Multi-Scale Description of Space Curves and Three-Dimensional Objects

Farzin Mokhtarian

Technical Report 87-38

November 1987

Department of Computer Science
University of British Columbia
Vancouver, B.C.
Canada V6T 1W5

Abstract

This paper addresses the problem of representing the shape of three-dimensional or space curves. This problem is important since space curves can be used to model the shape of many three-dimensional objects effectively and economically. A number of shape representation methods that operate on two-dimensional objects and can be extended to apply to space curves are reviewed briefly and their shortcomings discussed.

Next, the concepts of curvature and torsion of a space curve are explained. The curvature and torsion functions of a space curve specify it uniquely up to rotation and translation. Arc-length parametrization followed by Gaussian convolution is used to compute curvature and torsion on a space curve at varying levels of detail. Larger values of the scale parameter of the Gaussian bring out more basic features of the curve. Information about the curvature and torsion of the curve over a continuum of scales are combined to produce the curvature and torsion scale space images of the curve. These images are essentially invariant under rotation, uniform scaling and translation of the curve and are used as a representation for it. Using this representation, a space curve can be successfully matched to another one of similar shape.

The application of this technique to a common three-dimensional object is demonstrated. Finally, the proposed representation is evaluated according to several criteria that any shape representation method should ideally satisfy. It is shown that the curvature and torsion scale space representation satisfies those criteria better than other possible candidate methods.

I. Introduction

Most work done in computational vision on shape representation has focused either on planar curves and two-dimensional shapes [Hough 1962, Duda & Hart 1972, Ballard 1981, Pavlidis 1977, Mackworth & Mokhtarian 1984] or on three-dimensional objects and surfaces in 3-space [Brady *et al.* 1985, Faugeras & Ponce 1983, Weiss 1985]. This paper addresses the problem of describing the shape of three-dimensional curves. It is assumed that the curve is either directly computed from the image [Barnard & Pentland 1983, Watson & Shapiro 1982] or that it represents a surface reconstructed using stereo [Grimson 1985, Woodham 1984], "shape from" techniques [Ikeuchi & Horn 1981, Stevens 1982, Witkin 1981] or laser range finders [Faugeras *et al.* 1984].

Why study the problem of representing the shape of space curves? Space curves are useful to study for the following reasons:

a. Trajectories of objects in outer space and paths taken by atomic particles are space curves. Often, such an object or particle can be recognized by studying the shape of its path when subjected to specific forces.

b. Axes of generalized cones and cylinders [Agin & Binford 1973] are also space curves. A generalized cone or cylinder representation of a three-dimensional object can itself be efficiently represented by its axes.

c. Bounding contours of objects that consist of flat or nearly-flat surfaces are rich in information and can be used to represent the object effectively and economically. These bounding contours are space curves and can be extracted by *thinning* the object into lines and planes. An attempt to describe such objects using three-dimensional surfaces may not add much useful information but can significantly increase storage and processing requirements.

II. Criteria for a reliable representation

A reliable representation in computational vision should make it possible to obtain a reliable measure of the degree of match between two given objects. Various shape representation criteria have been proposed in [Marr & Nishihara 1978, Mokhtarian & Mackworth 1986, Mackworth 1987, Woodham 1987]. One set of criteria is presented here so that the shape representation proposed in this paper can be evaluated according to them in a later section.

a. **Efficiency:** The representation should be computable efficiently.

b. **Invariance:** Uniform scaling, rotation and translation are the transformations which do not alter the shape of an object. The representation should be invariant under these transformations.

c. Sensitivity: The degree of change to an object should correspond to the degree of the resulting change in its representation.

d. Uniqueness: There should be a one-to-one correspondence between objects and their representations. This requirement is only up to the class induced by criterion b above.

e. Detail: The representation should contain information about the object at varying levels of detail. This is important since features on an object usually exist at different scales.

f. Robustness: Arbitrary initial choices should not affect the representation. Furthermore, incomplete data should only change the representation locally.

Several shape description methods for planar curves exist which can be extended to apply to space curves as well. The Hough transform [Hough 1962] has been used to detect lines, circles and arbitrary two-dimensional shapes in images, and may be extended to detect space curves but such an extension would involve an explosion in the parameter space and would not meet requirements **a**, **b** and **e**. Chain encoding [Freeman 1974] and polygonal approximations [Pavlidis 1977] can be extended as representations for space curves. These methods don't satisfy requirements **b**, **d** and **e**. Fourier descriptors [Persoon & Fu 1974] may also be used. Criteria **e** and **f** are not satisfied by this class. Shape factors and quantitative measurements [Danielsson 1978] can be used to describe space curves but these methods involve a substantial reduction in data and do not meet conditions **c**, **d** and **e**. Splines [ballard & Brown 1982] can also be used to represent space curves. This method does not satisfy requirements **b**, **d**, **e** and **f**. Another class of methods is strip trees [Ballard & Brown 1982]. Hierarchical straight line approximations to a space curve can be obtained using an extension of these methods. However they don't meet criteria **b**, **c** and **f**. A final method is [Asada & Brady 1986]. They use a limited number of well-defined shape primitives which can be approximated well by analytical functions. The extension of this method to space curves would require a large number of primitives and would violate criteria **a** and **d**.

III. Multi-scale description of space curves

This section introduces the parametric representation of space curves and describes the Frenet Trihedron for space curves. Curvature and torsion of a space curve are then defined and geometrical interpretations given to them. Next it is shown how to compute curvature and torsion on a space curve at varying levels of detail. A multi-scale representation for a space curve which combines information about the curvature and torsion of the curve at varying levels of detail is then presented.

A. The parametric representation of a space curve

A space curve is the image of an interval under a continuous locally one-to-one mapping into the 3-space [Goetz 1970]. Therefore the set of points of a space curve are the values of the position vectors of the continuous vector-valued, locally one-to-one function:

$$\mathbf{r} = \mathbf{r}(u) = \mathbf{r}(x(u),\, y(u),\, z(u))$$

where $x(u)$, $y(u)$ and $z(u)$ are the components of $\mathbf{r}(u)$ and u is a monotonic function of arc-length s of the curve. s is also called the *natural parameter*. The function $\mathbf{r}(u)$ or the triple of functions $(x(u), y(u), z(u))$ is called a *parametric representation* of the curve.

