DECOMPOSITION AND RECOGNITION OF NON-ORTHOGONAL INTERACTING FEATURES USING AN SOFM NEURAL NETWORK

A.H. Zulkifli and S. Meeran*
Department of APEME
University of Dundee, Scotland, DD1 4HN
*To whom correspondences should be addressed

ABSTRACT

Recognition of interacting features has been a difficult task in many existing feature-recognition systems. The unique topological patterns of isolated features change drastically when they interact. Hence many surface-based methods encounter problems in accommodating these changes in their generic feature definitions. Recently, much effort has been concentrated on the volumetric approach. However, many of these systems suffer from a problem of combinatorial explosion as the interaction between features becomes more complex. This paper presents a simple and robust system, in which the non-orthogonal interacting features are decomposed into non-orthogonal regions using a Kohonen self-organizing feature map (SOFM) neural network. The feature patterns in these non-orthogonal regions are then used as input in a multilayer feedforward neural network to recognize the features. Self-organization, competitive learning and the clustering of data are some of the SOFM's attributes, exploited in this work to deal with interacting features.

Keywords: Feature recognition; Neural networks; Feature patterns; CAD/CAM

1. INTRODUCTION

Feature recognition is a process by which the data structure of a solid modeller is searched for geometric and topological information that corresponds to a predefined set of features. The features referred to here are the form-features which are identified by their shapes and hence are application dependent. Holes, slots, steps, pockets, ribs, bosses and grooves are typical examples of the features. Feature recognition is an important process for achieving the CAD/CAM integration, and for realizing a concurrent engineering environment that will have a direct impact on design lead time and quality, and in turn on the marketability of the product.

There have been many attempts at developing feature-recognition systems, especially by using pattern-matching algorithms such as syntactic pattern recognition, graph matching and neural networks. However, the recognition of interacting features has been found to be hard in most of the existing systems, because some of the faces that correspond to a feature may be entirely absent, partially missing, or fragmented into several regions because of interactions (Vandenbrande and Requicha, 1994).

The interacting features normally encountered in a part can be categorized into two types: a 'no-need-to-decompose' type, in which the features do not share any common bottom face, and a 'to-be-decomposed' type, in which they do, as shown in Figs 1a and 1b respectively. The feature patterns for the interacting features shown in Fig. 1a are independent of each other; however, the interacting features in Fig. 1b need to be decomposed into their constituent features before any recognition can be done, as their individual feature patterns are corrupted by their interaction. The first type of orthogonal and non-orthogonal interacting features ('no-need-to-decompose') is dealt in Zulkifli and Meeran (1998a), while the orthogonal interacting features of the second type ('to-be-decomposed') is given in Zulkifli and Meeran (1998b). This paper proposes a new approach, which is able to break down the non-orthogonal interacting features of the second type into non-orthogonal primitive features and then recognize them using neural networks.

2. LITERATURE REVIEW

Feature-recognition systems can generally be classified into surface-based and volumetric systems. The techniques employed in surface-based systems include graph matching, syntactic pattern recognition and neural networks. The techniques employed in the volumetric systems include cell/volume decomposition approach, convex-hull and rule-based approach. As for the interacting features, simple primitive features such as slots, steps, blind slots, blind steps and pockets, which are uniquely defined in a syntactic-pattern recognition system (Kyprianou, 1980; Chuang and Henderson, 1990), lose their patterns when they interact, and hence may not be recognized. Graph matching (Joshi and Chang, 1988; Gavankar and Henderson, 1990; Corney and Clark, 1991; Ji and Marefat, 1995), which matches the feature graphs (which express the topological and geometric information of features) with the B-rep graph of the solid model may face the problem of unsolvable graphs due to the changes in the topology of interacting features.

