Structural Representation of the Outline and its Application

Structural Representation of the Outline and its Application

Ao GUO-LI, Cui YU-JUN, Masao IZUMI and Kunio FUKUNAGA

(Received June 15, 1991)

Abstract Scale-space filtering has been discussed in order to represent waveforms in a hierarchical form. Under some particular conditions, however, the shape of the zero-crossing curve of second derived function (ZCSDF) suddenly changes even if the original waveform changes a little bit. In addition, when we represent the ZCSDF (zero-crossing curve of second derived function) by graphs (tree structures), there is an intermediate state in the case when the graph structure changes from encompassing structure to parallel structure and vice versa. It is difficult to define a measure of similarity between the waveforms by using the graph structure of ZCSDF defined in the previous works. To solve these problems, this paper introduces an equivalent transformation between tree structures of ZCSDF, and proposes the degree of similarity between waveforms by use of graph structures which provides a concise but hierarchical description of the waveforms. According to our experiments, the proposed definition of similarity between the waveforms can reflect our intuitive measure of similarity. In the last we discuss the possibility of application of the defined similarity to object recognition when we represent the outline of the object by a waveform function.

1. Introduction

When a human being recognizes an object, one firstly examines major features of the object to understand the scene surrounding the object and then pays one's attention to features of the object at the necessary level. The human being skillfully controls these two processes so as to recognize quickly the objects appeared in the scene. This type of recognition process suggests that one of the most efficient object recognition would be based on a hierarchical approach.

In order to represent the features of a waveform in the hierarchical form, Witkin introduced a qualitative signal description method[1] [5] which deals the waveform with various scales of scale-space filtering. The waveforms are generated by smoothing the original waveform in continuous variable "bandwidth" of Gaussian filtering. These waveforms are called generalized waveforms. Witkin paid his attention to the ZCSDF of generalized waveforms because it can describe the waveform in hierarchical form.

---

* Department of Electrical Engineering, College of Engineering.

** University of Osaka Prefecture.

When the shape of a waveform changes, the shape of ZCSDF may change in general. But under a particular condition there may be large difference between the shapes of the ZCSDFs even if the difference between the shapes of the original waveforms is not large. In other words, a small change of a waveform brings about a large change of the shape of ZCSDF in the scale-space. For this reason it is said that this method can only apply to those images in which the distortion is smaller. Besides sometimes ZCSDFs intersect each other in the scale-space. It is difficult to describe this shape of ZCSDF by the previous method.

The goals of this paper are

(1) To find a method which can describe two different shapes of ZCSDF generated by a small distortion of the waveform with the same expression.

(2) In order to represent the structure of ZCSDF in the scale-space, we introduce an expression of "tri-tree structure" which represents the feature of ZCSDF in a concise form.

(3) To define a measure of the similarity between the ZCSDFs using the tri-tree structure.

(4) As an application of the similarity measure, we examine the object recognition by using the degree of similarity.

2. Scale-Space Filtering

Scale-space filtering is a method that describes signals qualitatively, managing the ambiguity of scale in an organized and natural way[2] [6]. The gaussian convolution of a signal f(x) depends both on x, the signal's independent variable, and on σ, the gaussian's standard deviation. The convolution is given by

$$F(x,\sigma)=f(x) * g(x,\sigma)= \int_{-\infty}^{\infty} f(u) \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-u)^2}{2\sigma^2}} du, \tag{1}$$

where "$*$" denotes convolution with respect to x. This function defines a surface on the (x,σ)-plane, where each profile of the constant σ is a gaussian-smoothed version of f(x). The (x,σ)-plane is called as scale-space, and the function F, defined in Eq.(1) as the scale-space function of f(x).

At any value of σ, the extrema in the n-th derived function of the smoothed signal are given by the zero-crossing of the (n+1)-th derived function which can be given by the following relation.

$$\frac{\partial^n F}{\partial x^n}=f * \frac{\partial^n g}{\partial x^n}. \tag{2}$$

Although the method presented here can apply to any level of derived function, we will restrict our attention to those in the second derived function. These are extrema of slope of a function, i.e. inflection points. In terms of the scale-space function, the inflections at all values of σ are the points that satisfy $F_{xx}=0$, $F_{xxx}\neq 0$, here the subscript notation indicates partial derivatives. The outlines of $F_{xx}=0$ mark the appearance and motion of inflection points on the smoothed signal, and provide the raw material for a qualitative description over all scales, in terms of inflection points.

It is noticed that new zero-crossing points may appear as the change of σ from coarse to fine scale, but ones never disappear. In other words, the scale-space outlines are closed in the upward direction of the scale space. Consequently, extrema observed at any scale may be localized by their projections at the finest available scale, and the partitioning of the function by extrema moving from coarse to fine forms a strict hierarchy[3].

3. Structure Description of Outlines

3. 1 Curvature Function and Characteristic Function

Let us consider a conversion of an outline image expressed in the two dimensions into a function with one independent variable. When we recognize the objects based on the outline image, it is better for us to consider a function with one variable as the feature funciton by use of the scale space approach. This paper introduces a curvature function of the object outline as a feature function. Fig.1 shows a curvature function. Firstly the outline is approximated by a piecewise linear line. The angle function $\theta(t_0)$ of the outline denotes an angle between two line segments at the position $t=t_0$. In the case of Fig.1, $\theta(t)$ shows the angle between the line segment t_0 t_1 and the line segment t_1t_2 as follows.

