axis, the combination of constant geometric transformations with HENLT, and the possibility of using hierarchies of Extended Non-Linear Transformations.

4.1 Hierarchical Extended Non-Linear Transformations

Hierarchical Extended Non-Linear Transformations [21, 22] are a variation of geometric transformations; translation, rotations, scale and so on. In non-linear transformations, the transformation itself is changed depending on the position. The function that relates position and transformation is called a *control function*. A geometric transformation can be seen as a non-linear transformation, for which the control function is constant (henceforth referred to as constant transformations). More formally, given a transformation, T, which is constant for all vertices, a point (x, y, z) is transformed into $(x', y', z') = T(x, y, z)$. A selection axis is chosen for every transformation. The selection axis points to which coordinate x, y, o z, will be the independent variable in the control function.

A *control function* is a function that defines how the parameters that control the deformation change. This function depends on the value of a coordinate, which itself depends in turn on the selection axis. In general, the control functions depend on other functions which produces the variation in time.

5 OBSERVER DEPENDENT DEFORMATIONS

The main goal of this work is to provide a system that allows the illustrator to produce backgrounds and characters in a easy way, with the capability of reproducing the extreme changes of form and perspective that can be found in illustration, and with the visual appearance of being hand-made.

As mentioned earlier, the HENLT is the technique used for obtaining the expressive elements. But whilst in animation, the deformations are used to give the illusion of life to objects, creating movements, in this case the deformations, have another goal. That is, the deformations are applied to the objects for obtaining new forms of being seen. The objects, their form, is dependent on the observer's position. For example, an object is seen as a right cube with a perspective projection from one position but as a curved cube with a parallel projection from other. There may be many objects with their respective deformations and projections in a scene. So, in general, its not possible to obtain these kind of images with only a fixed-object multi-perspective approximation; even applying multi-perspective on an object-by-object basis may be not a solution due to that every camera has its own set of coordinates for the same object, which must be adjusted to produce a correct result (as [35]), or a multi-layered method has to be used to solve depth problems.

Our strategy is not to change the observer but to change how the

Figure 6: Sequence with observer movement and fixed deformations

objects are seen. We think that this is a more natural approach than introducing several observers with different types of projection, and also it allows more extreme results to be produced.

The solution proposed is the use of HENLT's dependent on the observer's position. The current implementation of HENLT is based on mono-evaluated control functions. Whilst in animation the control function is controlled by time-dependent functions, in this application the control function is dependent on the observer's position, though it is possible to include time-dependent functions for obtaining animation. The general function that relates the observer's position to the variation in the control function, which defines the deformation, is a 3D function, a infinite volume, such that $\alpha = f(x, y, z)$ with x, y, and z the coordinates of the observer's position, and α is the value that produces the change in the control function. The general process is shown in Figure 4.

One example of deformation is the simulation of the perspective projection. Given that we could only use the parallel projection, the perspective projection can be obtained with HENLT's applying first a constant transformation that moves the object, and secondly a scaling non-linear transformation which depends on z and is applied to x and y (with this formulation $x' = \frac{x*d}{z+d}$, $y' = \frac{y*d}{z+d}$). This is equal to the normalisation process for obtaining the canonical view volume.

Though there are many possibilities for defining such kinds of function, we are interested in those that can be easily understood by the user. Currently, we have developed functions that are dependent on the orientation or are dependent on the distance.

The orientation-dependent functions can be used for performing transformations in the model such as the one shown in Figure 5. The top row shows the figure without deformations, whilst the bottom row shows the character with the observer-dependent deformations applied. When the observer is in front the character, the eye and the ears are seen with the same form as when they are observed from one side. As we are only interested in the orientation, we can use a function that represents a boundary. If we normalise the observer's position vector, and the function is constant, it is represented by a sphere with a radius R, with R being the value that controls the deformation. The sphere-like function must be changed to produce deformations,i.e., the deformation will change if the distance from the origin to the surface changes. The more powerful possibility is using B-spline patches. In the example of Figure 5 the general function has been simplified to a kind of infinite tube-like function, the cross section of which is the desired deformation.