Recently many authors such as Tseng and Joshi (1994), Vandenbrande and Requicha (1994), Sakurai (1995) and Trabelsi and Meeran (1996) have used volumetric approaches in addressing interacting features. These approaches decompose these interacting features, mostly of the orthogonal type, into cells or smaller volumetric units by extending the faces or edges of the interacting features; this is referred to as 'entity growing'. Features are considered as aggregations of cells or volumes created by this face/edge extension. Different combinations of these volumetric units give rise to multiple (however, equally feasible) interpretations of the part, and can therefore be used as the basis for generating and evaluating alternative process plans. However, the generation of these multiple interpretations can lead to a problem of combinatorial explosion as the interactions between features become more complex. Hence, there is a need for a system which is simpler, but which can still generate valid interpretations of the model, especially for the one that have non-orthogonal interactions.

3. SYSTEM ORGANIZATION

The non-orthogonal primitive features are basically simple primitive features that are oriented in the non-orthogonal directions (in the *x* and *y* axes only, not the *z* axis). These features have vertical faces, not tapered ones as in the slanting features (i.e., wedges, tapered pockets, dovetail slots, inverted dovetail slots and V slots). Examples of the non-orthogonal primitive features are shown in Fig. 2a. Fig. 2b shows the possible boundaries of the features (using dashed lines) if they are viewed as simple primitive features. These boundaries can also be viewed as the machining volumes of the features if they are machined using, for example, canned cycles. Fig. 3a, on the other hand, shows a more-complex 'to-be-decomposed' non-orthogonal interacting feature. The task here is to decompose and recognize this type of interacting feature into their constituent features, which in this case are the non-orthogonal primitive features.

The feature-recognition system developed here uses a B-rep solid model as input. The system uses a cross-sectional layer method to find the existence of any interacting feature. Then a decomposition scheme, which uses a Kohonen SOFM neural network and Boolean operations, breaks down the interacting features into non-orthogonal primitive features, which are then recognized using a multilayer feedforward neural network. The methodology used in the system is described in the following sections.

3.1 The cross-sectional layer method

There are three steps involved in the cross-sectional layer method used to detect the interacting features in a part. They are as follows:

(i) Orient the part in an orthogonal direction, so that it is opened from the top. This orientation is similar to that of a part which is machined from the top.

(ii) Slice the part, layer by layer, at the vertex points in the *z* direction (i.e., in the *xy* plane).

(iii) Take two layers at a time, and subtract the cross-sectional area that bounds the vertices in the top layer from the cross-sectional area that bounds the vertices in the bottom layer. This operation is repeated for every pair of layers, in sequence from top to bottom.

The cross-sectional layer method actually searches for any volume that exists between two successive layers of the part. The volume is represented by a Resultant Area of the subtraction process, along with a scheme of 'dots and crosses' (Zulkifli and Meeran, 1998a), which is drawn up as follows. All the vertices in the top layer are denoted by 'crosses', and those in the bottom layer by 'dots'. The vertices of the Resultant Area obtained from the subtraction operation is compared against the vertices in the top and bottom layers that were the subjects of the subtraction operation. The vertices of the Resultant Area inherit 'dots' and 'crosses' from both the layers. Hence an 'overlapping cross and dot' indicates a vertical line, whereas a 'dot' without a 'cross' means a missing vertical line. In a prismatic domain, the scheme shows the existence of vertical edges, vertical faces and, in turn, the material and non-material volumes. Thus, the Resultant Area in Fig. 3b represents a non-material volume, which in this case is a 'to-be-decomposed' non-orthogonal interacting feature.

3.2 The decomposition scheme

The decomposition scheme on the interacting features involves two major operations; the use of Kohonen SOFM neural networks to cluster the vertices of the interacting features, and boolean operations to break down the interacting features based on the clustered vertices, into non-orthogonal primitive features.