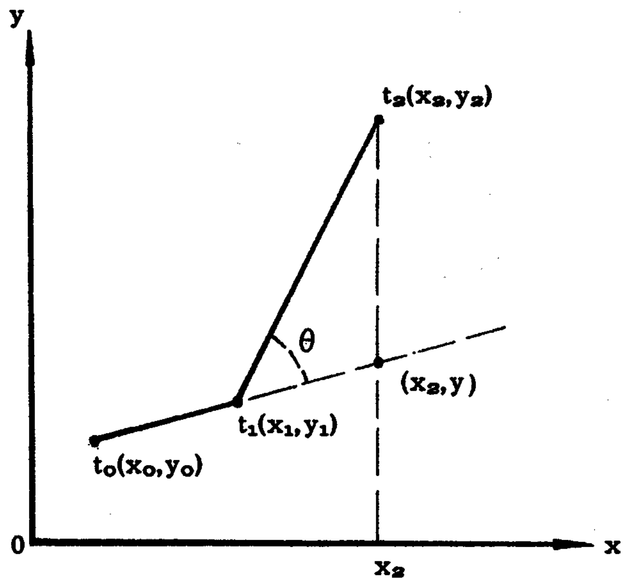


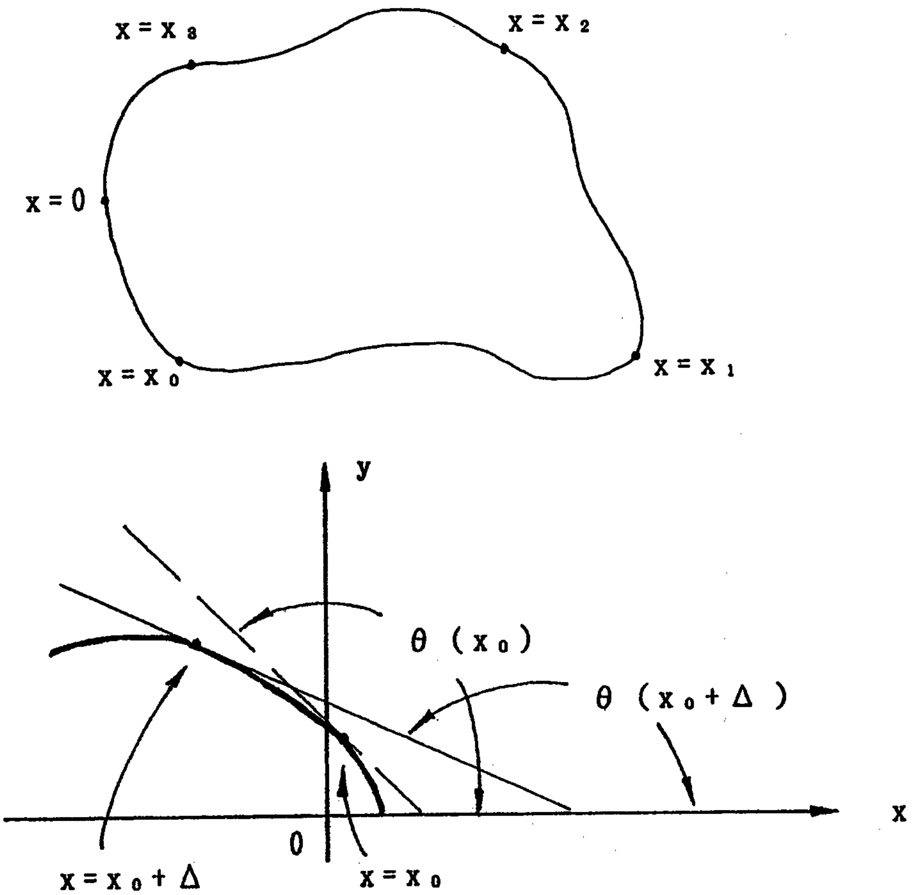
Fig.1 Calculation of θ

Definition 1 : Let us consider the extension line of the line segment t_0t_1. This extension line divides x-y plane into two portions. We suppose that the sign of the function θ is minus if the line segment t_1t_2 appeared in the left hand side when the direction is from point t_0 to point t_1, otherwise the sign of θ is taken as plus. The θ is set to be zero only if the point t_2 locates on the extension of the line segment t_0t_1. The absolute value of θ is given by

$$\theta=\pi-\cos^{-1}\frac{(t_0t_1)^2+(t_1t_2)^2-(t_0t_2)^2}{2\times(t_0t_1)\times(t_1t_2)} \tag{3}$$

The curvature function f(x) is defined as follows in the case when the outline is shown by Fig.2.

Definition 2 : Firstly, we set a point of the outline as starting point (x=0), then trace the outline in the clockwise direction. The curvature function f(x) is calculated by



$$f(x)=\lim_{\Delta\to 0}\frac{\theta(x+\Delta)-\theta(x)}{\Delta}=\frac{d\theta(x)}{dx}$$

Fig.2 An outline and its curvation

$$f(x)=\lim_{\Delta\to 0}\frac{\theta(x+\Delta)-\theta(x)}{\Delta}=\frac{d\theta(x)}{dx}, \tag{4}$$

The function f(x) is a periodic function where the period is the length L of the outline. We call the normalized curvature function f(x/L) "a feature function" of the object.

3. 2 Structure Tree of Feature Functions

In the next, we consider a representation of structural feature of zero-crossing curve in the feature function by a graph expression based on the tri-tree structure. Fig.3 shows an example of a structure of the tri-tree. In the tri-tree a son node incident to the middle branch is called a middle-node, a son node incident to a side branch is called a branch-node. The son node incident to the left branch is called a left-node and the son node incident to the rigth branch is called a right-node.

