AVC89: SURFACE DESCRIPTION BY LABELLED GRAPHS

CHRISTOPHER J. DAVIES

Computer Science Department
Queen Mary College
Mile End Road
London
E1 4NS

The aim of this paper is to propose a new way of describing three dimensional shape. The shapes to be described are non rigid shapes, like those of animal cells, which have no well defined shape. The shape descriptors are based upon Leyton's symmetry axis description for smooth contours. Like Leyton's contour description, a process grammar is proposed to allow the description of the change of shape over time,but here we can also deal with surfaces. We can extract these description from conventional three dimensional models through finding either the symmetry set of the object or the evolute. Initial computational schemes are described along with initial results for contours using two differing wave methods on an AMT DAP which is a SIMD parallel machine.

There are many ways available to describe surfaces in Computer Vision. The use of surface patches and CSG models have been imported from computer graphics with differing amounts of success. The generalised cylinder has been used extensively [1] for shape description, especially when a hierarchical shape model is being used. Superquadrics have also been used to model biological shape [2] due to the flexibility of the model primitives and the possibility of avoiding low level processing by extracting the model directly from the surface illuminance. All of the above shape descriptors are based upon representing shape in terms of primitives which have been glued together in some fashion. They can only be used to describe shapes which vary over time with considerable difficulty unless the shapes are composed of jointed sub-shapes. The aim of this document is to propose a global shape descriptor which will allow non jointed shape change over time in a manner similar to that proposed by Koenderink [3].

The description is based upon Leyton's Process Grammar [4] which is a way to describe the shapes of curves and how one curve can be deformed into another. The aim is to extend this description from curves to surfaces. To do this it will be necessary to simplify the description and initially, to ignore the links between the description and the symmetry axes that motivated the description. This is because, while it is clear what is mean by a force acting along a line, it is not obvious what is meant by a force acting through a sheet. To compute symmetry sets necessary for th descriptions, we need to obtain the description from low level surface models. Here it is done by running a wave process. This can either be done by using Fourier transforms or by using Laplacian masks [5] as described in the section on implementation. First, however, we describe the development of the representation scheme

A NEW WAY TO VIEW LEYTON'S PROCESS GRAMMAR

Leyton's curve descriptions are framed in term of local symmetry axes. However the descriptions do not use any facts about symmetry axes except that each symmetry axis terminates at a unique curvature extrema [6] and that for each curvature extrema there is a unique symmetry axis. The labelling of the different types of symmetry axis is derived from the properties of the associated curvature extrema. This means that it is possible to express Leyton's curve descriptions in terms of curvature extrema rather than in terms of the local symmetry axes.

The curves are represented by graphs consisting of nodes and edges where the nodes are associated with points on the curve which are curvature extrema. Two nodes are connected by an edge if the two curvature extrema are connected by a segment of the curve which contains no other curvature extrema. This means that all the graphs will be cyclic graphs. The nodes are labeled, and carry the same label as the associated point on the curve. These can be expressed in the following way: the node

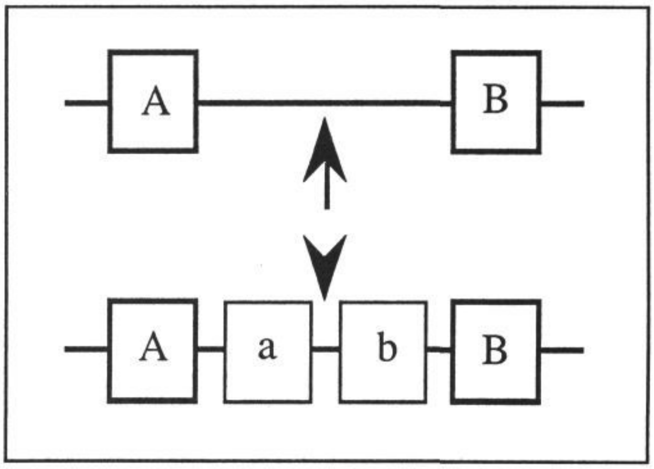

Figure 1: The Continuation Rules for Contour Descriptors

AVC 1989 doi:10.5244/C.3.11

label set is $\mathcal{N}=\mathcal{E}\times\mathcal{S}$ where $\mathcal{E}$={Maximum,Minimum} , and $\mathcal{S}$={Positive,Negative} where $\mathcal{E}$ represents whether the curvature extrema is a maximum or a minimum, and $\mathcal{S}$ represents the sign of the curvature at the relevant curvature extrema.