B. The Frenet Trihedron and Frenet formulas for a space curve

With every point P of a space curve of class C_2 is associated an orthonormal triple of unit vectors: the tangent vector $\mathbf{t}$, the principal normal vector $\mathbf{n}$ and the binormal vector $\mathbf{b}$ (Figure 1). The *osculating plane* at P is defined to be the plane with the highest order of contact with the curve at P. The principal normal vector is the unit vector normal to the curve at P which lies in the osculating plane. The binormal vector is the unit vector perpendicular to the osculating plane such that the three vectors $\mathbf{t}$, $\mathbf{n}$ and $\mathbf{b}$ in that order form a positively oriented triple. The plane containing $\mathbf{t}$ and $\mathbf{n}$ is the osculating plane. The one containing $\mathbf{n}$ and $\mathbf{b}$ is the *normal plane* and the one containing $\mathbf{b}$ and $\mathbf{t}$ is the *rectifying plane*.

The derivatives of $\mathbf{t}$, $\mathbf{n}$ and $\mathbf{b}$ with respect to the arc-length parameter give us:

$$\frac{d\mathbf{t}}{ds} = \kappa\mathbf{n}, \qquad \frac{d\mathbf{n}}{ds} = -\kappa\mathbf{t} + \tau\mathbf{b}, \qquad \frac{d\mathbf{b}}{ds} = -\tau\mathbf{n}.$$

These formulas are called the *Frenet* or the *Serret-Frenet* formulas. The coefficients κ and τ are called the *curvature* and *torsion* of the curve respectively.

Curvature is the instantaneous rate of change of the tangent vector to the curve with respect to the arc length parameter. There is no interpretation for the sign of curvature. Torsion is the instantaneous rate of change of the osculating plane with respect to the arc length parameter. A sign is assigned to the absolute measure of torsion as following:

Let point P correspond to value s of the arc length parameter and let point Q correspond to value $s+h$. Let line l be the intersection of the osculating planes at P and Q. Give line l the orientation of a vector $\mathbf{w}$ on l such that $\mathbf{t}.\mathbf{w}>0$. Consider the rotation about l through a non-obtuse angle which superposes the osculating plane at P on the osculating plane at Q. This rotation also superposes $\mathbf{b}(s)$ on $\mathbf{b}(s+h)$. If $\mathbf{b}(s)$, $\mathbf{b}(s+h)$ and $\mathbf{w}$ form a positively oriented triple, then torsion has positive sign, otherwise it has negative sign.

C. Computing curvature and torsion of a curve at varying levels of detail

Since the curve is represented in parametric form, in order to compute curvature and torsion at each point on the curve, we need to express those quantities in terms of the derivatives of $x(.)$, $y(.)$ and $z(.)$. In what follows, $\mathbf{r}(u)$ represents the parametrization of a space curve with respect to an arbitrary parameter and $\rho(s)$ represents the parametrization of that curve with respect to the arc-length parameter.

C.1. Curvature

In case of an arc-length parametrization, we simply have:

$$\kappa = |\ddot{\rho}|$$

In coordinate form

$$\kappa = \sqrt{\ddot{x}^2 + \ddot{y}^2 + \ddot{z}^2}$$

Given an arbitrary parametrization of the curve:

$$\kappa = |\mathbf{t}_s| = \frac{|\mathbf{t}_u|}{|\dot{\mathbf{r}}|} = \frac{\left|\frac{d}{du}(\dot{\mathbf{r}}/|\dot{\mathbf{r}}|)\right|}{|\dot{\mathbf{r}}|} = \frac{|\dot{\mathbf{r}} \mathrm{x} \ddot{\mathbf{r}}|}{|\dot{\mathbf{r}}|^3}$$

In coordinate form

$$\kappa = \frac{\sqrt{A^2 + B^2 + C^2}}{(\dot{x}^2 + \dot{y}^2 + \dot{z}^2)^{3/2}}. \tag{1}$$

where

$$A = \begin{vmatrix} \dot{y} & \dot{z} \\ \ddot{y} & \ddot{z} \end{vmatrix} \quad B = \begin{vmatrix} \dot{z} & \dot{x} \\ \ddot{z} & \ddot{x} \end{vmatrix} \quad C = \begin{vmatrix} \dot{x} & \dot{y} \\ \ddot{x} & \ddot{y} \end{vmatrix}.$$

C.2. Torsion

We will first derive an expression for the torsion of a space curve with arc-length parametrization. Multiplying both sides of the third Frenet formula by $\mathbf{n}$ results in

$$\tau = -\mathbf{b}_s\mathbf{n} = -(\mathbf{t}\mathrm{x}\mathbf{n})_s\mathbf{n} = -(\mathbf{t}_s\mathrm{x}\mathbf{n})\mathbf{n} - (\mathbf{t}\mathrm{x}\mathbf{n}_s)\mathbf{n} = \mathbf{t}\mathbf{n}\mathbf{n}_s$$

Note that $\mathbf{tnn}_s$ is the *mixed product* of vectors $\mathbf{t}$, $\mathbf{n}$ and $\mathbf{n}_s$ and is equal to $(\mathbf{t}\mathrm{x}\mathbf{n})\mathbf{n}_s$. We now make use of

$$\mathbf{t} = \dot{\rho}, \quad \mathbf{n} = \frac{\ddot{\rho}}{|\kappa|}, \quad \mathbf{n}_s = \frac{\dddot{\rho}}{\kappa} - \frac{\kappa_s}{\kappa^2}\ddot{\rho}$$

to obtain

$$\tau = \frac{\dot{\rho}\,\ddot{\rho}\,\dddot{\rho}}{\kappa^2} = \frac{\dot{\rho}\,\ddot{\rho}\,\dddot{\rho}}{\ddot{\rho}^2}$$

In coordinate form

$$\tau = \frac{\begin{vmatrix} \dot{x} & \dot{y} & \dot{z} \\ \ddot{x} & \ddot{y} & \ddot{z} \\ \dddot{x} & \dddot{y} & \dddot{z} \end{vmatrix}}{\ddot{x}^2 + \ddot{y}^2 + \ddot{z}^2}.$$

