EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling (2006)
Thomas Stahovich and Mario Costa Sousa (Editors)

Transformation Strokes

Aaron Severn[1], Faramarz Samavati[1] and Mario Costa Sousa[1]

[1]University of Calgary, Calgary, Canada

Abstract

In this paper, we present a sketch-based technique for specifying transformations for general models by means of a single stroke, offering a more streamlined form of user interaction. The shape of the stroke is interpreted to allow composition of translation, rotation and scaling. We extract two main directions from the input stroke using Principle Component Analysis and use them to obtain an appropriate transformation for the model. Our method helps to have a more natural and faster way of assembling 3D structures. It is general and does not depend on specific knowledge about the type of models. As such, it can fit in the majority of graphics systems or modelling techniques.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling: Modeling packages

1. Introduction

Today, 3D modelling systems are used in a wide variety of applications such as designing mechanical parts, architecture, and creating scenes in animated films. Computer models have replaced traditional drawings and physical models in many areas. The demands of precision modelling have led to complex interfaces, where parameters must be carefully supplied by the users, which often require significant skill and expertise to use effectively. These systems, while effective at producing precise three dimensional models, are not well suited to applications where exact precision is not essential.

One of the key components of any 3D modelling or viewing system is user controlled transformations. Transforming models is a fundamental operation in computer graphics. Traditional systems usually support transformations through a click and drag interface, where translation, rotation, and scaling are divided into three distinct operations, or by defining complicated relationships between models. Simply a mouse click in 3D is not enough to provide a reasonable three dimensional transformation. This presents difficulties, making even simple manipulations surprisingly hard to perform([TDM01], [CSSJ05]). Positioning two models for a Boolean operation, for instance, can involve a sophisticated set of transformations until the alignment of the models is as desired.

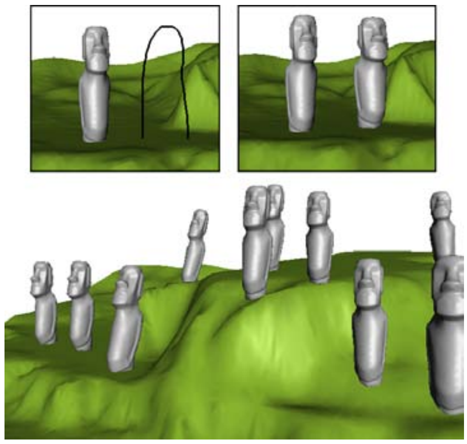

Figure 1: *A scene created in a few minutes with transformation strokes.*

We present a novel approach to performing transformations in a modelling system using a single stroke. This is a justifiable effort because a stroke provides more information than a mouse click. In addition, it not only gives a better visual impact but also approximates our regular drawing metaphor([ZHH96], [IMT99]). Our transformation stroke is designed to interpret the likely intensions of the user so that a previously difficult transformation can be performed easier. From a single stroke, we interpret the translation, rotation, and non-uniform scaling intended by the user and apply these to a model. When used in conjunction with traditional transformation interfaces our method may allow users to position models with less difficulty and effort (Figure 1).

It is possible to specify the transformation more accurately when we take into consideration more information from the models. However, our goal is to find a general method that can specify the transformation without any (or with a minimized amount of) specific knowledge about the type of models. This assumption helps to have a method general enough to be fit in any graphics system or modelling technique. On the other hand, it makes the problem harder with several ambiguities.

1.1. Related work

Few works have explored the use of simple sketch-based stroke indicators or commands for specifying linear transformations for modeling 3D objects. Pereira et al. [PJBF03] present a calligraphic sketching metaphor for a CAD/drawing system (GIDeS) providing a paper-like interaction for various modeling operations including translation. Gomis et al. [GACN04] use strokes for indicating 2D symmetry operators in a calligraphic editor for tile and textile design. Igarashi and Hughes [IH02] present a sketch-based indicator approach for putting clothes on a 3D character and manipulating them. The user paints freeform marks on the clothes that are then placed around the body so that corresponding marks match. Ijiri et al. [IOOI05] present a SBIM system used together with floral diagrams and inflorescences to provide the positional information for assembling individual flower components.

Our approach is based on a single stroke-based technique allowing translation, rotation and scaling into a single operation. We use our transformation strokes as a tool for transforming models, rather than modelling by transformation. As such, they are general and can be used in any kind of modelling.

1.2. System overview

Our transformation stroke supports manipulation of arbitrary models, freely and with respect to other models, in a three dimensional environment. We have employed it in a system that supports loading multiple mesh models, which can be manipulated and have operations performed on them such as Boolean operations and subdivision. Strokes are performed by tracing with the mouse along the intended stroke path. The mouse position is sampled at discrete intervals in order to create a polyline representation of the stroke.