The rewrite rules that Leyton produces are based on reasoning about basic perceptual psychology. The argument is that the local symmetry axes can be viewed as lines along which the shape has been deformed. This will be ignore here because what is being considered are all the general ways that a shape can be deformed. While Leyton presents 2 continuation rules, and 4 bifurcation rules, what is presented here are 4 continuation rules,16 introduction rules, and 16 destruction rules which are expressed as three families of rules. To make this possible we will first define some functions that we can apply to N. These rely on defining the negation operator for any set containing only two elements ie Maximum = -(Minimum), Positive= -(Negative) .

We can now define the two functions:

$\alpha:\mathcal{N}(a,b)\rightarrow\mathcal{N}(-a,b)$	Change extrema
$\beta:\mathcal{N}(a,b)\rightarrow\mathcal{N}(a,-b)$	Change sign

Table 1: Node Functions

There are also two equivalence relations which must be defined, extrema equivalence and sign equivalence.

$[(a,b)\approx_e(c,d)] \Leftrightarrow [\, a=c]$	Same Extrema
$[(a,b)\approx_s(c,d)] \Leftrightarrow [\, b=d\,]$	Same Sign

Table 2: Node Equivalence Classes

We are only considering general surfaces. Inflections of the curvature and maxima and minima which have zero curvature are not stable and so can be ignored. The four continuation rules can then be written as a single family of rules.

Continuation at node n: $n \rightarrow \beta(n)$

The bifurcation rules are rewritten as node introduction rules. Given two nodes joined by an edge labeled A and B the two nodes a and b are introduced into edge as shown in Figure 1.

Introduction of a,b between A,B possible only if:

$A=a(a)=b=a(B)$

This is a family of 16 separate introduction rules. The simplicity is gained from considering similarities between separatc changes. So for the introduction rules we write:

Introduction of ab in AB possible only if:

$AB \rightarrow AabB \Leftrightarrow (\, A\approx_e\alpha(a)=b\approx_e\alpha(B)\,)$

The last set of rules are the destruction rules which are simply the reverse of the introduction rules. Using Figure 1 again to explain we have:

Destruction of ab in AabB only possible if:

$AabB \rightarrow AB \Leftrightarrow (\, \alpha(a)=b\,)$

SURFACE DESCRIPTION BY LABELLED GRAPHS

Node Type	Unusual Ridge Point
P	Parabolic Ridge Point
F	Flat Ridge Point
H	Hyperbolic Ridge Point
E	Elliptic Ridge Point
X	Crossing of two Ridge Lines

Table 3: Surface descriptor Node types

In producing a three dimensional surface description the above method was extended. The vertices (curvature extrema) of the contours were replaced by the *Ridge lines* of the surface. The *Ridge lines* are composed of all the points which are curvature extrema of the lines of curvature of the surface[7][8]. There are two sets of *Ridge lines* because most surface points have two curvatures (maximum and minimum). Following Porteous lead we will colour these line red and blue to differentiate them. These lines can now be represented by a graph which is enscribed upon the surface. The *Ridge lines are* mapped onto edges of the graph while singular *Ridge Points* are mapped onto the nodes of the graph [9].