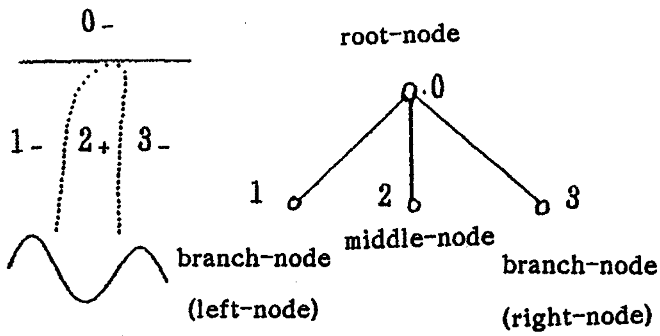
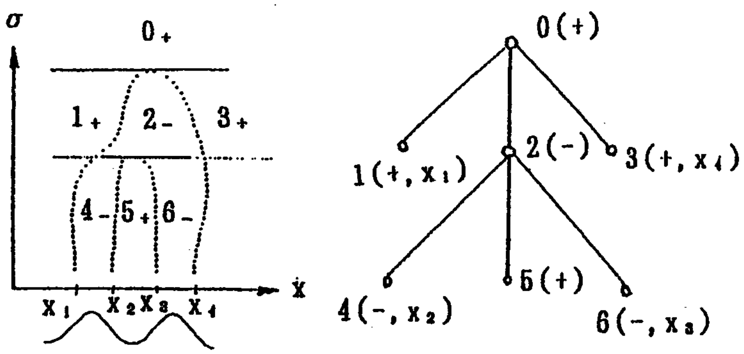


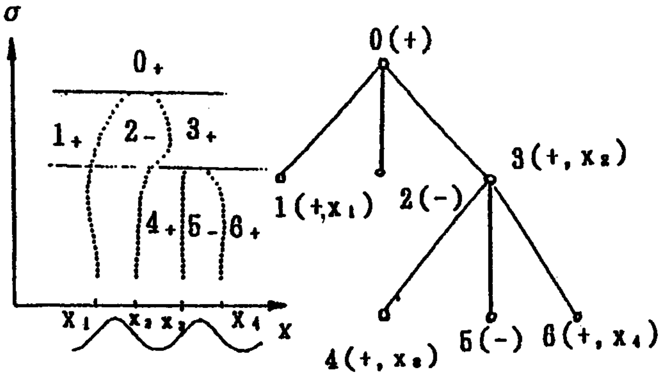
Fig.3 A tri-tree

In general, ZCSDF closes from downward to upward, but opens downward in the (x,σ) scale-space. The level of hierarchy in the representation of the tri-tree structure is related to the value of σ of gaussian filter when we obtain the ZCSDF. There is no zero-crossing point in the (x,σ) scale-space when the σ is sufficiently large. When the value of σ decreases, a zero-crossing point appears first. In this condition, we divide the (x,σ) scale-space into three domains like as Fig.3 so that a new level of hierarchy of the structure tree is set up. The domain surrounded by the ZCSDF and X-axis is expressed by the middle-node of the structure tree, the left domain is expressed by the left-node, and the right domain is expressed by the right-node. While σ is getting smaller, another zero-crossing point appears and the domain is subdivided, then the next hierarchy of the structure tree is set up. Tracking the ZCSDF in the (x,σ) scale-space from coarse to fine, we can divide the space into a number of domains. When we represent the relation of the domains in the (x,σ) scale-space by the structure tree, the feature of the original waveform (function) can be described in the hierarchical manner.

There are two fundamental structures in the structure tree described above. One is an encompassing structure and the other is a parallel structure. An encompassing structure consists of a parent tri-tree and a son tri-tree, and the node of the son tri-tree connects the middle-node of the parent tri-tree. A parallel structure connects the son tri-tree from the either of left or right side node of the parent tri-tree as shown in Fig.4.



(a) An encompassing structure



(b) A parallel structure

Fig.4 Two kinds of structure tree

Two parameters, sign parameter and position parameter, are considered to represent the feature of the node in our expression of the structure tree.

(1) Sign parameter

(a) Sign of the node represents a sign of curvature of the outline section expressed by the node. The curvature of the outline section is plus if the shape of outline is concave, and the curvature of outline section is minus in the case when its shape of outline is convex.

(b) As ZCSDF is a set of inflection points, two branch-nodes of the tri-tree have the same sign and are opposite to the sign of the middle-node.

(c) The sign of the root-node is opposite to the sign of the middle-node and is same as the branch-nodes.

(2) Position parameter

Besides a sign parameter there is a position parameter in branch-nodes. This value is the X-coordinate of projections of the inflection point expressed by the node.

4. Equivalent Transformation of Structure Tree

4. 1 Separating and merging of ZCSDF

According to the above discussion, a feature function is represented by the structure tree. In this expression, the structure of the tree reflects the the shapes of ZCSDF, and thus the similarity between the structure trees reflects the similarity between the shapes of ZCSDFs. But under some particular conditions there might be large difference between the shapes of ZCSDFs even if the difference between the shapes of feature functions of outlines is not large. It may happen by the separation or the merging of the concave-convex structures of feature function when ZCSDF changes from an encompassing structure to a parallel structure and vice versa. Fig. 5 shows the separation and the merging of concave-convex structures. The structures of similar shape of outlines should be represented in the same structure. The representation based on the previous works results in different structures (the encompassing structure and the parallel structure) even if between two similar feature functions. It is necessary to introduce an unified representation. In this paper, we propose an equivalent transformation between structure trees.

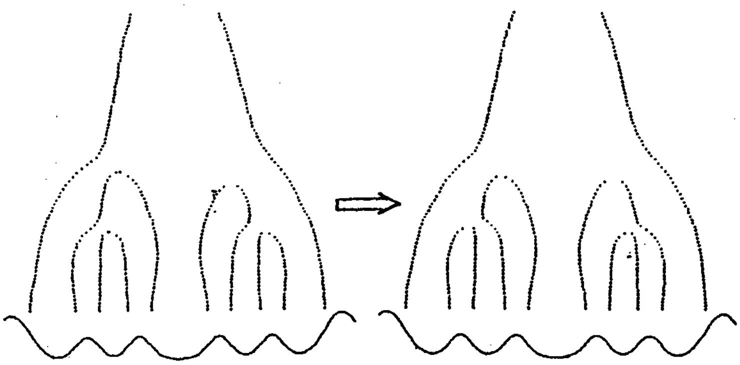


Fig.5 Separation and merging of Concave-convex structure

4. 2 Equivalent Transformation

There are two kinds of equivalent transformation. One is the transformation from an encompassing structure tree to a parallel structure tree, and the other transformation from a parallel structure tree to an encompassing structure tree. In this papar we transform the parallel structure tree into an encompassing structure tree if transformation is needed. In this case the difference between two similar outlines can be examined among the encompassing structure trees.

