Fast Customization of Geometric Models

MARCELO WALTER[1], CRISTIANO G. FRANCO[1]

[1]Mestrado em Computação Aplicada–Universidade do Vale do Rio dos Sinos
Av. Unisinos 950, São Leopoldo, RS, Brasil
`{marcelow,franco}@exatas.unisinos.br`

Abstract. We present in this paper a technique for customization of geometric models of animals based on pictures. The object to be customized is divided into a set of hierarchically related local coordinate systems which allow for independent transformations of different body parts. A picture of the desired result is used as a visual aid to drive the customization process. We show the results of applying this technique to customize a generic three-dimensional horse model to 3 different horse breeds.

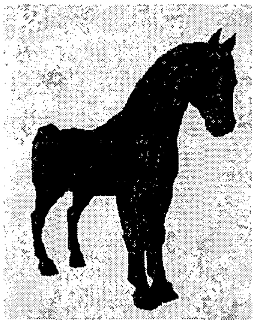
1 Introduction

The process of building complex geometric models such as animals and humans is still highly dependent upon artistic abilities of skilled designers. For this reason commercially available models, such as the ones made by Viewpoint (`www.viewpoint.com`), are expensive and too generic for specific applications.

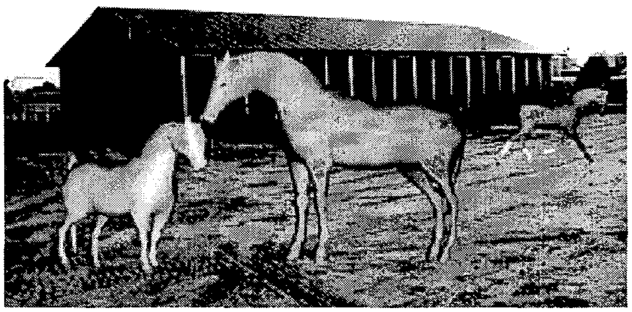
On the other hand, the need for customized models arise in many applications. Commercial customization is available but building a custom model is even more expensive than building a generic one. By customization we mean the process of creating a model according to the user's specification.

In this paper we propose a technique to help the user customize a pre-existent *generic* geometric model to fit its particular needs. The process of customization is driven by pictures representing the final desired shape of the models. The process is semi-automatic since a full automatic solution for this problem, based on pictures, is highly complex.

Our solution incrementally modifies the generic 3D model, until it resembles the one in the picture using local transformations. It should be pointed out, however, that the range of possible customizations is limited by factors explained later. Therefore the proposed technique is good for "fast" and coarse customizations. In our case a solution is possible because we assume that the image represents an animal of the same species as the 3D model. Figure 1 illustrates a result of our system. The three horses were all generated from the same generic horse model (shown in (a)).



(a) Original Horse Model



(b) Three different horse breeds

Figure 1: Three different horse models generated with our system

2 Previous Work

The main goal of our research is the controlled transformation of a *single* shape or model. The techniques for this task are usually referred to as *deformation* of geometric shapes, considering that the final result is a deviation of the origi-

0-7695-0878-2/00 $10.00 © 2000 IEEE

nal. Many modelling tasks usually have this same goal, but the emphasis is on the process of building the final object only, and not on the transformation process itself.

The technique we will use here to customize polygonal models will be similar to the feature-based 2D morphing techniques, for instance as described by Beyer and Neely [1]. The main differences are that we are operating in 3D, that we only have *one object* that is to be transformed, and that we want the set of features to be organized in a structure that will allow for arbitrary scaling and rotation of parts of the model. Therefore we review here the techniques generically known as *morphing* between two objects.