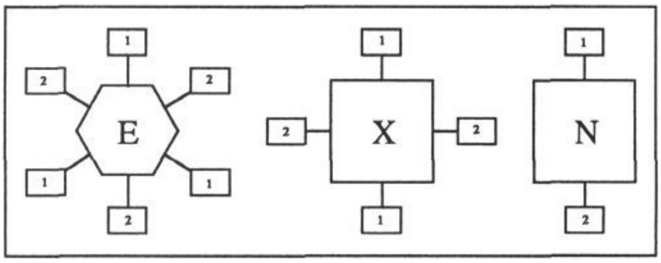

Figure 2: The Surface Graph Node Types

Both the edges and the nodes can now be labelled. The edges inherit their label from the *Ridge lines* that they represent (Red or Blue, Maximum or Minimum of curvature, Positive or Negative curvature). This is written as $\mathcal{L}=\mathcal{C}\times\mathcal{E}\times\mathcal{S}$. where $\mathcal{C}$= {R,B}, $\mathcal{E}$= {M,m} $\mathcal{S}$= {P,N}.

The nodes need no label, as all the information about them can be recovered from the edges which meet at the node. However, for convenience, the label they carry denotes the special point that they represent. The set of all node types is represented by $\mathcal{N}$={H,P,F,X,E}. Where these are defined in Table 3, and depicted in Figure 2. The node rules assume that L_1=(a,b,c) and L_2=(d,e,f) where the edge labels are depicted by the small rectangles. The node labels are the larger shapes and the label N is a meta-label which represents P, F and H. These node rules as depicted in Table 4.

Again as in the curve description functions and equivalence relations can be defined to aid in the description of both the

Node Type	Label Relationships		
P	a=d	b=-e	c=f
F	a=d	b=e	c=-f
H	a=-d	b=e=F	c=f
E	a=-d	b=e=F	c=f
X	a=-d		

Table 4:Surface Descriptor Label Relationships

surface graphs and the rewrite rules that can be imposed upon the surface graphs. The functions are from edge labels to edge labels. Below are the complete set the four functions that it is possible to define upon the edge labels using the negation operator.

$\mathbf{i}:\mathcal{L},(a,b,c) \rightarrow \mathcal{L},(a,b,c)$	Identity
$\alpha:\mathcal{L},(a,b,c) \rightarrow \mathcal{L},(-a,b,c)$	Change colour
$\beta:\mathcal{L},(a,b,c) \rightarrow \mathcal{L},(a,-b,c)$	Change extrema
$\gamma:\mathcal{L},(a,b,c) \rightarrow \mathcal{L},(a,b,-c)$	Change sign

Table 5: Edge Functions

Also three equivalence relations are defined upon $\mathcal{L}$ where (a,b,c) and (d,e,f) are two edge labels then;

$[(a,b,c)\approx_c(d,e,f)] \Leftrightarrow [\ a=d\ ]$	Same Colour
$[(a,b,c)\approx_e(d,e,f)] \Leftrightarrow [\ b=e\ ]$	Same Extrema
$[(a,b,c)\approx_s(d,e,f)] \Leftrightarrow [\ c=f\ ]$	Same Sign

Table 6: Edge Equivalence Classes

The enscription upon a surface allows a fourth and final relation to be applied to the edges of the graph Γ. If the graph is used to divide the surface in which it is embedded into regions, then two edges are neighbors if and only if they lie on the boundary to the same region. Unlike the other three relations this is not an equivalence relation and so cannot be used to divided up the edge set into separate partitions.

The rewrite rules were produced by considering the geometry of the graph structure. Only probable changes were considered. This means that if two features are introduced into a figure then they are introduced one at a time, and not both together.

As has been explained the node points represent special points on the ridge line. Here a link can be made to singularity theory. The ridge lines are the singular points of a distance map of the surface (evolute) projected back onto the surface. The node points are higher level singularities. To study the evolution of surfaces what is needed is a study of the evolution of distance maps over time. This will give a list of surface changes which have a non-zero probability of occurring. These are the changes that should be encapsulated in the rewrite rules which are provided for surfaces.

Unfortunately, this is not as easy as it sounds. The dimensions of the spaces being dealt with become very high. At these dimensions simple singularity theory breaks down as the algebra no longer exhibits a one to one correspondence with the topology. It may eventually be possible to prove these rewrite rules. At present they are viable suggestions which agree with what is known about the geometry of surfaces.