In case of an arbitrary parametrization, we make use of:

$$\dot{\rho} = \dot{\mathbf{r}}\frac{dt}{ds}, \qquad \ddot{\rho} = \ddot{\mathbf{r}}\left(\frac{dt}{ds}\right)^2 + \dot{\mathbf{r}}\frac{d^2t}{ds^2},$$

and

$$\dddot{\rho} = \dddot{\mathbf{r}}\left(\frac{dt}{ds}\right)^3 + 3\ddot{\mathbf{r}}\left(\frac{dt}{ds}\right)\frac{d^2t}{ds^2} + \dot{\mathbf{r}}\frac{d^3t}{ds^3}$$

to obtain

$$\tau = \frac{\dot{\mathbf{r}}\,\ddot{\mathbf{r}}\,\dddot{\mathbf{r}}}{|\dot{\mathbf{r}}|^6} \cdot \frac{|\dot{\mathbf{r}}|^6}{(\dot{\mathbf{r}} \times \ddot{\mathbf{r}})^2} = \frac{\dot{\mathbf{r}}\,\ddot{\mathbf{r}}\,\dddot{\mathbf{r}}}{(\dot{\mathbf{r}} \times \ddot{\mathbf{r}})^2}$$

In coordinate form

$$\tau = \frac{\begin{vmatrix} \dot{x} & \dot{y} & \dot{z} \\ \ddot{x} & \ddot{y} & \ddot{z} \\ \dddot{x} & \dddot{y} & \dddot{z} \end{vmatrix}}{A^2 + B^2 + C^2} \qquad (2)$$

where A, B and C are as before.

C.3. Curvature and torsion at varying levels of detail

In order to compute κ and τ at varying levels of detail of the curve Γ, functions $x(u)$, $y(u)$ and $z(u)$ are convolved with a Gaussian kernel $g(u,\sigma)$ of width σ [Marr & Hildreth 1980]:

$$g(u,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{u^2}{2\sigma^2}}$$

The convolved functions together define the *evolved* curve Γ_σ. The convolution of a function $f(u)$ and the Gaussian kernel is defined as:

$$F(u,\sigma) = f(u) \circledast g(u,\sigma) = \int_{-\infty}^{\infty} f(v) \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(u-v)^2}{2\sigma^2}} dv$$

Furthermore, it is known that

$$\dot{F}(u,\sigma) = \frac{\partial F(u,\sigma)}{\partial u} = f(u) \circledast \left[\frac{\partial g(u,\sigma)}{\partial u} \right]$$

$$\ddot{F}(u,\sigma) = \frac{\partial^2 F(u,\sigma)}{\partial u^2} = f(u) \circledast \left[\frac{\partial^2 g(u,\sigma)}{\partial u^2} \right]$$

and

$$\dddot{F}(u,\sigma) = \frac{\partial^3 F(u,\sigma)}{\partial u^3} = f(u) \circledast \left[\frac{\partial^3 g(u,\sigma)}{\partial u^3} \right].$$

These properties of convolution can be used to compute curvature and torsion on evolved versions of a space curve. Note that since the arc-length parameter s on a space curve is in general *not* the arc-length parameter on the evolved curve [Mackworth & Mokhtarian 1987], the most general expressions for κ and τ (formulae (1) and (2)) must be used. Figure 2 shows a space curve representing the shape of an armchair. Figure 3 shows an application of this method to that curve. If it is desired to describe an object at a specific scale, this method can be used to obtain that description.

D. A multi-scale representation for space curves

The curvature and torsion functions of a space curve specify that curve uniquely up to rotation and translation [Do Carmo 1976]. We therefore propose a representation for a space curve that consists of the *curvature scale space* and *torsion scale space* images of the curve. This representation is a generalization of the curvature scale space representation proposed for planar curves in [Mokhtarian & Mackworth 1986]. The scale space image was first proposed as a representation for one-dimensional signals in [Stansfield 1980] and developed in [Witkin 1983].

We first explain an algorithm to compute the torsion scale space image of a space curve and then show how the curvature scale space image of that curve can be computed by a slight modification of the algorithm.

Algorithm: torsion scale space:

1. Let σ, the width of the Gaussian mask used, be equal to σ_0, a small positive value.

2. Add a small positive value $\Delta\sigma$ to σ.

3. Compute masks representing the first three derivatives of the Gaussian function with σ as its width.

4. Convolve each of the three masks computed in step 3 with each of the coordinate functions of the curve.

5. Combine the results obtained in step 4 according to formula (2) for torsion to compute its value at every point on the evolved curve. Store the results in an array.

6. Scan the array of torsion values computed in step 5 to find the zero-crossing points. A zero-crossing point is between two adjacent values in the array with one of those values positive and the other one negative. Maintain a count of the number of zero-crossing points found. If this count is equal to zero, then STOP.

7. Mark each of the zero-crossing points discovered in step 6 in a coordinate system in which the horizontal axis represents u, the parameter along the curve and the vertical axis represents σ, the width of the Gaussian mask. As a result, each zero-crossing point p will have coordinates u_P and σ_P where u_P is the value of u at point P and σ_P is the value computed for σ in step 2.

8. Go to step 2.

End of Algorithm: torsion scale space.

The two-dimensional array computed by this algorithm is the torsion scale space image of the space curve. Figure 5 shows the torsion scale space image of the chair of figure 2. Note that this image is much more structured than the original curve.

The curvature scale space image of a space curve can also be constructed using an algorithm almost identical to the one described above. The only difference is that level-crossings rather than zero-crossings are searched for. This is because the curvature of a space curve has only magnitude and no sign.

Some care should be given to choosing a suitable value for level L. A first approximation to L is the average of the curvature values of all the sampled points on curve Γ. However, if this approximation is used, the number of level-crossing points found on curves Γ_σ drops quickly to zero as σ increases and the resulting curvature scale space image will not be very rich and therefore not suitable for matching purposes.

Therefore the actual value used for L is the average of curvature values of points of Γ_{σ_0} where $\sigma_0 \in [0,\sigma_t]$ and σ_t is the value of σ where the number of zero-crossings first drops to zero in the torsion scale space image of Γ. Using such a value ensures that the resulting curvature scale space image will be sufficiently rich for matching purposes and will represent roughly the same range of values of σ represented in the torsion scale space image of Γ. Figure 4 shows the curvature scale space image of the chair computed using this method for selecting a value for L.

Note that the curvature and torsion scale space images of a space curve can be *renormalized* using the procedure described in [Mackworth & Mokhtarian 1987]. In the renormalized curvature or torsion scale space image, the parameter u is always the arc-length parameter of the evolved curve.