Distance dependent functions are easily implemented because the distance can be used directly with the control function. This kind of functions allow us to obtain deformations that are non perspective-like, as non-linear zoom, and so on.

The two effects can be combined in a ordered way using the general mechanism for combination implemented in HENLT. This mechanism allows us to mix different classes of transformations, and to apply them in a hierarchical way.

Now, we can show the differences between our method and Rademacher's one. We do not transform directly the geometry of the model, but we use the HENLT's. HNLTS's use normally simple 1D control functions. The deformation of the model depends on the HENLT which itself depend on the control function whose parameters are controlled by the observer. So, We only need to define the function that controls the parameters. We can use the interpolation but only for defining the variation of the control function. Depending on the complexity of the model and movement, we may use 1D, 2D or 3D control functions, but, for example, all the figures shown have been obtained with 1D control functions.

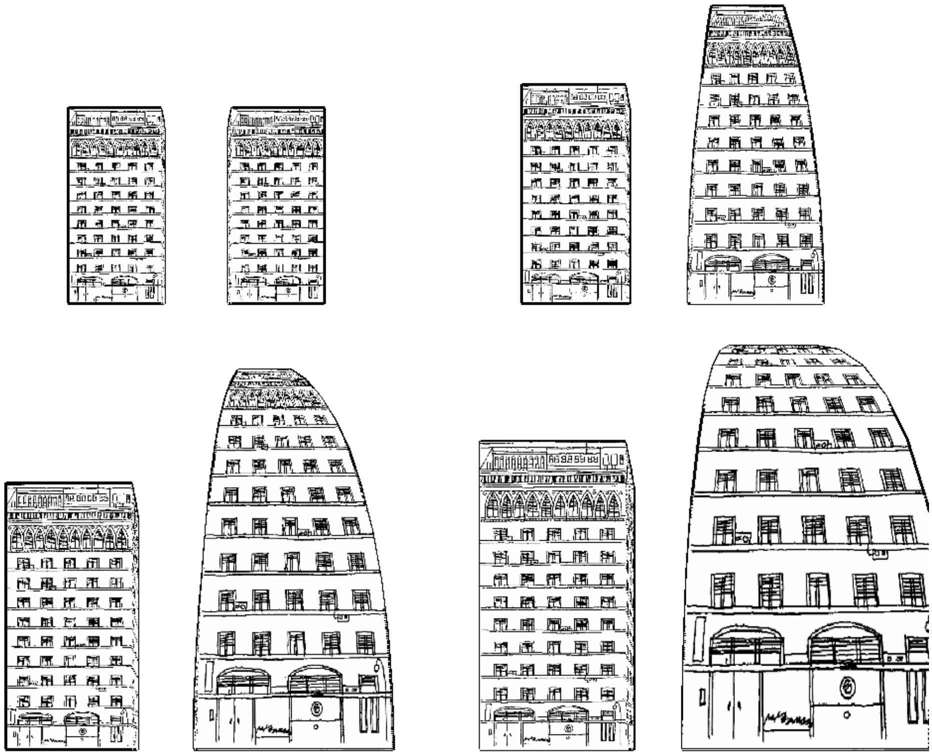

Figure 7: Example of distance-dependent deformations

6 RESULTS

The Alhambra system has been extended to include the capabilities mentioned earlier. The test-bed has been the image of Figure 2. We have produced a synthetic version of the background, taking into account that,in this case, the goal is to save the extra work done when a static background is used many times but with little changes (Figure 3). In the Figure 6 there is a sequence with no observer-dependent deformations, but fixed ones. In Figures 7 and 8 there are two deformations that are applied when the observer is moving. The first one is like a zoom, but the function that controls the size of the flat is non linear. It has also applied a function that makes the flat to stretch as it is near the observer, obtaining an effect that looks like a projection with an eye-fish lens. The transformation used to produce the result is a combination of a scaling and a rotation, with both control functions depending on the distance. The second deformation, Figure 8, is an approximation produced with a non-linear scaling transformation. In both cases, the flat on the left is shown with no transformation applied. In Figures 9 and 10, there is an example of an orientation-dependent deformation. As the observer turns around the flat (Figure 9, it changes its form. This deformation is achieved with only one transformation. In Figure 10, there is the same example, but the flat has its own independent movement. 1D transformation. The