Swapping Rules

There are nine rewrite rules which involve the swapping of edges or nodes in the graph. The number of edges and the number of each type of labelled edge remains the same. This means that the two graphs, the new and the original, can be regarded as being as complex as each other. The introduction and destruction rules change the complexity of the graph.

The first four rewrite rules are all represented in the top box of Figure 3. These involve the swapping of two nodes which share a common edge. The node are swapped and the edge which lies in between them is relabelled so that $\mathbf{L_4= F(L_3)}$, where **F** is the label function defined in Table 7 and $\mathbf{L_l}$ represents the edge labels on Figure 3. It should be noted that these are the only swapping routines that can occur across a single edge.

The fifth rewrite rule **Sw** does not involve nodes at all. The four grey squares in Figure 3 represent any nodes. The edges between AB and CD may be replaced by edges between AD and BC if the labels of edges AB and CD are the same and if edge AB next to edge CD is true. When this happens the labels given to AD and BC are the same as possessed by AB and CD. The final 4 rules may be dealt with together. The point where the six edges meet is the elliptic node point. The black boxes represent 3 hyperbolic nodes when dealing with **HEa** and **HEs**, while they represent flat nodes when dealing with **FEs** and **FEa**. The edge labels $\mathbf{L_c}$ and $\mathbf{L_d}$ are defined in terms of the edge

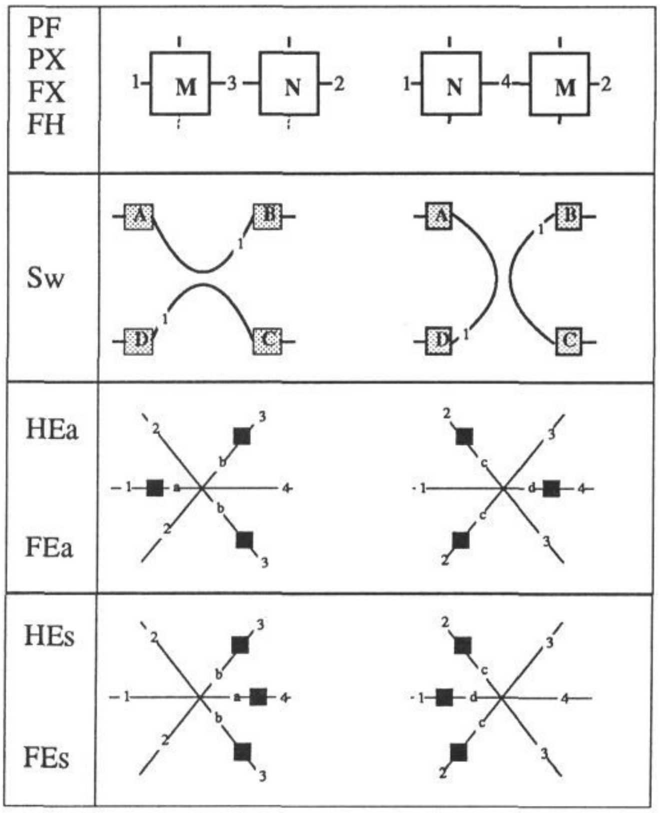

Figure 3: The Swapping Rules

Name	Node Types		Label Function
PF	P	F	βγ
PX	P	X	β
FX	F	X	γ
FH	F	H	αγ
HEa	3*H	E	I
HEs	3*H	E	I
FEa	3*F	E	αγ
FEs	3*F	E	αγ

Table 7: Node Swapping Functions

labels $\mathbf{L_a}$ and $\mathbf{L_b}$ thus; $\mathbf{L_c=F(L_b)}$ and $\mathbf{L_d=F(L_a)}$.