In order to match a space curve against another, the torsion scale space images of both are constructed and matched against each other using the algorithm described in [Mokhtarian & Mackworth 1986]. If the resulting cost of match is low, then one curve is transformed according to the transformation parameters predicted by the match so that both curves exist at the same scale. The curvature scale space images of both curves are then constructed and matched using the same algorithm. The final cost of match is a combination of the two costs.

IV. Discussion

If the curve to be represented is closed, then its coordinate functions are assumed to be periodic. This eliminates all edge-based problems during computation of convolutions.

The computation of a multi-scale representation for a space curve involves computing derivatives of functions. While this process may be sensitive to noise in the raw data, it is quite stable for a slightly evolved version of the curve.

We can now evaluate our space curve representation method according to the criteria set forth in section II:

Criterion a: Efficiency

The construction of the curvature and torsion scale space images typically involves the computation of a large number of convolutions. Convolutions involving Gaussians of large widths can be expressed in terms of convolutions involving Gaussians of small widths only. Since the filters used to approximate these functions are also small, a significant reduction in computation time can be achieved in this way. An alternative way to render the computation efficient is to use Fast Fourier Transforms.

Criterion b: Invariancy

The torsion scale space image consists of zeroes of torsion at varying levels of detail and is therefore essentially invariant under rotation, uniform scaling and translation of the curve. The curvature scale space image consists of curvature level-crossings and in invariant under rotation and translation of the curve but not under uniform scaling of it. Since the torsion scale space image is used for initially matching a pair of space curves, this is not a shortcoming of this representation.

Criterion c: Sensitivity

Small changes to the shape of the curve usually result in small changes in its representation since smaller values of the scale parameter will be sufficient to smooth out the change.

Criterion d: Uniqueness

It has been shown by Yuille and Poggio [1983] that almost all signals can be reconstructed up to an equivalence class from their scale space images. Although this property has not been proven for the curvature and torsion scale space images, these images are very rich in information and very unlikely to be identical for two curves of different shapes.

Criterion e: Detail

Since the representation combines information about the curve at varying levels of detail, criterion e is also satisfied.

Criterion f: Robustness

If the curve is closed, an arbitrary starting point should be chosen on the curve for parametrization. This only results in a horizontal shift in the scale space image but will not change its structure. If the curve is open, the natural starting point is one of its endpoints. Incomplete data near the endpoints does change the structure of the scale space image but the change is local.

V. Conclusions

The problem of finding a representation for space curves was addressed in this paper and a number of criteria for any solution method were proposed. A technique for describing a space curve at varying levels of detail was developed and a multi-scale representation based on that technique was proposed. It was shown that the proposed representation satisfies those criteria better than other possible candidate methods.

Acknowledgment

I am grateful to Dr. Alan Mackworth for helpful discussions. This work was supported by a graduate scholarship from the University of British Columbia.

References

Agin, G.J. and T.O. Binford, "Computer description of curved objects," *IJCAI-73*, pp. 629-640, Palo Alto, CA, 1973.

Asada, H. and M. Brady, "The curvature primal sketch," *IEEE PAMI*, vol. 8, pp. 2-14, 1986.

Ballard, D.H., "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, pp. 111-122, 1981.

Ballard, D.H. and C.M. Brown, *Computer vision,* Prentice Hall, Englewood Cliffs, NJ, 1982.

Barnard, S.T. and A.P. Pentland, "Three-dimensional shape from line drawings," *Proc. Eight International Joint Conference on Artificial Intelligence*, pp. 1062-1064, Vancouver, B.C., 1983.

Brady, M., J. Ponce, A.L. Yuille, and H. Asada, "Describing surfaces," AI-Memo-822, MIT-AI, Cambridge, MA, 1985.

Danielsson, P.E., "A new shape factor," *CGIP*, vol. 7, pp. 292-299, 1978.

DoCarmo, M., *Differential geometry of curves and surfaces,* Prentice-Hall, Englewood Cliffs, NJ, 1976.

Duda, R.O. and P.E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *CACM*, vol. 15, pp. 11-15, 1972.

Faugeras, O.D., *et al.* , "Object representation identification and positioning from range data," in *First International Symposium on Robotics Research*, ed. R. Paul, pp. 425-446, MIT Press, 1984.

Faugeras, O.D. and J. Ponce, "Prism trees: a hierarchical representation for 3-D objects," *Proc. Eight International Joint Conference on Artificial Intelligence*, pp. 982-988 , Vancouver, B.C., 1983.

Freeman, H., "Computer processing of line drawing images," *ACM-CS*, vol. 6, pp. 57-97, 1974.

Goetz, A., *Introduction to differential geometry,* Addison Wesley, Reading, MA, 1970.

Grimson, W.E.L., "Computational experiments with feature based stereo algorithm," *IEEE-PAMI*, vol. 7, pp. 24-27, 1985.

Hough, P.V.C., "Method and means for recognizing complex patterns," U.S. patent 3 069 654, 1962.

Ikeuchi, K. and B.K.P. Horn, "Numerical shape from shading and occluding boundaries," *Artificial Intelligence*, vol. 17, pp. 141-184, 1981.

Kecs, W., *The convolution product and some applications,* D. Reidel Publishing Co., Boston, MA, 1982.

Mackworth, A. K., "Update on computational vision: shape representation, object recognition and constraint satisfaction," Technical Report 87-29, Department of Computer Science, University of British Columbia, Vancouver, BC, 1987.

Mackworth, A. K. and F. Mokhtarian, "Scale-based description of planar curves," *CSCSI-84*, pp. 114-119, London, Ont., 1984.

Mackworth, A. K. and F. Mokhtarian, "The renormalized curvature scale space and the evolution properties of planar curves," Technical Report 87-37, Department of Computer Science, University of British Columbia, Vancouver, BC, 1987.

Marr, D. and E.C. Hildreth, "Theory of edge detection," *Proc. R. Soc. Lond. B*, vol. 207, pp. 187-217, 1980.

Marr, D. and H. K. Nishihara, "Representation and recognition of the spatial organization of 3D structures," *Proc. R. Soc. Lond. B*, vol. 200, pp. 269-294, 1978 .

Mokhtarian, F. and A. K. Mackworth, "Scale-based description and recognition of planar curves and two-dimensional shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, pp. 34-43, 1986.

Pavlidis, T., "Polygonal approximations by Newton's method," *IEEE-C*, vol. 26, pp. 800-807, 1977.

Persoon, E. and K.S. Fu, "Shape discrimination using Fourier descriptors," *Proc. Second International Joint Conference on Pattern Recognition*, pp. 126-130, Copenhagen, Denmark, 1974.