The rules of equivalent transformation from a parallel structure tree to an encompassing structure tree are given in the following :

Rule 1 Transformation of a simple parallel structure tree

A parallel structure tree which has not any parent tree, son tree and brother tree is called a simple parallel structure tree as shown in Fig.6. The four branch-nodes of a simple parallel structure tree called the outside-node of the younger brother tree (node 4), the inside-node of the younger brother tree (node 6), the outside-node of

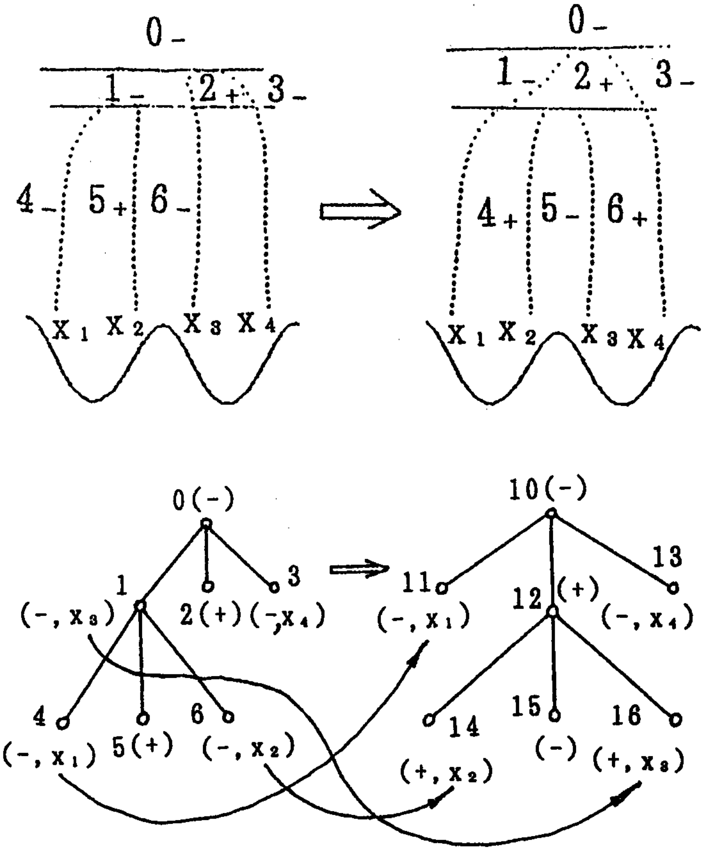


Fig.6 Equivalent transformation of a simple tree

the elder brother tree (node 3) and the inside-node of the elder brother tree (node 1) respectively. The four branch-nodes of transformed encompassing structure tree are called the younger brother side-node of the parent tree (node 11), the elder brother side-node of the parent tree (node 13), younger brother side-node of the son tree (node 14) and elder brother side-node of the son tree (node 16) respectively. When a transformation is performed, it first reverses the sign parameters of the younger brother tree, then moves the tree from the branch-node to the middle-node of the elder brother tree. In the last, it moves six parameters from the simple parallel structure tree to the encompassing structure tree as follows :

(1) Move the parameters of the outside-node (node 4) in the younger brother tree of the parallel tree to the younger brother side-node (node 11) in the parent tree of the new transformed encompassing tree.

(2) Move the parameters of the inside-node (node 6) in the younger brother tree of the parallel tree to the younger brother side-node (node 14) in the son tree of the new transformed encompassing tree.

(3) Move the parameters of the outside-node (node 3) in the elder brother tree of the parallel tree to the elder brother side-node (node 13) in the parent tree of the

new transformed encompassing tree.

(4) Move the parameters of the inside-node (node 1) in the elder brother tree of the parallel tree to the elder brother side-node (node 16) in the son tree of the new transformed encompassing tree.

(5) Move the parameter of the middle-node (node 2) in the elder brother tree of the parallel tree to the middle-node (node 12) in the parent tree of the new transformed encompassing tree.

(6) Move the parameter of the middle-node (node 5) in the younger brother tree of the parallel tree to the middle-node (node 15) in the son tree of the new transformed encompassing tree.

Rule 2 Transformation of a parallel tree in the case when it contains an outside brother free (Fig.7)

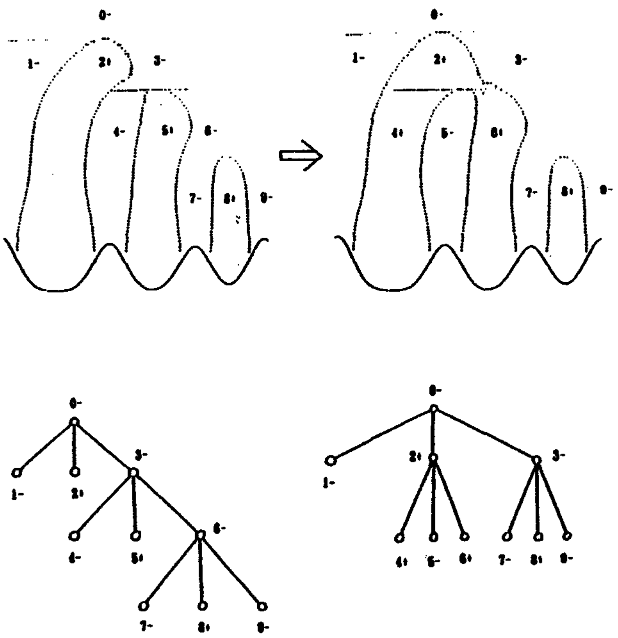


Fig.7 Equivalent transformation of a tree with an outside brother tree

When there is a brother tree outside the parallel tree, we first transform the parallel tree into an encompassing tree according to the rule 1, then move the outside brother tree to the branch-node in the same side of the new transformed parent tree. It becomes a brother tree of the new parent tree.