3.2.1 The Kohonen SOFM neural network

The Kohonen SOFM can capture the topology and distribution of the input data and, hence, can provide a clustering analysis for the raw data set. The clustered data can then be subjected to classification. The network is used in this work to cluster the vertices of the Resultant Area discussed earlier, with the aim of identifying constituent features from these clusters. As this work deals with 'to-be-decomposed' type of interacting features, which by definition have common bottom faces, the scheme attempts to decompose these faces into the individual bottom faces of non-orthogonal primitive features, before they are extracted for recognition. To start with, the Resultant Area of the interacting feature shown in Fig. 3b is translated into an *xy* graph as a collection of points, as shown in Fig. 4a. These points represent the vertices of the bottom layer, which are represented by the 'dots' in the Resultant Area. The vertices of the top layer, which are represented by the 'crosses', are not required to be considered, because only features of uniform depth are dealt with in this paper. Then, this collection of points is presented to the SOFM with 2 x 2 output array as input. The SOFM produces an output vector that represents the gravitational points (weights) of the clusters, which can be plotted as a feature map, shown in Fig. 4b. The map, which learned the topology and distribution of the points in the graph during the competitive training, has clustered the points into four clusters, as indicated in the figure. The map can also be imagined as an elastic rubber band, covering the input space of the points.

3.2.2 The boolean operations

The clustered points generated by applying the SOFM in the previous section can now be subjected to classification, taxonomy or any user-defined operation. In this work, boolean operations are used on the clustered points that provide the decomposition of the interacting features into non-orthogonal primitive features. The methodology used here is described in the following steps:

(i) Generate Maximal Convex Regions (MCRs).

An MCR (shown by a darker area in Fig. 5) is the maximal convex region which is formed by the edges that bound the clustered vertices. Each vertex point in a cluster of vertices has two associated edges, and each edge is defined by two vertex points. An edge is referred to as a 'complete' edge if it is defined by two vertices existing in a same cluster. On the other hand, if an edge is defined by two vertices existing in two different clusters, then this edge is refered to as an 'extendible' edge. The inner diagram drawn in the middle of Fig. 5 shows the two types of edges: a 'solid line', which represents a 'complete' edge, and a 'dashed line', which represents an 'extendible' edge.

An edge extension rule is then applied to the both ends of an 'extendible' edge. This edge is extened both ways until it intersects the boundary of the part. Hence, an MCR is generated as a convex region which is bounded by 'complete' and 'extendible' edges associated with the vertices of a cluster. The outer diagrams of Fig. 5 show the MCRs that are generated from the clusters of vertices which are obtained from the SOFM. However, the MCR for cluster 2 is not part of the Resultant Area, and no MCR can be generated for cluster 4. At least two diagonal vertex points in a cluster are required to generate an MCR.

(ii) Decompose the Resultant Area into non-orthogonal primitive regions.

The generated MCRs are intersected with the Resultant Area as shown in Fig. 6a. The intersected areas are called *non-orthogonal primitive regions*, and correspond to the bottom faces of non-orthogonal primitive features.

(iii) Find the Remaining Region.

The Remaining Region obtained from subtracting the non-orthogonal primitive regions from the Resultant Area is shown in Fig. 6b. The Remaining Region is used to test the convergence of the decomposition process by checking to see whether it is a non-orthogonal primitive region, or still represents a 'to-be-decomposed' type of interacting feature. If latter is the case, the region is further decomposed into non-orthogonal primitive regions by repeating the whole process of clustering the remaining region using another SOFM, and then intersecting the newly generated MCRs with the Remaining Region as in steps (i) and (ii) above. On the other hand, if the Remaining Region is a non-orthogonal primitive region, then the decomposition process is terminated, as the Resultant Area has now been decomposed totally into non-orthogonal primitive regions.

The Remaining Region in Fig. 6b, which in this case still represents a 'to-be-decomposed' type of interacting feature, is shown in Fig. 7a as a graph. The neural network used to cluster the points (vertices) of this Remaining Region this time is an SOFM with a 2-neuron output. The MCRs generated from the clustered points are shown in Fig. 7b. The decomposition processes of this Remaining Region as in steps (ii) and (iii) above are shown in Fig. 7c. There is no region remains after this decomposition process. The complete decomposition of the Resultant Area into non-orthogonal primitive regions is shown in Fig. 8a. The SOFM shown in Fig. 4 is called the 1st stage SOFM, while the one shown in Fig. 7a is called the 2nd stage SOFM. The decomposition process shown in Fig. 6 is called the 1st stage decomposition process (1 SDP), while the process shown in Fig. 7c is called the 2nd stage decomposition process (2 SDP).