Our system supports an *active model*, which can be selected by the user if desired. If an active model is selected, transformation strokes will use it as a reference for aspects of the transformation that are difficult to interpret using two dimensional input, such as the desired depth of a model.

2. Method

2.1. Stroke interpretation

A 2D stroke is not enough to determine all parameters and freedoms of a 3D transformation. We need to come up with natural and simple interpretations for ambiguities. While our system accepts and interprets any generic stroke, for clarity of understanding we can perceive the strokes to be U-shaped, with the height of the U being greater than its width (Figure 2). We need to determine three sets of information from our stroke: the target position of the model; the target orientation; and the target scale. To obtain these transformations we extract four measurements from the stroke. We find a vector from the base to the top of the U, which we call the *major axis*, and determine its magnitude, which will be used to determine the target scale. We find a vector perpendicular to the major axis in the plane of the stroke, the *minor axis*, and determine its magnitude, also for use in scaling. We use the centre of the stroke to determine the target position, and the orientation of the major axis for the target orientation.

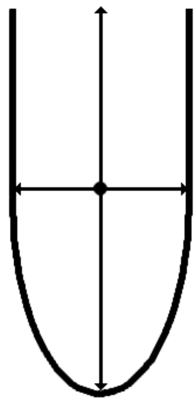

Figure 2: *Interpretation of the transformation stroke.*

The major and minor axes of the stroke are computed using principle component analysis [DH73]. We compute the covariance matrix

$$M = \frac{1}{n}\sum_{i=0}^{n-1}(P_i - C)(P_i - C)^T$$

where P_i denotes the points of the stroke, and C denotes the mean of those points. Since the stroke lies in the xy-plane, this results in a 2x2 matrix. The principle components, which will form the axes of the stroke, are the eigenvectors of M

and can be computed by solving a quadratic equation in the 2x2 case. These two orthogonal vectors are the axes of maximum and minimum variance, thus indicating the orientation of the stroke. We determine the magnitude of the axes by projecting each of the points of the stroke on to each of the axes to determine how far the stroke extends in each direction. By taking the midpoint of the extents along each axis we can determine the centre of the stroke (Figure 3).

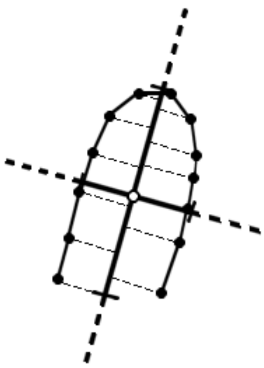

Figure 3: *Stroke extents are determined by projecting each stroke point on to the computed axes, the stroke centre is at the midpoint along each axis.*

2.2. Translation

Our translation works by moving the centre of a model to the centre of the stroke. We compute the centre of the model as the centre of an axis-aligned bounding box around the model since this is the fastest and easiest approach, and typically gives good results. Alternately, it could also be computed as the centre of an oriented bounding box, the centre of mass, or any other means that is appropriate.

Since the stroke is defined only in the xy-plane, this simple translation does not deal with the depth of the model. We make use of two heuristics for resolving this issue. The first is to simply maintain the current depth of the model. The second involves selecting another model for reference, which we call the *active model*, and using its depth.

2.2.1. Active model

When using an active model to determine the intended depth, the depth of the active model will vary across its surface. Thus we interpret the desired behaviour in three different ways.

- If the start and end points of the stroke lie over the active model (Figure 4) then we determine the stroke as follows. We cast two rays from the view point, through the start and end points of stroke, to determine where the rays enter and exit the active model. We then take the average of these four points and use its depth as the new depth of the model that is being transformed. This method allows the user to define a region of interest in the active model, to which the transformed model should be moved.
- If the start and end points do not lie over the active model then we approximate the desired depth using the centre of a bounding box around the active model (Figure 5). Here, the active model is used as a kind of reference for the depth.
- The user is allowed to select a point on the surface of the active model, which we call the *active point*, and if this point is selected it will define the depth (Figure 6).