Stansfield, J.L., "Conclusions from the commodity expert project," AI-Memo-601, MIT-AI, Cambridge, MA, 1980.

Stevens, K.A., "Implementation of a theory for inferring surface shape from contours," AI-Memo-676, MIT-AI, Cambridge, MA, 1982.

Watson, L.T. and L.G. Shapiro, "Identification of space curves from two-dimensional perspective views," *IEEE Pattern Analysis and Machine Intelligence*, vol. PAMI-4, pp. 469-475, 1982.

Weiss, I., "3-D shape representation by contours," *Proc. Ninth International Joint Conference on Artificial Intelligence*, pp. 969-972, Los Angeles, CA, 1985.

Witkin, A.P., "Recovering surface shape and orientation from texture," *Artificial Intelligence*, vol. 17, pp. 17-45, 1981.

Witkin, A.P., "Scale space filtering," *IJCAI-83*, pp. 1019-1023, Karlsruhe, W. Germany, 1983.

Woodham, R.J., "Photometric method for determining shape from shading," in *Image Understanding 1984*, ed. W. Richards, pp. 97-125, Ablex Publishing Corp., Norwood, NJ, 1984.

Woodham, R. J., "Stable representation of shape," Technical Report 87-5, Department of Computer Science, University of British Columbia, Vancouver, BC, 1987.

Yuille, A.L. and T. Poggio, "Fingerprint theorems for zero crossings," AI-Memo-730, MIT-AI, Cambridge, MA, 1983.

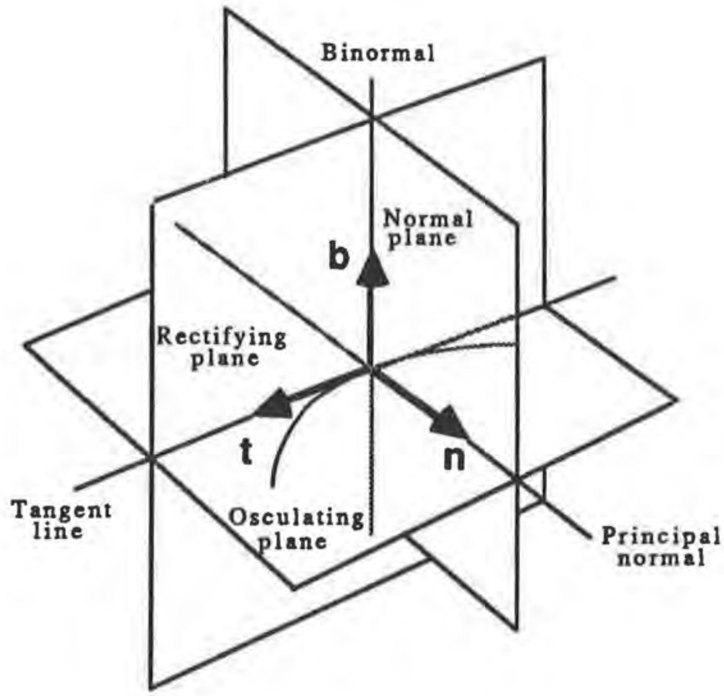


Figure 1. The Frenet trihedron for a space curve.

Figure 2. A space curve representing an armchair.