The Kohonen SOFM neural network used in this work has to deal with a variety of interactions; some of these are complex (Fig. 3a), and others are simple ones (Fig. 6b). The task here is, hence, to design the right dimension of the output layer of the SOFM that can deal effectively with these varieties of interactions. If too many neurons are used, such as a 3 x 3 output array, then many unnecessary clusters containing insufficient points (which fail to generate any MCR) may be formed. On the other hand, if too few neurons (for example, two) are used in the output array, then only two MCRs will be generated, which could be insufficient to decompose many complex interacting regions (Resultant Areas). As a balance, an SOFM with a 2 x 2 array is selected to be used in this work. Using this 2 x 2 SOFM, the system will be able to find only four clusters of vertices, which in turn can generate only four possible MCRs. More stages of the 2 x 2 SOFM needed to be used to decompose more complex Remaining Regions. A later stage of SOFM operates on the Remaining Region resulting from applying boolean operations between the Resultant Area and the MCRs generated from the previous stage. This necessity for the multi-stage SOFM also becomes apparent when in the 1st stage SOFM the clustered vertices do not lead to generating MCRs because of insufficient vertices in clusters. The 2nd stage SOFM with diffirent configuration with smaller number of neurons may be able to find sufficient number of vertices to generate the required MCRs. It is found that to deal with most parts, albeit not covering all complex interactions, it is sufficient to use only two stages: an SOFM with a 2 x 2 output layer at the first stage, and an SOFM with a choice of 2 or 3 neurons in its output layer in the second stage.

3.3 Recognition of the non-orthogonal primitive features

The non-orthogonal primitive regions decomposed from the Resultant Area shown in Fig. 8a can now be recognized. The edges and vertices of the volumes corresponding to these regions (refer to Fig. 8b) are processed, and then established into feature patterns. The feature patterns are translated into input matrices to a multilayer feedforward neural network, which is used to recognize the features. The above processes of establishing feature patterns and translating them into input matrices are described in a companion paper (Zulkifli and Meeran, 1998a). The feature patterns for the decomposed non-orthogonal primitive regions are given in Fig. 8c. The feature patterns for region 1, 2 and 3 are recognized as non-orthogonal blind slots, while region 4 as a non-orthogonal step. This demonstrates that the aim of the work (to decompose an interacting feature into non-orthogonal primitive features and to recognize them) has been achieved.

3.4 System implementation

The decomposition and recognition system developed in this work, the block diagram of which is shown in Fig. 9, has the following four major phases:

(a) In Phase 1, the system accesses the solid model, which is created using AutoCAD® Advance Modeling Extension™ (AME™ R2.1), and extracts the vertices data of the model to an output file using an AutoLISP® program.

(b) In Phase 2, the system searches the non-orthogonal interacting features in the solid model using the cross-sectional layer method described above. The procedure is implemented in a C++ program which reads the data from the output file discussed above.

(c) In Phase 3, the system clusters the vertices of the interacting features, using a Kohonen SOFM neural network available in the MATLAB® Neural Network Toolbox, and then decomposes them using boolean operations, which are implemented in another AutoLISP® program.

(d) In Phase 4, the feature patterns of the decomposed non-orthogonal primitive features are established and recognized. The multilayer feedforward neural network used this time is called NT5000® .

4. RESULTS AND DISCUSSION

The decomposition and recognition system developed here has been tested on two different parts as shown in Fig. 10. The results are discussed as follows:

Part 1: The MCRs generated after applying the edge extension rule on the clustered vertices that are obtained from the 1st stage SOFM are shown in the figure. The Remaining Region after the 1st stage decomposition process represents a pocket and, hence, the decomposition process is terminated. Two non-orthogonal steps and a non-orthogonal pocket have been recognized.