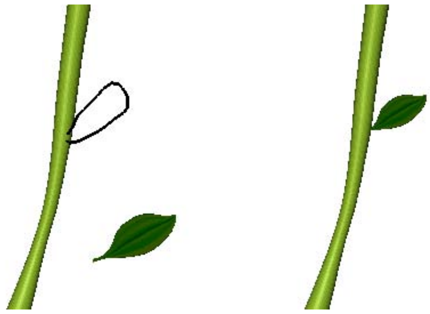

Figure 4: *When the stroke is drawn with its end points above the active model, this defines a region of interest on the active model. Here, the leaf is placed on the stem.*

Figure 5: *The active model (the sphere) is used to bring the cylinder to the front.*

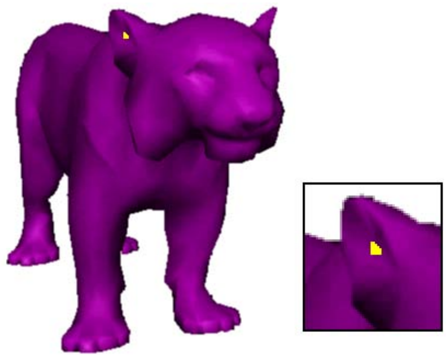

Figure 6: *An active point can be used when the centre of the model doesn't provide an accurate enough depth. Here, the active point is on the left ear.*

After a depth reference is determined we project the stroke centre to the desired depth, using an appropriate projection to match up with the viewing transformation, and then compute the translation from the model centre to the stroke centre at that depth. It is important to note that simply translating the stroke centre to the desired depth often will not produce the desired result, such as when a perspective transformation is being used.

2.3. Rotation

When determining the rotation we have more degrees of freedom than what can be expressed in the stroke, thus we must choose an appropriate interpretation. The intention of our rotation is to align the longest axis of the model to the stroke's major axis. This approach gives a logical and predictable result that is useful in practice. In our default interpretation, we assume that the model will be axis-aligned in some sense, so that the longest axis of an axis-aligned bounding box will yield a meaningful orientation. This is often true of manufactured objects such as pipes or nails, but can also be true for natural objects. We determine the longest axis out of the x-, y-, and z-axes using an axis-aligned bounding box which, in the case of our system, has already been computed around each model. We have used two different methods for finding the axis of rotation. In the first, the axis of rotation is the cross product of the longest axis of the model and the major axis of the stroke, and the angle of rotation is the inverse cosine of the dot product of those same two vectors, normalized (Figure 7). In the second method we can simply use the normal to the screen as the direction of the axis. In the first method the proportion of the source and the target axes is important, while in the second method the orientation of the model towards the viewer is kept unchanged.

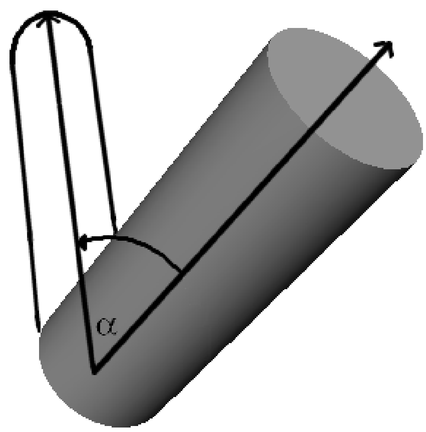

Figure 7: *Rotation from the longest axis of the model to the stroke major axis.*

We would like to be able to predict which direction the model will be oriented based on the direction of its main axis and some property of the stroke. Here, we choose the curved portion of the U-stroke to indicate which way the major axis is pointing. Principle component analysis produces a vector in the first or fourth quadrant (having a positive x coordinate) for the axis of maximum variance, so there is a danger that our stroke's major axis will be oriented in the opposite direction of what we want. To determine whether or not this is the case, we project the start point of the stroke and a middle point of the stroke on to the major axis. If the dot product between the first point of the stroke and the major axis is greater than the dot product between a middle point and the major axis then our axis must be reversed, so we correct the orientation by multiplying the major axis by -1.

In some cases using a main axis aligned to either the x-, y-, or z-axis will perform poorly, since many freeform models are not axis-aligned. To handle these situations we also allow the user to define the main axis directly. It may seem appropriate to use the longest axis of an oriented bounding box around a model when computing a rotation, however the axis-aligned approach performs well in many common cases. It is also quicker and works better with our scaling method, in which scaling is applied only along the x-, y-, or z-axes, thus we prefer to use an axis from the standard basis to define the rotation.

2.4. Scaling

Scaling is also a difficult transformation to fit into our framework for transformation strokes. When combined with rotation it creates an ambiguity: the model could either be scaled to the dimensions suggested by the stroke without rotation, or it could be rotated first and then scaled. Also, since our two dimensional strokes are intended to represent the non-uniform scaling of a three dimensional model we are missing information along one axis.