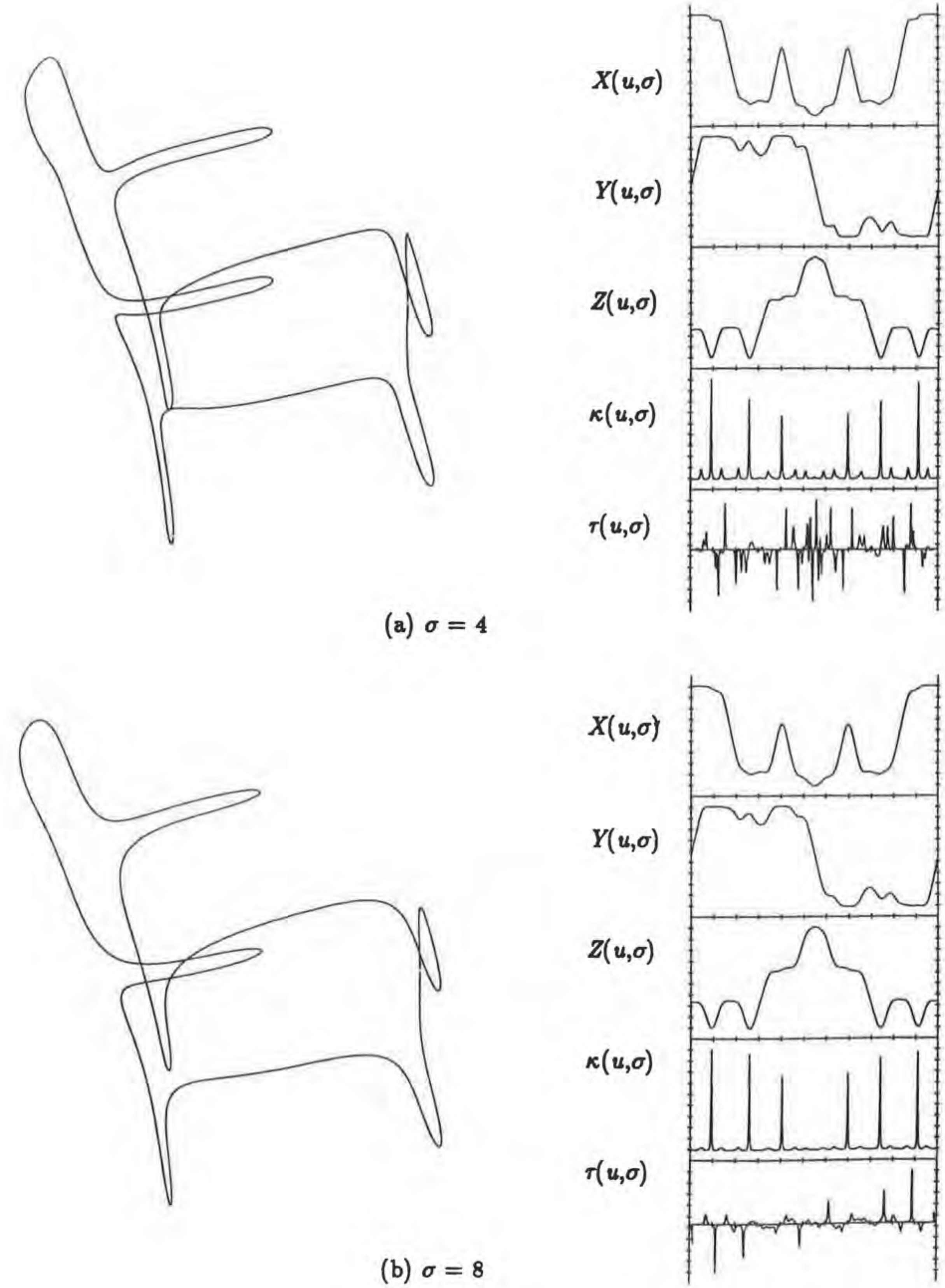
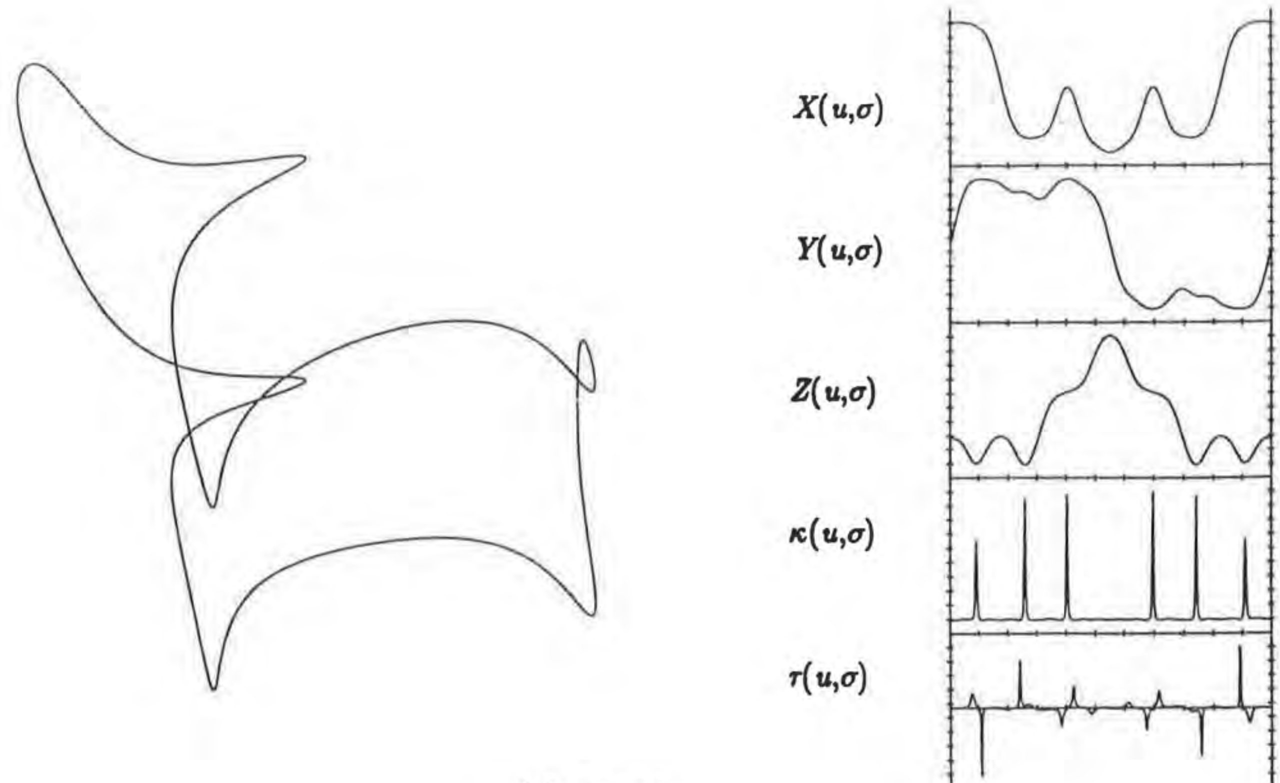
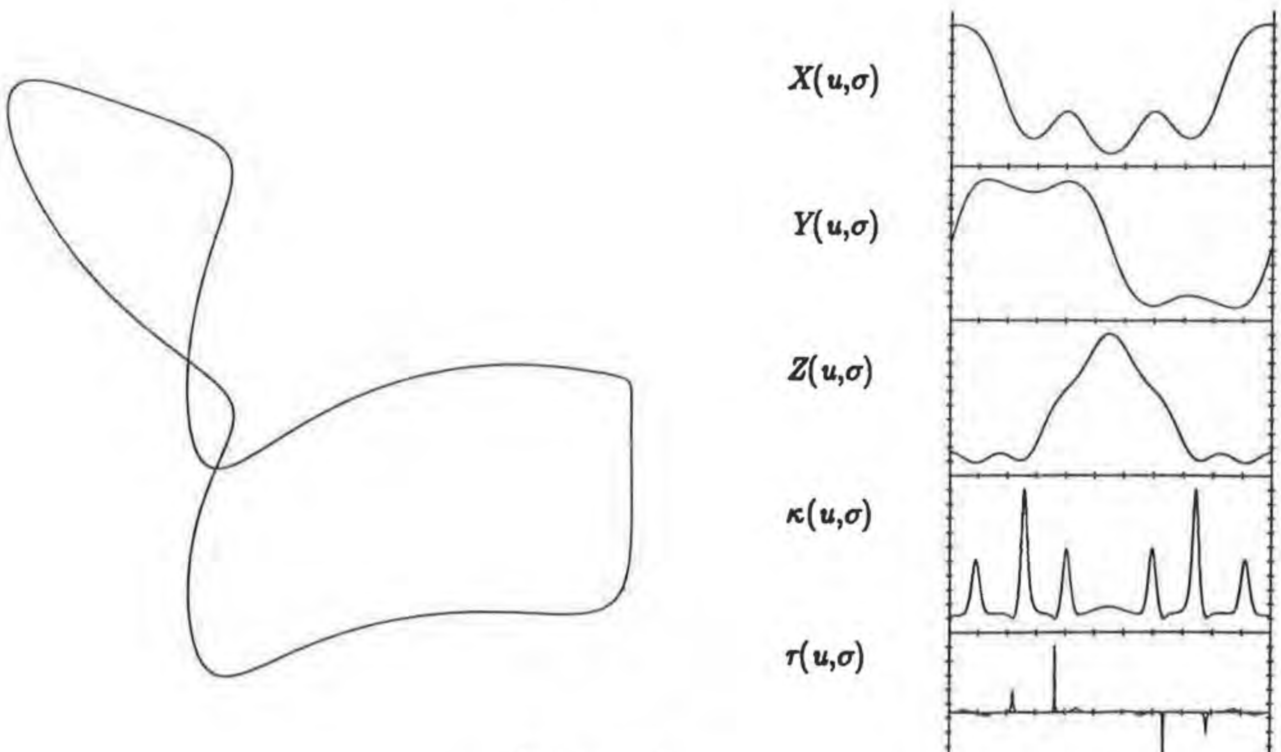


Figure 3. Multi-scale description of the chair.