Part 2: The four clusters of vertices obtained in the 1st stage SOFM manage to generate four MCRs. The part has been decomposed and recognized as four non-orthogonal blind slots and a non-orthogonal pocket.

The results show that the Kohonen SOFMs are capable of generating the required clusters for the MCRs needed in the decomposition processes. The 2 x 2 network used in the 1st stage SOFM manages to cluster the vertices of many generic shapes of complex interacting regions, which are then decomposed either completely into non-orthogonal primitive regions or at least into simple ones. The vertices of the simple interacting regions are clustered further, using the 2nd stage SOFM. However, to deal with more complex interacting regions, more stages of 2 x 2 SOFMs can be added, which shows the flexibility and adaptiveness of the system.

The results also show that the decomposition processes (1 SDP and 2 SDP), which used boolean intersection and subtraction operations, manage to decompose the interacting features successfully into non-orthogonal primitive features. The procedures used in these decomposition processes, which are arguably similar to the CSG-based solid modelling (although regions have been used here instead of solids), are robust as they can check whether the remaining region is a non-orthogonal primitive region or not and, hence, can decompose interacting regions totally into non-orthogonal primitive ones. Generating the MCRs is a simple and straightforward operation, as it is carried out on the clustered vertices found in the interacting regions, whereas generating the cells used in the volume decomposition approaches (Tseng and Joshi, 1994; Sakurai, 1995; Trabelsi and Meeran, 1996) can involve complicated processes of face or edge extensions. Also, these cells need to be created for the whole delta volume, or for the part itself.

5. CONCLUSIONS

The non-orthogonal interacting features considered in this paper are of the ‘to-be-composed’ type, and are characterized by having a common bottom face. This face plays an important role in this work, as the cross-sectional layer method used to find the interacting volume, the clustering of the vertices using the Kohonen SOFM, and the decomposition processes using boolean intersection and subtraction operations, all deal with this face. An advantage of using and manipulating this common face entity is that this two-dimensional framework is much simpler than using the three-dimensional entities manifested in the cell or volume decomposition techniques. Even though the system presented here also extracts volume features, this is done from 2 D patterns. It utilizes mostly the *x* and *y* points of the vertices in the cross-sectional layer method, the Kohonen SOFM and the boolean operations. In conclusion, a robust and novel system using the framework of a self-organizing feature map is presented in this paper, and makes a positive contribution in the decomposition and recognition of non-orthogonal interacting features.

AutoCAD®, AME™ and AutoLISP® are registered trademarks of Autodesk Inc, MATLAB® is registered trademark of MathWorks Inc and NT5000® is registered trademark of Neural Technologies Limited.

REFERENCES

AutoCAD, 1992, AME 2.1 Reference Manual, Autodesk, Neuchatel, Switzerland.

Chuang, S.H., Henderson, M.R., 1990, Three-dimensional shape pattern recognition using vertex classification and vertex-edge graphs. Computer-Aided Design, 22, 377-387.

Corney, J., Clark, D.E.R., 1991, Method for finding holes and pockets that connect multiple faces in 21/2D objects. Compute- Aided Design, 23, 658-668.

Gavankar, P., Henderson, M.R., 1990, Graph-based extraction of protrusion and depressions from boundary representations. Computer-Aided Design, 22, 442-450.

Ji, Q., Marefat, M.M., 1995, Bayesian approach for extracting and identifying features. Computer-Aided Design, 27(6), 435-454.

Joshi, S.B., Chang, T.C., 1988, Graph-based heuristics for recognition of machined features from a 3D solid model. Computer-Aided Design, 20(2), 58-66.

Kyprianou, L.K., 1980, Shape classification in computer aided design. Ph.D. thesis, Kings College, University of Cambridge, UK.