7 CONCLUSIONS

There are works that have simulated the illustration, but only in its mechanical part, in the sense that they simulate the visual results obtained when a pencil or a pen and ink are used. The results obtained are good. The next stage is to obtain the expressive capabilities of the illustration, which implies transforming the models. The need of transforming objects in extreme ways is augmented, in relation to animation. The goals are also different. The use of HENLT dependent on the observer's position is a solution for the problem. The results obtained are good, as can be seen in the Figures.

Though the technique has been developed for illustration, the roots of the system allows us to use the new capabilities with animation. In this case, the use of observer-dependent deformations may opens new possibilities to the the classical and "un-natural" form of looking the world, changing the location of the camera, allowing the capability of adapting the reality to the narration.

We are currently working in the definition of other kind of space-dependent functions. We are studying space partition schemes, like octrees, as well as the mapping of zones of the sphere with control functions.

References

[1] A. Barr. Global And Local Deformations Of Solid Primitives. *Proceedings of SIGGRAPH*, 18(3):21–30, 1984.

[2] N. Burntnyk and M. Wein. Interactive Squeleton Techniques For Enhancing Motion Dynamics In Keyframe animation. *Communications of the ACM*, 19(10):564–569, October 1976.

[3] E. Catmull. The Problems Of Computer-Assisted Animation. *ACM Computer Graphics*, pages 348–353, 1978.

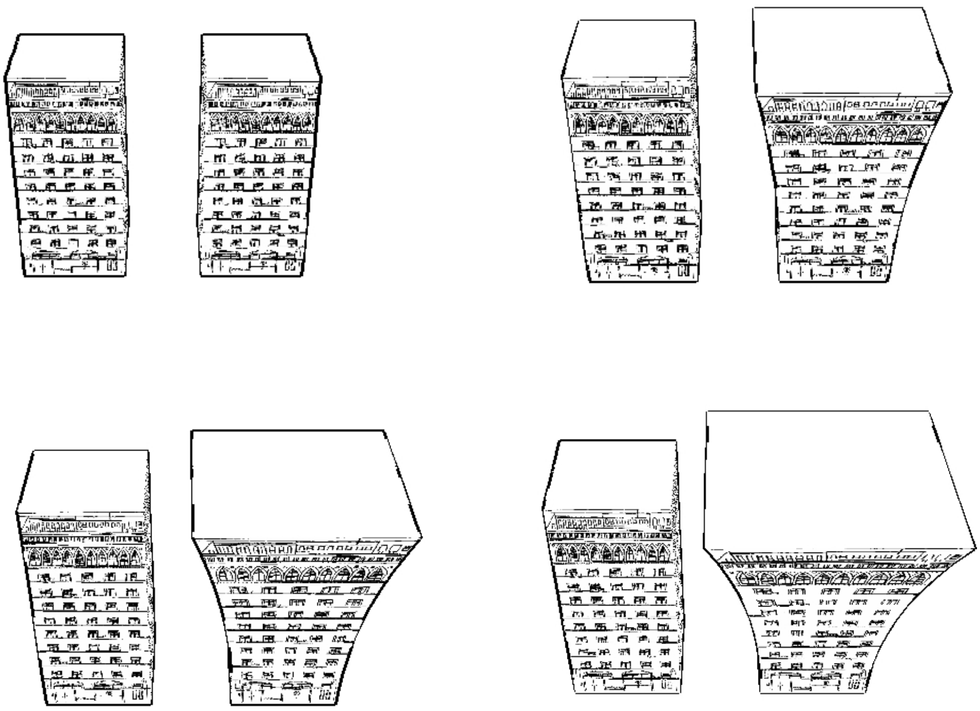

Figure 8: Another example of distance-dependent deformations

[4] P.H. Christensen. Contour Rendering. *Computer Graphics*, 33(1):58–60, February 1999.