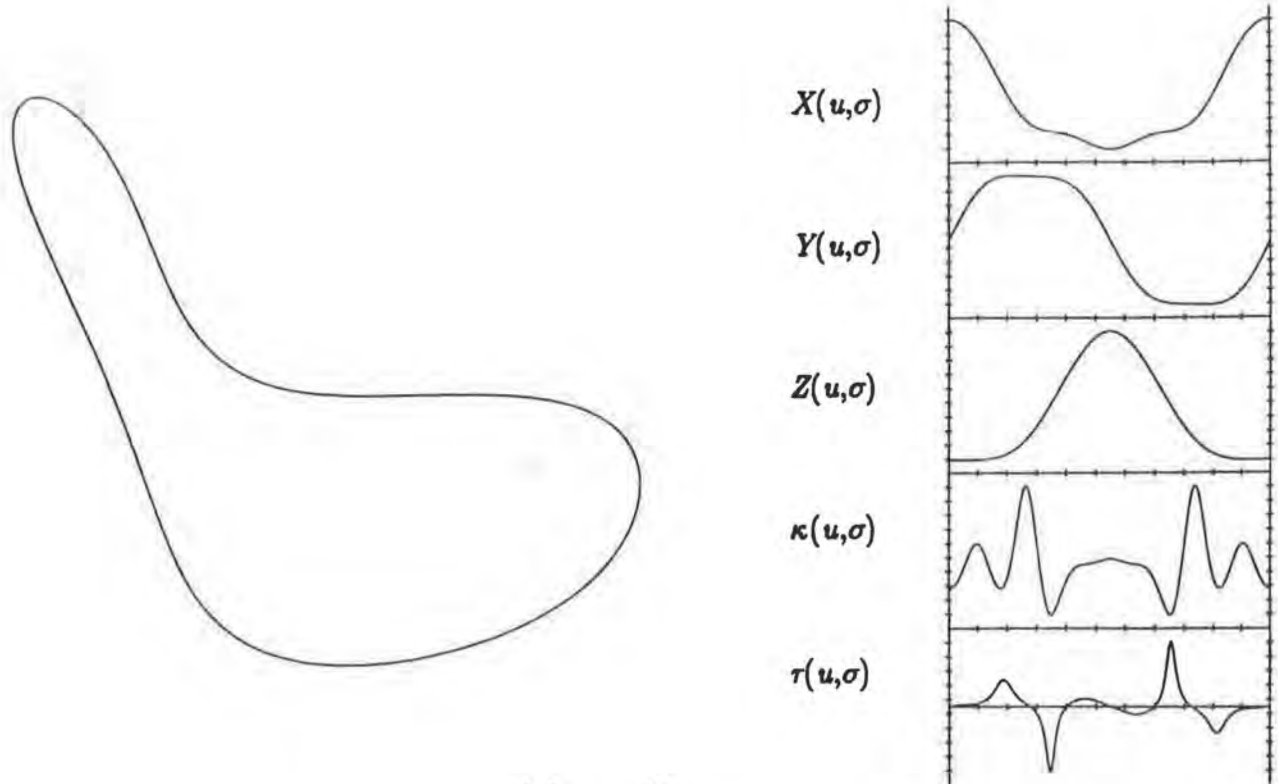


(a) $\sigma = 16$

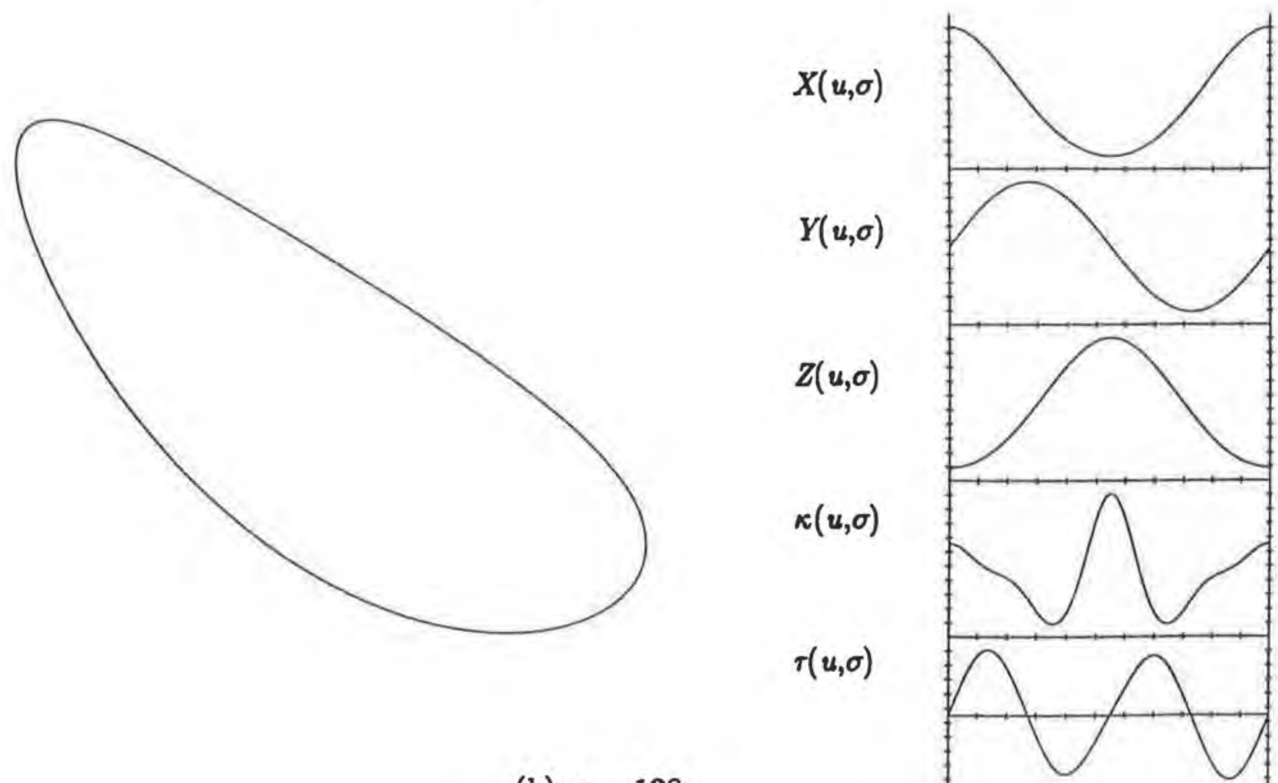


(b) $\sigma = 32$

Figure 3. (Continued) Multi-scale description of the chair.