These techniques input two objects and generate a set of intermediate ones such that the sequence conveys the idea that the first object was transformed into the second. Basic references go as far back as Burtnyk [2] and Reeves [11] and these techniques are referred to as *morphing*, *metamorphosis*, *key-framing*, and *in-betweening*. For a polyhedral representation the morphing problem is reduced to finding for a given point in the first object the corresponding point in the second one. Different approaches then only differ on how to achieve this correspondence. In the approach by Hong *et al.* [5] for example, more than one face in the source object can be mapped to the same face in the target object and faces can degenerate into points. The approach taken by Kent, Carlson and Parent [8, 7] is to use an intermediate object to establish the correspondence, usually a sphere. Their solution first positions the objects to be transformed in the center of a canonical sphere and their vertices are projected towards the sphere. This creates two sets of projected vertices, one set from each original object. The algorithm then maps back in the objects the vertices which were not originally there. This creates the necessary correspondence. Chen presented an approach that morphs two objects by morphing their 2D parametric descriptions [3]. The main drawback of the technique is the conversion to parametric representation from a polygonal one. A solution proposed by Lazarus and Verroust [9] explores the idea of features to compute a morph between 2 poliedra. In their work the features are defined as a 3D coordinate system inside each object (the same as a skeleton for the objects). A sampling process on these skeletons is used to discretize the objects and provides a parametrization for the same. A morph between the two objects is built by interpolating between the 2 parametrizations.

There are other solutions for the shape transformation problem when the objects are represented by volumetric data. Hughes [6], for example, transforms the data into the frequency domain and the low frequencies from the source object are interpolated into the low frequencies of the target one. Finally, Lerios *et al.* [10] have extended the idea of 2D features from Beier and Neely [1] to 3D volumetric represented objects. A set of features defining fields of influence for the source and target objects is defined and the morphing process uses these features to warp voxels from the source object into the target one.

3 Transformation of 3D Models

Since our solution is an improvement over the technique presented by Walter and Fournier [13] we briefly review it here. In that work they presented a technique to transfer growth data to polygonal models of animals. For each section of the body that will be transformed independently, a cylindrical coordinate system is attached to it. The cylinder is positioned so that it encloses that part of the body it controls. These cylinders are predefined by the user, and are designed so that they segment the object into *coarse groups* of geometry that represent the object in a higher structured level. For each cylinder it is also defined a pair of *features* which indirectly control the cylinder's scale. Once this structure of cylinders and features is in place, the user can modify any body part by acting on the features. In the case of growth simulation the features are normally chosen so that they correspond to the available or obtainable growth data.

The cylinders are part of a classical modelling hierarchy. Figure 2 shows an example of the 18 cylinders defined for a horse's model. Figure 3 shows the features associated with the same model. Each feature is a length measurement between two specific points on the body.

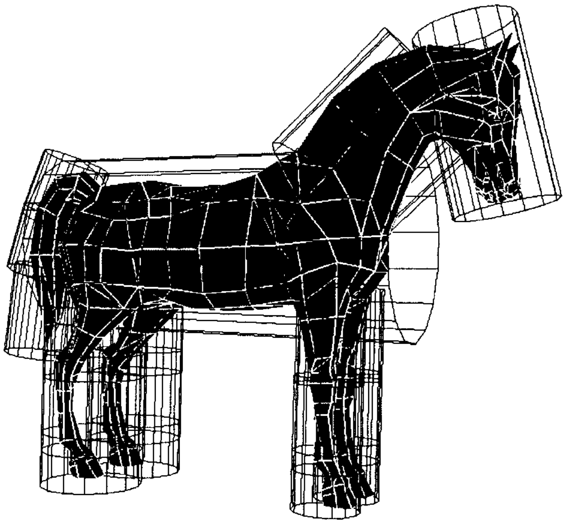


Figure 2: The 18 local coordinate systems defined for a horse model

The main difference between a normal transformation hierarchy and the one used for controlled transformation is that since the growth data available is generally expressed in absolute terms, only translation and rotation are inherited from the ancestors in the growing hierarchy, but the scaling is local and in absolute terms. Each cylinder A has its own canonical coordinate system, and an associated matrix to transform to the world coordinate system:

$$M_{W\leftarrow A} = [T_{W\leftarrow A} R_{W\leftarrow A} G_{W\leftarrow A}]$$

where $T_{W\leftarrow A}$ is a translation matrix, $R_{W\leftarrow A}$ is a rotation matrix and $G_{W\leftarrow A}$ is a scaling matrix, the growth matrix. The growth matrix is given by:

$$G_A = \begin{bmatrix} L_A & 0 & 0 & 0 \\ 0 & R_A & 0 & 0 \\ 0 & 0 & R_A & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

where R_A and L_A are the world radius and length of the cylinder, which can be easily derived from such measurements as the length and girth of features of the real animal or its pictures (see for instance Table 1 with measurements for a quarter horse taken from Cunningham [4]).