[5] S. Coquillart. Extended Free Form Deformation: A Sculpturing Tool For 3D Geometric Modeling. *ACM Computer Graphics*, 24(4):187–196, August 1990.

[6] S. Coquillart and P. Jancene. Animated Free Form Deformation: An Interactive Animation Technique. *ACM Computer Graphics*, 25(4):23–26, July 1991.

[7] Mario Costa and John Buchanan. Computer Generated Graphite Pencil Rendering Of 3D Polygnal Models. *Computer Graphics Forum*, 18(3):C195–C207, September 1999.

[8] S. García. *Amura*. Ed. Glénat, España, 1994.

[9] S. García. *Geografía Marciana, Part. 1 Utopía*. Ed. Dargaud, Brussels-Paris, 1996.

[10] S. García. *Geografía Marciana, Part. 2 Mons Olympus*. Ed. Dargaud, Brussels-Paris, 1996.

[11] S. García. *Geografía Marciana, Part. 3 Domus*. Ed. Dargaud, Brussels-Paris, 1998.

[12] S. García. *Comix 2000*. Ed. L' Association, Paris, 1999.

[13] S. García. *Les Trois Chemins*. Ed. Delcourt, Paris, 1999.

[14] A. Gooch, B. Gooch, P. Shirley, and E. Cohen. A Non Photo Realistic Lighting Model For Automatic Technical Illustration. *Proceedings of SIGGRAPH*, pages 447–452, July 1998.

[15] P. Hall. Nonphotorealistic Rendering By Q-mapping. *Computer Graphics Forum*, 18(1):41–56, March 1999.

[16] S.C. Hsu and I.H. Lee. Drawing And Animation Using Skeletal Strokes. *Proceedings of SIGGRAPH*, pages 109–118, July 1994.

[17] S. Karan and E. Fiume. Wires: A Geometric Deformation Technique. *Proceedings of SIGGRAPH*, pages 405–414, 1998.

[18] J. Lasseter. Principles Of Traditional Animation Applied To 3D Computer Animation. *Proceedings of SIGGRAPH*, 21(4):35–44, July 1987.

[19] P.C. Litwinowicz. Inkwell: A $2\frac{1}{2}$D Animation System. *Proceedings of SIGGRAPH*, 25(4):113–122, July 1991.

[20] L. Markosian et al. Real-Time Nonphotorealistic Rendering. *Proceedings of SIGGRAPH*, pages 415–419, 1997.

[21] D. Martín. *Alhambra, Un Modelo Para La Producción De Animación Bidimensional*. PhD thesis, Escuela Técnica Superior de Ingeniería Informática, 1999.

[22] D. Martín and J.C. Torres. Alhambra: A System For Producing 2D Animation. *Computer Animation 99*, pages 100–114, 1999.

[23] D. Martín and J.C. Torres. Virtual Lights: A Method For Expressive Visualization. *EG99-SP*, pages 67–70, September 1999.