Introduction Rules

The first three node introduction rules, **IF,IP,IH**, can be dealt with together. The nodes are introduced along an edge labelled $\mathbf{L_1}$. The edges leading from the node pair are labelled $\mathbf{L_1}$, while the edge between the node pair is $\mathbf{L_2= F(L_1)}$ where **F** is the label function defined in Table 8. To introduce a pair **X** node there are two choices available as can be seen in Figure 4. The **X** nodes are represented by the points where four edges meet. The simpler of the two is introduction upon a pair of edges, IX2. This can occur when edges $\mathbf{L_1}$ *next to* $\mathbf{L_2}$ is true and $\mathbf{L_1 \approx_c \alpha(L_2)}$. The new edges created are defined thus; $\mathbf{L_3=L_1}$ and $\mathbf{L_4=L_2}$. The introduction of a pair of **X** nodes on a single edge is more of a family of rules rather than a single rule. The two edges that are introduced are not defined by the edge in the graph as in all the previous rules, they are only constrained. The edge $\mathbf{L_3 = L_1}$ but all that is defined about the edge $\mathbf{L_2}$ is that $\mathbf{L_2 \approx_c \alpha(L_1)}$. The family of rules can be written as $\mathbf{IX1(L_2)}$, where this is the introduction of a pair of **X** nodes on an edge with the two edges coloured differently to $\mathbf{L_1}$ being $\mathbf{L_2}$.

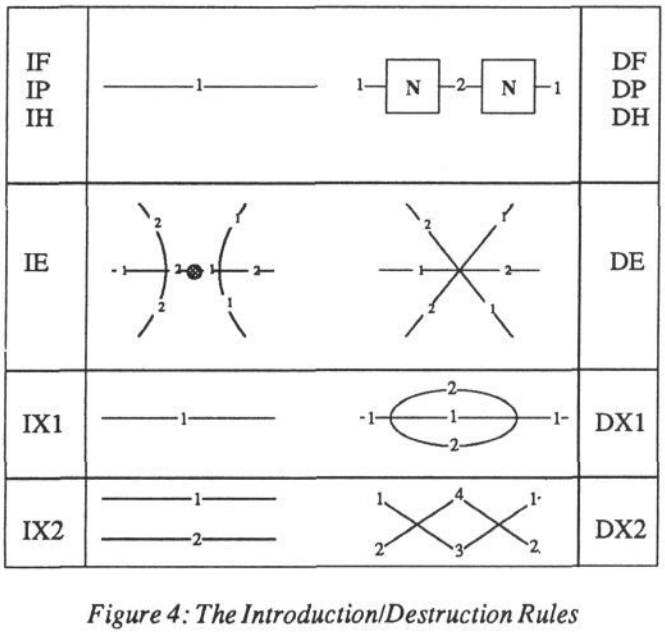

Figure 4: The Introduction/Destruction Rules

Name	Node Type	Label Function
IF	F	γ
IP	P	β
IH	H	α
IX1	X	Not Applicable
IX2	X	i
IE	E	Not Applicable

Table 8: Node Creation Functions

The final rewrite rule is **IE**. This has no edges to define at all because, unlike in all the other introduction rules, edges are destroyed rather than created. It Figure 4 the grey circle represents a hyperbolic umbilic, again the four lines meeting at a point represents an **X** node and the six line meeting at a point represent an **E** node. The relationship between the edge labels $\mathbf{L_1}$ and $\mathbf{L_2}$ is that $\mathbf{L_2=\alpha(L_1)}$.

Destruction Rules

These are simply the inverses of the introduction rules. For **DF, DH, DP**, if the edges labels for the edges either side of the node pair are the same, call it L, and the edge in between the edges, labelled $\mathbf{L_2}$, is such that $\mathbf{L_2=F(L_1)}$ where **F** is the labelling function for the equivalent introduction rule then the nodes may be replaced by a single edge labelled $\mathbf{L_1}$. Similar processes will give **DE, DX1** and **DX2.**

This classification uses only local rewrite rules to describe the ways that a shape can change. However these rules can be used to provided a language to describe both local and global changes in amorphous shapes, as well as be used for object recognition and deformation prediction.

IMPLEMENTATION

It is necessary to find the ridge lines of a surface to create a 3D description. These ridge line could be found by calculating the curvatures at each point and then searching the surface for the maxima,minima and zeros of curvature. This however would be extremely time consuming. The method suggested to find the ridge lines is to extract them using either the full symmetry set, or the evolute of the surface.