Sample data for quarter horse (inches)				
Age (months)	Length from elbow to ground	Diameter of cannon bone	Length of hind leg	Width of head
0	25.0	1.43	18.1	5.6
12	33.1	2.19	22.8	7.8
24	35.9	2.39	23.0	8.6
36	35.1	2.39	23.2	8.9
48	34.9	2.42	23.1	9.1
60	35.8	2.51	23.3	8.8

Table 1: Some measurements for a quarter horse.

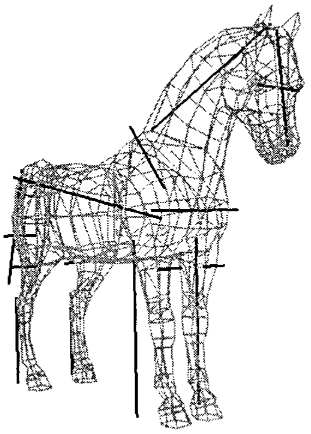


Figure 3: Features defined for the horse model

The operation to transform a point P_B in feature B (associated with a cylinder B) to a point P_A in the coordinate system of its parent feature A (associated with cylinder A) is:

$$P_A = [T_{A\leftarrow B} G_A^{-1} R_{A\leftarrow B} G_B] P_B$$

where:

$$T_{A\leftarrow B} = [M_{A\leftarrow W} T_{W\leftarrow B} T_{A\leftarrow W} M_{W\leftarrow A}]$$

One then applies these transformations to convert to the coordinate system of the ancestor of A, until the root is reached, at which point we have the point in world coordinate system. The growth process then consists in applying these transformations to each needed vertex for each time at which we have measures for the feature. The growth data can be interpolated between time using any suitable interpolating formula.

To guarantee continuity as the shape changes, and to achieve a degree of smoothness in the resulting surface, the cylinders have to overlap, and therefore one has to decide how to weight the influence of the cylinders on a vertex which belongs to more than one. The solution proposed uses a weight inversely proportional to the distance from the vertex to the axis of the cylinder.

It is important to note that applying the technique has two practical consequences: any measurement from a real animal once applied to the polygonal model has the effect of changing the proportions of the original model (which could be an idealized or otherwise fictitious animal) to the proportions of the real measured animal. That means that for a single model the technique can create individual bodies with their own measurements, which we explore in this paper.

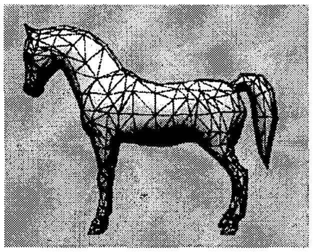
The second point is that the same hierarchy can be used for animation, since it allows for independent rotations and translations to be applied to the various local coordinate systems enclosing the body.

4 Customization of 3D Models

The extension on the previous work proposed in this paper is the use of pictures of real animals to derive the measurements needed for customization. The process of customization starts with the user specifying a source image and a source object to be customized as illustrated in Figure 4.

As mentioned before, the image and 3D object should be of the same animal that we are customizing. Since we will be deriving the measurements from the image, it is important the image to be as close as possible to a sideview of the animal. This is not strictly necessary but it is more intuitive for the user. When the image is not a perfect sideview we could apply a correction factor to compensate for perspective foreshortening. This would demand estimating camera parameters for the image, a task beyond the scope of the current work. We will be exploring this in future work, as part of a more sophisticated fully automatic solution.