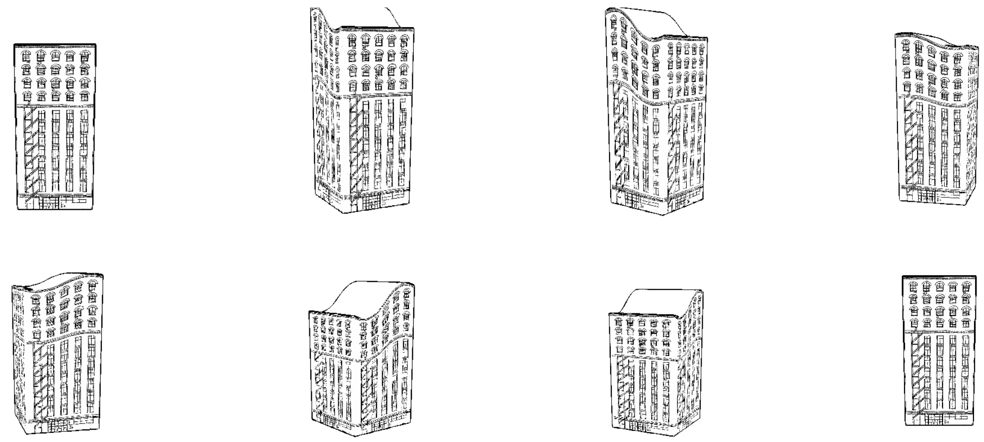

Figure 9: Example of an orientation-dependent deformation

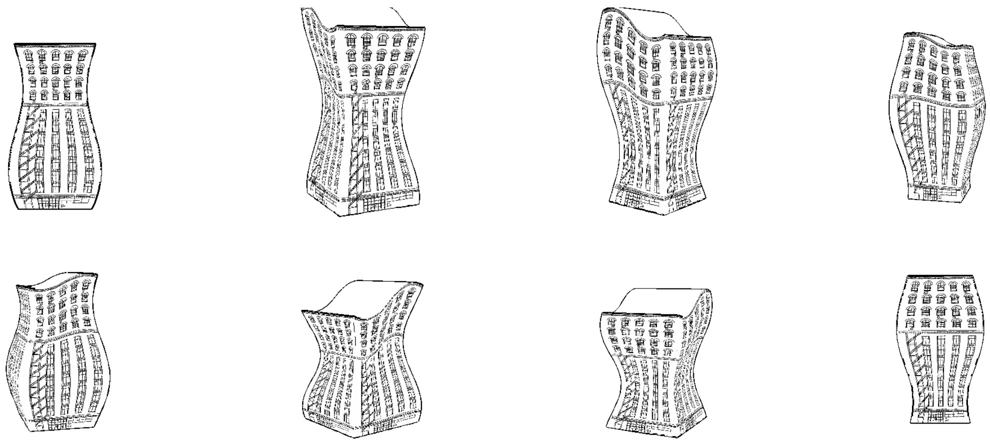

Figure 10: The same example but adding independent movement to the object

[24] D. Martín and J.C. Torres. A General Framework For Non-Linear Transformations. *submitted to SIGGRAPH2000*, 2000.

[25] P. Rademacher. View-Dependent Geometry. *Proceedings of SIGGRAPH*, pages 439–446, 1999.

[26] T. Saito and T. Takahashi. Comprehensible Rendering Of 3D Shapes. *ACM Computer Graphics*, 24(4):197–206, August 1990.

[27] M.P. Salisbury, C. Anderson, D. Lischinski, and D.H. Salesin. Scale-Dependent Reproduction Of Pen-And-Ink Illustration. *Proceedings of SIGGRAPH*, pages 461–468, August 1996.

[28] M.P. Salisbury, S.E. Anderson, R. Barze, and D.H. Salesin. Interacting Pen And Ink Illustration. *Proceedings of SIGGRAPH*, pages 101–108, July 1994.

[29] M.P. Salisbury, M.T. Wong, J.F. Hughes, and D.H. Salesin. Orientable Textures For Image-based Pen And Ink Illustration. *Proceedings of SIGGRAPH*, pages 401–406, August 1997.

[30] S. Schofield. Piranesi:A 3-D Paint System. *Eurographics U.K.*, 1996.

[31] T. Sederberg and S.R. Parry. Free Form Deformation Of Solid Geometric Models. *Proceedings of SIGGRAPH 86*, 20(4):151–160, 1986.

[32] F. Thomas and O. Johnston. *The Illusion Of Life: Disney Animation*. Hiperion, 1995.

[33] A. Watt and M. Watt. *Advanced Animation And Rendering Techniques. Theory And Practice*. Addison-Wesley, 1992.

[34] G. Winkenbach and D.H. Salesin. Computer Generated Pen And Ink Illustration. *Proceedings of SIGGRAPH*, pages 469–476, July 1994.

[35] Daniel N. Wood, Adam Finkelstein, J.F. Hughes, Craig E. Thayer, and David H. Salesin. Multiperspective Panoramas For Cel Anmation. *Proceedings of SIGGRAPH*, pages 243–250, August 1997.

[36] D. Zorin and A. H. Barr. Correction Of Geometric Perceptual Distortion In Pictures. *Proceedings of SIGGRAPH*, pages 257–264, August 1995.