Rule 3 Transformation of a parallel tree in the case when it contains an inside brother tree between the two trees (Fig.8)

When there is a brother tree between two trees of the parallel tree, we first transform the parallel tree into an encompassing tree according to the rule 1, then move the brother tree between two parallel trees to the middle-node of the new transformed son tree. It becomes a son tree of the new son tree.

Rule 4 Transformation of a parallel tree in the case when the parallel tree contains a son tree (Fig.9)

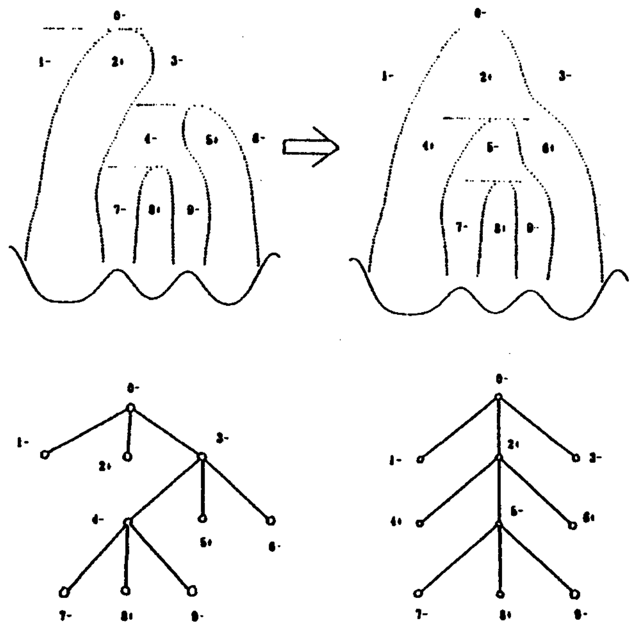


Fig.8 Equivalent transformation of a tree with an inside brother tree

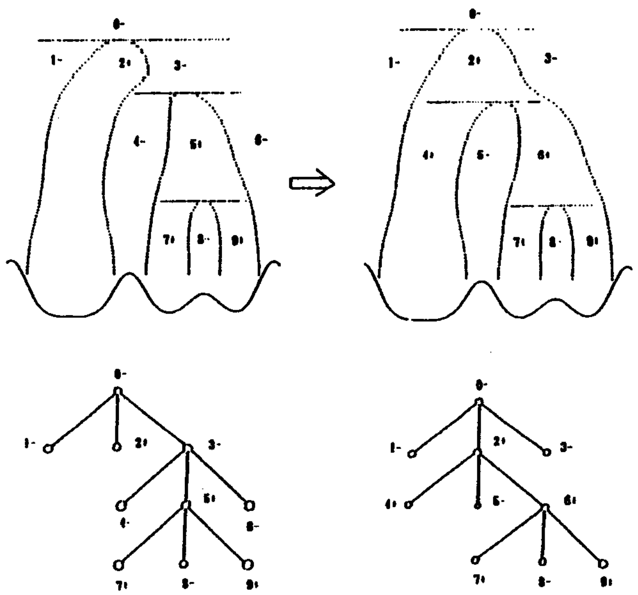


Fig.9 Equivalent transformation of a tree with a son tree

When there is a son tree in either of the two trees of the parallel tree, the transformation first converts the parallel tree into an encompassing tree according to the rule 1, then move the son tree in either of the two trees of the parallel tree to the branch-node in the same side (the elder brother side or the younger brother side) of the new transformed son tree. It becomes a brother tree of the new son tree.

As shown in Fig.4(b) (x,σ) scale-space is divided into three domains by ZCSDF of a pair of parallel structure trees, i.e. inside of the elder tree (2_-), inside of the younger tree (5_-) and outside of the brother trees (0_+, 1_+, 3_+, 4_+, 6_+). If a tree is inside of an elder brother tree, it is a son tree of the elder brother tree. If a tree is inside of a younger brother tree, it is a son tree of the younger brother tree. And if a tree is outside of two brother trees, it is a parent tree or a brother tree of the two brother trees. But in the midway of the equivalent transformation the parent tree can not be considered because in this paper the equivalent transformation always starts from its root node (parent node) and proceeds toward its leaf nodes (son nodes). Therefore we can reach the following conclusions.

(1) All parallel trees can be generated by extending a pair of trees of a simple parallel tree iteratively according to the following three forms.

(a) A son tree grows from an elder brother tree.

(b) A son tree grows from a younger brother tree.

(c) A brother tree grows from an elder brother tree or a younger brother tree or both.

(2) The equivalent transformation for all kinds of the tri-tree can be realized by using the four rules of equivalent transformation iteratively, because above-mentioned rule 1 is a transformation of the simple parallel tree, rule 2 and 3 are transformations for the situation (c), and rule 4 is a transformation for the situations (a) and (b).

As an example of equivalent transformation, Fig.10(a) shows an outline and its ZCSDF of an elephant image, (b) shows an outline and its ZCSDF of another elephant image which is an image of the same model taken from another angle. The structure trees are shown in (c) and (d) respectively. There are large differences between two structure trees in (c) and (d) although its outlines in (a) and (b) are similar. According to the rules of equivalent transformation the parallel structure tree in (d) can be transformed into an encompassing structure tree (f), and trees in (c) and (f) have same structure although its position parameters are different. It is more confident that the difference between the outline (a) and (b) is examined by structure trees of (c) and (f) rather than by structure trees of (c) and (d).