Sakurai, H., 1995, Volume decomposition and feature recognition: Part 1 - polyhedral objects. Computer-Aided Design, 27(11), 833-843.

Tseng, Y.J., Joshi, S.B., 1994, Recognizing multiple interpretations of interacting machining features. Computer-Aided Design, 26(9), 667-688.

Vandenbrande, J., Requicha, A., 1994, Geometric recognition of interacting machining features. In Advances in Feature Based Manufacturing, J.J.Shah, M.Mantyla and D.S. Nau (eds), Elsevier, Amsterdam.

Zulkifli, A.H. Meeran, S.M., 1998a, Feature patterns in recognizing non-interacting and interacting prismatic, circular and slanting features using a neural network. International Journal of Production Research (to be published).

Zulkifli, A.H., and Meeran, S., 1998b, Decomposition of interacting features using a Kohonen self-organizing feature map neural network. Engineering Applications of Artificial Intelligence (to be published).

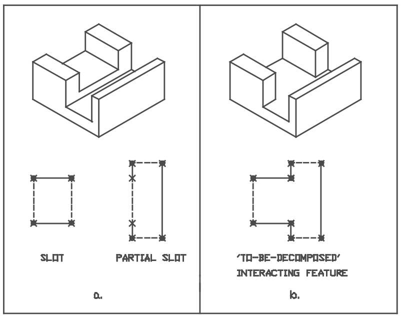


Fig. 1. Two types of interacting features.

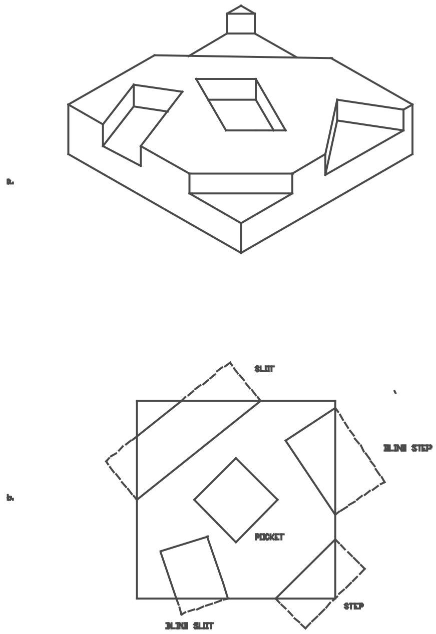


Fig. 2 The non-orthogonal primitive features.

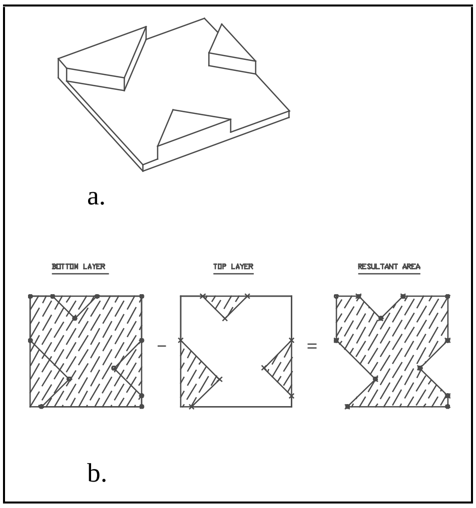


Fig. 3 A 'to-be-decomposed' non-orthogonal interacting feature.

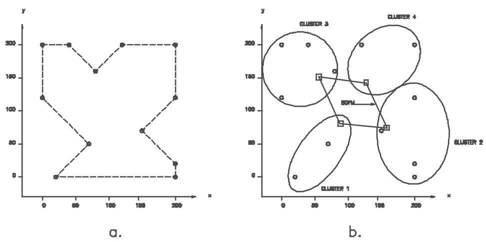


Fig. 4 The 1st stage SOFM.