If the symmetry set of the surface is used to find the description then projecting back the lines which terminate the symmetry sheets onto the surface will give the ridge lines. The curvature at each point is the radius associated with the sphere centered at the point on the symmetry set which touches the point when projected back onto the surface. If the evolute is used to find the surface description then the rib line of the evolute can be projected back in an identical manner. (This is because the rib lines of the evolute are the same as the bounding lines of the full symmetry set.)

At present it is assumed that the surface will be provided as in a binary voxel representation. The idea is to try applying both the method of Scott to produce the symmetry set together with

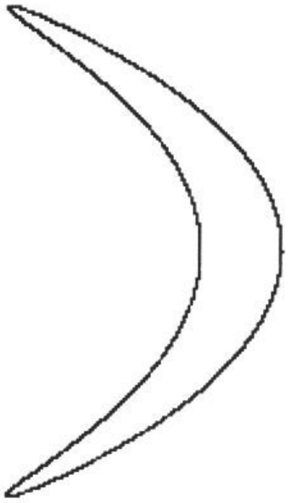

Figure 5: The Contour

the a method for extracting the Evolute using Fourier transforms. At present both are implemented on 2D binary pixel images using an AMT DAP which is an SIMD array processor.

The Scott method is uses a two step wave/diffusion process where at each step the wave process is approximated by convoluting with a Laplacian mask to solve the wave equation, while the diffusion step is performed to remove the high frequency terms of the wave which get left behind. This produced the symmetry axes, however the larger the radius of curvature the larger the area of localization to the end points of the symmetry set. Also the larger the radius of the symmetry sets the harder it becomes to detect them. This is implemented on the DAP by performing the convolution simultaneously at different pixels. The routine is provided with a binary contour(as shown in Figure 5. The resulting output is an image. At each pixel is stored the maximum value of the energy of the wave-diffusion process taken over time. This is show in Figure 6.