We have resolved the first problem by choosing to always rotate first and then scale. This choice is appropriate since it results in a close aspect ratio between the original model and the transformed version, which is typically what the user would expect. The second problem is resolved by assuming that the model should only be stretched or compressed along the longest axis, while the aspect ratio between the other two-axes should be maintained. We only need two scaling factors to define such a scale, which is what we have from the stroke.

Initially, we use the magnitude of the stroke major axis to scale along the longest axis of the model. This is the same axis that we used to rotate the model. The scale factor will be the magnitude of the stroke major axis divided by the magnitude of the longest model axis. This scale stretches or compresses the model to the same length as the stroke.

Next, we determine the aspect ratio of the two remaining axes, v_0 and w_0, which is calculated as $a = |v_0|/|w_0|$. We intend to scale the diagonal of the rectangle defined by v_0

and w_0 to the magnitude of the stroke minor axis (Figure 8), we will call this magnitude d. Thus, we can solve for the scaled magnitude of v_0 and w_0

$$|w| = \sqrt{d^2/(a^2+1)}$$

$$|v| = a * |w|$$

where v and w represent the scaled vectors. We then compute the scaling factors as $|v|/|v_0|$ and $|w|/|w_0|$ respectively.

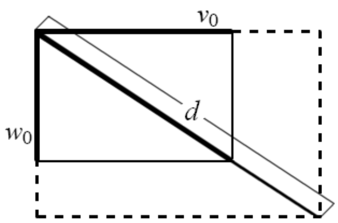

Figure 8: *We scale the diagonal of the shorter two sides to the magnitude of the stroke minor axis.*

Figure 9: *A dead tree is constructed by positioning copies of a primitive using strokes.*

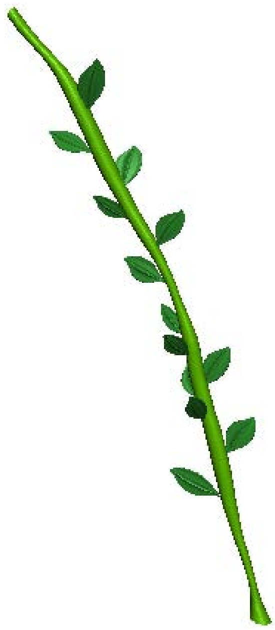

Figure 10: *Many leaves can be quickly added to a stem to produce a vine.*

3. Results and discussion

We have used our transformation stroke to position and resize models in a wide variety of modelling applications. Figure 1 depicts a scene where ten statues were positioned on a landscape using only ten strokes. It makes use of the active model feature to position the models so that their bases are located on the part of the terrain lying under the stroke end points. Since precise positioning of the statues is unnecessary to create the desired scene (and may even be detrimental to the realistic appearance) a sketch-based approach works well.

Transformation strokes can be used to quickly place models. We have used them to assemble larger models from their parts, for both natural objects such as the dead tree in Figure 9, which was created from a single simple primitive, and the vine in Figure 10, and mechanical objects such as the various gears shown assembled in Figure 11. Note that the gears were placed without using scaling since the parts were already precisely sized.

Our method is best suited to models that have a well defined main axis, which is true of many real world objects. This property can be seen in man-made objects, such as nails and cars, and also in natural objects like leaves and trees (Figure 12). The ability to specify a user-defined main axis extends the usefulness of our transformation stroke to additional objects, however those where no reasonable main axis exists, such as spherical objects, present difficulties.

4. Conclusion and future work

We have presented a stroke based method for performing transformations in a modelling system. Our transformation stroke is capable of interpreting the desired translation, rotation, and non-uniform scaling in many common situations. By building a complex transformation from a single stroke, we allow for quicker and easier manipulation of models than