(a) Source Image



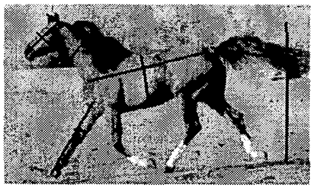
(b) Source Original 3D Object

Figure 4: Start of Customization Process

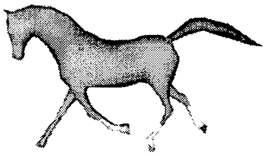
The process continues with the user selecting one feature from the 3D model which she wants to control with a corresponding line in the 2D image. This establishes a correspondence between the two. The first correspondence is more important since it establishes the scale between the 2 "worlds", the 2D world and the 3D one. We call this the *scale correspondence*. Basically, we need to define that a given distance d_{3D} in the 3D world corresponds to d_{2D} pixels in the 2D image. The scale is then computed as $s_{2D\leftarrow 3D} = d_{3D}/d_{2D}$. Therefore, the new length l_{3D} for a given feature in the 3D world will be derived from a corresponding length l_{2D} in the 2D world as $l_{3D} = l_{2D}s_{2D\leftarrow 3D}$.

Any line in the image together with any feature in 3D would work to establish the scale correspondence. Nevertheless, we have found useful to use the feature defined for the length of the body as the basis to establish such a correspondence. For this feature we create a first line in 2D (the larger vertical line shown in Figure 5(a)). The purpose of this line is to help the user fine tune the best scale without affecting the 3D model. After the scale correspondence is made the user can ask the system to customize the model up to that point. This is done continuously and interactively until the user considers the result satisfactory.

The same line in the image can, and usually does, correspond to more than one feature in the 3D object. This is normally the case, for instance, when one single line controls the features for both front legs of a horse. Not necessarily all features of the 3D model are customized. The body parts not affected by the process remain the same. After a few iterations, where the user can fine tune the lengths of lines in the 2D image, the model is customized.



(a) Source Image with Lines



(b) Customized Horse

Figure 5: Final Result of Customization

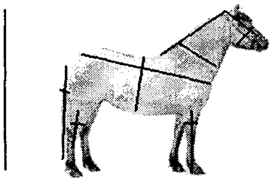
Since we are using cylinders with a circular base, we can afford to have only 2 measurements for each cylinder and still modify the 3D shape accordingly. This means that we can not modify depth information. The modified value for the depth or z-coordinate will be proportional to the transformation applied to the y-coordinates. In future work we plan to use cylinders with a non-circular base as primitives and therefore be able to control the z-coordinate as well.

In Figure 5 we can see that besides scale, the final customized horse has the same posture as the one in the image, a horse galloping. This is accomplished independently from the scaling modifications since for each cylinder the user can specify a rotation to be applied.

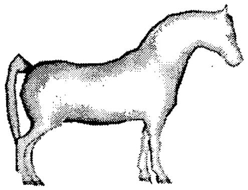
5 Results

We show in this section two results of applying the technique to customize a 3D horse model. The original 3D general horse model is shown in Figure 1(a).

We have chosen 2 horse breeds with distinguished body differences. The first one is the *Highland Pony* shown in Figure 6. It has shorter than usual legs and a compact body, as usual for ponies. Figure 7 shows the results for a *Don* type of horse. This type of horse has a more balanced structure and leaner body. For the Highland Pony we have used



(a) Source Image



(b) Customized Horse

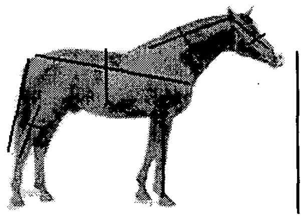
Figure 6: Source Image and Result for a *Highland Pony*

as many lines as features in the 3D model whereas for the Don horse there was no need to customize the width of the foreleg.

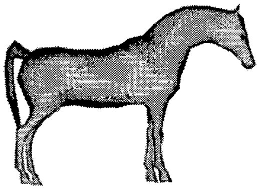
As we can see from the results, the technique allows for coarse body modifications only. Of course, the same initial distinctive body characteristics of the original 3D model are maintained. This is evident in the neck area of the original horse model, which has a strong curvature also present in the customized versions. This could be fixed at the expense of introducing more cylinders in the hierarchy. We could, for instance, represent the neck with 2 cylinders instead of one, and force the underlying geometry representing the neck to "straight the neck up". We should point out also that we are using models with a relatively low count of triangles, only 1344 triangles. A more detailed mesh would allow for smoother customizations. In a sense the results here could be interpreted as the "worst case". Figure 8 shows another result of composing the individual horses into a single scene.