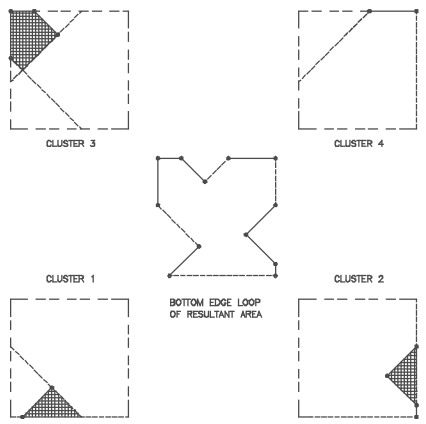


Fig. 5 The Maximal Convex Regions.

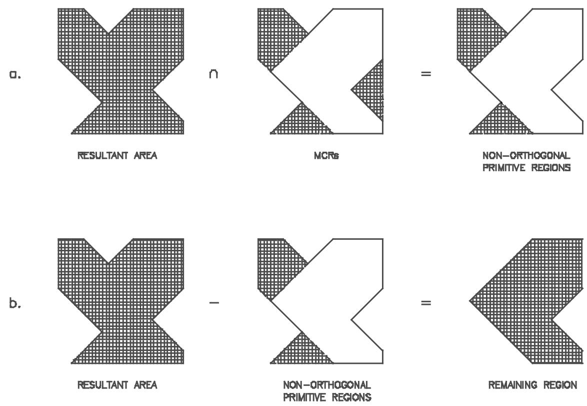


Fig. 6 The 1st stage decomposition process.

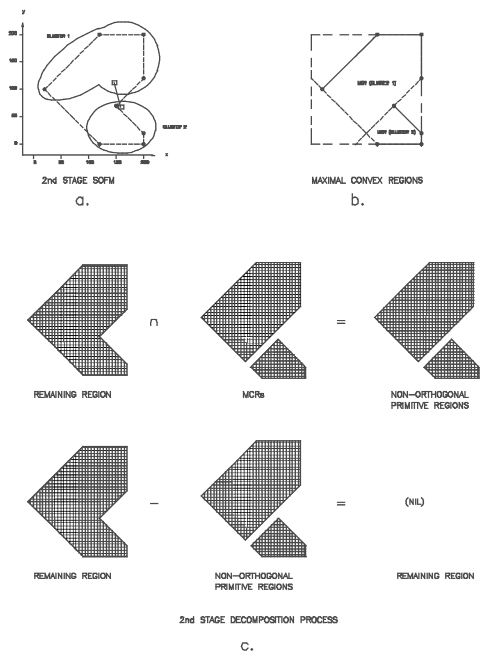


Fig. 7 The 2nd stage SOFM and decomposition process.

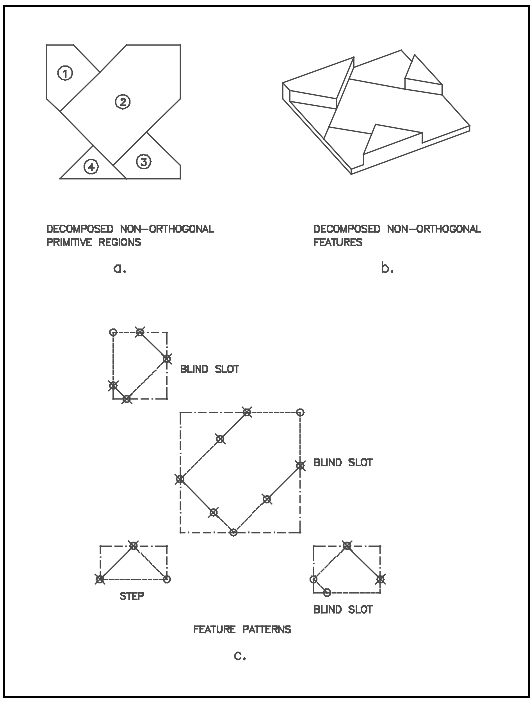


Fig. 8 Feature patterns of the decomposed non-orthogonal primitive features.