Of course the equivalent transformation is performed under a particular condition where the shapes of two waveforms are similar. In this paper the equivalent transformation is performed only when the length L of the outlines are regularized as 1, and all of the differences between four pairs of position parameters of two trees in this hierarchy are smaller than 0.05, otherwise equivalent transformation is not performed.

4. 3 Intermediate Structure

While the concave-convex structure of a waveform changes, the shape of its ZCSDF in the scale-space changes. There are three types about the changes of ZCSDF of waveform.

(1) Moving.

(2) Increasing and decreasing.

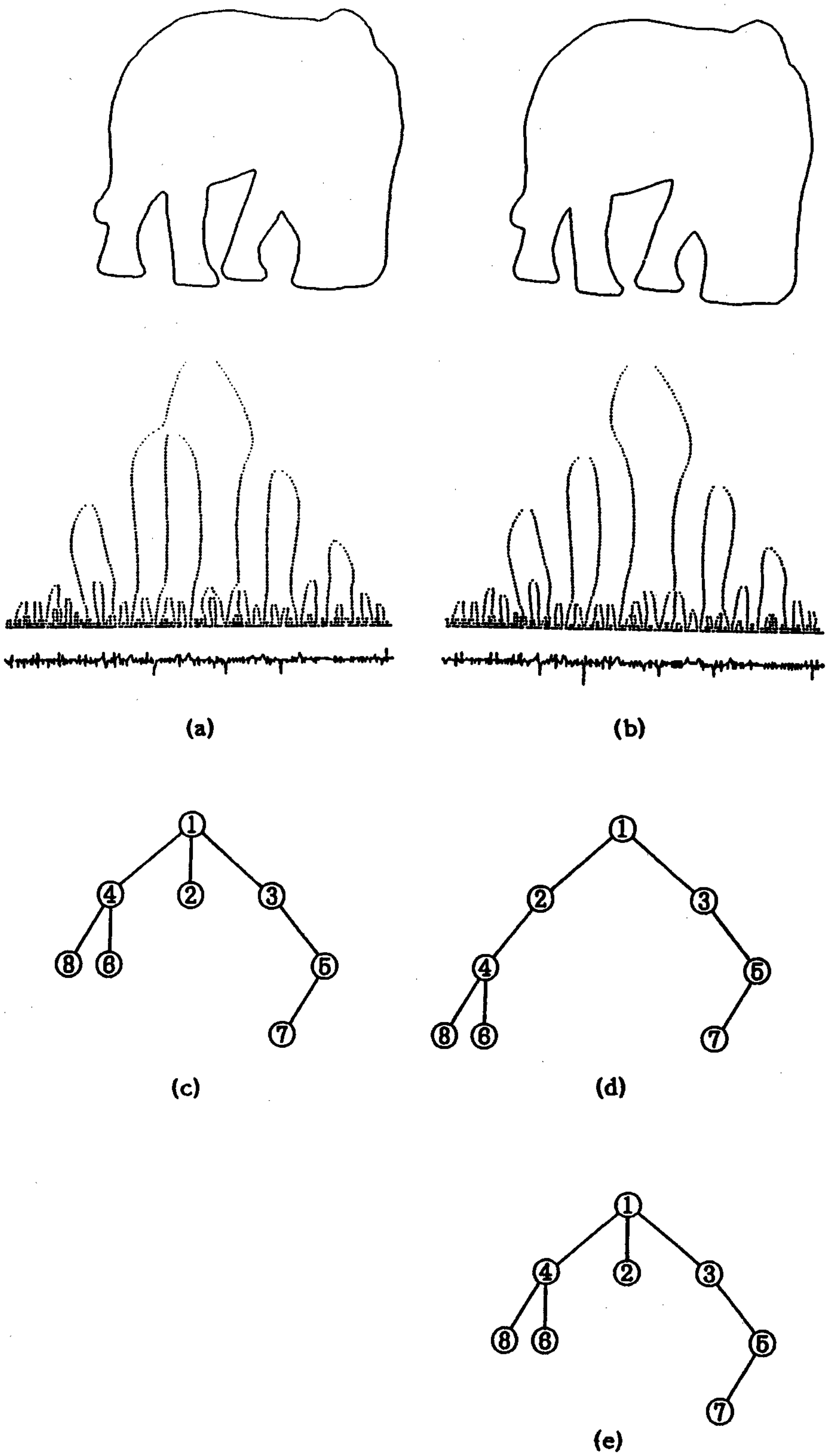
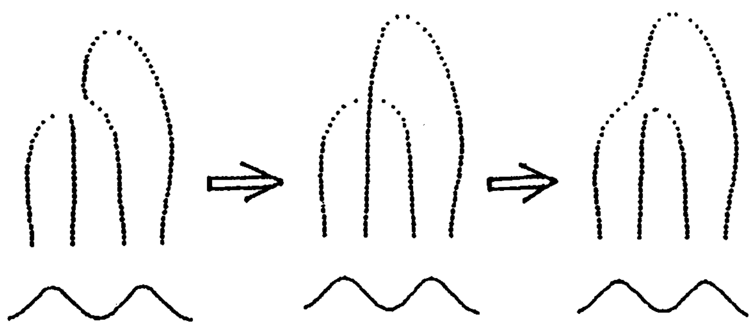


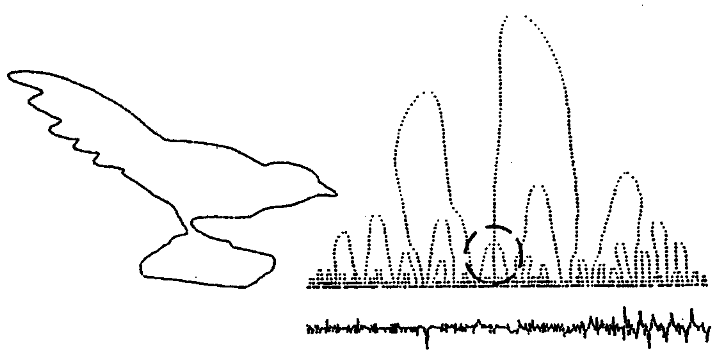
Fig.10 An example of equivalent transformation

(3) Separation and merging.