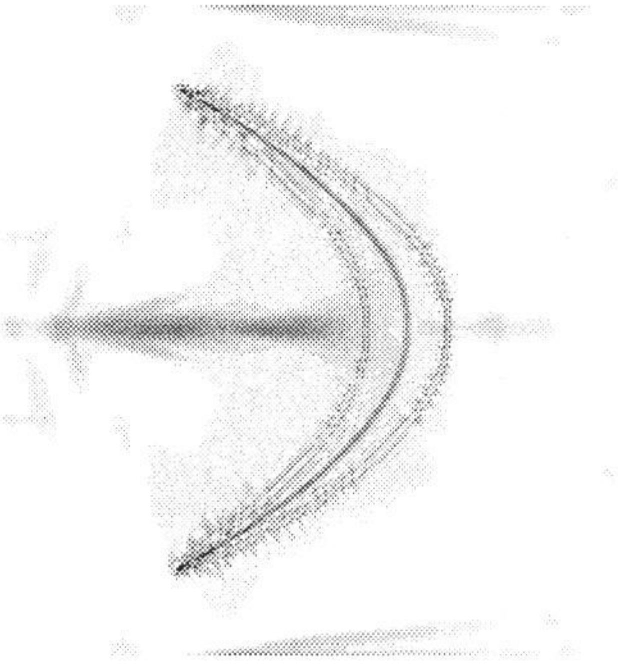

Figure 6: The Symmetry set of a Contour

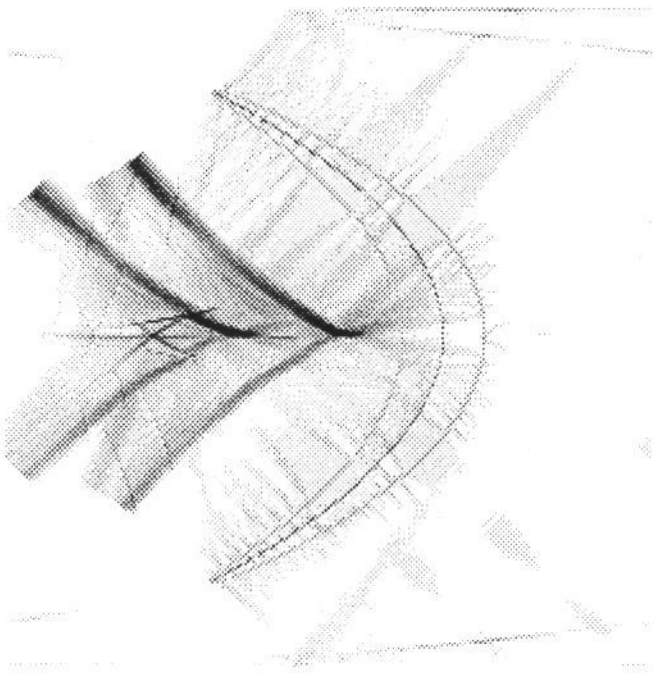

Figure 7: The Evolute of a Contour

The second method is to apply the wave process in the Fourier domain. This method more exactly simulates the wave process and so the smoothing of the contour need only be performed at the start rather than at each time step. This leads to the production of the evolute (and a weak version of the symmetry set) instead of just the symmetry axis as shown in Figure 7.

This work surgests severage areas for work to proceed. The AMT DAP can be used to find thes descriptions from shapes which are provided to it as binary voxels.

A three dimessional hieracical shape space can be produced, similar to the two dimensional shape space produced by Leyton. This will encapsulate both the different complexities of shapes and the process history which will transform one shape into another.

Finally since the shape descriptor is a simple graph. Model matching may be accomplished using existing model matching techniques.

CONCLUSION

It may be possible to check if these rewrite rules are mathematically correct by examining the critical points of the evolute of the surface as it changes over time. While this is has been done for the case of 2D contours[10], it has not been done for 3D surfaces. This is due to the fact that problems in singularity theory tend to explode when the dimension of the problem gets too large. The increase of dimension between contours and surfaces is that of moving from a five dimensional space to a seven dimensional one. While the mathematics of the descriptors remains unchecked it is not possible to produce a systematic way to catorgarise all the possible rules and descriptions.

Despite this it is also possible to examine simple shapes and to then produce descriptions for them. These will then be used to check that the rules work, and to try to find any rules that have been missed.

The symmetry sets of the surfaces must also be found. The nature of the two 2D algorithms means that the 3D algorithms will be very similar, For the Wave/Diffusion process all that is required is a 3D convolution, while the Fourier transform is decomposable and this means that increasing the dimension will not complicate the program.

REFERENCES

1: **D. Marr, H. K. Nishihara**, "Representation and recognition of the spatial organization of three-dimensional shapes", *Proceedings of the Royal Society - London B*, vol. 200, pp. 269-294, 1978

2: **A. P. Pentland**, "Perceptual Organization and the Representation of Natural Form", *Artificial Intelligence*, vol. 28, pp. 293-331, 1986

3: **J. J. Koenderink, A. J. van Doorn**, "Dynamic Shape", *Biological Cybernetics*, vol. 53, pp. 383-396, 1986

4: **M. Leyton**, "A Process-Grammar for Shape", *Artificial Intelligence*, vol. 34, pp. 213-247, 1988

5: **G. Scott, S. C. Turner, A. Zisserman**, "Using a Mixed Wave / Diffusion Process to Elicit th Symmetry Set", *AVC 88*, pp. 221-228, 1988.

6: **M. Leyton**, "Symmetry-Curvature Duality", *Computer Vision, Graphics, and Image Processing*, vol. 38, pp. 327-341, 1987

7: **I. R. Porteous**, "The normal singularities of surfaces in Rsup4(3)", *Proceedings of Symposia in Pure Mathematics*, vol. 40 Part 2, pp. 379-393, 1983

8: **I. R. Porteous**, "The normal singularities of a submanifold", *Journal of Differential Geometry*, vol. 5, pp. 543-564, 1971

9: **C. J. Davies**, "Surface Description by Labelled Graphs", *Forthcoming technical report*

10: **J. W. Bruce, P. J. Giblin, C. G. Gibson**, "Symmetry sets", *Proceedings of the Royal Society of Edinburgh*, vol. 101A, pp. 163-186, 1985