6 Conclusions and Future Work

We have presented a method for customization of geometric models representing animals. The user inputs an image of the desired result and traces lines on this image which indirectly control local shape transformations on the 3D model. The 3D model is represented by a set of hierarchically re-



(a) Source Image



(b) Customized Horse

Figure 7: Source Image and Result for *Don Horse*

Figure 8: A rural scene

lated local cylindrical coordinate systems.

This technique is attractive since it uses a picture as a visual aid to modify the shape of a 3D object. The method is useful for "fast" customization of a pre-existent model. In our experience, once the support for transformations is in place (cylinders plus features), changing a given model can take from a few minutes to half an hour, depending on how detailed the user wants the morph to be.

The current version of the system allows for a wide range of modifications without restrictions. In some cases it could be interesting, however, to have a "biologically correct" result (in terms of proportions). We plan to implement this feature based on animal studies [12] which give ideal proportions for animals, and restrict the modifications according to these rules.

The ultimate goal here is to input an image and a 3D model and let the system do the morph automatically, without any user intervention. This is an ambitious goal which we will pursue in future work, possibly exploring and integrating image processing and computer graphics techniques.

Acknowledgments

We would like to thank Tatiana Evers (for her expertise on 3DStudio) and Alessandro Amora from Unisinos who helped putting together parts of this work. This work has been partially supported by FAPERGS through grants 99/-0453.8 and 99/50455.8.

References

[1] T. Beier and S. Neely. Feature-based image metamorphosis. In Edwin E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 35–42, July 1992.

[2] N. Burtnyk and M. Wein. Interactive skeleton techniques for enhancing motion dynamics in key frame animation. *Communications of the ACM*, 19:564–569, 1976.

[3] D. Chen, A. State, and D. Banks. Interactive shape metamorphosis. In *1995 Symposium on Interactive 3D Graphics*, pages 206–215, April 1995.

[4] K. Cunningham and S. H. Fowler. A study of growth and development in the quarter horse. Technical Report 546, Lousiana State University - Agricultural and Mechanical College, November 1961.

[5] T. M. Hong, N. Magnenat-Thalmann, and D. Thalmann. A general algorithm for 3D shape interpolation in a facet-based representation. In *Proceedings of Graphics Interface '88*, pages 229–235, June 1988.

[6] J. F. Hughes. Scheduled Fourier volume morphing. In Edwin E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 43–46, July 1992.

[7] J. Kent, R. E. Parent, and W. E. Carlson. Establishing correspondences by topological merging: A new approach to 3-D shape transformation. In *Proceedings of Graphics Interface '91*, pages 271–278, June 1991.

[8] J. R. Kent, W. E. Carlson, and R. E. Parent. Shape transformation for polyhedral objects. In Edwin E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 47–54, July 1992.

[9] F. Lazarus and A. Verroust. Feature-based shape transformation for polyhedral objects. In *5th Eurographics Workshop on Animation and Simulation*, pages 102–110, September 1994.

[10] A. Lerios, C. D. Garfinkle, and M. Levoy. Feature-Based volume metamorphosis. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 449–456. ACM SIGGRAPH, August 1995. held in Los Angeles, California, 06-11 August 1995.

[11] W. T. Reeves. Inbetweening for computer animation utilizing moving point constraints. *Computer Graphics (SIGGRAPH '81 Proceedings)*, 15(3):263–269, August 1981.

[12] E.E. Thompson. *Anatomy of Animals*. Random House UK Ltd., 1896 (reprinted in 1996).

[13] M. Walter and A. Fournier. Growing and animating polygonal models of animals. *Computer Graphics Forum (Eurographics'97)*, 16(3):C–151–C–158, September 1997.