If the concave-convex structures of waveform changes independently, its ZCSDF may move, increase or decrease. If the change of the concave-convex structures has an effect on other concave-convex structures, its ZCSDF may be separated or merged so that its hierarchical structure changes. As mentioned above, there are two fundamental structures about structure tree. One is a parallel structure and the other is an encompassing structure. Separation of ZCSDF is a change of structure tree from an encompassing structure to a parallel structure, and merging of ZCSDF is a change of structure tree from a parallel structure to an encompassing structure. But there is an intermediate structure between the parallel structure and the encompassing structure in the case when the structure changes from a parallel structure to an encompassing structure and vice versa. Fig.11(a) shows a intermediate structure when the structure changes from a parallel structure to an encompassing structure, and Fig.11(b) shows an outline of a model of a bird and its ZCSDF in which an intermediate structure appears. An intermediate structure can be considered to be an intersection of two ZCSDFs each other[4]. According to the principle of equivalent transformation, three structures (an encompassing structure, a parallel structure and an intermediate structure) which describe similar waveforms can be considered to



(a)



(b)

Fig.11 An intermediate structure tree

be an equivalent structure, and can be transformed into same kind of structure. In this paper if an intermediate structure appears, first of all we substitute the intermediate structure by an encompassing structure, then go to the next steps.

5. The Degree of Similarity concerning to the Structure Tree

5. 1 Definition of the Degree of Similarity

When the feature function of an outline is described by a structure tree, the degree of similarity between outlines can be examined by the degree of similarity between their structure trees. For this reason we propose a definition of the degree of similarity between two structure trees as follows.

Definition 3 : When a node t_i is expressed by a parameter node $t_i(f,h)$, the degree of similarity between the tree T1 starting from node $t1_i$ and the tree T2 starting from node $t2_i$ is given by

$$S^i_j = \begin{cases} F^i_j \times [0.5 \times H^i_j + 0.5 \times (1/M^i_j) \times \sum_{y=1}^{M^i_j} S_y^{i+1}] & \text{if } M^i_j \neq 0 \\ F^i_j \times H^i_j & \text{if } M^i_j = 0 \end{cases} \tag{5}$$

Where $F^i_j = |f1^i_j + f2^i_j|/2$, f^i_j is the sign of curvature in the domain surrounded by X-axis and ZCSDF expressed by node t^i_j, $f^i_j \in \{-1, 1\}$, i.e. $F^i_j \in \{0, 1\}$. $F^i_j = 0$ if the concave-convex shapes of two feature functions are opposite. It means that two feature functions are not similar to each other at all.

$H^i_j = |h1^i_j - h2^i_j| / \max(h1^i_j, h2^i_j)$, h^i_j is the absolute value of the difference between position parameters of two branch nodes, i.e. the width of inflection points of the feature function expressed by the node t^i_j.

$h^i_j = |1 - r|$, 1 and r are position parameters of left-right branches.

$M^i_j = \max(m1^i_j, m2^i_j)$, m^i_j is the number of son node of node t^i_j, $m^i_j = 0$ for a leaf node of the structure tree.

By this definition a weight is attached automatically to all nodes from the root node to the leaf nodes in order of the hierarchy, and the degree of similarity between two structure trees is given by a sum of the degree of similarity of its all nodes in a recursive form.

5. 2 Some Examples of the Degree of Similarity

Example 1 : As shown in Fig.12(a), S=1 when all of the parameters f, h of nodes in two trees have same value.

Example 2 : As shown in Fig.12(b), S=0.75 when the parameters f, h of nodes in two trees have same value except that a node does not exist in the second hierarchy of the right tree.

Example 3 : As shown in Fig.12(c), S=0.9375 when the parameters f, h of nodes in two trees have same value except that a node does not exist in the third hierarchy of the left tree.

Example 4 : As shown in Fig.12(d), S=0.925 when the parameters f, h of nodes in two trees have same value except that in the second hierarchy $h_a = 0.7 \times h_b$.

Example 5 : As shown in Fig.12(e), S=0.98125 when the parameters f, h of nodes in two trees have same value except that in the third hierarchy $h_a = 0.7 \times h_b$.

Example 6 : The outlines of three chairs are shown in Fig.13(a), (b), (c), their ZCSDFs are shown by Fig.13(d), (e), (f). The degrees of similarity among three chairs are shown in table 1. It is clear that the degree of similarity between chair(a) and chair(b) is larger than between chair(a) and chair(c) because chair(a) is more similar to chair(b) than chair(c).