(a) $\sigma = 64$



(b) $\sigma = 128$

Figure 3. (Continued) Multi-scale description of the chair.

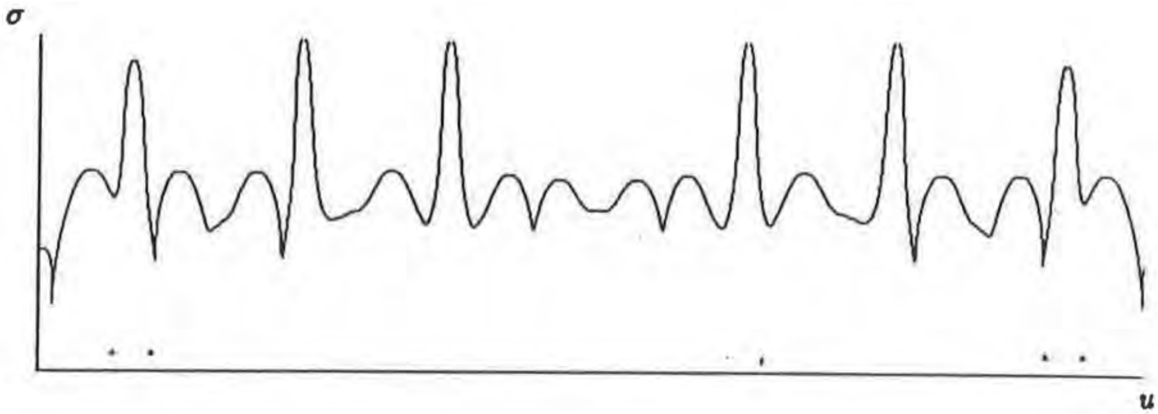


Figure 4. The Curvature Scale Space Image of the chair.

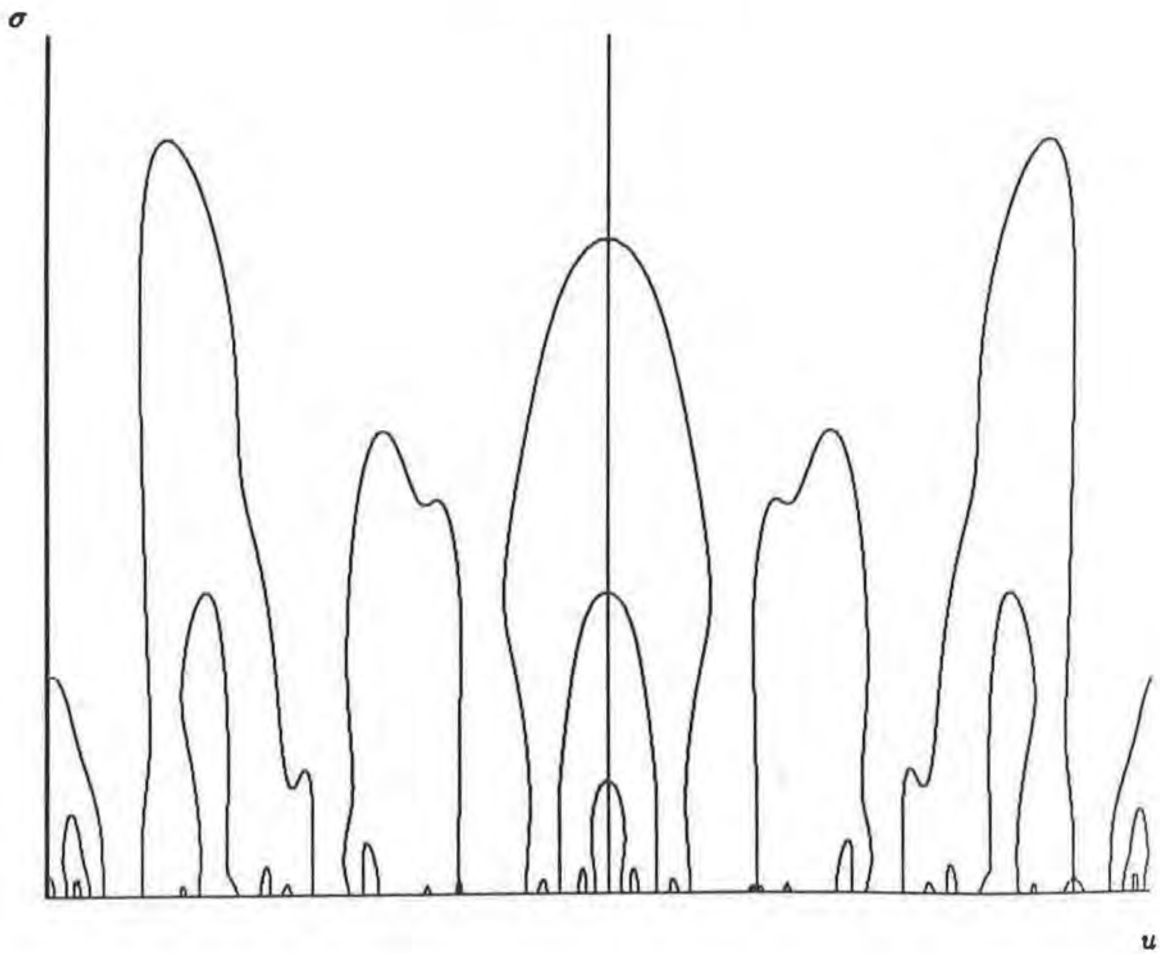


Figure 5. The Torsion Scale Space Image of the chair.