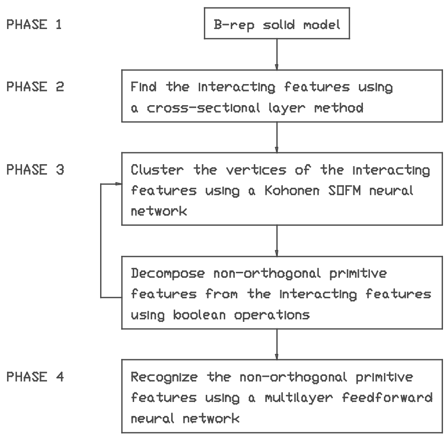
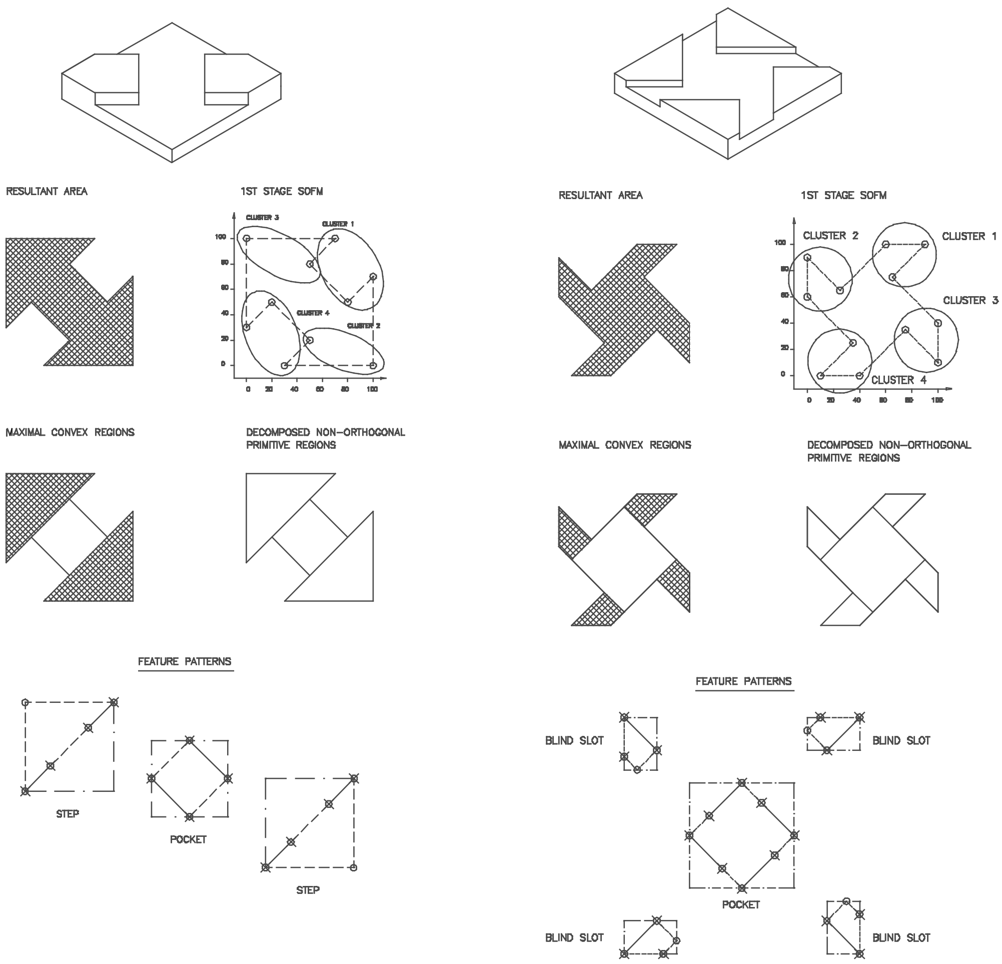


Fig. 9 System organization.



a. Part 1 b. Part 2

Fig. 10 The decomposition and recognition results.