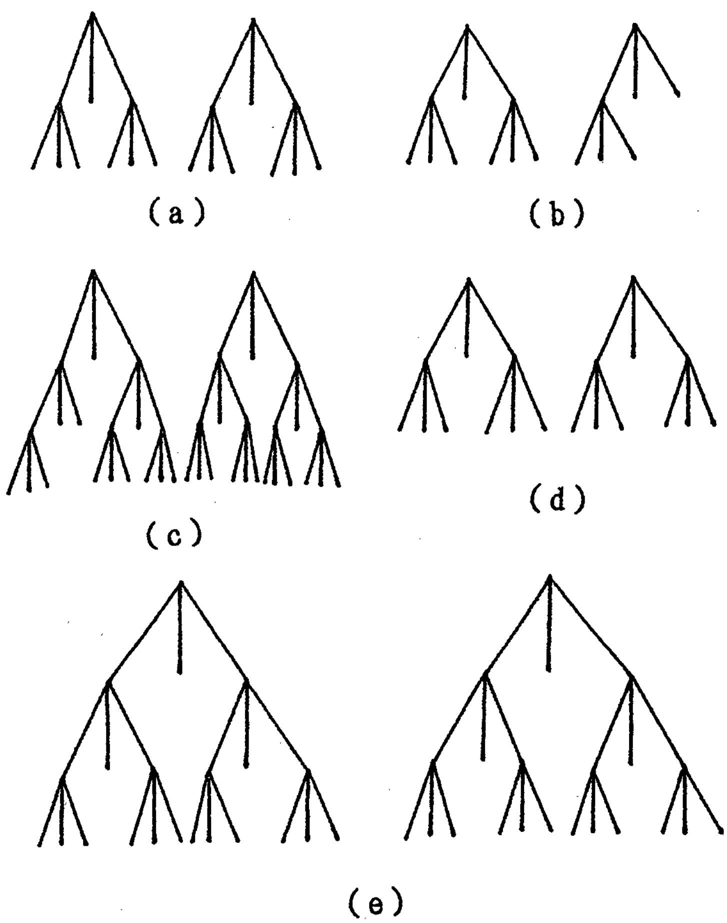


Fig.12 Examples of the degree of similarity

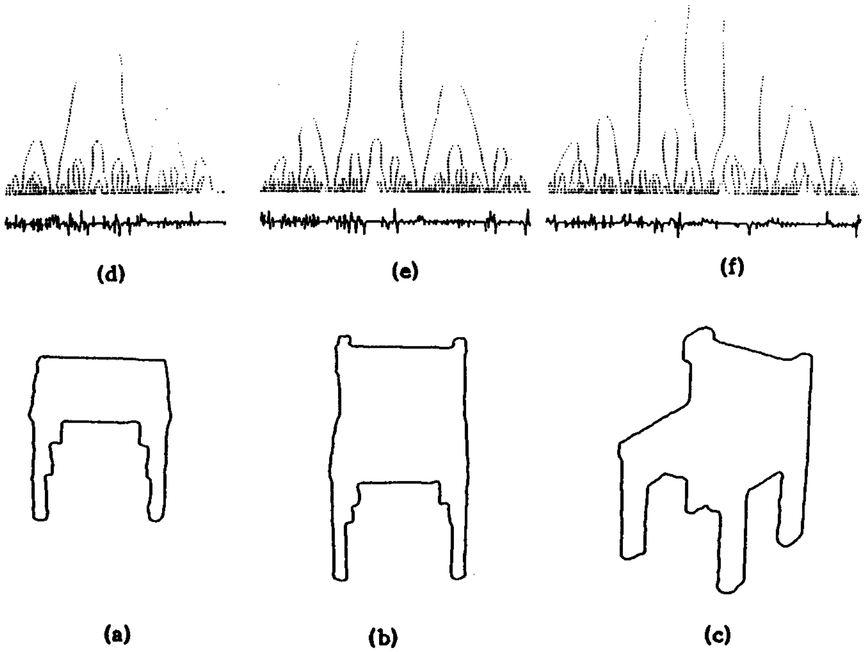


Fig.13 The degree of similarity of chairs

table 1

	c1	c2	c3
c1	1	0.6930	0.5689
c2		1	0.6820
c3			1

6. Conclusions

In scale-space filtering, it is natural that the structures of ZCSDFs should be similar in the case when the shapes of waveforms are similar, and the structures of ZCSDFs should be different when the shapes of waveforms are different. But under some particular conditions, the separation or the merging of the concave-convex structures of waveforms brings about a modification of the structures of ZCSDFs although its shapes of waveforms are still similar. The equivalent transformation proposed in this paper is a method of solving this kind of questions. By using equivalent transformation, the different structures generated from a small change of a waveform can be dealed as same structures, and it can lead to a more confident matching.

Scale-space filtering is based on inflection points of the curve at each level of smoothing, and describes the waveform from coarse to fine. The structure of the tri-tree is a concise but complete qualitative description about the hierarchical structure of ZCSDF. The degree of similarity between structure trees proposed here is an efficient method of examining the difference between structure trees.

References

1) A. P. Witkin : "Scale space filtering", Proc. of 8th International Joint Conference on Artificial Intelligence, Karlsruhe, pp.1019-1022, (1983).

2) Alan L, Yuille and Tomaso A.Poggio : "Scaling Theorems for Zero Crossings", IEEE Trans. PAMI, Vol. PAMI-8, No.1, pp.16-25, January 1986.

3) Jean Babaud, Andrew P.Witkin, Michel Baudin : "Uniqueness of the Gaussian Kernel for Scale-Space Filtering", IEEE Trans. PAMI, Vol. PAMI-8, No.1, pp.26-33, January 1986.

4) Farzin Mokhtarian and Alan Mackworth : "Scale-Based Description and Recognition of Planar Curve and Two-Dimensional Shapes", IEEE Trans. PAMI, Vol. PAMI-8, No.1, pp.34-43, January 1986.

5) A. P. Witkin, D. Terzopoulis and M. Kass : "Signal matching through scale space", Proc. Fifth National Conf. on Artificial Intelligence, Philadelphia, Penn, pp.714-719, (1986).

6) F. Mokhtarian and A. Mackworth : "Scale-based description and recognition of planar curve and two-dimensional shapes", IEEE Trans. Pattarn. Anal. & Mach. Intell., PAMI-8, No.1, pp. 34-43(1986).