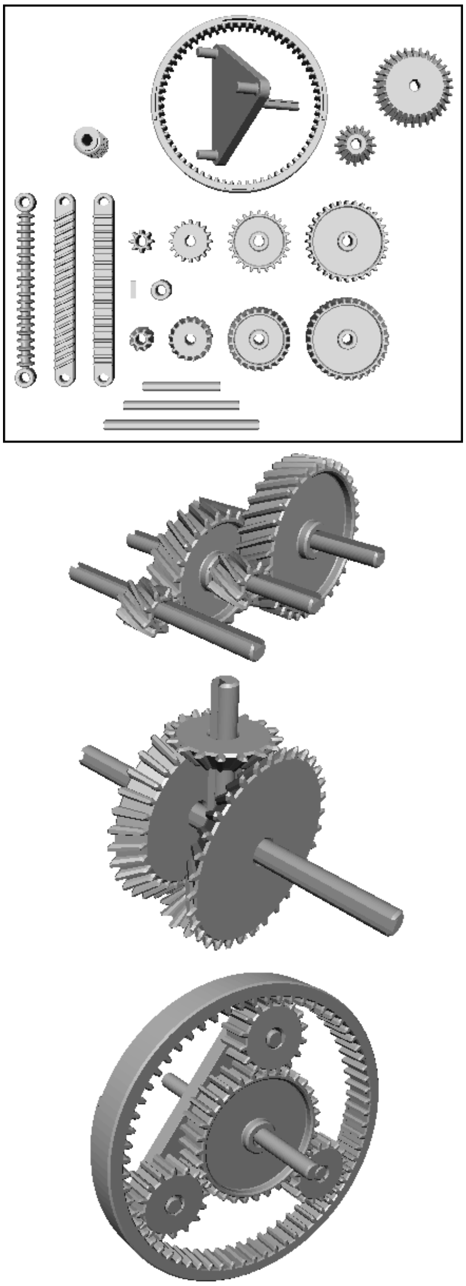

Figure 11: *The gears were assembled using transformation strokes in conjunction with standard transformation techniques.*

was previously possible with a mouse. This transformation stroke is relatively easy to implement and does not involve a high computational cost.

While our transformation stroke interprets many of the desired manipulations of a user, it is not yet comprehensive.

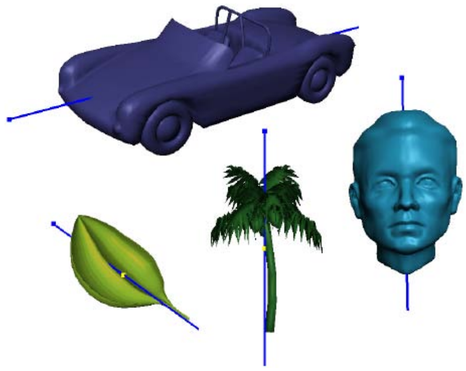

Figure 12: *Many real world objects have a well defined main axis, such as the ones shown here.*

Our stroke is best suited for models with a well defined main axis, and while this encompasses a wide variety of models, further work is required to produce a transformation stroke that will interpret the user's intentions under more circumstances. Although using a bounding box's axes and the major axis of the stroke can provide a useful default interpretation of the user's intension, it is not always correct and requires additional information that might be directly supplied by the user. However, it would be better if all necessary properties could be determined automatically by the system.

Many aspects of our system involve choosing an appropriate interpretation among a variety of possibilities. These choices have a profound impact on the behaviour of our transformation strokes, in particular for rotation and scaling. While we have attempted to choose the best interpretation for all ambiguous situations, some other interpretation may prove to be more appropriate. These alternate interpretations require further study to determine the best choice. User studies need to be conducted to determine the best interpretations and to assess the overall effectiveness of the method.

References

[CSSJ05] CHERLIN J. J., SAMAVATI F., SOUSA M. C., JORGE J. A.: Sketch-based modeling with few strokes. In *21st Spring Conference on Computer Graphics* (2005).

[DH73] DUDA R. O., HART P. E.: *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.

[GACN04] GOMIS J. M., ALBERT F., CONTERO M., NAYA F.: Calligraphic editor for textile and tile pattern design system. In *Proceedings of SmartGraphics '04. Lecture Notes in Computer Science vol. 3031* (2004), pp. 114–120.

[IH02] IGARASHI T., HUGHES J. F.: Clothing manipulation. In *15th Annual Symposium on User Interface Software and Technology, ACM UIST* (2002), pp. 91–100.

[IMT99] IGARASHI T., MATSUOKA S., TANAKA. H.: Teddy: A sketching interface for 3d freeform design. In *SIGGRAPH 99 Conference Proceedings* (1999).

[IOOI05] IJIRI T., OKABE M., OWADA S., IGARASHI T.: Floral diagrams and inflorescences: Interactive flower modeling using botanical structural constraints. *ACM Transactions on Graphics (Proc. of SIGGRAPH '05) 24*, 3 (2005), 720–726.

[PJBF03] PEREIRA J. P., JORGE J. A., BRANCO V. A., FERREIRA F. N.: Calligraphic interfaces: Mixed metaphors for design. In *Interactive Systems: Design, Specification and Verification, DSV-IS 2003 Proceedings* (2003), pp. 154–170.

[TDM01] TOLBA O., DORSEY J., MCMILLAN L.: A projective drawing system. In *ACM Symposium on Interactive 3D Graphics* (2001).

[ZHH96] ZELEZNIK R., HERNDON K., HUGHES J.: Sketch: An interface for sketching 3d scenes. In *SIGGRAPH 96 Conference Proceedings* (1996).